Gaussian Adaptive Attention is All You Need: Robust Contextual Representations Across Multiple Modalities

Georgios Ioannides[1,2] , **Aman Chadha**[3,4*] , **Aaron Elkins**[1]
[1]James Silberrad Brown Center for Artificial Intelligence
[2]Carnegie Mellon University
[3]Stanford University
[4]Amazon GenAI
gioannid@alumni.cmu.edu, hi@aman.ai, aelkins@sdsu.edu

Abstract

We propose the Multi-Head Gaussian Adaptive Attention Mechanism (GAAM), a novel probabilistic attention framework, and the Gaussian Adaptive Transformer (GAT), designed to enhance information aggregation across multiple modalities, including Speech, Text and Vision. GAAM integrates learnable mean and variance into its attention mechanism, implemented in a Multi-Headed framework enabling it to collectively model any Probability Distribution for dynamic recalibration of feature significance. This method demonstrates significant improvements, especially with highly non-stationary data, surpassing the state-of-the-art attention techniques in model performance (up to approximately +20% in accuracy) by identifying key elements within the feature space. GAAM's compatibility with dot-product-based attention models and relatively low number of parameters showcases its adaptability and potential to boost existing attention frameworks. Empirically, GAAM exhibits superior adaptability and efficacy across a diverse range of tasks, including emotion recognition in speech, image classification, and text classification, thereby establishing its robustness and versatility in handling multi-modal data. Furthermore, we introduce the Importance Factor (IF), a new learning-based metric that enhances the explainability of models trained with GAAM-based methods. Overall, GAAM represents an advancement towards development of better performing and more explainable attention models across multiple modalities[‡].

1 Introduction

Attention mechanisms, as exemplified in the Transformer model [1], have significantly advanced the field of sequence modeling, particularly in Natural Language Processing (NLP) and various branches of Signal Processing such as Speech Signal Processing and Digital Image Signal Processing. These mechanisms are adept at capturing dependencies within the context length, although their effectiveness can vary based on the relative placement of tokens and the inherent limitations in handling long-range dependencies due to quadratic complexity [2], [3], [4]. Ongoing research continues to address these challenges, seeking more efficient ways to model long sequences and capture global context dependencies.

In recent years, the self-attention mechanism (or variations of it that are also dot-product based) integral to Transformer layers has become central to the encoders of Pre-Trained Models (PTMs) trained via Self-Supervised Learning (SSL) methods. Examples include WavLM [5] and HuBERT [6] for Speech Encoding, Llama 2 [7] for Text Encoding, and BEiT [8] for Digital Image Encoding. These PTMs excel in generating contextualized embeddings, surpassing traditional feature engineering methods [9], and are versatile for various downstream tasks tailored to their respective training modalities.

The impetus for this work stems from the broad spectrum of downstream use cases that can benefit from an enhanced attention mechanism due to the inherent limitations of the self-attention mechanism (or other dot-product attention mechanisms variations of it) in Transformer models, which relies on normalized dot-product, and the potential advantages of adopting a more robust and explainable approach.

Self-attention's fixed-length context window can lead to sub-optimal performance [10], especially for long sequences where distant elements may not be relevant. Without inductive biases like locality [11], self-attention layers might require more data to learn patterns that are more easily captured by other methods. Despite its theoretical capability, self-attention can struggle with capturing long-term dependencies in practice, particularly as sequence length increases [12]. The interpretability of self-attention mechanisms is challenging (since we can only derive correlation-based activations and hence adopting all of correlation's drawbacks such as primarily focusing on pairwise similarities), making it difficult to understand why certain parts of the input are prioritized and while self-attention allows each token to attend to all others, it may not always effectively capture the most relevant context [13].

In our work, we introduce a significant enhancement to the Transformer model's Attention mechanism: the (Multi-Head) Gaussian Adaptive Attention Mechanism (GAAM). GAAM is designed to improve upon the standard self-attention mechanism in Transformers. Unlike conventional attention

*Work does not relate to position at Amazon.

‡Code: https://github.com/gioannides/Gaussian-Adaptive-Attention

in the Transformer, which calculates weights based on dot-product between different weight matrices, GAAM employs a Gaussian-based modulation of input features instead. This approach enables the model to concentrate on the most pertinent features in a context-sensitive manner, thereby improving its capability to interpret and process sequential and spatial data.

GAAM's attention mechanism, applied in various domains like multimedia recommendation (as in [14]), image classification (aligning with Patrick et al.'s [15] robustness strategies), and text classification (enhancing accuracy in contexts like e-commerce as shown by [16]), can significantly enhance model performance. Its ability to dynamically recalibrate feature significance based on Gaussian parameters proves particularly beneficial, offering improved accuracy, robustness, and user experience across diverse and challenging real-world applications. Furthermore, GAAM's Gaussian-based modulation offers a more interpretable framework for Artificial Intelligence (AI), addressing the critical need for transparency and trustworthiness in real-world AI systems [17].

Our proposed GAAM mechanism learns both the mean and variance of input features in a Multi-Headed setting. This mechanism operates independently across different heads, each focusing on distinct, non-overlapping subspaces of the input features. By employing Gaussian modulation, GAAM assigns varying levels of importance to each feature, effectively generating local attention outputs from each head. These outputs are then combined to construct a comprehensive Global Attention map. Each head independently adjusts its mean and variance, allowing for a focused approach to different skewness aspects in data subsets capturing a broader range of data characteristics, including asymmetries, and collectively, non-Gaussian traits. Unlike previous approaches in the literature wherein no parameters of the Gaussian distribution are learned, and are thus hard-coded making them non-specific to the data they are used on [18], [19], only multiplicative parameters like the scaled variance are learned [20], [21], a pre-defined Amplitude that is updated during training [22] or approaches that are limited because their attention framework can only explicitly model Gaussian traits behavior [23], [24].

Advantages of Learning in Multi-Head GAAM

Dynamic Contextual Adaptation: GAAM's dual learning strategy, encompassing both additive (mean offset, δ) and multiplicative (variance-based scaling factor, ξ) Gaussian learnable parameters, offers significant advantages [25]. The mean offset allows for a dynamic shift in the Gaussian distribution's mean, recalibrating the central focus of attention based on input context. This shift enhances the model's sensitivity to deviations and makes it a more contextually relevant center. Concurrently, the variance scaling factor adjusts the Gaussian curve's spread, enabling the model to adaptively concentrate on features with varying degrees of dispersion. This multiplicative adjustment ensures that the attention mechanism is not just centered correctly but also appropriately scaled, optimizing the model's performance for specific tasks. Finally, the Multi-Headed formulation allows each head to capture different aspects of the data distribution, making it possible to collectively mimic non-Gaussian traits.

