Dissecting the RetNet Hype:
A comparative study between Retentive Networks and Transformers

Keagan Pinto
keaganjp

Anmol Mansingh
anmolmsg

Saurav Telge
sauravt

1 Abstract

Through this project, we aim to explore the Retentive Network (RetNet) and compare the retention mechanism proposed in (Sun et al., 2023) to the well-known transformer architecture initially proposed in (Vaswani et al., 2017). Transformers have proved to be game-changers in the language modeling space. While transformers have proved to be efficient in terms of training parallelism and have exhibited remarkable performance on a plethora of tasks. Transformers have failed to provide fast inference speeds. This gave rise to the impossible triangle problem (1), attempts have been made to overcome this problem to little effect. The Retention mechanism introduced in (Sun et al., 2023) claims to have finally solved this problem. To explore this exciting work we compare the performance of a relatively small-size RetNet Architecture with the GPT-2 architecture that has already been seen to be performing well. We train these models from **scratch** that have ~150 M parameters on wikitext-2 and wikitext-103, well-known datasets for pre-training language models, and subsequently generate some outputs to gauge their performance on the text generation task. We also try to emulate the ablative study in (Sun et al., 2023) and discuss all our results in the various sections below.

2 Introduction

The landscape of language modeling has been revolutionized by various models featuring diverse architectures, training methodologies, and processing techniques. Central to these advancements is the Transformer model, introduced by (Vaswani et al., 2017), which has become a foundational architecture in this domain. Originally designed to address the limitations of sequential training in recurrent neural network architectures, the Transformer excelled in language tasks and offered the benefit of training parallelism, albeit with less efficient inference.

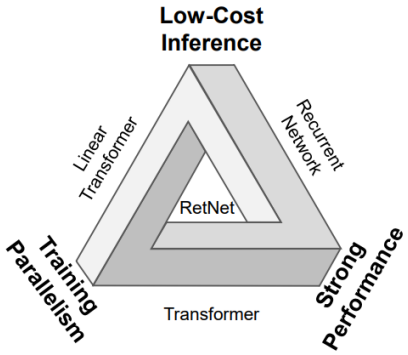

Figure 1: RetNet makes the "impossible triangle" possible, which achieves training parallelism, good performance, and low inference cost simultaneously

Addressing the challenges of balancing training parallelism, cost-effective inference, and maintaining performance – a dilemma often dubbed the "impossible triangle" – has been a focal point of recent research. In this context, the introduction of the Retentive Network (RetNet) by (Sun et al., 2023) marks a significant development.

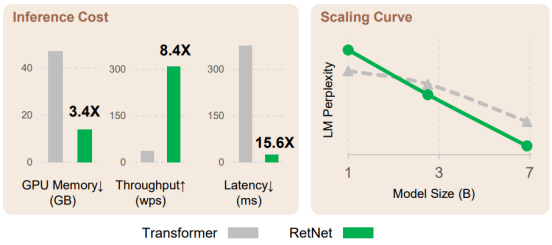

Figure 2: Retentive network (RetNet) achieves low-cost inference (i.e., GPU memory, throughput, and latency), training parallelism, and favorable scaling curves compared with Transformer. Results of inference cost are reported with 8k as input length

RetNet's architecture is touted for its low-cost inference, efficient handling of long sequences, performance on par with the Transformer, and capabil-

ity for parallel model training. Central to this architecture is the novel retention as well as multi-scale retention mechanism, an innovative substitute for the traditional attention and multi-head attention mechanisms. This mechanism encompasses three computational strategies: parallel, recurrent, and chunkwise recurrent representations. The parallel strategy leverages GPU capabilities to maximize training efficiency, while the recurrent approach focuses on efficient inference, reducing memory and computational demands, thus lowering deployment costs and latency. The chunkwise recurrent method is adept at effectively managing long sequences.

