# Modalities got latent: A novel approach to use latent spaces for more efficient usage of resources in multi-modal models.

Srijito Ghosh

Undergrad, Computer Science and Engineering

University Institute of Technology, The University of Burdwan

Bardhaman, West Bengal, India

srijito2work@gmail.com

*Abstract*— **In recent times, models like DeepSeek V3 have taken not just the AI community, but the world at large by storm, all thanks to their novel approach of Multi-head Latent Attention (MLA for short)[2], a significant upgrade to the traditional transformer architecture[1] (proposed in the 2017 paper, “Attention is all you need”, by Vaswani et al.) with the introduction of latent spaces when caching the Key and Value vectors to ensure less space usage, without compromising on performance. However, one question that crossed our minds is- Why don’t we perform the entire process (including attention and embedding layers) using just the reduced latent vectors, only to upscale it later to get the outputs? That way we can not only reduce the need for more space but also ensure faster training and inference times, thereby overcoming the space-time tradeoff to quite an extent. For this project, we will need the following frameworks: namely- Pytorch, Scikit-learn, and Pandas. The workflow involves generating the relevant embeddings, compressing them into latent spaces using linear layers, before entering the multi-head attention block to output the attention score matrix, which is then upscaled into the original representation for residual connections and passed onto the following feed-forward layers. Performing the operations at a lower scale helps us to achieve a lower computational usage (wastage) when computing at the original scale, while also preventing huge memory caching. Proving itself as a viable approach in a world where computing is becoming more expensive with each passing day, raising climatic concerns of an increased carbon footprint, due to increased resource consumption in AI servers. Any contribution to making a better model, while also conserving resources, that also saves money tenfold in return is the need of the hour, and so far our proposed solution seems to click all the boxes!**

Keywords—Latent, Attention, Climate, Energy, Resources.

# Introduction

Inspired by the advancements in memory-efficient latent caching methods, particularly in models like DeepSeek-V3, we introduce Decompressive Multi-Head Attention (DMHA) as a novel approach to enhance the efficiency of attention mechanisms in neural networks. In traditional multi-head attention mechanisms, the model processes input embeddings in their full dimensionality, which can lead to substantial computational costs and high memory usage, especially when scaling to larger models or vast datasets. DMHA addresses these inefficiencies by leveraging learnable linear layers to compress input embeddings into lower-dimensional latent spaces prior to applying the multi-head attention block. This compression serves a dual purpose: it allows the model to operate on a more manageable amount of data and promotes a focus on the most salient features inherent in the embeddings. By calculating attention scores within these compressed latent spaces, DMHA reduces the complexity and overhead associated with the attention computations. Once the attention scores are established, the outputs from each attention head are subsequently decompressed back to the original dimensionality. This decompression process is designed to ensure that no important information is lost during the compression phase, allowing the final output to retain the richness of the original data while benefiting from the efficiency of working in lower dimensions throughout the computation. The final outputs from the individual heads, after they have been recombined into their original dimensions, are summed to generate the cohesive attention output. This structured approach significantly alleviates both computational overhead and memory consumption tied to traditional multi-head attention setups. By confining the core attention operations to reduced dimensions, DMHA navigates the trade-offs between performance and efficiency, striving to uphold the robustness and accuracy of attention mechanisms. Moreover, the implications of DMHA extend into the realm of sustainable AI development. As the demand for large-scale models continues to rise, employing effective strategies that minimise resource usage becomes essential. DMHA not only aims to enhance the performance of AI systems but also does so in a manner that minimises their environmental footprint and resource consumption. By fostering a methodology that keeps efficiency at its core, DMHA represents a forward-thinking step towards the evolution of more sustainable large-scale AI systems, ensuring that as our capabilities grow, our resource usage can be managed responsibly. Our work thoroughly outlines the methodology employed in DMHA, detailing the processes of compression and decompression, as well as the anticipated benefits. These include improved computational efficiency, reduced memory requirements, and the ability to maintain high performance across various tasks. Through our proposed approach, we aim to contribute significantly to the ongoing research on optimising large-scale AI models, thereby paving the way for more effective and sustainable AI technologies in the future.