Enhanced Model Interpretability and Explainability: In a multi-head setting, GAAM's integration of learnable Gaussian parameters (mean and variance) in each attention head enables nuanced, data-specific attention interpretation. GAAM's heads adaptively focus on statistical features of the input, representing distinct statistical patterns or anomalies. This approach enhances interpretability, allowing each head's attention to be analyzed in terms of statistical significance and deviations.

The paper is structured as follows: Section 2 reviews relevant attention mechanisms; Section 3 explains GAAM and its integration with Grouped Query Attention; Section 4 discusses experimental findings and future research; Section 5 concludes. Key contributions are summarized below:

OUR CONTRIBUTIONS

- We propose a novel fully learnable probabilistic attention framework, the Multi-Head Gaussian Adaptive Attention Mechanism (GAAM) and the Gaussian Adaptive Transformer (GAT). This approach incorporates learnable mean and variance parameters, designed within a Multi-Headed framework enabling the model to approximate any Probability Distribution. This design improves the model's ability to dynamically recalibrate feature importance.
- We introduce the Importance Factor (IF) as a new learning-based metric to enhance explainability in models trained using GAAM-based methods. This measure quantitatively evaluates the significance of features, according to the model's end-goal task enhancing its interpretability and explainability.
- We use a frozen Pre-Trained Model (PTM) as an Encoder for Embedding extraction, and conduct extensive experiments to validate the effectiveness of GAAM within GAT across various downstream modeling tasks in multiple modalities such as Speech, Text and Vision. Our results demonstrate its superiority, particularly in handling data with highly non-stationary attributes, over conventional dot-product attention in SSL models (e.g. in Transformer) and earlier Gaussian-based attention mechanisms.
- We demonstrate the integration of GAAM with Grouped Query Attention (selected for its superior computational efficiency, close performance to MHA [26] and benefits of hierarchical learning [27]), showcasing how this combination leads to a compatible integration of our proposed attention mechanism with dot-product based ones, that are the most popular amongst recent PTM models in the literature (e.g., WavLM, HuBERT, LLama2, BEiT etc.) without compromising performance, but instead, improving it with only a relatively marginal increase in learnable parameters.

2 Related Work

2.1 Attention Mechanisms

An exploration of the diverse landscape of attention mechanisms in the literature reveals a rich tapestry of approaches, each contributing uniquely to advancements in sequence modeling.

The multi-head self-attention mechanism, integral to the Transformer architecture, enhances sequence modeling by parallelizing attention across multiple 'heads'. Each head independently computes attention scores using the scaled dot-product attention:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

where d_k is the dimensionality of the keys. The Multi-Head Self-Attention (MHA) is defined as:

$$\text{MHA}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_H)W^O, \tag{2}$$

where each head is defined as

$$\text{head} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{3}$$

This parallelism allows the model to capture relationships within the data, as each head focuses on different aspects of the input sequence. The outputs from all heads are concatenated and linearly transformed, resulting in a comprehensive representation that encapsulates varied contextual information. This mechanism forms the backbone of Transformer architectures, leading to enhanced capabilities in tasks such as machine translation and text summarization, offering a more nuanced and efficient approach to understanding sequential data.

The Grouped Query Attention (GQA) [26] mechanism serves as an innovative intermediary between MHA and Multi-Query Attention (MQA) [28] in neural networks. GQA, however, divides the entire set of query heads into G groups, each sharing a single key and value head, effectively interpolating between MHA and MQA. This is represented as:

$$\text{GQA}(Q, K, V, G) = \text{Concat}(\text{group}_1, \dots, \text{group}_G)W^O, \tag{4}$$

with each group group_i having a shared key and value head. This mechanism allows for a balance in computational efficiency and model capacity, making GQA a versatile approach.

The Gaussian Context Transformer (GCT) [20] offers an approach in channel attention mechanisms, leveraging a Gaussian function for contextual feature excitation, contrasting with traditional methods that learn relationships between global contexts and attention activations. GCT employs a predetermined relationship modeled by a Gaussian function. This approach is encapsulated in the equation:

$$GCT(x) = exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right) \tag{5}$$

where x represents the input features, μ the mean, and σ^2 the variance. GCT's efficiency is underscored by its reduced parameterization, yet it achieves better performance compared to many other existing attention blocks. In the parameter-free GCT, all parameters of the Gaussian distribution are provided. In the Parametrized GCT, σ^2, is the only learnable parameter. The authors in [21] combine a similar methodolgy as in GCT with self-attention. Furthermore, Chen et al. [23] extend the GCT framework to also learn the mean. The mean becomes learnable via the parameter θ, where $\mu = \mu \times (1 + \tanh\theta)$, while [24], learn the mean and variance via Sigmoid-Gaussian approximation. However, all of these approaches can only model a single Gaussian Distribution at a time and are therefore restricted to modelling Gaussian-only traits.

2.2 General Purpose Encoder Models

WavLM [5] operates as a large-scale pre-trained model that advances speech processing by utilizing 94,000 hours of diverse audio inputs. It extends the Hidden Unit BERT (HuBERT) [6] framework, a method primarily focused on masked speech prediction, by also incorporating denoising modeling. This dual approach enables WavLM to effectively handle a wide range of speech-related tasks. At its core, WavLM employs self-supervised learning techniques, where the model learns to predict masked portions of audio inputs, thereby gaining a deeper understanding of speech patterns, nuances, and contextual information.

Llama 2 [7] represents an evolution in large-scale language models (LLMs), leveraging an optimized auto-regressive transformer architecture. A key enhancement includes training on a token volume of 2 trillion tokens from publicly sourced data. Notably, Llama 2 doubles the context length from previous iterations to 4096, enhancing text sequence comprehension and generation. The integration of Grouped-Query Attention (GQA) improves inference scalability, particularly for larger models.

BEiT (Bidirectional Encoder Representations from Image Transformers) [8] is a self-supervised learning model for vision tasks, adopting a novel approach akin to BERT [29] in NLP. It leverages masked image modeling (MIM) for pre-training vision transformers, tokenizing images into discrete visual tokens and employing a blockwise masking strategy. The model predicts the original visual tokens from these masked patches, focusing on learning higher-level semantic representations from raw pixels. This methodology facilitates effective performance on downstream tasks like image classification and semantic segmentation, outperforming other pre-training methods in both performance and fine-tuning stability.