Despite these advantages, RetNet's theoretical ability to handle long-range dependencies is questioned. The retention mechanism, while efficient, might not replicate the expansive reach of the self-attention mechanism in Transformers, which is crucial for capturing long-range dependencies in sequences.

In our project, we aim to harness these methodologies to train a language model on the wikitext-2 and wikitext-103 datasets and evaluate its performance on the text generation task. A crucial aspect of our study involves benchmarking the performance of RetNet against an established large language model - GPT-2, providing a comprehensive understanding of its capabilities in comparison to these well-known models.

2.1 Related Works

In the evolving landscape of neural network research, the comparison between Retentive Networks and Transformer models, including GPT-2, represents a significant area of inquiry. This section examines pivotal contributions to this field, highlighting advancements and contextualizing the position of Retentive Networks within the broader spectrum of neural architectures.

The inception of the Transformer model by (Vaswani et al., 2017) marked a paradigm shift in neural network design, moving away from recurrent layers to focus on attention mechanisms. This foundational work has been instrumental in shaping subsequent developments in language modeling, setting the stage for the emergence of models like GPT-2. Furthering this, (Brown et al., 2020) demonstrated the Transformer model's capacity as a few-shot learner in their development of GPT-2. This study was pivotal in showcasing the adaptability of Transformer models to a range of language tasks without requiring task-specific training data, underscoring the model's versatility and efficacy.

Efficiency in processing long sequences has been a critical focus in recent research, especially relevant to Retentive Networks. (Katharopoulos et al., 2020) contributed significantly to this discourse by introducing a fast autoregressive Transformer with linear attention, offering a more efficient approach to handling long sequences. In a similar vein, (Huang et al., 2021) explored efficient attentions for long document summarization, aligning with the primary goals of Retentive Networks in managing extensive sequence data. These studies underscore the ongoing efforts to enhance efficiency in neural network architectures, a key aspect where Retentive Networks also seek to make impactful contributions.

The exploration of alternatives to traditional Transformer models has also been a topic of interest. (Poli et al., 2023) delved into larger convolutional language models, providing a fresh perspective on alternatives to the conventional Transformer architecture. This research is particularly relevant in understanding the competitive landscape in which Retentive Networks operate. Additionally, (Orvieto et al., 2023) revisited recurrent neural networks for processing long sequences, highlighting the enduring relevance of RNNs and their potential intersection with Retentive Networks.

Innovations in language modeling and neural network components have continually influenced the development of new architectures. The introduction of Gaussian Error Linear Units (GELUs) by (Hendrycks and Gimpel, 2016) is a notable example, widely adopted in modern neural networks, including Transformer-based models. Such innovations are crucial in understanding the underlying elements that might impact the performance of Retentive Networks. Further, the concept of Foundation Transformers explored by (Wang et al., 2022) adds to the understanding of scalable and robust transformer models, providing a valuable context for comparing Retentive Networks with state-of-the-art Transformer models in scalability and adaptability.

In conclusion, the literature reviewed here highlights the dynamic and rapidly evolving field of neural network architectures. The foundational work on Transformers and subsequent innovations in efficiency and alternative architectures offer a comprehensive backdrop for evaluating and comparing Retentive Networks. This ongoing exploration signifies a vibrant research area ripe with

Architectures	Training Parallelization	Inference Cost	Long-Sequence Memory Complexity	Performance
Transformer	✔	$O(N)$	$O(N^2)$	✔✔
Linear Transformer	✔	$O(1)$	$O(N)$	✘
Recurrent NN	✘	$O(1)$	$O(N)$	✘
RWKV	✘	$O(1)$	$O(N)$	✔
H3/S4	✔	$O(1)$	$O(N \log N)$	✔
Hyena	✔	$O(N)$	$O(N \log N)$	✔
RetNet	✔	$O(1)$	$O(N)$	✔✔

Figure 3: Model comparison from various perspectives. RetNet achieves training parallelization, constant inference cost, linear long-sequence memory complexity, and good performance.

potential for novel breakthroughs in efficiently and effectively handling complex neural network tasks.