# Methodology

This section details our proposed Decompressive Multi-Head Attention (DMHA) approach, designed to enhance the efficiency of multi-head attention mechanisms. DMHA achieves this by performing the core attention operations in a lower-dimensional latent space.

*Overview*:

DMHA modifies the standard multi-head attention mechanism by introducing compression and decompression steps. The input embeddings are first compressed into a lower-dimensional latent space. Attention scores are then computed within this compressed latent space. Finally, the output of each attention head is decompressed back to the original dimension before being concatenated to produce the final output.

*Compression*:

The input embeddings are compressed into a lower-dimensional latent space using learnable linear layers.

Eg. nn.Linear(in\_features=<number of tokens>, out\_features=<compressed size>)

This compresses the original embedding matrix with ‘n’ number of tokens to ‘n/2’ number of tokens (at 50% compression).

*Multi-head attention in latent space:*

The multi-head attention mechanism is applied to compressed embeddings. For each head (i), the queries (Q\_i), keys (K\_i), and values (V\_i) are derived. Post which, they go for the self-attention mechanism, which’s multiplying the Q\_i and K\_i values, scaling the attention scores gained upon multiplication of these values with the square root of the embedding dimension, passing the result through a softmax activation layer to convert the attention scores into a probability distribution, before multiplying it with V\_i to get the context values from each attention head.

*Decompression*:

The output of each attention head is decompressed back to the original dimension using a learnable linear layer.

Eg: nn.Linear(in\_features=<compressed size>, out\_features=< number of tokens>)

This decompresses the latent context with ‘n/2’ number of tokens, back into ‘n’ number of tokens.

*Output aggregation*:

The decompressed outputs of all heads are then concatenated before performing the residual operation of adding them to the original decompressed embeddings, and passing through a final feed-forward layer and softmax function to produce the final output.

P.T.O.

*Advantages of DMHA:*

This approach aims to reduce computational overhead and memory usage by performing attention calculations in a lower-dimensional space. By compressing the embeddings before the attention mechanism and decompressing them only at the final stage, DMHA offers a more efficient alternative to traditional multi-head attention.

# Experiments

Models trained:

Latent-CLIP -> A Vision Language Model for Visual Question Answering. It uses the CLIP architecture to project text and image encodings into a shared embedding space, aiming for cosine similarity calculation between the images and related text (questions and answers), aiming for contrastive learning. It too applies the compression, decompression method as mentioned in this paper for more efficient usage of computing resources. This project truly showcases the multi-modal capabilities of this approach, that we promise. We aim to work further on this project to extend its capabilities to scene understanding and reasoning. and Vision Language Action Modelling for usage in robotics.

Latent-Shakespeare -> A Text Generation model. Inspired by the capabilities of OpenAI’s GPT-2 architecture, we aimed to design our own… with a twist however. Inspite of the traditional transformer decoder architecture followed by OpenAI, we tried using our much aspirational DMHA. However, we admit that the benchmarking with the original GPT-2 model is yet to be done, but the best part is, we are confident of exhibiting in-par performance with GPT-2, even with a much smaller model of ours, while using far fewer resources, if given a bit more time to polish our architecture even more.

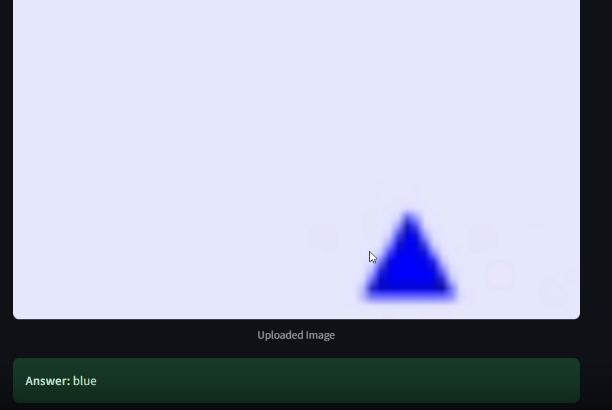
Datasets used:

EasyVQA (for Latent-CLIP)[5]: A shape identification dataset for hypothesis testing and validation before finally moving onto the actual implementation. This not only helped us with the overhead burden of training the model on an even more complex dataset, but also let us focus on the more important aspects of this research, such as designing an appropriate model for proper demonstration of our approach.

