ISSN 1870-4069

Transformer Architecture Reconfiguration Using Heterogeneous Attention

Xamanek Martínez-Marín, José Adan Hernández-Nolasco,
Noel Zacarias Morales

Universidad Juárez Autónoma de Tabasco,
División Académica de Ciencias y Tecnologías de la Información,
México

xamanekmtz@gmail.com, adan.hernandez@ujat.mx,
n1.zkmt@gmail.com

Abstract. Each attention mechanism in neural network architectures specializes in capturing specific and relevant aspects of input data sequences, allowing the model to focus on the most significant parts for the task at hand. However, this focus is performed in a "homogeneous" manner; that is, the model applies a uniform approach across all training epochs, which can result in output tensors containing incomplete or biased information. This research explores a new direction by proposing the incorporation of various attention mechanisms within a single model to create what we call a "heterogeneous" attention mechanism. For this purpose, heterogeneous attention mechanisms (HAM) A and B are implemented, which make use of “Soft Attention” as a complementary mechanism and integrate the output tensors using “Hadamard Product” and “Element-Wise Addition” respectively. The Vision Transformer (ViT) is used in image classification as a task to evaluate the performance of three models, ViT HAM A, ViT HAM B and ViT Classic. The results show an improvement in convergence time of almost 9.5% and better accuracy of ViT HAM B with respect to ViT Classic. This research opens the door to new methods to improve the efficiency and effectiveness in the training of deep learning models through the use of diversified attention mechanisms.

Keywords: Attention mechanism, neural networks, deep learning, transformer.

1 Introduction

The introduction and continuous improvement of attention mechanisms has led to significant advances in the field of artificial intelligence, laying the groundwork for the development of more advanced architectures, such as Transformers, which rely exclusively on attention mechanisms to efficiently and effectively process data streams. These advances have revolutionized not only machine translation but also a wide range of applications in natural language processing (NLP), computer vision, and beyond, making the attention mechanism one of the most influential innovations in the recent history of machine learning (ML).

Given the computational complexity of the Transformer, the goal is to make efficient use of the resources used for training, so reducing the total training time with minimal hardware is an attractive option. Taking this into account, an architectural improvement process is carried out, proposing a variant of the attention mechanism applied in deep learning research.

The traditional Transformer architecture employs an attention mechanism called "Self Attention," which performs "homogeneous" focusing, that is, it applies a uniform focus across all training epochs, which can result in output tensors containing possibly incomplete or biased information. It is proposed to incorporate various attention mechanisms within a single model to create what we call a "heterogeneous" attention mechanism. The central idea is that, by integrating multiple types of attention that operate in a complementary manner, the model can capture relevant information in a more diverse and complete way during each training epoch. This variability in the focus of attention allows the model to avoid excessive dependence on certain features, thus improving the quality of the representation of the input data. Furthermore, it is proposed that this heterogeneous approach not only improves the model's ability to generalize, but also reduces the overall training time. By capturing relevant information more efficiently in each iteration, the model requires fewer epochs to converge, optimizing the use of computational resources and making the training process more efficient in terms of time and processing.

This study focuses on analyzing the performance of the proposed attention mechanism using the Vision Transformer architecture in image classification tasks using the CIFAR-10 dataset. Evaluation in other visual domains and a comprehensive comparison with traditional convolutional architectures are not considered. The scope is limited to analyzing the model's accuracy, computational efficiency, and ability to visually identify the regions representing the subject of interest in the images within the defined experimental context. Future research could extend this approach to segmentation or object detection tasks in more complex datasets and in other domains such as natural language processing (NLP) tasks.

2 Related Works

The attention mechanism has evolved and diversified into various implementations that have demonstrated different degrees of complexity, efficiency and effectiveness, in 2017 Vaswani et al [1] implemented the Transformer model that makes use of the "Self Attention" attention mechanism to calculate the attention scores of the sequence, it has also been implemented in the Vision Transformer (ViT) in 2020 by Alexey Dosovitskiy et al [2] to do image recognition.