3 Approach

3.1 Architecture

3.1.1 RetNet

The architecture of the Retentive Network (RetNet) is centered around the Retention Mechanism.

As stated earlier, the retention mechanism has three computational paradigms:

- **Recurrent Retention**
- **Parallel Retention**
- **Chunkwise Retention**

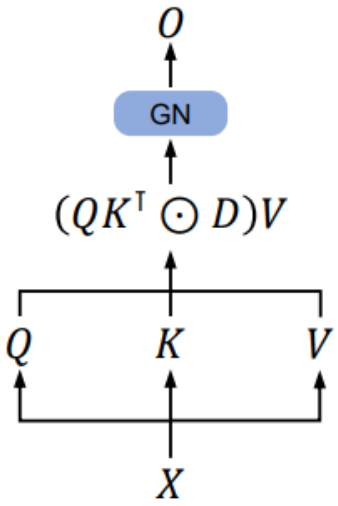

Figure 4: parallel representation

The **recurrent retention** (figure 5) mechanism maps the current input through states s_n. This defines how the *'retained'* state is updated for each token. This typically involves the combination of the current input token, the previous retained token

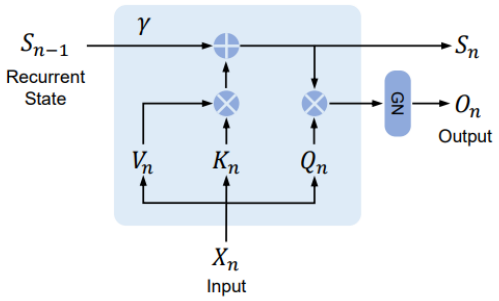

Figure 5: recurrent representation

state, and contextual information through learnable W matrices as stated in (Sun et al., 2023). This computational paradigm is responsible for the quick inference.

The **parallel representation** (figure 4) of retention is the dual of the recurrent representation. It combines causal masking and exponential decay along relative distance in terms of contextual awareness. Similar to transformers, this representation is responsible for efficient training. The **chunkwise recurrent** representation is sort of a hybrid of the aforementioned representation. As the name states, this representation divides the inputs into chunks. Within these chunks, a parallel representation is followed to conduct the computation. In-contrast cross-chunk information is passed following the recurrent representation.

3.1.2 GPT-2

GPT-2 (Radford et al., 2019) represents a major advancement in natural language processing, built upon the Transformer architecture and distinguished by its vast scale of 1.5 billion parameters. This extensive parameter count is crucial for the model's deep learning capabilities, enabling it to capture intricate linguistic patterns and subtleties. The model utilizes advanced layer normalization techniques, applied meticulously at the inputs of each sub-block and after the final self-attention

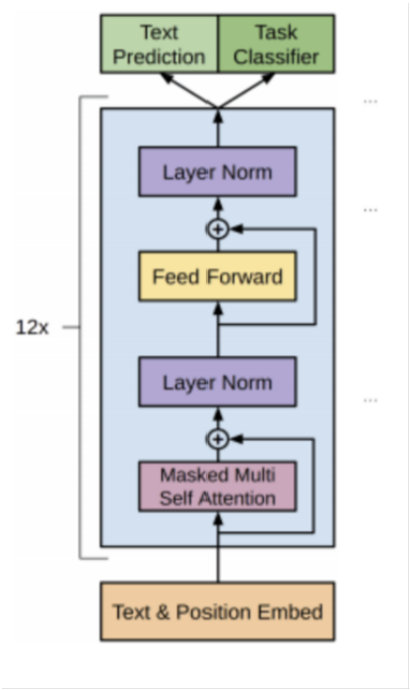

Figure 6: Architecture of GPT-2

block, drawing from the principles of pre-activation residual networks. This innovation is key to ensuring learning stability and efficiency, particularly important for a network of this magnitude.

