PCAM: Parametric Channel Attention Module

Abstract

We introduce the Parametric Channel Attention Module (PCAM), a lightweight architectural enhancement designed to improve attention mechanisms in convolutional neural networks. PCAM replaces fixed design assumptions in prior modules by (i) learning a data-driven fusion weight for global average and max pooling, and (ii) employing a parametric sigmoid activation to adaptively control attention sharpness. We verify that these modifications promote more stable gradient flow and flexible feature recalibration. Evaluated on CIFAR-10 and STL with ResNet-18, PCAM achieves lower classification error and reduced gradient norm variance compared to CBAM and baseline models. We further conduct a comprehensive analysis of six activation functions within attention modules, showing that parametric sigmoid yields improved performance and more balanced attention distributions. PCAM is lightweight, end-to-end trainable, and readily applicable to existing architectures.

Keywords: image classification, attention mechanism, gated convolution

1 Introduction

Convolutional Neural Networks (CNNs) have achieved strong performance in visual recognition by learning hierarchical features through local filters [12]. However, their inherent locality restricts global context modelling [5], which is essential for holistic reasoning and resolving fine-grained details. Attention modules such as Squeeze-and-Excitation [14] rescale channels via global pooling, and CBAM adds spatial gating [28]. Both assume fixed average and max pooling fusion and use a static sigmoid that can saturate early, causing potential vanishing gradients. Despite their success, these modules rely on fixed pooling and static sigmoid activations, which may limit their expressivity and learning.

Recent work has questioned these design choices. Fixed sigmoid gating can saturate, impeding gradient flow [29]. Learnable gating functions [25] and adaptive pooling combinations [19] have shown improvements in both accuracy and training stability.

In this work, we propose the Parametric Channel Attention Module (PCAM), which learns a per-block scalar α to adaptively balance global average and max-pooled channel descriptors. Integrated into ResNet-18, we found that PCAM reduces Top-1 error by 0.45% over CBAM on CIFAR-10 and STL-10 datasets, lowers gradient norm variance by 0.8%, and yields more interpretable attention maps.

1.1 Our Contribution

We perform an empirical comparison of six activation functions, sigmoid, parametric sigmoid, swish, scaled tanh, softmax and sparsemax, within both channel and spatial attention modules, evaluating their effects on classification accuracy, gradient-norm stability and attention-map interpretability across CIFAR-10 and STL-10 datasets. We show that applying a parametric sigmoid in the spatial-attention branch yields the best overall performance.

We further introduce a learnable pooling-fusion weight α in the channel-attention branch and demonstrate that combining this learned fusion with parametric sigmoid gating outperforms both CBAM and a vanilla ResNet-18 baseline.

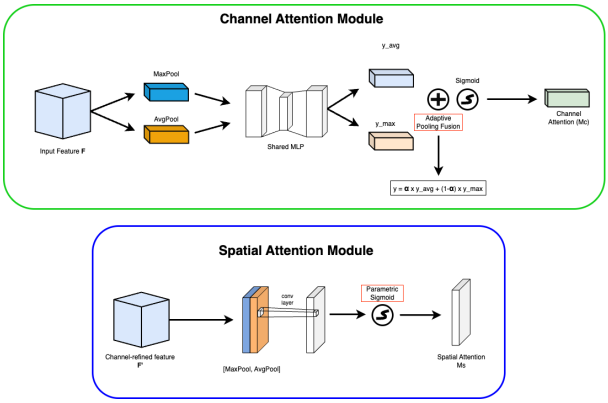


Figure 1: PCAM architecture. Channel attention fuses pooled descriptors via a learnable weight and applies sigmoid gating. Spatial attention uses a learnable-temperature sigmoid for adaptive focus.

2 Related Work

2.1 Convolutional Attention

Convolutional neural networks (CNNs) are known for extracting local patterns via fixed-size kernels well, however, this locality limits their ability to model global context [5], which is critical for tasks requiring holistic reasoning and for clarifying fine details. Convolutional representations are opaque, limiting per-feature interpretability, such that although we know that a CNN can make accurate predictions, we rarely know why on a per-feature basis. Originally from sequence models [27], attention has been added to CNNs to enable global receptive fields and reveal where the network focuses, improving transparency of a model's internal reasoning.

The Squeeze-and-Excitation (SE) block was introduced to enhance a network's representational power by explicitly modelling the relationships between its feature-map channels [14]. In practice, an SE block "squeezes" each channel's spatial information via global average pooling, then applies two fully-connected layers and a sigmoid activation to produce per-channel scaling factors. These factors 'recalibrate' the original feature-maps, amplifying channels that carry discriminative information and suppressing those that do not. By making channel responses input-dependent, SE blocks allow convolutional layers to adapt their feature detectors to each example, yielding richer and more discriminative representations than static filter outputs.