This made "Self Attention" the most famous attention mechanism today, Vaswani introduced it in his work and replaces recurrent networks with this mechanism as the basis for sequential data processing; it has been the basis of current natural language processing models such as GPT, BERT among others. Various works have been derived from the transformer that have sought to make Vaswani's Transformer more efficient, for example the Linformer [3] that reduces the complexity to O(n) by approximating the attention matrices, the Longformer [4] being very similar to the standard Transformer introduces a combination of local and global attention to reduce the complexity to O(n) in long sequences, the Performer [5] makes use of an approximation technique based on kernel functions called FAVOR+ (Fast Attention Via positive Orthogonal Random features) with which it reduces the complexity of the attention calculation to O(n).

3 Methodology

3.1 Heterogeneous Attention Mechanisms

The proposed mechanism is based on the premise that multiple viewpoints can provide a more robust and effective solution to a problem than a single approach could offer. In the context of attention mechanisms within neural networks, this idea translates into the integration of different types of attention, such as "soft attention" [6], "global attention" [7], "local attention" [7], and so on, which act as complementary perspectives. These perspectives support the "self-attention" mechanism, allowing the model to not only focus on global relationships within the input sequence, but also capture finer details and contextual nuances that could be essential for a more complete and accurate understanding of the data.

Vision Transformer. A ViT is implemented with the following characteristics in its architecture and as shown in Fig 1 (Classic ViT Diagram).

3.2 Heterogeneous Attention Mechanism A (HAM A)

This mechanism implements "soft attention" (See Fig 2) as a complementary mechanism to "self attention" and integrates them into one using Hadamard Product [8], which consists of multiplying each element of the first tensor with the corresponding element of the second tensor at the same position, this operation is usually denoted as:

$$C = A \odot B,$$

where:

$$A, B \in \mathbb{R}^{m \times n},$$
$$C_{ij} = A_{ij} \cdot B_{ij}.$$

Hadamard product does not mix cross-information between different positions, it only locally affects each element.

The incorporation of the attention mechanism is done within the attention head which is within the Multi-Head Head (MHA) which is found within each of the four blocks that make up the Encoder and at the end of which the "soft attention" output tensor is integrated with the "self attention" output tensor.

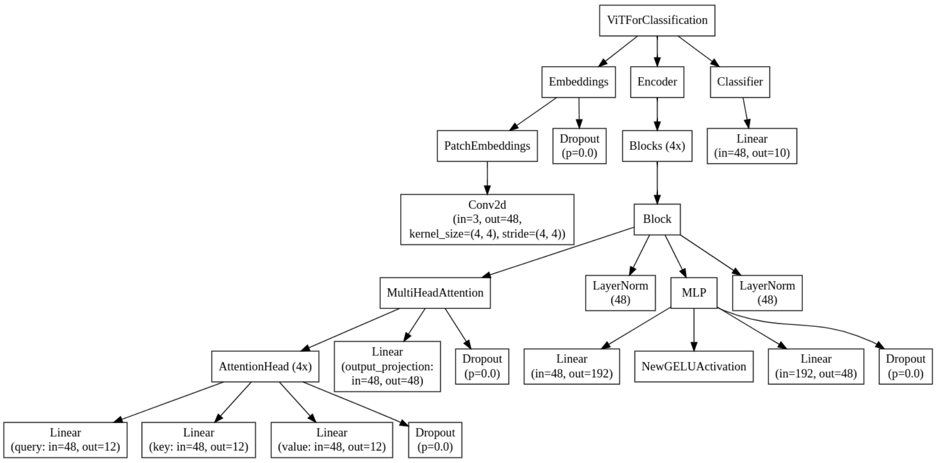

Fig. 1. Structural diagram of the Classic Vision Transformer (ViT) model used for image classification tasks.