3 Methods

3.1 Multi-Head GAAM

In the Gaussian Adaptive Attention Mechanism (GAAM), the attention weights are computed using a Gaussian probability density function, where the scaled variance is a learned parameter and the mean is adjusted by a learned offset. This approach allows the model to dynamically adapt the focus of attention, based on the input data's distributions. Each input feature vector x undergoes a process to compute its sample mean, $\overline{\mu}$ (Equation 6, *left*), and sample variance $\overline{\sigma^2}$ (Equation 6, *right*). This step is crucial for understanding the distribution of input features.

$$\overline{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i, \quad \overline{\sigma^2} = \frac{1}{N}\sum_{i=1}^{N} x_i^2 - (\mu)^2 \tag{6}$$

The sample mean, $\overline{\mu}$, is then adjusted by a learnable offset, δ, to get ψ (Equation 7). This approach allows the model to

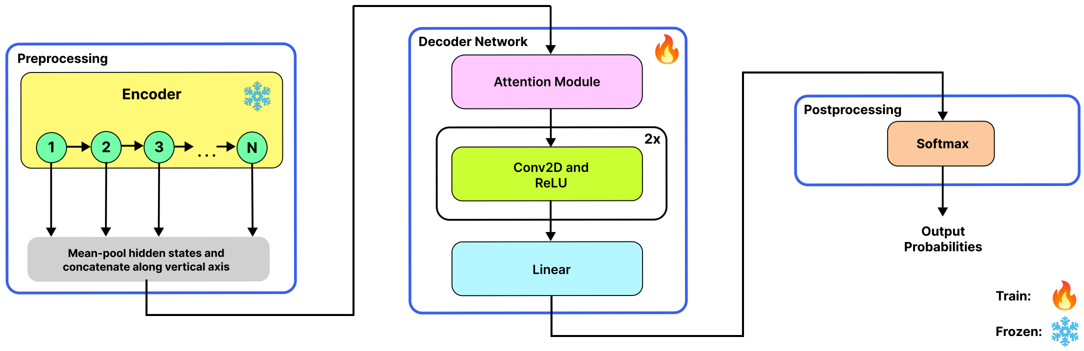


Figure 1: Proposed Model Architecture, showcasing a Pre-Trained Model (i.e. the Encoder) for feature extraction (i.e. Embeddings) via its (N) Transformer layers, followed by the Attention Module within the Decoder Network for selective emphasis, and concluding with probability output. The process flow is marked with the trainable and non-trainable (frozen) states.

dynamically adapt the focus of the attention, based on the input data's contextual structure. Its primary role is to learn a mean that is not necessarily reflective of the current batch's characteristics and attempts to approximate the population mean from the sample mean.

$$\psi = \overline{\mu} + \delta, \tag{7}$$

$$x_{\text{norm}} = \frac{x - \psi}{\sqrt{\sigma^2 + \varepsilon}} \tag{8}$$

The feature vector x is normalized using the adjusted mean, ψ, and sample variance (Equation 8) where $\varepsilon > 0$ for stability. This normalization is vital for stabilizing the learning process and enhancing model performance. The GAAM applies a Gaussian function to compute attention weights for each feature in the input vector, considering the norm and learnable scaled variance, ξ, thus exploiting the advantages of both additive and multiplicative learnable parameters (Equation 9).

$$\text{GAAM}(x_i) = \exp\left(-\frac{{x_{\text{norm}}}^2}{2\xi}\right) \tag{9}$$

In a multi-head configuration, the GAAM process is independently applied across different heads, each focusing on distinct subspaces of the original input features. The final output is computed as an element-wise multiplication (Hadamard product) of the input features and the Gaussian attention weights (Equation 10).

$$\text{Output}_j = x_{ij} \odot \text{GAAM}_j(x_{ij}) \tag{10}$$

where $j \in \{1, K\}$ and K is the total number of heads and $\odot$ denotes the element-wise multiplication operation. This process enhances the model's ability to focus on contextually relevant information in the input sequence. All head outputs are stacked vertically, forming the Gaussian Attention map.

3.2 GAAM with Grouped Query Attention

Following the integration of Multi-Head GAAM, we investigate its compatibility with dot-product-based attention mechanisms (e.g. MHA, MQA, GQA). Our focus on Grouped Query Attention (GQA) is driven by its comparable performance to MHA, superior computational efficiency [26] and advantages of its hierarchical learning structure [27]. This approach is termed as Grouped Query Gaussian Adaptive Attention Mechanism (GQGAAM).

3.3 Encoder and Decoder models

We apply different Attention mechanisms on SSL-based PTMs acting as Encoders to extract Embeddings from. Specifically, we utilize the pre-trained model weights from three distinct Encoders: (i) WavLM-Large, (ii) Llama2-13B, and (iii) BEiT-Large.

Libri-light [30] GigaSpeech [31] and English parts of VoxPopuli [32] have been used for pre-training (i). ImageNet-1k [33] has been used for pre-training (iii). Conversely, while (ii) has undergone pre-training on undisclosed, publicly sourced textual data, this aspect does not impact our research. The downstream application we employ is different from the model's original pre-training task. The specific datasets used in this work are described in further detail in Section 4.1.

The role of PTMs within the proposed model architecture, as depicted in Figure 1, is crucial during the inference phase (post-training). It is important to note that in this study, PTMs are utilized in their original pre-trained state, eschewing any further re-training during the preprocessing stage. For each PTM under consideration, the Encoder component remains static (frozen), allowing the focus to be on training and subsequently evaluating the performance of the newly proposed Decoder on the designated downstream task. This approach ensures that the intrinsic properties and learned representations of the PTMs are preserved, while the Decoder adapts and fine-tunes to the specific requirements of the task at hand [34].

The output from each transformer layer (in the Encoder) undergoes mean pooling across the time dimension (sequence length), followed by concatenation of these pooled outputs. These concatenated outputs then serve as input embeddings for the Attention Module, which employs either (i) Multi-Head Self-Attention, (ii) Multi-Head Gaussian Adaptive Attention, or (iii) Multi-Head Grouped Query Gaussian Adaptive Attention where (ii) and (iii) are contributions of this work. **When (ii) or (iii) are used, the Decoder Network is termed as the Gaussian Adaptive Transformer**.

In mathematical terms, the embeddings are represented as

Mechanism	Heads	Learn Params.
GQGAAM	$g:8, q:8, kv:2$	1.00M - 3.16M
GQA	$q:8, kv:2$	0.984M - 3.08M
GAAMv1	$g:8$	0.016M - 0.082M
GAAMv2	$g:1$	0.002M - 0.010M

Table 1: Comparison of minimum and maximum learnable parameters in millions (M) for different Attention Mechanisms across different Pre-Trained Models (standalone GQA is not used and is only shown for comparison with the proposed GQGAAM).

$X \in \mathbb{R}^{N\times d}$, where each x_i is a vector in a d-dimensional space, with d taking values in the set $\{1024, 5120\}$. Here, N signifies the total count of transformer layers in the Encoder, which are kept in a static (frozen) state. The attention mechanism of the module then produces a new, contextualized representation $C \in \mathbb{R}^{N\times d}$ for the input sequence. Subsequently, convolutional layers are utilized to distill features (pertaining to speech, text, or image data) from the context matrix generated by the attention mechanism. By employing 2-dimensional convolution layers (with parameters kernel_size $= (3, 3)$, stride=1, padding=1), the model adeptly processes the array of context tensor outputs from each transformer layer.