Extending beyond channel recalibration, Woo et al. (2018) [28] proposed the Convolutional Block Attention Module (CBAM), which refines both what to attend (channels) and where to attend (spatial locations). They observe that average-pooled statistics capture overall activation trends but can miss "peak activations" (the single strongest responses) that often correspond to the most salient features, whereas spatial context (which is the arrangement of activations across H×W) governs precise localisation. Hence, average pooling conveys overall trends while max pooling highlights the most salient features. A shared two-layer MLP processes both pooling streams, and their outputs are merged to yield channel attention weights. Recognizing that knowing which channels to amplify is insufficient without where in the spatial map those channels should focus, CBAM's spatial branch pools across channels using both average and max operations to form two single-channel maps, concatenates them, applies a 7×7 convolution for broad context, and then a sigmoid to produce a soft mask that highlights salient regions. This shows that CBAM consistently improves top-1 accuracy by over 1% on ImageNet-1K relative to SE-only ResNet-50, demonstrating that jointly modelling both channel interdependencies and spatial contexts yields richer feature refinements.

Bello et al. (2020) [5] subsequently proposed Attention-Augmented Convolutional Networks. This fuses 2D relative self-attention with convolution, preserving equivariance while capturing global context. From the attention branch, each position can now see and integrate information from the entire image (a much larger receptive field). Following this approach, the authors show that augmenting a ResNet-50 architecture in this way produces a 1.3% top-1 accuracy boost, confirming that sequence-style self-attention can serve as a foundation within CNNs.

More recently, channel-attention methods have evolved to learn optimal combinations of pooling cues rather than assume equal weighting. Li et al. (2022) [19] proposed Fused Max-Average Attention (FMAttn), which concatenates global max and average pooled descriptors and passes them through a convolution that learns how best to fuse these features. This richer channel summary captures both global trends and salient activations, improving on SE-style gating and yielding up to 2.21% accuracy gains on CIFAR-100 and ImageNet-100.

Motivated by these advances, we integrate lightweight attention into ResNet-18 and evaluate its impact on performance and gradient flow, aiming to understand how attention mechanisms can enhance CNNs

2.2 Activation Functions

Activation functions introduce the nonlinear transformations that enable deep networks to model complex input-output relationships. In the context of attention modules, they serve as gates, converting pooled feature descriptors into weighting coefficients that modulate feature importance.

In lightweight attention modules such as SENet [14] and CBAM [28], the sigmoid activation function has become the default choice for converting pooled descriptors into attention weights. Its bounded output in (0,1) and computational simplicity make it ideal for gating feature channels. However, recent work has called into question the universality of the standard sigmoid gate, demonstrating that both the form and parameters of the activation function can materially influence attention distributions, convergence behaviour, and final performance.

Ying et al. (2021) [29] introduced the Parametric-Sigmoid SE (PSE) block, in which the fixed sigmoid in the original SE module is replaced by a learnable "temperature" parameter. This allows the network

to dynamically adjust the sigmoid function's slope during training. Across a variety of image sets, like CIFAR-10/100 and Tiny-ImageNet, PSE consistently outperformed its standard-sigmoid counterpart, confirming that a trainable activation can enhance the expressivity of channel attention.

More recently, self-gated functions like Swish

$$\text{Swish}(x) = x \cdot \sigma(\beta x)$$

blend identity mapping with sigmoid gating, retaining small negative outputs and offering a non-monotonic, smooth profile that often yields faster convergence and higher accuracy than both ReLU and vanilla sigmoid in attention contexts [25].

Misra (2020) introduced the Mish activation.

$$\text{Mish}(x) = x \cdot \tanh(\ln(1 + e^x))$$

This demonstrates a 0.4% increase in top-1 accuracy on CIFAR-100 when replacing ReLU or Swish in SE-ResNet18. Alhazmi and Altahhan (2020) [2] conducted a comparison of ReLU, ELU, and a custom "ELU+" in ResNet50 models with CBAM and BAM modules for facial expression recognition. They found that ELU+ improved performance significantly, with accuracy gains of up to 30%, implying that the shape and slope of activation functions can critically affect the learning within attention layers.

Chen et al. (2020) [7] found that sigmoid gating in CNN-Transformer hybrids yielded more flexible attention maps than softmax. As softmax enforces a strict sum-to-one constraint, it can impose unwanted competition among features, whereas sigmoid allows parallel emphasis across multiple regions. Martins et al. (2021) [22] subsequently introduced the sparsemax activation, which produces exact zeros in its output, thereby concentrating attention on a small subset of salient features. While sparsemax enhances interpretability by yielding sparse attention distributions, the authors caution that excessive sparsity can fragment the attention map and destabilise training, particularly in spatial contexts.

To address the impact of activation-function choice on the structure of channel attention maps, our work analyses channel-attention behaviours under various activations. By plotting attention magnitude against channel index for each activation type, and quantifying metrics such as attention entropy and rank-order stability, we shed light on how the activation shape influences both the focus and diversity of attention.

2.3 Gradient Flow in Deep Networks

In many convolutional-attention modules, such as the SE block [14] and CBAM [28], the network computes attention weights by passing intermediate activations through a sigmoid function, which maps real-valued inputs into the interval (0, 1). However, as these activations saturate toward 0 or 1, the sigmoid's derivative rapidly approaches zero, causing back-propagated gradients through the attention weights to vanish and thereby stalling their learning. This is a classic vanishing-gradient phenomenon that can impair the network's ability to adapt its focus, since attention parameters receive negligible updates once the gating outputs become too extreme, thus limiting overall accuracy. To address this, a parametric sigmoid introduces a learnable temperature parameter, T, that controls the gating curve's steepness. By initialising T to a high value, thereby essentially producing a "soft" sigmoid with substantial slope, early training preserves informative gradients even when outputs stray toward extremes. As confidence in the learned attention patterns grows, T can be gradually reduced, sharpening the sigmoid and yielding more discriminative gating without sacrificing trainability [29].