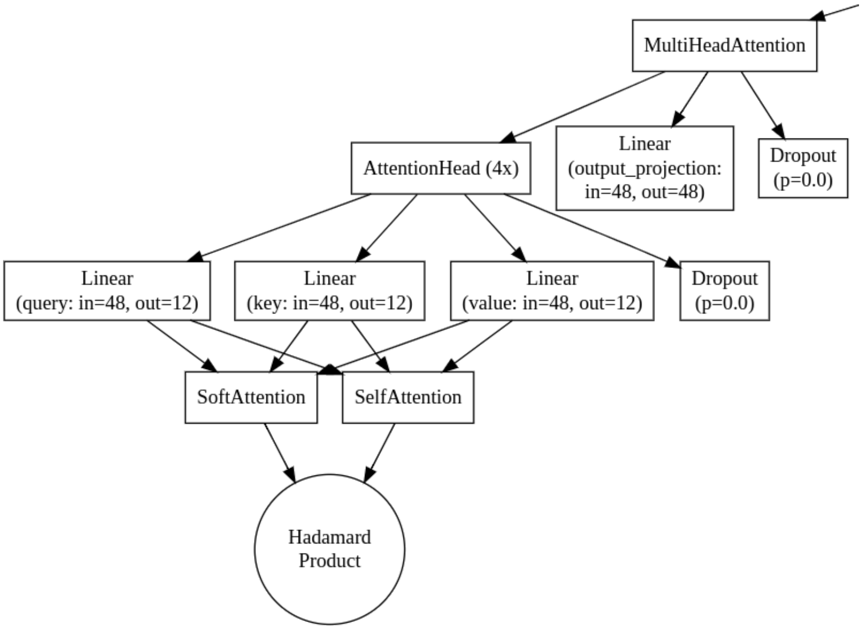
MultiHeadAttention
AttentionHead (4x)
Linear (output_projection: in=48, out=48)
Dropout (p=0.0)
Linear (query: in=48, out=12)
Linear (key: in=48, out=12)
Linear (value: in=48, out=12)
Dropout (p=0.0)
SoftAttention
SelfAttention
Hadamard Product

Fig. 2. Structural diagram of the Vision Transformer HAM A (ViT) model used for image classification tasks.

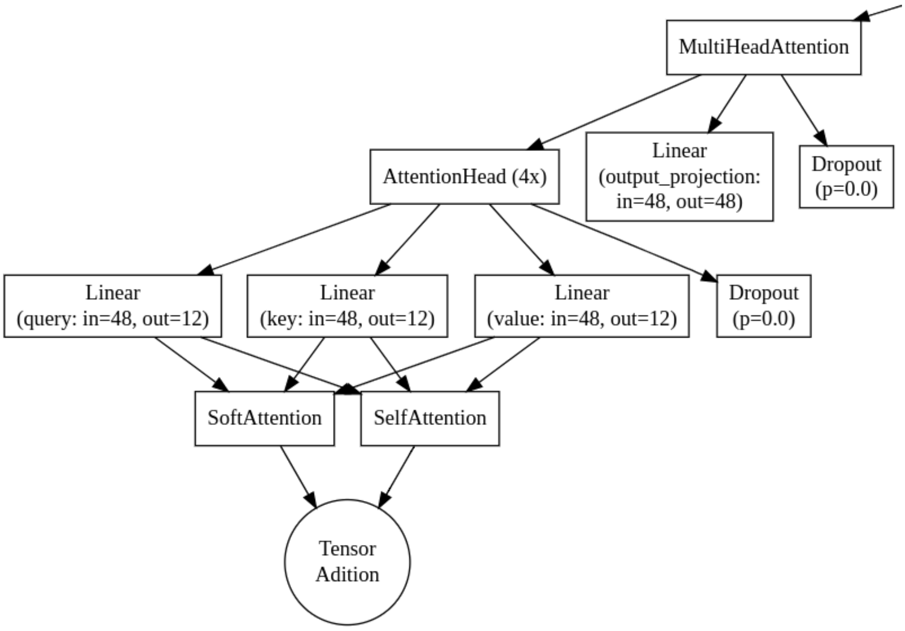
MultiHeadAttention
AttentionHead (4x)
Linear (output_projection: in=48, out=48)
Dropout (p=0.0)
Linear (query: in=48, out=12)
Linear (key: in=48, out=12)
Linear (value: in=48, out=12)
Dropout (p=0.0)
SoftAttention
SelfAttention
Tensor Adition

Fig. 3. Structural diagram of the Vision Transformer HAM B (ViT) model used for image classification tasks.