Table 1 lists the attention mechanism parameters for the proposed GAAM-based decoders of WavLM-Large, Llama2-13B, and BEiT-Large encoders. Here, g denotes GAAM-based head count, with higher values indicating a higher number of learned Gaussian Distributions. q and kv are the counts of query and key-value heads, respectively. The embedding dimensions are 1024 for WavLM and BEiT, and 5120 for Llama2.

All decoder network models are trained for 35 epochs and their layer weights (except their respective attention module) are initialized using Xavier initialization [35]. Adam [36] is used as the optimizer, with both weight decay factor of 0.1 and an initial learning rate of 10^{-4} (except for when Llama 2 is used as an Encoder, in which case it is 5×10^{-5}). For the SER experiments, Focal Loss [37] is used, where $\gamma = 2.5$. For the text and image classification experiments the Cross-Entropy Loss is used. All GAAM-based attention module are initialized with a mean offset, $\delta = 0$, where $\delta \in 1 \times d$ and scaled variance, $\xi = 2$, where $\xi \in 1 \times d$. A batch size of 8 is used for SER and a batch size of 32 for Text and Image Classification. Across all downstream tasks, mixed precision training is utilized. Regarding the SER downstream task – during training and evaluation, audio files are split to a maximum of 5 second clips. If an audio file exceeds 5 seconds in duration, a new audio file will be generated containing the excess audio. Each audio file is passed through the trained Encoder model. For the text classification downstream task, text is tokenized with maximum context length of 4096 during both training and evaluation. For the image classification downstream task, images are resized to 224×224 dimensions during both training and evaluation.

3.4 Evaluation Metrics

In this study, the primary evaluation metric is Accuracy (Acc.), calculated as the ratio of correct predictions to total predictions. Additionally, the Importance Factor (IF) is introduced, calculated using Gaussian Attention weights (GA) from the attention module. IF is $\frac{\mathrm{GA}_{ij} - \min(\mathrm{GA})}{\max(\mathrm{GA}) - \min(\mathrm{GA})}$ with IF $\in [0, 1]$, indicating the relative importance of features in the model's decision process. Higher IF values indicate more significant features and vice versa. IF-based heatmaps are created by taking the arithmetic average of the generated Gaussian Attention maps during validation and then applying the IF formula. They visually depict feature importance.

4 Results and Discussion

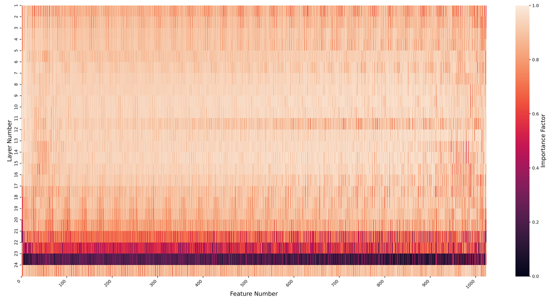
4.1 Datasets

The frozen Encoder of each of the three PTM implementations (as described in Section 3) is used to train and evaluate a Decoder on the *IEMOCAP* [38], *AG News* [39] and *CIFAR100* [40] datasets to assess the applicability of the newly proposed attention mechanisms across speech, text and image modalities. For the *IEMOCAP* dataset, we employ 5-fold cross-validation, training on 4 sessions and validating on 1. We focus on the emotion categories *neutral*, *happiness* (merging *happiness* and *excited*), *anger*, and *sadness*. This dataset includes diverse speech samples across speakers and genders, with a sampling rate of 16 kHz for both training and validation audio files. *AG News* dataset is employed in our study. It comprises of a comprehensive collection of web-based news articles. Our dataset construction focuses solely on the title and description fields of these articles. In terms of data distribution, each category (out of four) contributed 30,000 articles to the training set and 1,900 articles to the validation set. The *CIFAR-100* dataset consists of 60,000 color images, grouped into 100 classes, each containing 600 images. The dataset provides a diverse range of everyday objects, animals, and vehicles. For our analysis, we use the following division of the CIFAR-100 dataset: 50,000 images for training and 10,000 for validation.

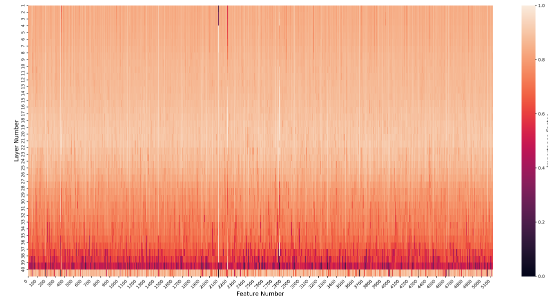
4.2 Current benchmarks

Method	F1	F2	F3	F4	F5	Avg.
MHA	0.627	0.599	0.617	0.613	0.657	0.623
MHA $\rightarrow$ BN	0.627	0.599	0.629	0.648	0.666	0.634
SGA	0.646	0.594	0.658	0.652	0.656	0.641
GAAMv2	0.661	0.600	0.663	0.652	0.654	0.646
bi-SGA	0.668	0.606	0.654	0.671	0.661	0.652
GQGAAM	0.665	**0.654**	**0.687**	0.659	0.668	0.667
GAAMv1	**0.672**	0.646	0.681	**0.679**	**0.690**	**0.674**

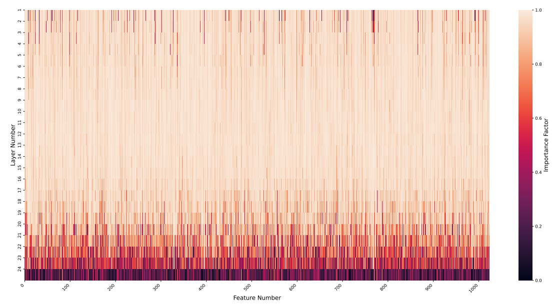
Table 2: IEMOCAP 5-fold validation (F1-F5) using WavLM-Large as Encoder.



(a) Importance Factor values for Speech Processing with *WavLM-Large* using GAAM.



(b) Importance Factor values for Text Processing with *Llama2-13B* using GQGAAM.



(c) Importance Factor values for Digital Image Processing with *BEiT-Large* using GQGAAM.

Figure 2: Importance Factor values for different processing tasks with their respective models using GAAM and GQGAAM.

Method	Best Acc.
MHA	0.604
MHA $\rightarrow$ BN	0.630
GAAMv1	0.799
GQGAAM	0.800
SGA	**0.802**
bi-SGA	**0.802**
GAAMv2	**0.802**

Table 3: CIFAR100 using BEiT-Large as Encoder.