Empirical studies of temperature-controlled sigmoid gating demonstrate its concrete benefits in combating vanishing gradients and improving attention learning. Ying et al. [29] replaced the standard sigmoid in the SE block with a parametric sigmoid that effectively implemented a learnable temperature and evaluated its impact across multiple architectures and tasks. On CIFAR-10 classification, a ResNet model implemented with the parametric sigmoid SE block saw its training accuracy rise by approximately 0.9%. Hence, modifying the sigmoid's shape to avoid early saturation can reduce the vanishing gradient problem in deep attention blocks and lead to higher recognition accuracy.

3 Implementation Details

3.1 Convolutional Attention

Classical convolutions integrate channel and spatial information indiscriminately within a fixed local receptive field, limiting their ability to capture global context and fine-grained localisation. Squeeze-and-Excitation (SE) networks first addressed inter-channel dependency through global pooling and gating [14]. CBAM then extended this concept to two successive submodules, channel then spatial attention, yielding consistent performance gains across architectures [28]. The following sections detail each component of our enhanced CBAM, building on these concepts.

3.1.1 Channel Attention with Learnable Pooling Weights

In the original CBAM formulation, Woo et al. explain that average pooling softly encodes the extent of the target object, whereas max pooling captures the degree of the most salient part by focusing on peak activations. CBAM's Channel Attention module, therefore, combines Global Average Pooling (GAP) and Global Max Pooling (GMP) equally:

$$\mathcal{M}_c(F) = \sigma(\text{MLP}(\text{GAP}(F)) + \text{MLP}(\text{GMP}(F)))$$

However, depending on the dataset and task, one may be more informative than another. Rather than summing these features equally, to allow the network to adaptively weigh these features, a learnable scalar, $\alpha \in [0, 1]$, now mediates their contribution:

$$\mathcal{M}_c(F) = \sigma\left(\alpha \cdot \text{MLP}(\text{GAP}(F)) + (1 - \alpha) \cdot \text{MLP}(\text{GMP}(F))\right)$$

With α learned end-to-end and initialised to 0.5 to recover the original CBAM when no preference emerges. α is also passed through a sigmoid to ensure bounded outputs. The module applies both GAP and GMP to the input feature map, yielding two descriptors, average-pooling and max-pooling. Each descriptor is processed by a shared two-layer MLP (implemented via 1×1 convolutions with intermediate ReLU), producing two channel-wise vectors which are then fused by α.

This data-adaptive attention fusion allows the network to learn, from the training data itself, the optimal weighting (α) between these two pooling streams—rather than treating them as equally important a priori. Hence, the fusion of GAP and GMP-based descriptors becomes adaptive, driven by back-propagated gradients, such that the resulting attention map reflects the feature statistics most predictive for the task at hand. By learning α, the network transcends CBAM's equal-weight assumption. This modification adds only one trainable parameter per attention block, retains compatibility with CBAM's lightweight design, and enables post-hoc inspection of α to interpret whether global context or peak activation dominates in different network layers. Details are in Appendix 1.

3.1.2 Activation Functions and Their Comparative Analysis

To evaluate the efficacy of activation functions in channel and spatial attention, and to identify the most suitable option for our models, we conducted controlled experiments across six candidate functions: softmax, scaled tanh, parametric sigmoid, swish, sparsemax, and sigmoid. Performance was assessed using top-1, top-5 accuracy, and average precision, with experiments structured to isolate the impact of each function. For channel attention evaluation, spatial activation was fixed to sigmoid, and vice versa, ensuring observed differences stemmed solely from the varied activation under test. Pooled descriptors (average and max) were transformed into attention weights using each candidate function, enabling direct comparisons. To ensure robustness, experiments were conducted under constant hyperparameters (e.g., learning rate, batch size), and explicit documentation of the dataset (e.g., CIFAR-10) and model architecture (e.g., ResNet-18). Activation functions were selected based on theoretical benefits: sparsemax for sparse, focused attention; scaled tanh for tunable output ranges; parametric sigmoid for learnable slope adaptation; and swish for smooth gradient propagation.