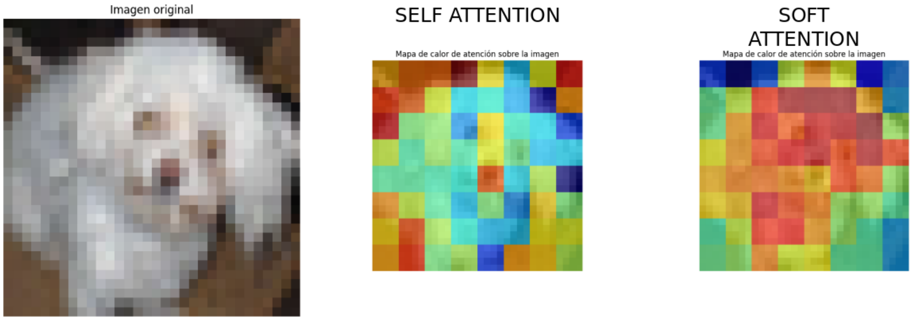

Fig. 4. Comparison between the original image (CIFAR-10) and attention maps of each mechanism separately.

3.3 Heterogeneous Attention Mechanism B

This mechanism implements "soft attention" (See Fig 3) as a complementary mechanism, but integrates them with a one-to-one addition of tensors (Element-Wise Addition) [9], which is a mathematical operation in which two tensors of the same size are added position by position, each element of the resulting tensor is the sum of the corresponding elements of the original tensors.

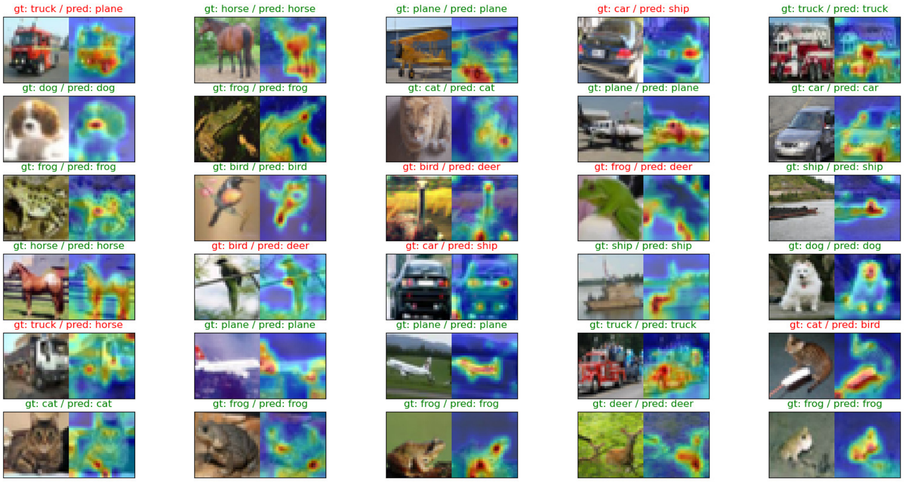

Fig. 5. Heat map generated by the attention captured by the Classic ViT model after training.