Method	Best Acc.
MHA	0.944
GAAMv2	0.944
MHA $\rightarrow$ BN	0.945
SGA	0.945
bi-SGA	0.945
GAAMv1	0.945
GQGAAM	**0.948**

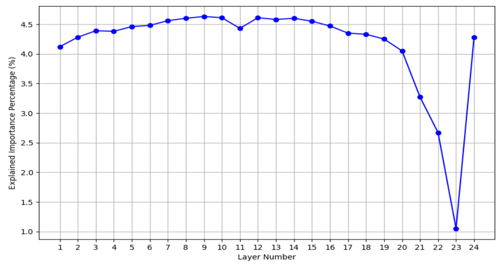
Table 4: AGNews using Llama2-13B as Encoder.

Modality	**Mean Offset**	**Scaled Variance**
Speech	[-0.06, 0.10]	[1.88, 2.06]
Text	[-0.05, 0.07]	[1.94, 2.02]
Vision	[-0.02, 0.02]	[1.98, 2.03]

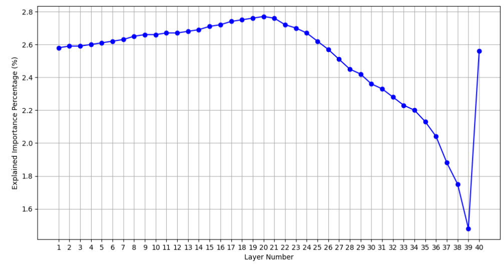
Table 5: Range of learned Gaussian parameters for normalized features (for best performing GAAM-based models using $g : 8$)

In the analysis of benchmark results across different modalities in Tables 2-4, showcases that the aspects of GAAM emerges as the overall superior approach consistently across Speech, Text and Vision modalities (see Table 1 for details on attention models). GQGAAM is a pivotal aspect of our work. It demonstrates that existing PTMs utilizing dot-product attention can be *enhanced* with GAAM – only adding a comparatively small number of additional parameters ($+2.66\%$) from a standalone GQA module (see Table 1).

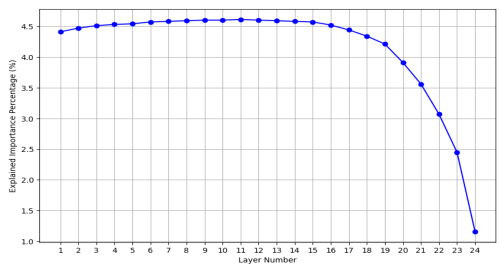
In Tables 2-4, we also contrast GAAM-based methods with those using Batch Normalization (BN), Skewed Gaussian Attention (SGA) and bi-Skewed Gaussian Attention (bi-SGA) [23]. BN operates under the assumption of independent and identical distribution across mini-batches [41] and we position BN immediately after MHA to match the normalization order in the original Transformer. GAAM, with its Multi-Headed nature, adeptly handles variations in feature distribution, including shifts in mean offset, variance, and modelling a mixture of Gaussian distributions with varying attention weights as evidenced in Table 5 and Figure 2 respectively. This adaptability leads to GAAM's superior performance over methods reliant on static feature distributions such as BN, SGA and bi-SGA. SGA and bi-SGA assume data strictly follows a single Gaussian distribution at a time, limiting its ability to model non-Gaussian traits by not having a learnable mixture of them. In contrast, GAAM can model multiple Gaussian distributions with varying parameters, enabling it to approximate any Probability Distribution. Results show that GAAM outperforms in scenarios with significant variation in Gaussian parameters (i.e. need for modelling non-stationary data, perhaps having non-Gaussian traits overall), while in cases with minimal variation, outperforms MHA and matches (bi-)SGA, depending on the number of heads used. Observing Table 5, Speech data exhibits high variability in both the central tendency (mean offset, δ) and the spread (scaled variance, ξ). This reflects the highly non-stationary nature of speech [42]. Attention mechanisms must adaptively adjust both the focus (μ) and the



(a) Percentage Contribution of each layer to attention weights for Speech Processing with *WavLM-Large* using GAAM.



(b) Percentage Contribution of each layer to attention weights for Text Processing with *Llama2-13B* using GQGAAM.



(c) Percentage Contribution of each layer to attention weights for Image Processing with *BEiT-Large* using GQGAAM.

Figure 3: Contributions of each layer to attention weights in different downstream tasks (for best performing GAAM-based models using $g:8$).

width (σ) to effectively capture the rapidly changing features in speech, essential for tasks like emotion recognition. Text data shows high mean variation perhaps due to changing semantic contexts, but the variance remains moderately stable, reflecting the structured nature of language. Attention mechanisms should primarily focus on tracking the shifting mean (μ) to align with the changing semantic and syntactic focal points, while the moderate variance variation allows for a relatively consistent attention span. In Vision, both the mean and variance variations are low, indicating stable and consistent feature locations and spreads. Simpler attention mechanisms can be effective, maintaining a consistent focus (μ) and width (σ), suitable for tasks where feature variability is minimal.

4.3 Analysis of Encoder Layer Contribution

We utilize heatmap visualizations of the Importance Factor revealing the interplay of features within the frozen PTMs that drive decision-making. We plot the IF heatmap of the best performing *multi-head* attention mechanism (refer to Tables 2-4) across each data modality. A concentration of higher IF values is observed in specific feature space areas for different tasks utilizing GAAM and GQGAAM attention methods.

For *Speech Processing with WavLM-Large using GAAM* in Figure 2(a), the dense population of higher IF values at the lower layers suggests these layers' active role in modulating the input sequence. This observation implies that fundamental speech features are likely captured initially, while upper layers refine these for more abstract representations.

Conversely, *Text Processing with Llama2-13B using GQGAAM* in Figure 2(b), exhibits a more uniform IF distribution across all layers with a slight concentration at the earlier layers. This pattern indicates a balanced hierarchical feature extraction approach, with both lower and higher-level features playing a significant role, particularly those extracted by the early to middle layers.

Similarly, *Digital Image Processing with BEiT-Large using GQGAAM* in Figure 2(c) emphasizes lower layer features, reflecting the necessity of early-stage feature extraction in visual tasks, such as identifying edges and textures. These variations in IF value distribution underscore the distinct information processing requirements of each modality. Speech and image processing appear to rely on primary feature extraction, while text processing demands both fundamental and complex feature identification. The insights provided by IF analysis enhance the explainability of the models, offering a quantifiable measure of feature significance. This can lead to more informed decisions in model refinement, focusing on attention mechanisms that align with the specific demands of the data type and task at hand, thus efficiently utilizing the knowledge gained for improved model performance and understanding.