Key research questions addressed whether parametric sigmoid, with its learnable slope parameters, could outperform static activations (e.g., standard sigmoid) by adaptively balancing attention sharpness and flexibility. We also explore whether sparsemax's sparsity-enhanced performance is achieved by prioritising salient channels or overly restricted feature integration, and whether swish's continuous gradients improve convergence over sigmoid. Quantitative metrics (top-1/top-5 accuracy for classification, average precision for localisation) were complemented by qualitative analysis of attention maps, revealing how functions modulated weight distributions. For instance, sparsemax produced sharply peaked activations, while swish generated smoother, distributed patterns. To further quantitatively evaluate these features, we added three metrics:

(1) Entropy, quantifying the uncertainty and spread of attention. Lower entropy indicates sharper focus, linked to interpretability in prior work [15]

$$H = -\sum_{i=1}^{N} a_i \log(a_i) \quad \text{where } \sum_{i=1}^{N} a_i = 1$$

(2) Mean Attention Value, reflecting global activation strength

$$\mu = \frac{1}{N} \sum_{i=1}^{N} s_i \quad \text{(raw attention scores)}$$

(3) Focus Area Ratio, measuring spatial concentration of high-attention regions, pixels exceeding 50% of maximum attention, inspired by saliency detection [24].

$$\text{FAR} = \frac{\sum_{i=1}^{N} \mathbb{I}\left(a_i \geq 0.5 \cdot \max(\mathbf{a})\right)}{N}$$

(thresholded at 50% of max attention)

(4) Gini Coefficient: Evaluated inequality in attention weight distribution, where values closer to 1 indicate extreme concentration (e.g., a single dominant channel) and values near 0 reflect uniformity [3].

$$G = \frac{\sum_{i=1}^{N}(2i - N - 1)a_i^{\uparrow}}{N}$$

Sorted ascending weights a_i

More details about activation functions can be found in Appendices 3 and 4.

3.2 Gradient Flow

We track two core metrics to analyse gradient flow during training: the L2-norm of gradients [18] and its variance [10] across epochs. The L2-norm measures the overall magnitude of gradient updates, calculated as the square root of the sum of squared gradients across all trainable parameters, expressed as

$$\text{Gradient Norm} = \sqrt{\sum_{i=1}^{N}(\nabla\theta_i)^2}$$

where $\nabla\theta_i$ denotes the gradient of the i-th parameter. High norms may indicate unstable updates and risk overshooting minima, while low norms can signal vanishing gradients, leading to stalled learning. To assess consistency, we also compute the variance of gradient norms over training epochs, defined as

$$\text{Variance} = \frac{1}{T-1}\sum_{t=1}^{T}(g_t - \bar{g})^2$$

where g_t is the gradient norm at epoch t and $\bar{g}$ is the mean gradient norm. Our implementation involves collecting the L2-norm of gradients after each epoch, logging these values throughout training, and calculating their mean and variance. We visualise trends using line plots (e.g., Fig. 2) and summarise variance values in tables (e.g., Table 2) to compare across models. Healthy gradient flow is characterised by moderate, stable norms and low variance, whereas common pathologies such as vanishing or exploding gradients are reflected by rapidly decaying or spiking norms, respectively.

4 Experiments

The experiments are organised into two stages. First, we conduct extensive evaluations across various activation functions, comparing their performance and investigating the explainability of their learned representations by probing activation maps. Based on these results, we identify the most effective activation functions and then perform ablation studies, comparing our proposed models against CBAM and a vanilla ResNet-18 baseline.

4.1 Ablation studies

In this subsection, we empirically validate the effectiveness of our design choices. We conduct ablation studies on the CIFAR-10 [16] and STL [8] datasets, using ResNet-18 as the base architecture. The CIFAR-10 dataset contains 50,000 training images and 10,000 testing images across 10 object categories. For training, we apply the same data normalisation and resize test images to 224×224 to align with the input size requirements of pretrained models. The learning rate is fixed at 0.0001, and all models are trained for 10 epochs. Following the evaluation protocols of [12], we report classification errors on the validation set.

4.1.1 Image Classification Performance

To evaluate the effectiveness of our proposed module, we conduct classification experiments on both CIFAR-10 and STL, following the protocol outlined in Section 4.1. We integrate PCAM into ResNet architectures and compare it against baseline models.

As shown in Table 1, networks with PCAM consistently outperform the baselines in terms of Top-1 and Top-5 error across both datasets. On CIFAR-10, PCAM also achieves higher Average Precision compared to ResNet and CBAM. However, on STL, although PCAM improves accuracy, it does not surpass the vanilla ResNet in Average Precision. These results show the potential benefit of our design. The learnable temperature in the parameterised sigmoid enables adaptive control over the sharpness of the attention map. Furthermore, the adaptive pooling fusion balances complementary spatial information from average and max pooling, resulting in more discriminative feature representations.

To assess component contributions in PCAM, we performed a 9-configuration ablation study, varying: pooling weights in channel/spatial attention (included/excluded), activation functions (standard vs. parametric sigmoid), and baseline ResNet comparisons. Details are in Appendix 5.

4.1.2 Gradient Flow

To further investigate the impact of PCAM on optimisation, we analysed the gradient L2 norms during training. Figure 2 plots the gradient norms across epochs for Vanilla ResNet, ResNet with CBAM, and ResNet with our proposed PCAM module. We also computed the variance of the gradient norms across epochs (Table 2).