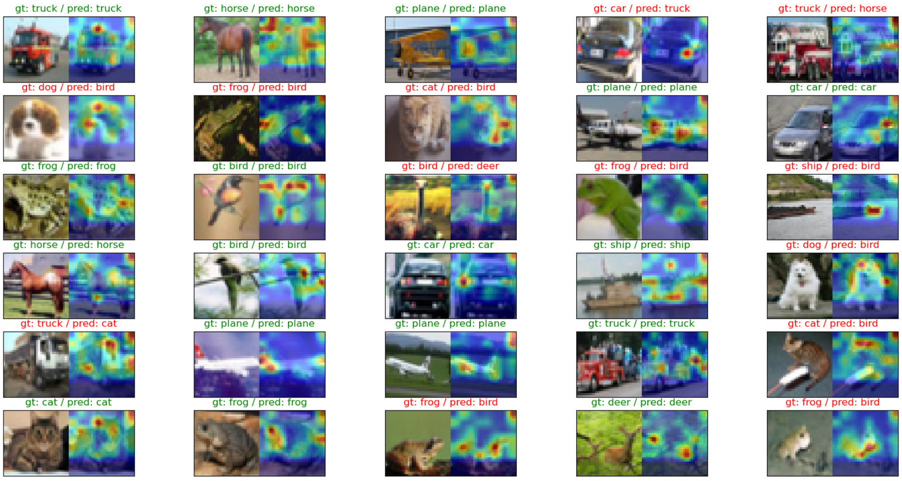

Fig. 6. Heat map generated by the ViT HAM A model's captured attention after training.

Given two tensors A and B of the same shape $m \times n$ their element-by-element sum is defined as:

$$C_{ij} = A_{ij} + B_{ij},$$

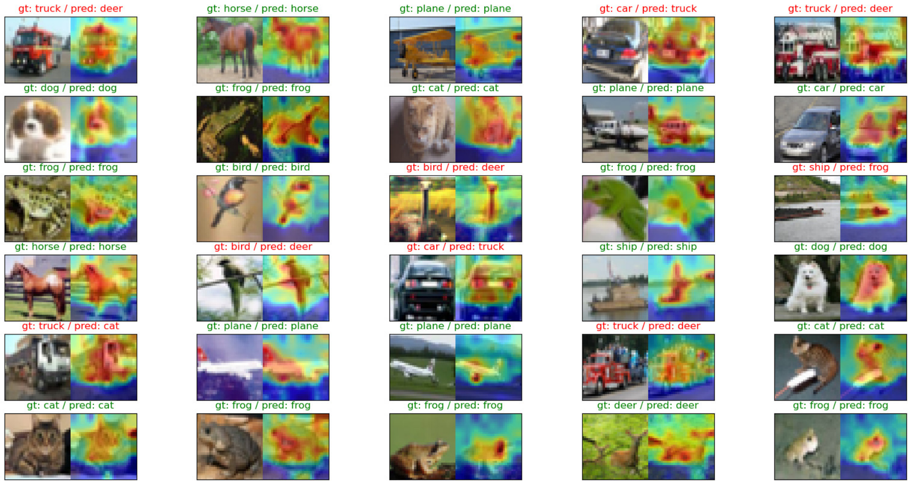

Fig. 7. Heat map generated by the ViT HAM B model's captured attention after training.

the result C is also a tensor of shape $m \times n$. As can be seen in Fig 3, the configuration of the ViT architecture is the same as the classical model and model A, with the difference being the integration of the output tensors of both mechanisms.

3.4 Pytorch

For the implementation and training of ViT, Pytorch (v2.3.0) is used, which is an open source framework for numerical computing and machine learning developed by Meta AI Research [10], Pytorch allows defining DL models such as ViT from scratch or using predefined models, it is possible to modify the number of classes, size of the patches, number of layers, etc. It allows the preprocessing of images by transforming them into the appropriate format, resizing, conversion to tensors and normalization.

Using built-in tools like "DataLoader" and "Dataset" allows you to load images in mini-batches and apply automatic transformations during training. It also enables the ability to define a loss function and an optimizer. It also allows us to use specialized hardware (GPUs) and local distributed computing (multiple GPUs or multiple computers on a network).

For this case, a ViT was implemented from scratch, Figs 1, 2 and 3 were created from the output of the "print(model)" command.