The model's vocabulary is notably comprehensive, encompassing 50,257 words, which allows GPT-2 to process a diverse range of linguistic expressions and styles, significantly enhancing its language comprehension and generation capabilities. In addition, GPT-2 features an increased context size of 1024 tokens, a strategic enhancement that empowers the model to effectively manage longer-term dependencies and maintain coherent narratives over extended text sequences. This is a critical improvement for tasks involving long passages or complex discourse.

Furthermore, GPT-2 is equipped with a specialized initialization technique, carefully designed to address the challenges of training such a deep neural network. This approach involves adjusting the scaling of residual layers during initialization, a measure that balances the contributions of each layer, ensuring efficient and stable training. The model also benefits from modifications in the attention mechanism and feed-forward layers, optimized to handle the increased parameter count and context size. Together, these architectural features – the massive parameter count, refined layer normalization, extensive vocabulary, elongated context size, and advanced initialization method – endow GPT-2 with an exceptional capability for generating text that is not only coherent and contextually accurate but also rich in stylistic variation. These advancements make GPT-2 a highly influential and versatile model in the realm of natural language understanding and generation, pushing the frontiers of AI-driven language processing.

3.2 Datasets

In our study, we utilized the Wikitext-2 and Wikitext-103 datasets, introduced in (Merity et al., 2016), sourced from Hugging Face's open-source platform. These datasets, drawn from Wikipedia's Good and Featured articles, boast a rich collection of over 100 million tokens. They offer a comprehensive vocabulary and maintain original text features like case, punctuation, and numbers, unlike the Penn Treebank used for our homework assignments. Their extensive size, with Wikitext-2 being twice and Wikitext-103 over a hundred times larger than the processed Penn Treebank, makes them ideal for models leveraging long-term dependencies.

We used the raw versions instead of the pre-processed versions for pre-training since we want the model to the nuances of language. The wikitext-2 dataset consists of **44.8k** rows and wikitext-103 consists of **1.81M** rows with the following splits:

Split	Rows
Train	36,718
Validation	3,760
Test	4,358

Table 1: Wikitext-2 Dataset Splits

Split	Rows
Train	1,801,350
Validation	3,760
Test	4,358

Table 2: Wikitext-103 Dataset Splits

However, we used **1/3rd** of the training split for analyses explained further in the report. Shown below are the first three rows sampled randomly from this split:

WikiText Random Sample: *'Perhaps because Abraham Lincoln had not yet been inaugurated as '*

'President, Captain Totten received no instructions from his '
'superiors and was forced to withdraw his troops. He agreed to '
'surrender the arsenal as long as the governor agreed to three '
'provisions : ',
' The governor would take possession of the arsenal in the name of ' 'the United States. '

Coming to the data processing process, we used a pre-trained **GPT-2 Byte Pair Encoding(BPE)** tokenizer for our models namely RetNet and GPT-2 for simplicity, as they follow a similar decoder-based architecture.

3.3 Implementation

We trained the RetNet and the GPT-2 models with the following hyper-parameters: AdamW optimizer (betas=0.9, 0.98), hidden dim=768, number of layers=12, head dim=8, weight decay=0.05, learning rate=1e-5 as mentioned in the table (3).

Table 3: Model parameters for RetNet and GPT2

Hyperparameters	Value
AdamW optimizer	betas = 0.9, 0.98
Hidden dimensions	768
Number of layers	12
head dimension	8
Weight decay	0.05
Learning rate	1e-5

This resulted in both models having approximately 150M parameters.

4 Evaluation

4.1 Comparative Analysis of Different Approaches and Configurations

The evaluation section focuses on comparing various approaches and configurations of RetNet and GPT-2, as well as benchmarking against baseline results from previously published work. This comparison is based on the different configurations and methodologies employed in the training and fine-tuning of these models.