PCAM, extending CBAM, introduces two key modifications: a parametric sigmoid activation and learnable pooling weights that adaptively combine average and max pooling. These enhancements

Datasets	Architecture	Top-1 error (%)	Top-5 error (%)	Average Precision
CIFAR-10	ResNet	5.61	0.15	0.9845
CIFAR-10	ResNet + CBAM	5.94	0.13	0.9845
CIFAR-10	ResNet + PCAM	**5.49**	**0.11**	**0.9856**
STL	ResNet	4.837	0.17	0.989
STL	ResNet + CBAM	4.987	0.175	0.988
STL	ResNet + PCAM	**4.537**	**0.125**	**0.989**

Table 1: Classification performance on CIFAR-10 and STL for ResNet, CBAM, and PCAM (ours). PCAM consistently achieves the lowest Top-1 and Top-5 error rates, with improved or matched average precision across both datasets.

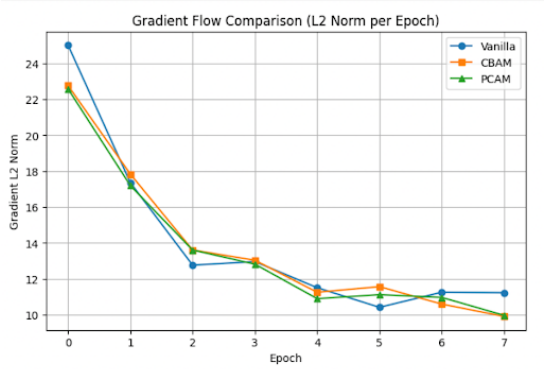


Figure 2: Gradient flow comparison across models on CIFAR-10. All methods show a steep early decline in L2 norm, with PCAM maintaining slightly lower gradients in later epochs, suggesting more stable training.

Model	Gradient Norm Variance
ResNet	23.1853
ResNet + CBAM	16.7743
ResNet + PCAM	**15.9660**

Table 2: Gradient norm variance during CIFAR-10 training.

aim to mitigate gradient saturation and vanishing updates, common issues in conventional attention modules. Prior studies [14] [28] observed that standard sigmoid activations can cause vanishing gradients, as saturation near 0 or 1 reduces the derivative toward zero, stalling the learning of attention weights. To address this, a parametric sigmoid with a learnable temperature T [29] controls the gating curve's steepness: a higher initial T preserves gradient flow during early training, while a gradual sharpening improves discriminative focus as learning progresses.

Our empirical results support these theoretical insights. As shown in Figure 2, PCAM maintains consistently healthy gradient magnitudes throughout training, comparable to or even slightly better than CBAM. More notably, the variance of gradient norms (Table 2) is lowest for PCAM (15.966) compared to CBAM (16.7743) and Vanilla ResNet (23.1853). Lower variance indicates more stable and smooth optimisation, reducing the likelihood of erratic updates and promoting more consistent learning.

These findings suggest that the introduction of a parametric sigmoid in PCAM potentially alleviates vanishing gradient issues within the attention module. The adaptive pooling fusion further contributes to stabilising the learning by allowing the model to flexibly adjust how spatial information is aggregated. Together, these improvements enhance the trainability and representational quality of PCAM, leading to improved classification performance as demonstrated in our CIFAR-10 and STL experiments.

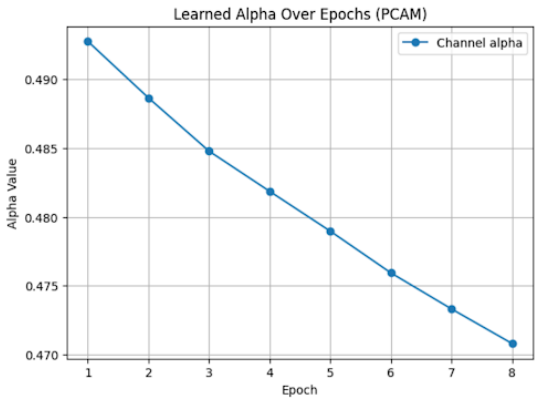


Figure 3: Learned channel alpha values over epochs using PCAM. Alpha steadily decreases, indicating a growing reliance on max pooling relative to average pooling as training progresses.

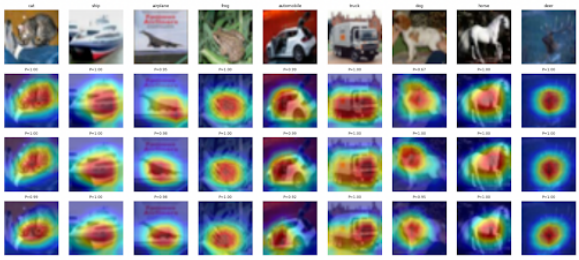
The learned channel alpha parameter reflects the relative emphasis placed on average pooling within the channel component of the PCAM. The observed decline in channel alpha across training epochs suggests a gradual reduction in the reliance on average pooling. Correspondingly, the weight for max pooling, 1- spatial alpha, increases. This suggests a progressive shift toward prioritising max pooling for spatial feature aggregation. This implies that as training advances, the model increasingly favours max pooling to extract spatially discriminative patterns. A plausible explanation lies in the inherent properties of max pooling: by emphasising prominent activations (e.g., edges, textures), it enables the model to focus on more distinctive features during later training stages.