4.4 Ablation Study

We validate that the IF from GAAM and GQGAAM attention weights accurately identifies key feature extraction regions affecting model performance. This is achieved by reassessing experiments focused on layers with low and high IF scores, aiming to understand the link between IF scores and the significance of the highlighted features.

Layer	F1	F2	F3	F4	F5	Avg.
9 (High IF score)	**0.659**	**0.601**	**0.644**	**0.627**	**0.670**	**0.640**
23 (Low IF score)	0.628	0.589	0.632	0.620	0.645	0.623

Table 6: Comparison of Layers with High and Low Importance Factor Scores for IEMOCAP 5-fold Validation (F1-F5) using WavLM.

Dataset	High IF score Layers	Low IF score Layers
AGNews	**0.949** (19, 20, 21)	0.947 (37, 38, 39)
CIFAR100	**0.726** (10, 11, 12)	0.647 (22, 23, 24)

Table 7: Best Accuracy for High and Low Importance Factor Score Layers using Llama2-13B (AGNews) BEiT-Large (CIFAR100).

Across Speech, Text, and Vision modalities, higher IF scores consistently align with improved model performance and vice versa. For the Speech downstream task, High IF layer (Layer 9) consistently outperforms low IF layer (Layer 23) across all folds (Table 6). For the Text and Vision downstream tasks (Table 7), layers with higher IF achieve better performance, especially in Vision. In MHA, attention weights primarily indicate the level of correlation between different parts of the input sequence [43]. Each element's weight reflects its relevance to every other element within the same sequence. However, this approach does not directly translate to the performance on downstream tasks. For instance, the authors in [34] derive normalized self-attention weights for SER on IEMOCAP using WavLM-Large, identifying layer 23 as pivotal. While useful for their use case, this only indicates inter-layer correlation, and not a direct link to better or worse performance and would be misleading to use these weights for that use case (as shown in Table 6 using GAAM for the same task). In contrast, GAAM-based learning dynamically adjusts attention weights tailoring attention to improve feature representation aligned with the model's end goal. Analysis of Figure 3 indicates earlier layers, exhibit more meaningful features, and contribute more to model performance, suggesting potential overparameterization in later layers [44]. Future work should explore GAAM in additional tasks, datasets, grounding experiments [45], and beyond feature extraction, including model compression using attention weights during training (crucial for resource-limited applications) [46].

5 Conclusion

In this work, we introduce the Multi-Head Gaussian Adaptive Attention Mechanism and the Gaussian Adaptive Transformer. We demonstrate their effectiveness in enhancing model performance, particularly with highly non-stationary data. Results show that combining learnable mean and variance for every Gaussian Distribution enables dynamic feature significance recalibration and approximation of any Probability Distribution across multiple modalities. Integrating this mechanism with the dot-product attention mechanism enhances performance with a minimal increase in parameters. Finally, we introduce the Importance Factor for improved model explainability.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[2] L. Wang, "Rrwkv: Capturing long-range dependencies in rwkv," *arXiv preprint arXiv:2306.05176*, 2023.

[3] Y. Zhuang, J. Zhang, and M. Tu, "Long-range sequence modeling with predictable sparse attention," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 271–281.

[4] H. He, "A unified view of long-sequence models towards million-scale dependencies," *arXiv preprint arXiv:2307.03172*, 2023.

[5] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, X. Yu, and F. Wei, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[6] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," 2021.

[7] H. Touvron and L. M. et al., "Llama 2: Open foundation and fine-tuned chat models," 2023.

[8] H. Bao, L. Dong, S. Piao, and F. Wei, "Beit: Bert pre-training of image transformers," 2022.

[9] S. Chen, J. Xie, and J. H. L. Hansen, "Fearless: Feature refinement loss for ensembling self-supervised learning features in robust end-to-end speech recognition," in *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea, 18-22 September 2022*, H. Ko and J. H. L. Hansen, Eds. ISCA, 2022.

[10] Y. Li, "Unlocking context constraints of llms: Enhancing context efficiency of llms with self-information-based content filtering," *arXiv preprint arXiv:2304.12102*, 2023. [Online]. Available: https://dx.doi.org/10.48550/arXiv.2304.12102

[11] B. L. Edelman, S. Goel, S. Kakade, and C. Zhang, "Inductive biases and variable creation in self-attention mechanisms," *International Conference on Machine Learning (ICML)*, 2022. [Online]. Available: https://arxiv.org/abs/2110.10090

[12] M. Hahn, "Theoretical Limitations of Self-Attention in Neural Sequence Models," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 156–171, 2020. [Online]. Available: https://doi.org/10.1162/tacl_a_00306

[13] M. Bhan, N. Achache, V. Legrand, A. Blangero, and N. Chesneau, "Evaluating self-attention interpretability

through human-grounded experimental protocol," in *Explainable Artificial Intelligence*, L. Longo, Ed. Cham: Springer Nature Switzerland, 2023, pp. 26–46.

[14] Z. Tao, X. Liu, Y. Xia, X. Wang, L. Yang, X. Huang, and T.-S. Chua, "Self-supervised learning for multimedia recommendation," *IEEE Transactions on Multimedia*, 2022. [Online]. Available: https://dx.doi.org/10.1109/TMM.2022.3187556

[15] D. Patrick, M. Geyer, R. Tran, and A. Fernandez, "Reconstructive training for real-world robustness in image classification," in *IEEE Winter Conference on Applications of Computer Vision Workshops (WACVW)*, 2022. [Online]. Available: https://dx.doi.org/10.1109/WACVW54805.2022.00031

[16] F. M. Yıldırım, A. Kaya, S. Öztürk, and D. Kılınç, "A real-world text classification application for an e-commerce platform," in *International Symposium on Advanced Electrical and Communication Technologies (ISAECT)*, 2019. [Online]. Available: https://dx.doi.org/10.1109/ASYU48272.2019.8946337

[17] W. Jin, X. Li, and G. Hamarneh, "Rethinking ai explainability and plausibility," *arXiv preprint arXiv:2303.17707*, 2023. [Online]. Available: http://arxiv.org/pdf/2303.17707

[18] W. You, S. Sun, and M. Iyyer, "Hard-coded gaussian attention for neural machine translation," in *Annual Meeting of the Association for Computational Linguistics*, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:218487704

[19] M. Guo, Y. Zhang, and T. Liu, "Gaussian transformer: A lightweight approach for natural language inference," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 6489–6496, Jul. 2019. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/4614

[20] D. Ruan, D. Wang, Y. Zheng, N. Zheng, and M. Zheng, "Gaussian context transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 15 129–15 138.

[21] J. Kim, M. El-Khamy, and J. Lee, "T-gsa: Transformer with gaussian-weighted self-attention for speech enhancement," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6649–6653.