3.5 Dataset

For the image classification task in which the training and subsequent evaluation of the trained models was carried out, the CIFAR-10 Dataset is used, which was developed by Alex Krizhvsky, Vinod Nair and Geoffrey Hinton [11] at the University of Toronto as part of the deep learning in computer vision project. This dataset has a total of 60,000 images that are grouped into 10 classes with a total of 6,000 images per class, at a resolution of 32 x 32 pixels and 3 channels (RGB), the storage format used was the binary matrix.

The dataset is divided into 50,000 images for training and 10,000 for testing, each set of images from all classes with a uniform distribution. The 10 object classes are "airplane", "automobile", "bird", "cat", "deer", "dog", "frog", "horse", "ship", "truck", the dataset presents an intermediate difficulty given the small image size with a lot of background noise and visual variability, this represents a reasonable challenge for simple models and a good benchmark for modern architectures such as ViT.

3.6 Hyperparameters

These are external configurations to the model that are not directly learned during training, but control the behavior of the learning process. Unlike model parameters (such as layer weights), hyperparameters must be defined before training and affect both the architecture and the dynamics of the learning process.

For ViT training, an optimized architecture was configured to operate with low-resolution images, specifically the CIFAR-10 Dataset, whose dimensions are 32 x 32 pixels and contain three channels corresponding to the RGB format. The implemented architecture is characterized by a patch size of 4 x 4, which allows each image to be represented by a sequence of 64 tokens. The size of the hidden embedding was set to 48 dimensions, and the model was structured with four layers of multi-head attention blocks, each with four parallel attention heads. Inside each block, a multi-layer perceptron was used with an intermediate dimension of 192 nodes, which corresponds to four times the size of the embedding. The GELU activation function was used, commonly used in Transformer-type models due to its smoothness and nonlinear modeling capacity (See Table 1).

Regarding image preprocessing, standard computer vision transformations were applied to enhance the model's generalization capabilities. The images were resized and subsequently subjected to random crops varying in both scale and aspect ratio, followed by random horizontal mirroring with a 50% probability. These techniques, which are aligned with data augmentation strategies, seek to simulate natural variations in the data without altering its semantics. Finally, the tensors were normalized using the mean and standard deviation of the CIFAR-10 set, which contributes to stabilizing the learning process by centering the distribution of the input values (See Table 2).

This set of hyperparameters and transformations was selected to balance the model's representational capability with computational efficiency, making training feasible in resource-limited environments without sacrificing the model's ability to capture relevant patterns in the data.

The model was trained for 400 epochs using the AdamW optimizer, known for its effectiveness in combining weight decay techniques with adaptive update steps. The learning rate was set to 0.01, a value that, although high in absolute terms, can be adequately managed using learning programming strategies such as progressive decay or warm-up schemes, although neither was used in this case. A batch size of 256 samples per iteration was used, which favors a more stable estimation of the gradients, especially in the presence of large-batch regularization techniques.

4 Results

In Fig 4 we can observe a visual comparison between the original input image randomly extracted from the CIFAR-10 dataset and the attention maps generated by ViT in a single step using a separate attention mechanism.

Table 1. Hyperparameters used for training the Vision Transformer model.

Hyperparameter	Value (applied style)
Image size	32×32
Input channels	3
Number of classes	10
Patch Size	4
Embedding Size	48
Layers	4
Attention Heads	4
Intermediate MLP size	192
Attention Dropout	0.0
Hidden Dropout	0.0
Initialization of weights	0.02
Activation function	GELU
Learning rate	0.01
Batch Size	256
Epochs	400
Optimizer	AdamW

In the Self Attention map, each token (image patch) directly attends to every other token using learned attention weights, allowing for global interactions. Regions highlighted in red are primarily boundaries of the dog's body or very sharp contrasts (such as a dark background versus a white coat); these contrasts can usually be detected as useful discriminants and therefore receive high attention weights. Unlike convolutional neural networks (CNNs) [12], ViTs do

Table 2. Transformations applied during image preprocessing.