We first focused on all three aspects of the "impossible pyramid", but quickly realized that we wouldn't be able to evaluate the "training parallelism" pillar, due to computational limitations with procuring multiple GPUs with Great Lakes. Hence, we shifted our focus to two main tasks for evaluation:

1. Training baseline architectures from scratch, followed by initial evaluation (untuned) on a prompt-based text generation task
2. Fine-tuning on a smaller dataset (tiny-shakespeare), and human evaluation of output quality and computing perplexity score

4.2 Approaches and Configurations

RetNet: Utilized the retention mechanism with three computational paradigms: Recurrent Retention, Parallel Retention, and Chunkwise Retention. It was pre-trained on the Wikitext-2 and Wikitext-103 datasets, and fine-tuned on the tiny-shakespeare dataset.
GPT-2: Followed a traditional Transformer architecture, also trained and fine-tuned on the same datasets. This comparison provided an understanding of how different architectural choices impact model performance.

Both models were scaled to approximately 150M parameters, with consistent hyperparameters. Ensuring similar scaling and hyperparameters allowed for a fair comparison of the inherent architectural efficiencies and limitations of each model.

4.3 Baseline Comparisons

RetNet: In the foundational study (Sun et al., 2023), RetNet demonstrated advantages in larger model sizes, particularly over 2B parameters, suggesting its scalability.
GPT-2: Known for robust performance across various sizes, as reported in its foundational papers. The baseline comparisons indicate that while RetNet shows promise in larger configurations, GPT-2 maintains a broad spectrum of efficiency across different scales. As we lack the computational resources to scale the model to those levels, we focus on evaluating the peformance on smaller models on a text-generation task, which was not a subject of focus in the reference paper.

4.4 Ablation Studies

The impact of different architectural components was analyzed, particularly focusing on the head dimensions and the activation functions in RetNet. These studies are critical in understanding how specific components in RetNet contribute to its overall performance and where improvements might be necessary.

Preliminary observations suggested that RetNet might offer slightly quicker inference times for the same number of tokens, but the difference was not

substantial. This aspect highlights the importance of optimizing both models for efficiency, especially in real-world applications where inference speed is crucial.

4.5 Conclusion from Evaluation

Architectural Impact: The different approaches, especially the retention mechanism in RetNet versus the traditional Transformer model in GPT-2, show distinct impacts on performance and efficiency. **Model Scaling**: Both models exhibit varying strengths when scaled, with RetNet potentially showing more promise in larger configurations. However, as highlighted in the results section, due to computational constraints, we were not able to verify on long-range dependencies **Ablation Insights**: The ablation studies shed light on the importance of specific architectural choices in RetNet, providing a pathway for future optimizations. This evaluation, focusing on the comparison between different approaches and configurations, lays the groundwork for the subsequent discussion of results, providing a context for understanding the performance and potential of RetNet in comparison to GPT-2.

5 Discussion of Results

5.1 Pre-Training

As mentioned earlier, we adopted a trial-and-error approach, where the results obtained from each run of training influenced the experiment setup of the next. To begin, we scaled both Retnet and GPT-2 to 150M parameters and trained them from **scratch** on Wikitext-2. The intermediate results obtained are shown below:

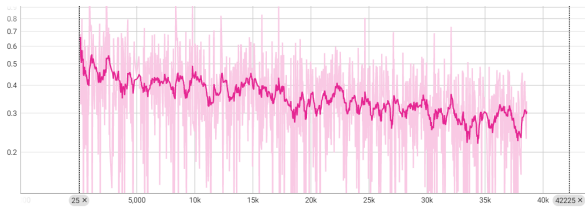

Figure 7: Training loss curve for RetNet

As we can see from the train and val loss plots, the GPT-2 model began overfitting after around 5000 iterations. We observed similar behavior in the RetNet model as well. At this point, as stated earlier, we decided to remain flexible and pivoted to the much larger Wikitext-103. After similar data preprocessing, we resumed training from the