[22] A. Luo, F. Yang, X. Li, L. Nie, C. Lin, H. Fan, and S. Liu, "Gaflow: Incorporating gaussian attention into optical flow," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 9642–9651.

[23] C. Chen and B. Li, "An interpretable channel-wise attention mechanism based on asymmetric and skewed gaussian distribution," *Pattern Recognition*, vol. 139, p. 109467, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S003132032300167X

[24] J. Xie, Z. Ma, D. Chang, G. Zhang, and J. Guo, "Gpca: A probabilistic framework for gaussian process embedded channel attention," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 8230–8248, 2022.

[25] J. Fluri, T. Kacprzak, A. Lucchi, A. Schneider, A. Réfrégier, and T. Hofmann, "Full $w$cdm analysis of kids-1000 weak lensing maps using deep learning," *Physical Review D*, 2022.

[26] J. Ainslie, J. Lee-Thorp, M. de Jong, Y. Zemlyanskiy, F. Lebron, and S. Sanghai, "GQA: Training generalized multi-query transformer models from multi-head checkpoints," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 4895–4901. [Online]. Available: https://aclanthology.org/2023.emnlp-main.298

[27] M. Chen, Y. Bai, J. D. Lee, T. Zhao, H. Wang, C. Xiong, and R. Socher, *Towards Understanding Hierarchical Learning: Benefits of Neural Representations*. Curran Associates Inc., 2020.

[28] N. Shazeer, "Fast transformer decoding: One write-head is all you need," 2019.

[29] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *North American Chapter of the Association for Computational Linguistics*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:52967399

[30] J. Kahn and M. R. et al., "Libri-light: A benchmark for asr with limited or no supervision," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[31] G. Chen and S. C. et al., "Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio," in *Interspeech*, 2021.

[32] C. Wang, M. Riviere, A. Lee, A. Wu, C. Talnikar, D. Haziza, M. Williamson, J. Pino, and E. Dupoux, "VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation," in *International Joint Conference on Natural Language Processing*, 2021.

[33] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 12 2015. [Online]. Available: https://doi.org/10.1007/s11263-015-0816-y

[34] G. Ioannides, M. Owen, A. Fletcher, V. Rozgic, and C. Wang, "Towards Paralinguistic-Only Speech Representations for End-to-End Speech Emotion Recognition," in *Proc. INTERSPEECH 2023*, 2023, pp. 1853–1857.

[35] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, Y. W. Teh and M. Titterington, Eds., vol. 9. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 249–256. [Online]. Available: https://proceedings.mlr.press/v9/glorot10a.html

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[37] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," 2018.

[38] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower Provost, S. Kim, J. Chang, S. Lee, and S. Narayanan, "Iemocap: Interactive emotional dyadic motion capture database," *Language Resources and Evaluation*, 2008.

[39] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf

[40] A. Krizhevsky, "Learning multiple layers of features from tiny images," Canadian Institute For Advanced Research, Tech. Rep., 2009.

[41] Y. Wang, Q. Shi, and T.-H. Chang, "Batch normalization damages federated learning on non-iid data: Analysis and remedy," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[42] R. Jaiswal and D. Romero, "Implicit wiener filtering for speech enhancement in non-stationary noise," in *2021 11th International Conference on Information Science and Technology (ICIST)*, 2021, pp. 39–47.

[43] G. Lovisotto, N. Finnie, M. Muñoz, C. K. Mummadi, and J. H. Metzen, "Give me your attention: Dot-product attention considered harmful for adversarial patch robustness," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15 213–15 222, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:247748735

[44] Q. Zhang, S. Zuo, C. Liang, A. Bukharin, P. He, W. Chen, and T. Zhao, "Platon: Pruning large transformer models with upper confidence bound of weight importance," 2022.

[45] K. Li, J. Li, D. Guo, X. Yang, and M. Wang, "Transformer-based visual grounding with cross-modality interaction," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 19, no. 6, may 2023. [Online]. Available: https://doi.org/10.1145/3587251

[46] J. Back, N. Ahn, and J. Kim, "Magnitude attention-based dynamic pruning," 2023.

Appendix

Extended Results

In order to address the lack of fold-based performance evaluation, similar to what is available when using the IEMOCAP dataset, we conduct additional runs on the CIFAR100 and AG News datasets. This approach is necessary to ensure a robust validation of our methods, as the original datasets do not provide the opportunity for a fold-based analysis. We present the results of these additional runs, offering a more comprehensive view of the performance consistency and effectiveness of GAAM-based methods. In Tables 8 and 9, the first run (i.e. R1) depicts the original results from Tables 3 and 4, while the rest of the runs depict the extended results.

Method	R1	R2	R3	R4	R5	**Avg.**
MHA	0.604	0.619	0.621	0.620	0.621	0.617
MHA $\rightarrow$ BN	0.630	0.671	0.695	0.639	0.670	0.661
GQGAAM	0.800	0.801	0.801	0.806	0.800	0.801
GAAMv1	0.799	0.802	0.802	**0.807**	0.807	0.803
SGA	**0.802**	**0.807**	0.807	0.801	0.808	0.805
bi-SGA	**0.802**	**0.807**	0.807	0.801	0.808	0.805
GAAMv2	**0.802**	0.804	**0.810**	0.803	**0.810**	**0.806**

Table 8: CIFAR100 5 Run validation (R1-R5) using BEiT-Large as Encoder.

For CIFAR100, as demonstrated in Table 8, GAAMv2 overall outperforms other methods, with an average accuracy rate of 0.806. This performance is followed by SGA and bi-SGA, both yielding an average accuracy rate of 0.805. Notably, all GAAM-based methods (GQGAAM, GAAMv1, GAAMv2) significantly surpass the MHA and MHA $\rightarrow$ BN models, underlining the superiority of GAAM frameworks in handling the Vision modality.

Method	R1	R2	R3	**Avg.**
GAAMv2	0.944	0.945	0.946	0.945
MHA $\rightarrow$ BN	0.945	0.945	0.947	0.946
MHA	0.944	0.945	0.948	0.946
SGA	0.945	0.945	0.947	0.946
GAAMv1	0.945	0.945	0.947	0.946
bi-SGA	0.947	0.945	0.947	0.946
GQGAAM	**0.948**	**0.949**	**0.949**	**0.949**

Table 9: AG News 5 Run validation (R1-R5) using Llama2-13B as Encoder.

Similarly, in the AG News dataset, as shown in Table 9, GQGAAM emerges as the top performer with an average accuracy rate of 0.949, outperforming other methodologies. This consistency in high performance across different runs for GAAM-based methods, compared to MHA and MHA $\rightarrow$ BN, as well as (bi-)SGA methods, validates the hypothesis of GAAM's effectiveness in the Text modality context.