4.1.3 Network Visualisation with Grad-CAM

To interpret how attention mechanisms influence feature localisation, we employ Grad-CAM visualisations on an intermediate convolutional layer, generating class-specific activation maps that highlight regions critical to the model's predictions. Grad-CAM uses gradient flow to weigh spatial locations in convolutional layers, producing heat maps that reflect the network's focus for a given class. By comparing activation patterns across models, we assess how effectively each architecture aggregates discriminative features. We evaluate three variants: Vanilla ResNet-18, ResNet-18+ CBAM, and ResNet-18 + PCAM.

As shown in Fig. 5, the Grad-CAM heatmaps and their corresponding softmax prediction scores reveal distinct behaviours: The baseline and CBAM-integrated models exhibit diffuse attention. CBAM-integrated networks show improved localisation, but activation boundaries remain coarse, occasionally including background clutter. PCAM-integrated networks produce more precise activation maps, tightly enclosing target objects (e.g., dog heads or monkey heads) while suppressing irrelevant regions. While differences are subtle (potentially limited by training epochs), PCAM's channel-weighted parametric sigmoid appears to enhance feature aggregation, suggesting a path toward more interpretable and precise attention mechanisms.

A) Grad-CAM comparisons on the CIFAR-10 image dataset.



B) Grad-CAM comparisons on the STL-10 image dataset.

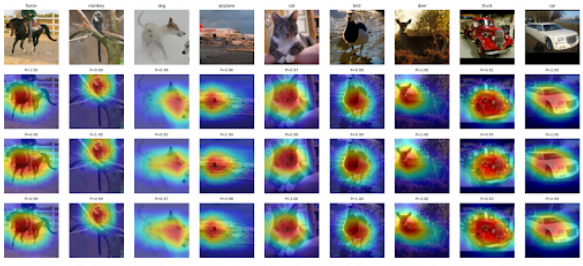


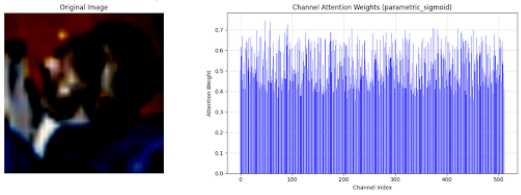
Figure 4: Grad-CAM comparisons of (a) ResNet18, (b) ResNet18+CBAM, and (c) ResNet18+PCAM (ours), showing final-layer activations. Ground-truth labels and prediction scores (P) demonstrate PCAM's superior focus on key features (e.g., dog's head) versus alternatives.

4.2 Activation Maps Probing

Building on the Grad-CAM visualisations that highlighted differences in class-discriminative focus across model variants, we next perform a systematic, quantitative probing of the internal attention maps generated by channel attention modules. While Grad-CAM captures high-level gradients linked to classification decisions, attention maps reveal how intermediate feature representations are modulated throughout the network. This layer-wise probing offers a complementary perspective into how and where attention mechanisms refine feature localisation and shape model behaviour.

A key focus of this analysis is the activation function used within channel attention modules. Activation functions determine how the raw channel descriptors are transformed into normalised attention weights, directly influencing whether attention is distributed smoothly or sparsely. While most existing works (e.g., [28]) rely on the standard sigmoid due to its simplicity and bounded output, recent studies (e.g., [25]; [19]) suggest that adaptive and learnable alternatives may yield better representational control. To isolate the effect of activation choice, we evaluated six activations: standard sigmoid, parametric sigmoid, swish, scaled tanh, softmax, and sparsemax, replacing only the attention module's activation while keeping the rest of the network fixed with the default sigmoid. The goal is to disentangle how different activation dynamics modulate attention weights and influence downstream performance.

A) Parametric Sigmoid



B) Standard Sigmoid

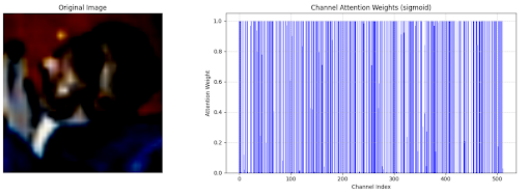


Figure 5: Comparison of channel attention weights between (a) parametric sigmoid (ours) and (b) standard sigmoid. The parametric sigmoid demonstrates smoother adaptability across channels, with attention weights mostly ranging between 0.4 and 0.7.

Figure 5 visually compares the standard sigmoid and our parametric sigmoid. The standard sigmoid yields a wide spread of attention weights ranging

from 0.0 to 0.8, resulting in sharp and sometimes binary-like gating. Some channels are strongly suppressed, while others dominate. This sparse pattern can be useful for hard feature selection but may limit adaptability in complex scenarios. In contrast, the parametric sigmoid exhibits a narrower and smoother distribution, with most weights concentrated between 0.4 and 0.7. This smoother profile avoids extreme values and indicates more balanced modulation across channels. Crucially, this distribution is not arbitrary. The parametric sigmoid's learnable parameters allow the network to adaptively control the sharpness of attention based on context, potentially capturing a more relevant representation.

Activation Function	Average Attention Score	Focus Area (%)	Gini	Test Accuracy (%)
Sigmoid	0.626	62.304	0.3738	89.58
Parametric Sigmoid	0.624	61.718	0.3729	90.16

Table 3: Channel attention weights for (a) parametric sigmoid (ours) and (b) standard sigmoid. Despite similar attention scores and focus areas, the parametric sigmoid yields higher test accuracy (90.16%) and lower Gini (0.3729), indicating more balanced attention and better performance.