Transformation	**Parameters (applied style)**
Random Crop	Tamaño: 32, Padding: 4
Random Horizontal Flip	Probability: 0.5
Random Resized Crop	Scale: (0.8, 1.0), Ratio: (0.75, 1.333)
Resize	32 × 32
ToTensor	—
Normalize	Mean: (0.4914, 0.4822, 0.4465), Standard Deviation: (0.2023, 0.1994, 0.2010)

Table 3. Conceptual comparison between attention mechanisms in Vision Transformer.

Criterion	**Self-Attention**	**Soft Attention**
Spatial dispersion	*High, with peaks at the edges*	*Low, concentrated in the center*
Multiple foci of attention	*Yes, can serve multiple regions*	*No, primarily unimodal focus*
Contextual Flexibility	*Global, allows full interaction between tokens*	*Partial, with focus on a dominant region*
Interpretability	*Complex but rich in spatial relationships*	*Clear, with easily interpretable localized focus*
Observed pattern	*Dispersed attention including periphery*	Focused circular attention with smooth decay

Table 4. Comparative results of accuracy and loss in training and testing for three variants of the ViT model.

Name	**Best Accuracy & epoch**	**Last Accuracy**	**Best Train Loss & epoch**	**Last Train Loss**	**Best Test Loss & epoch**	**algo borre**
ViT Classic	**0.7977** **Epoch 377**	**0.7902** **Epoch 400**	**0.4863** **Epoch 390**	**0.5011** **Epoch 400**	**0.6123** **Epoch 377**	**0.6156** **Epoch 400**
ViT HAM A	**0.7864** **Epoch 393**	0.7723 Epoch 400	0.5275 Epoch 398	0.5424 Epoch 400	0.6165 Epoch 393	0.6641 Epoch 400
ViT HAM B	**0.8107** **Epoch 342**	0.7693 Epoch 400	0.5015 Epoch 396	0.5130 Epoch 400	**0.5558** **Epoch 342**	0.6882 Epoch 400

not have an inductive bias towards local structures, meaning that the network does not automatically favor the center or assume that nearby regions are related. Therefore, it is perfectly plausible that distant areas can receive high attention if they help build a useful representation.

It is emphasized that this attention map corresponds to a single step without additional processing; these early layers tend to learn low-level patterns or global geometric structure. The attention to edges reflects that at this stage, the network has not yet refined its focus toward the central semantic region (the dog's face). This is a natural behavior of the "Self Attention" mechanism.

The "Soft Attention" attention map (right in Fig 4) weighs different regions of the same image, with attention strongly concentrated in the center of the image, particularly in the areas of the dog's face, which includes the snout, eyes, and part of the forehead. Unlike the "Self Attention" attention map, here we observe a gradual and symmetrical decrease in attention towards the edges of the image; the corners and margins present cooler colors, indicating that the model does not consider them "informative" in this first and only step. The "circular" shape of the attention is reminiscent of a "Gaussian window" type approach centered on the region of greatest visual density (representative of the subject in the image).

This behavior is due to the design of the attention mechanism, which generally consists of a normalized distribution of weights (via Softmax) applied to spatial or embedded features, which forces attention to focus strongly on a few regions and quickly decay outside of them.

This map reflects how "Soft Attention" prioritizes the central region of the image (where relevant semantics are located) in a more direct and localized way than "Self Attention." This central attention will remain constant across multiple layers, reinforcing classification based on a salient area (and reinforcing the attention captured by "Self Attention").

Table 3 shows a conceptual comparison between both attention mechanisms according to the previous analysis.

In Table 4 we have the ViT HAM B implementation reaching a maximum accuracy of 0.8107 at epoch 342 being higher than the other two ViT implementations, indicating that HAM B can capture patterns more effectively at some point during its training, this makes it the most accurate model during this stage making it a more efficient implementation in terms of convergence, reaching its peak performance before Classic and HAM A, however we see a considerable drop in the last epoch indicating overfitting at the end of training.