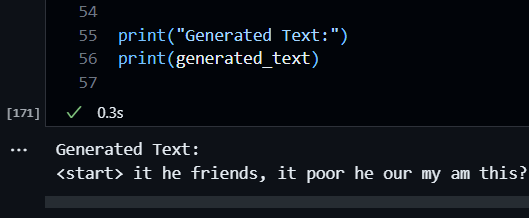
Tiny Shakespeare (for Latent-Shakespeare)[6]: This dataset contains 40000 lines of Shakespeare from a variety of Shakespeare’s plays. Featured in Andrej Karpathy’s video to build a GPT-like model from scratch[7]. Primarily built for developing a model to generate new works in the style of William Shakespeare. Our project aims to accomplish just that using the same DMHA architecture we have proposed in this paper.

# Results

Both the Latent-CLIP and Latent-Shakespeare models successfully demonstrate Visual Question Answering and Text generation capabilities, respectively.



Latent-CLIP output, on being asked, “What color is the triangle?”



Latent-Shakespeare output in next word(token) prediction after the special token, “<start>”.

## Discussion.

Reducing computational waste is crucial, especially in the field of Generative AI, where multi-modal models are becoming more common, particularly in edge AI applications. One promising solution to this problem is Decompressive Multi-Head Attention. However, like any approach, it has its limitations. One issue is that increasing the number of linear layers can enhance parallel processing. While this helps optimise GPU usage (which benefits from parallelisation), it may also lead to greater cache memory consumption. This could undermine the benefits of the compression and decompression algorithms. Additionally, using latent spaces for compression can result in the loss of important information from the original embedding space if the compression is too aggressive. Therefore, it’s essential to monitor any compression beyond 50% to prevent these potential issues.

# Conclusion

Inspired by the efficiency of latent space utilisation in recent models, this paper introduced Decompressive Multi-Head Attention (DMHA), a novel approach to enhance the efficiency of the multi-head attention mechanism. DMHA strategically compresses input embeddings into lower-dimensional latent spaces before performing attention computations and decompresses them only at the output stage.

This methodology offers the potential to significantly reduce computational overhead and memory usage, addressing the growing concerns about resource consumption and the environmental impact of large-scale AI models. By performing the core attention operations in a compressed space, DMHA contributes to the development of more sustainable and cost-effective AI solutions.

Further research and experimentation will fully validate the effectiveness of DMHA across various tasks and datasets, potentially paving the way for its adoption in future transformer-based architectures.

##### Acknowledgement

I want to extend my gratitude to the organising committee of Ideathon ’25, especially Assistant Professor Sumit Gupta, for providing enthusiastic engineering students the opportunity to explore their ideas. Thanks also to the content creators for simplifying complex topics on transformers and the latest trends[3][4][7]. Lastly, a heartfelt thank you to my parents for their unwavering support throughout this journey.

##### References

[1] Ashish, V. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*, I.

[2] Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., ... & Piao, Y. (2024). Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

[3]Illustrated guide to transformers - Step Explanation by Michael Phi:https://medium.com/data-science/illustrated-guide-to-transformers-step-by-step-explanation-f74876522bc0

[4] How DeepSeek Rewrote the Transformer [MLA]: <https://youtu.be/0VLAoVGf_74?feature=shared>

[5] EasyVQA dataset: <https://github.com/vzhou842/easy-VQA/tree/master>

[6]Tiny Shakespeare dataset: <https://raw.githubusercontent.com/karpathy/char-rnn/master/data/tinyshakespeare/input.txt>

[7]Let's build GPT: from scratch, in code, spelled out.: <https://youtu.be/kCc8FmEb1nY?feature=shared>

# Biographies

**Srijito Ghosh** is currently pursuing Bachelor of Engineering in the Computer Science and Engineering Department, University Institute of Engineering, The University of Burdwan, Bardhaman, West Bengal, India. His main research work focuses on developing resource-efficient Generative AI architectures for their primary application in robotics.