Despite similar average attention scores between the two functions, the parametric sigmoid yields a higher accuracy (90.16% vs. 89.58%), as shown in Table 3. This suggests that the improved performance is not simply due to higher attention magnitudes but may stem from better contextual generalisation. Furthermore, its slightly reduced attention spread (61.7% vs. 62.3%) points to a more focused emphasis on task-relevant channels, avoiding harsh suppression or excessive amplification. Supporting this, the lower Gini coefficient under parametric sigmoid indicates a more equitable and less skewed attention allocation across channels.

Compared to softmax (see Appendix 3), which enforces strict competition via normalisation and often leads to overly sparse attention, the parametric sigmoid produces a more balanced distribution without suppressing diversity. This aligns with insights from attention literature ([4] [27], which caution against overly peaky or exclusive attention, particularly when requiring integration across multiple features.

Taken together, these findings underscore the significance of activation choice in shaping the internal dynamics of attention. The parametric sigmoid offers a compelling trade-off: it supports adaptive gating while avoiding the pitfalls of excessive sparsity or rigid fixed gating. More detailed comparisons with the remaining activations are presented in Appendix 6.

5 Future Enhancements

While PCAM enhances CBAM's design, our experiments have several limitations that suggest future work. Firstly, we evaluated PCAM only on ResNet-18. Extending to deeper or structurally distinct architectures, such as VGG19 or InceptionV3 (for which preliminary implementations already exist in the experiment source code) would test its generality and reveal how depth, filter granularity, and attention dynamics interact. Furthermore, all runs used 8-10 epochs, a fixed learning rate, and no systematic sweep of α for learned pooling weights. Longer schedules, fine-tuned learning-rate strategies, and grid searches over α could further boost performance and ensure improved convergence. The datasets used in the experiment were CIFAR-10 and STL-10. Applying PCAM to larger-scale datasets such as ImageNet-1K or domain-specific corpora (e.g. medical imaging) would assess its robustness with high-resolution images, class imbalances, and within different domains. In this model, PCAM is inserted in every residual block. Future work might explore selective or conditional placement, such as integrating PCAM in deeper layers or when feature statistics exceed a defined threshold. In this manner, we can determine the optimal trade-off point between accuracy gains and computational cost, for example. Addressing these limitations will further our understanding of adaptive attention in CNNs and broaden PCAM's applicability across architectures and domains.

6 Conclusion

PCAM introduces two learnable parameters per block, fusion weight and sigmoid temperature T, yielding a 0.45% Top-1 error reduction and a 0.8% decrease in gradient-norm variance, positioning it as a lightweight enhancement in both performance over fixed-design attention modules. By introducing two additional learnable parameters per module, PCAM transcends the fixed fusion and static gating assumptions of prior attention blocks, allowing networks to adapt their focus dynamically. We hope that our insights into adaptive pooling fusion and temperature-controlled gating will inspire further innovations in attention design, advancing the performance of convolutional architectures across various tasks and domains.

7 Appendices

This appendix provides a complete blueprint for replicating our experiments and validates the efficacy of PCAM in enhancing both performance and interpretability in CNNs.

7.1 PCAM: Parametric Channel Attention Module Implementation Details

The attention mechanism is based on the CBAM (Convolutional Block Attention Module) framework, but with modifications to enable channel attention (channel_act) and spatial attention (spatial_act). For each type of attention, we allow the activation function to be either a standard sigmoid or a learnable temperature parameter (parametric sigmoid). Each attention block is inserted after the convolutional output and before the residual addition. To improve the flexibility of the attention mechanisms, we use a parametric sigmoid activation defined as:

$$\sigma_\beta(x) = \frac{1}{1 + e^{-\beta x}}$$

Where is a learnable temperature parameter that modulates the sharpness of the attention map. A higher value would allow the model to form sharper attention boundaries. This is applied separately to both channel and spatial attention branches, depending on the configuration. In channel attention, we extend the standard CBAM formulation by introducing adaptive channel pooling (channel_pool). Both global average pooling and global max pooling are applied over the spatial dimensions. The resulting descriptors are combined using learnable weights, which are trained end-to-end to emphasise the most informative pooling strategy. Formally, for a channel c:

$$F_c = w_{\text{avg}} \cdot \text{AvgPool}(X)_c + w_{\max} \cdot \text{MaxPool}(X)_c$$

Where w_{avg} and $w_{\max}$ are learnable scalar parameters. This enables the models to balance smooth global context (average pooling) with sharp, high-response activations (max_pooling). For spatial pooling (spatial_pool), spatial attention is enhanced using a multi-scale fusion strategy, where feature maps are pooled using different kernel sizes to capture textures at various resolutions. Each scale output is passed through a convolution and normalised. The outputs are fused using learnable weights, allowing the model to dynamically select the most relevant scale for spatial feature aggregation.

$$F_s = \sum_{i=1}^{n} \alpha_i \cdot \text{Conv}\left(\text{Pool}_{k_i}(X)\right)$$

Where α_i are the learnable weights for each scale k_i. The attention modules (with pooling strategies) are inserted into ResNet after each residual block's final convolutional output, and before the addition of the identity (residual) connection. Each experimental configuration selectively enables or disables these components (channel_pool, spatial_pool, activation type) to assess their contribution. All of these components are implemented in PyTorch, leveraging a modular class-based architecture. The parametric sigmoid is implemented using torch.nn.Parameter for the learnable. Furthermore, training uses the same hyperparameters across experiments to isolate the effect of architectural changes.