The attention shown by the heat maps in figures 5, 6 and 7 indicate that the "HAM B" model focuses more globally to capture image features, suggesting that it has a better ability to capture specific details or patterns in some examples, reflected in its better accuracy in some categories. In the images we observe a better focusing on the correct areas (for example, in the "frog" and "bird" images) this is a sign that the implemented attention mechanism has a better performance in terms of prioritizing the most relevant features in those classes.

5 Conclusions

The poor performance of the HAM A implementation compared to the other two models is mainly due to the fact that tensor integration using the Hadamard Product is not reversible, meaning that different combinations of values can give the same product (e.g., $2 \odot 3 = 6$, as well as $1 \odot 6$). Information is lost if one of the elements is zero, which causes the contribution of the other tensor to be completely lost, and finally, it does not preserve the original scales of the tensors.

The overfitting present in the HAM B model in its last epoch can be mitigated using several techniques in this type of tasks, starting by adding more "data augmentation", other transformations that were not initially included such as Vertical Flip, CutMix or MixUp can be included, this serves to create greater diversity in the input data to reinforce the model to generalize. Implementing regularization in the model can also help mitigate overfitting, such as "dropout" in layers within the MLP, the Stochastic Depth (DropPath) that prevents all layers from being activated in each training step, Weight Decay with L_2 penalty to the size of the weights in the optimizer.

Given the positive results, the HAM B implementation demonstrates that complementary attention mechanisms help capture additional features to converge more quickly in this case by incorporating the inherent nonlinearity of the soft attention mechanism to capture more complex features of the input data to calculate output attention scores. Further testing and implementation of other attention mechanisms are necessary to achieve more comprehensive and satisfactory results.

Acknowledgments. This research was not funded by private entities. The work was conducted as part of graduate studies with support from the National Program for Quality Graduate Studies (PNPC) of the National Council for Humanities, Science, and Technology (CONAHCYT) in Mexico.

References

1. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30, pp. 5998–6008 (2017)
2. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby: An Image is Worth 16x16 Words. Transformers for Image Recognition at Scale, arXiv preprint arXiv:2010.11929 (2020)
3. Wang, S., Zhou, B., Jiang, J., Chen, Y.: Linformer: Self-Attention with Linear Complexity. arXiv preprint arXiv:2006.04768 (2020). `https://doi.org/10.48550/arXiv.2006.04768`
4. Beltagy, I., Peters, M.E., Cohan, A.: Longformer: The Long-Document Transformer. arXiv preprint arXiv:2004.05150 (2020). `https://doi.org/10.48550/arXiv.2004.05150`
5. Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., et al.: Rethinking Attention with Performers. In: International Conference on Learning Representations (ICLR), arXiv preprint arXiv:2009.14794 (2021). `https://doi.org/10.48550/arXiv.2009.14794`
6. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2015). `https://doi.org/10.48550/arXiv.1409.0473`
7. Luong, M.-T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1412–1421 (2015). `https://doi.org/10.18653/v1/D15-1166`
8. J.-H. Kim, K.-W. On, W. Lim, J. Kim, J.-W. Ha, and B.-T. Zhang: Hadamard Product for Low-Rank Bilinear Pooling. In: Proceedings of the 5th International Conference on Learning Representations (ICLR) (2017) [Online]. Available: `https://openreview.net/pdf?id=r1rhWnZkg`
9. K. He, X. Zhang, S. Ren, J. Sun:Deep Residual Learning for Image Recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016)

10. Meta AI: PyTorch Foundation: A New Home for the Open Source AI Framework. Meta AI Blog (2022) [Online]. Available: `https://ai.meta.com/blog/pytorch-foundation/`
11. A. Krizhevsky: Learning Multiple Layers of Features from Tiny Images. Technical Report, University of Toronto (2009) [Online]. Available: `https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf`
12. A. Krizhevsky, I. Sutskever, G. E. Hinton: ImageNet Classification with Deep Convolutional Neural Networks. In: Advances in Neural Information Processing Systems, vol. 25, pp. 1097–1105 (2012)
13. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2021). `https://doi.org/10.48550/arXiv.2010.11929`