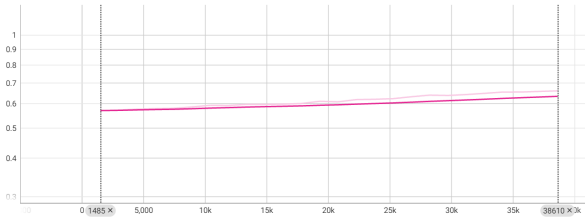

Figure 8: Validation loss curve for RetNet

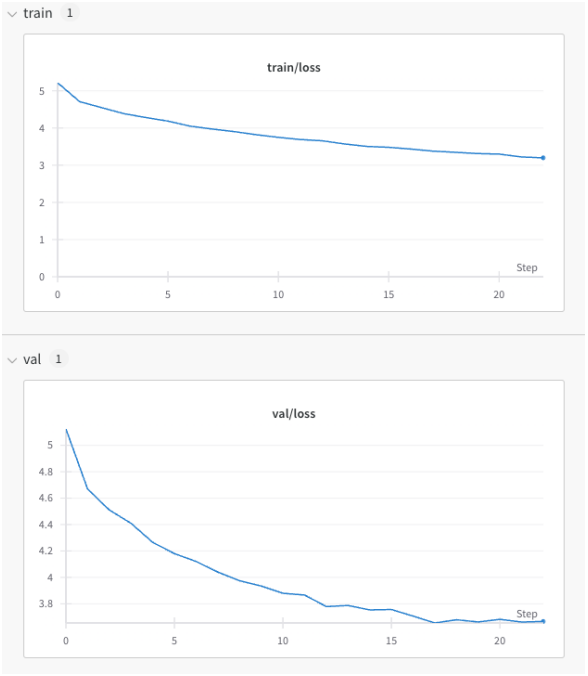

Figure 9: Training and validation loss curves for GPT-2

saved checkpoints for both models and trained it for another 35000 iterations. It is also important to note that the RetNet model continued to overfit however GPT-2 showed much better results. After pre-training, we first observed the un-tuned text generation outputs from both outputs (64 tokens).

Input prompt: William Shakespeare

RetNet:*William Shakespeare's mother has noted that it is the "great little woman in the world".*

GPT-2:*William Shakespeare's works. From the original, the play was so popular that it was revived several months after its release*

From these results, it is apparent that the RetNet model performs significantly worse than GPT-2 given similar training conditions, input prompt and hyper-parameters. This is consistent with the

original paper's claims that RetNet's performance doesn't improve significantly at a low parameter count. Hence, rather than computing the perplexity score here, we decided to fine-tune both models on the tiny-shakespeare dataset first (more details in next section).

5.2 Fine-Tuning

Next, we fine-tuned both models on tiny-shakespeare for a downstream text generation task. The results are shown below.

RetNet:

RICHARD: Will you have drawn'd on his brother, He which is king, if it be?

CATESBY: Why, my lord, my lord.

RICHARD: Why, all, my lord and I fear me not.

POMPEY: Lo, sirrah: I go, sir, yes. How do you know how you have done?

BENVOLIO: Nay, fair, you know not what in but ever thou'll care to make a gentleman tell him.

POMPEY: So

GPT-2:

Nurse: Come, we have enough time to proceed.

KING HENRY VI: To save us: so my mind hath been not made much impression and now an aspect of my life And the effect of your sorrow Plends with and off; but that knows no virtue shall be so.

CEO: Why, look you in to-Third Coriolanus. Now, Camillo, I will ask her, I guess it may be so, That ever I am sorry, so look on, And follow her that she is more distemper'd I love

From these outputs, we can say that both models had roughly similar performance on the smaller shakespeare dataset. With these fine-tuned models, we calculated the perplexity scores as another metric of performance to support the results on the text generation task shown above.