Gaussian Adaptive Attention Mechanism (GAAM)

In this section, we delve into a more comprehensive analysis of the Gaussian Adaptive Attention Mechanism (GAAM), elucidating its operational intricacies and the pivotal role it plays in enhancing model performance. The GAAM operates on the principle of dynamically adjusting focus on different segments of the input tensor, utilizing learned statistical properties. This adaptive mechanism allows for a nuanced and context-sensitive interpretation of complex data modalities, leading to more accurate and efficient model outcomes. The algorithm for GAAM can be found in listing 1. The algorithm of the Multi-Headed framework and how it utilizes GAAM can be found in listing 2.

Algorithm 1 Gaussian Adaptive Attention Mechanism (GAAM)

Require: x (input tensor), normDimSize, normAxis, c, eps
Ensure: Attention-modified tensor
Initialize **c** to a tensor of size $(1, \text{normDimSize})$ filled with c
Initialize $\mathbf{meanOffset}$ to a tensor of size $(1, \text{normDimSize})$ filled with zeros
for each batch in x **do**
  $\mathbf{mean} \leftarrow \text{mean}(x, \text{dim} = \text{normAxis})$
  $\mathbf{var} \leftarrow \text{mean}(x^2, \text{dim} = \text{normAxis}) - \mathbf{mean}^2$
  $\mathbf{var} \leftarrow |\mathbf{var}| + 10^{-8}$
  $\mathbf{adjustedMean} \leftarrow \mathbf{mean} + \mathbf{meanOffset}$
  $\mathbf{yNorm} \leftarrow (x - \mathbf{adjustedMean})/\sqrt{\mathbf{var} + 10^{-5}}$
  $\mathbf{yTransform} \leftarrow \exp(-(\mathbf{yNorm}^2/(2 \cdot \mathbf{c})))$
  $x \leftarrow x \cdot \mathbf{yTransform}$
end for
return x

Parameters and their Roles

- x (Input Tensor): The data on which the attention mechanism is applied.
- **normDimSize**: Specifies the size of the dimension for normalization.
- **normAxis**: Determines the axis of x along which the mean and variance are computed.
- c: A learnable parameter controlling the scale of Gaussian normalization.
- **clear numbers**: A small constant for numerical stability.

Learning Aspects:

- **c** and $\mathbf{meanOffset}$ are learnable parameters adjusting the normalization's impact and bias.

Operational Steps:

1. Compute mean and variance along **normAxis**.
2. Apply Gaussian normalization to each feature, re-weighted by learned parameters.
3. The output is the input tensor modulated by the learned attention.

Algorithm 2 Multi-Head Gaussian Adaptive Attention

Require: x (input tensor), normDimSize, numHeads, normAxis, c
Ensure: Concatenated attention output tensor

Initialize an array $\mathbf{attentionHeads}$ of size numHeads
for $i = 1$ to numHeads **do**
  $\mathbf{attentionHeads}[i] \leftarrow$ GAAM(**normDimSize**, **normAxis**, **c**, **eps**)
end for
Initialize an empty list $\mathbf{outputs}$
for each $\mathbf{head}$ in $\mathbf{attentionHeads}$ **do**
  $\mathbf{headOutput} \leftarrow$ apply($\mathbf{head}, x$)
  Append $\mathbf{headOutput}$ to $\mathbf{outputs}$
end for
$\mathbf{output} \leftarrow$ concatenate($\mathbf{outputs}$, dim = normAxis)
return $\mathbf{output}$

Multi-Head Gaussian Adaptive Attention

GAAM applies multiple instances of GAAM in parallel, allowing simultaneous focus on different parts or aspects of the data.

Parameters and their Roles in Multi-Head GAAM:

- x (Input Tensor): The data for multi-head attention application.
- **normDimSize**: Size of the normalization dimension.
- **numHeads**: Number of parallel GAAM mechanisms.
- **normAxis**: The axis for normalization.
- c: The scale parameter for Gaussian normalization, shared across heads.

Learning Aspects:

- Each head in $\mathbf{attentionHeads}$ applies GAAM independently.

Operational Steps:

1. Initialize and apply multiple GAAM instances to the input.
2. Concatenate the outputs from each head along **normAxis**.
3. The result is a combined attention output, integrating multiple focus areas.

6 Multi-Head Gaussian Adaptive Attention Mechanism Extension

This section presents an extension of the Multi-Head Gaussian Adaptive Attention Mechanism (GAAM), focusing on enhancing the stability of the training process and the model's efficiency by reducing the number of learnable parameters. The proposed method integrates multi-head attention mechanisms with Gaussian mixtures and skip connections to provide a more refined and adaptable approach to handling complex datasets.

The extended GAAM incorporates multiple attention heads, each with its Gaussian mixture model, to process different segments of the input tensor in parallel. This approach allows for a more diverse and comprehensive understanding of the data, leading to increased model robustness and efficiency.

6.1 Algorithmic Details

Gaussian Adaptive Attention

This algorithm which forms the core of the extended GAAM, implementing the Gaussian mixture model within each attention head can be found in listing 3. Key elements include initialization of Gaussian parameters and mean offsets, and a forward pass handling.

Algorithm 3 Gaussian Adaptive Attention Mechanism

Require: x (input tensor), normAxis, numGaussians, eps
Ensure: Attention-modified tensor

Initialize an array **meanOffsets** of size numGaussians
Initialize an array **c** of size numGaussians
mean $\leftarrow$ mean(x, axis = normAxis)
var $\leftarrow$ var(x, axis = normAxis) + eps
mixture $\leftarrow 1$
for $i = 0$ to numGaussians $- 1$ **do**
  adjustedMean $\leftarrow$ mean + **meanOffsets**[i]
  yNorm $\leftarrow$ (x − adjustedMean)$/\sqrt{\text{var}}$
  gaussian $\leftarrow \exp(-(\text{yNorm}^2)/(2\cdot \mathbf{c}[i]^2))/\sqrt{2\pi\mathbf{c}[i]^2}$
  mixture $\leftarrow$ mixture · gaussian
end for
Normalize mixture across normAxis
x $\leftarrow$ x · yTransform
return x

Multi Head Gaussian Adaptive Attention

This part of the framework remains the same as before (refer to listing 2).

Gaussian Block

The Gaussian Block (listing 4) encapsulates multiple layers of Multi-Head Gaussian Adaptive Attention, each layer processing the input tensor independently integrating a skip connection between them to enhance stability during training and inference.

Algorithm 4 Gaussian Block

Require: x (input tensor), normAxes, numHeads, numGaussians, paddingValue, eps
Ensure: Final modified tensor

for each layer in MultiHeadGaussianAdaptiveAttention **do**
  $x \leftarrow$ layer(x) + x
end for
return x