- Perplexity score for RetNet: **93.4**
- Perplexity score for GPT-2: **81**

Here, we observe that RetNet has a higher perplexity score (hence, worse performance) on this text generation task than GPT-2.

5.3 Ablation Studies

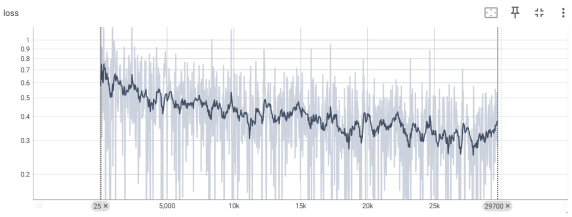

Figure 10: training loss curve when the number of heads is changed from 8 to 4.

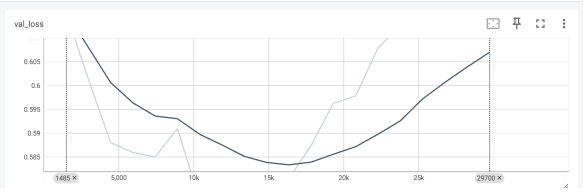

Figure 11: validation loss curve when the number of heads is changed from 8 to 4.

In our comprehensive study, we successfully conducted performance comparisons and ablation studies on the RetNet model. This study included two main components similar to (Sun et al., 2023):

- Ablating the head dimensions from 8 to 4
- Ablating the swish gate optimizer

In the ablation studies of the RetNet architecture, which consists of L identical blocks reminiscent of the Transformer layout, we delved deeply into its two primary modules: the Multi-Scale Retention (MSR) module and the Feed-Forward Network (FFN) module. Specifically, we focused on the impact of ablating the swish gate and the head dimensions from the architecture. This involved a critical examination of how these components contribute to the model's overall efficiency and effectiveness.

In order to reduce the head dimensions(h_{dim}) where $h_{dim} = d_{model}/num_{heads}$ where d_{model} is the embedding dimensions. We first reduced the number of heads to 4 from 8. As we can see in figure 10, the training loss first decreased and then started increasing again. Similar trend can be seen for the validation loss in figure 11. Next, we tried changing the number of heads from 8 to 16 to check if there is a reverse effect.

Next, for the ablation analysis of the *swish* gate, we replaced the swish gate with the *relu* activation function. However, although we saw the performance drop we did not see a significant drop. The

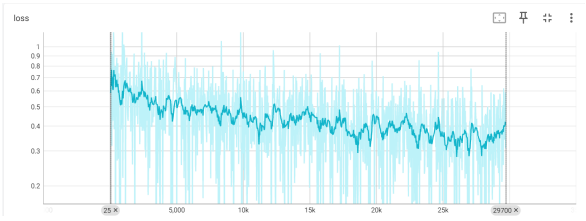

Figure 12: training loss curve when the number of heads is changed from 8 to 16.

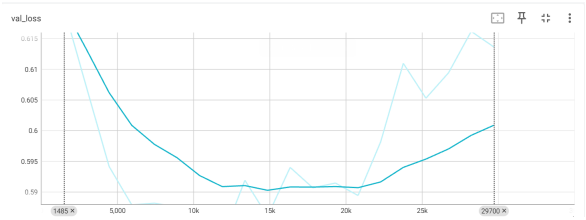

Figure 13: validation loss curve when the number of heads is changed from 8 to 16.

training loss remained similar in figure 14 but the validation loss increased significantly as seen in figure 15.

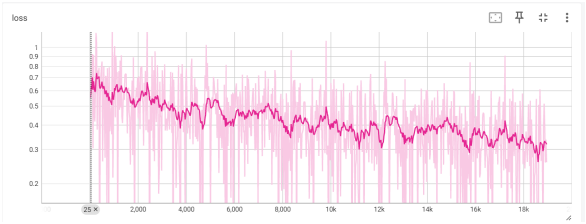

Figure 14: Training loss after ablating the swish gate by replacing it with ReLU

Through these rigorous examinations, we gained a nuanced understanding of the RetNet architecture, revealing how each component distinctly influences the overall performance. Our findings not only substantiate the claims made about RetNet but also provide valuable insights into optimizing network architectures for enhanced performance with resource efficiency.

We observed a surprising jump in performance with very little additional pre-training.

6 Conclusion

Through a number of systematic experiments, we reached two main conclusions:

- RetNet outperforms GPT-2 in terms of inferencing speed
- GPT-2 still outperforms RetNet in text generation output quality

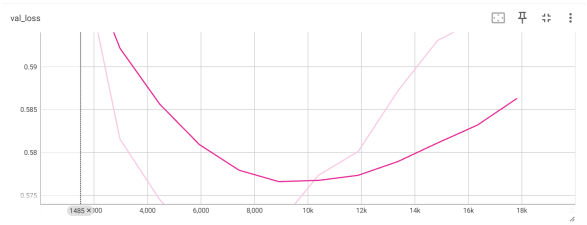

Figure 15: Validation loss after ablating the swish gate by replacing it with ReLU

Table 4: Analysis of inference times for RetNet and GPT-2.

Model	Number of tokens	Inference time (sec)
RetNet	1024	26.64
	512	13.21
	256	6.60
GPT-2	1024	31.11
	512	12.65
	256	4.53

We are cognizant of some of the conditions in the RetNet paper regarding the size of the model in order to significantly improve the model performance in certain cases. When we proposed this project we did not foresee RetNet to outperform GPT-2, however, we expected the output quality to be comparable. Apart from quicker inferencing for the selected size of the model, we did not see very comparable results. The RetNet from the fine-tuning over the tiny-shakespeare as mentioned above showed higher perplexity scores than GPT-2. However, it did take into account the prompt and generated and generated something related to it. We can observe that the generated text was not at par with the GPT-2 output.

Another point to note is the convergence while training. We can see through the plots that GPT-2 model converged significantly better than the RetNet, with the same dataset under similar training conditions and data pre-processing. This can be attributed to the fact that GPT-2 has been around for a while and there have been many iterations to the code of the architecture compared to the newer RetNet.

7 Future Work

After exploring and comparing Retentive networks with GPT-2 architecture, we feel there are more steps that we can take to further solidify our comparative study. Few research that we would like to

explore are -

- Train the Retentive network and GPT-2 model for more number of epochs and using a larger dataset with around 150M parameters.
- Explore the effectiveness of training parallelism, once we have access to multi-GPUs setup.
- Further finetune and evaluate on more downstream tasks such as text summarization, text abstraction, title generation etc.

8 Author Contribution

The proposed project has been executed by three individuals. We performed the work as follows (authors are in order of names cited at the beginning of the paper):

- All authors contributed equally to the most intensive tasks; i.e. data pre-processing, loading and writing the report. In addition,
- Author 1 also worked on training RetNet from scratch, and fine-tuning on text generation along with evaluation
- Author 2 also worked on training GPT-2 from scratch, and fine-tuning on text generation along with evaluation
- Author 3 also worked on performing comprehensive ablation studies on RetNet.

9 URL

Github link to our project - https://github.com/anmolmansingh/EECS595-RetNet-Study

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. *arXiv preprint arXiv:2104.02112*.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR.

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

Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. 2023. Resurrecting recurrent neural networks for long sequences. *arXiv preprint arXiv:2303.06349*.

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. 2023. Hyena hierarchy: Towards larger convolutional language models. *arXiv preprint arXiv:2302.10866*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. 2023. Retentive network: A successor to transformer for large language models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Hongyu Wang, Shuming Ma, Shaohan Huang, Li Dong, Wenhui Wang, Zhiliang Peng, Yu Wu, Payal Bajaj, Saksham Singhal, Alon Benhaim, et al. 2022. Foundation transformers. *arXiv preprint arXiv:2210.06423*.