7.2 Spatial Attention with Parametric Gating

The spatial attention gate begins by producing a pre-activation map $M \in \mathbb{R}^{1\times H\times W}$ via a 7×7 convolution on concatenated channel-pooled descriptors (Woo et al., 2018), whereby CBAM's spatial branch pools across channels to form a 2-channel map, convolves with a 7×7 filter to capture spatial context, and applies a standard sigmoid:

$$M_s(F) = \sigma\left(\text{Conv}_{7\times7}\left([\text{AvgPool}_C(F)\ ;\ \text{MaxPool}_C(F)]\right)\right)$$

Large kernels are shown necessary to capture sufficient context for precise localisation, as "a broad view (i.e. large receptive field) is needed for deciding spatially important regions." (Woo et al., 2018). However, gating outputs that saturate near 0 or 1 suffer vanishing gradients. To control the gate's steepness, or the sharpness with which it transitions from 0 to 1, we introduce a learnable temperature parameter T. The final spatial attention map is then:

$$M_s(F) = \sigma\left(\frac{T}{M}\right)$$

When T is large, the sigmoid's input is effectively down-scaled, keeping it in an almost-linear regime where gradients do not vanish. This soft gating facilitates stable exploration of attention patterns early in training. As the model converges, a smaller T steepens the sigmoid, making it resemble a near-binary threshold, such that the attention mask can highlight salient spatial locations and suppress background noise. This dynamic modulation of sigmoid steepness via T balances trainability and precision in spatial attention, addressing

vanishing-gradient concerns whilst enabling precise localisation.

7.3 Activation Function Details

Attention modules employ non-linear gating functions to convert pooled feature descriptors into per-channel or per-spatial weighting coefficients. These gating functions critically determine the sparsity and selectivity of attention maps. Highly saturating, sigmoidal mappings induce focalised weighting that sharply emphasises salient features, whereas more linear functions yield diffuse attention distributions that spread emphasis across many activations [14]. Moreover, the choice of nonlinearity influences gradient propagation within the attention branch. Activation functions with vanishing regions can impede back-propagated gradients, slowing convergence or destabilising optimisation [11], whereas smoother, self-gated mappings, like Swish, can preserve gradient flow, facilitating stable, efficient training [25]. To assess the effects of these non-linear activation-gating units on attention sharpness, interpretability, and overall performance, we examine six representative functions spanning probabilistic transformations, parameterised sigmoids, self-gating formulations, and sparse normalisations.

Sigmoid

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

The sigmoid function maps to (0, 1), yielding a natural probabilistic interpretation for each gating coefficient. In classification contexts, sigmoid is the standard for binary outputs, constraining predictions to [0, 1] and supporting cross-entropy training. Within channel-attention modules (e.g. Squeeze-and-Excitation blocks), sigmoidal gating softly suppresses less informative features while accentuating prominent ones, with minimal computational overhead and full differentiability [14]. Its interpretability, whereby each weight can be read as the probability of "selecting" a feature, makes it useful in settings requiring explainable attention maps.

Parametric Sigmoid

$$\sigma(x) = \frac{1}{1 + e^{-\alpha(x-\beta)}}$$

Generalising the standard sigmoid with learnable steepness () and midpoint () parameters, the parametric logistic enables the model to adjust gating sensitivity and bias across layers or tasks. By adapting the activation's slope and threshold per layer or task, the network gains fine-grained control over gating sensitivity. This flexibility is particularly useful in multi-domain or hierarchical attention architectures where attention distributions vary widely in scale or require dynamic calibration [9]. Empirically, parameterised sigmoids have been shown to improve convergence and representational expressivity compared to their rigid vanilla counterparts.

Swish

$$\text{Swish}(x) = x \cdot \sigma(\beta x) = x \cdot \frac{1}{1 + e^{-\beta x}}$$

The Swish activation combines linear identity with logistic gating in a self-modulated fashion. Empirical studies demonstrate that Swish outperforms ReLU-like units in deep vision architectures, improving both accuracy and convergence smoothness [25]. Its non-monotonic shape allows negative inputs to contribute adaptively, enriching representational capacity within attention pathways. Its allowance for negative outputs and smooth gradient profile facilitates richer feature representations and more stable training. In attention branches, this translates to both finer focus and faster convergence.

Scaled Hyperbolic Tangent (Scaled Tanh) The scaled hyperbolic tangent enhances the basic tanh function by applying scale factors to its output range and slope.

$$\text{ScaledTanh}(x) = a \cdot \tanh(bx)$$

The scaled hyperbolic tangent extends the symmetric tanh (range 1 to 1) by learnable scale (a) and slope (b) factors. Tanh's zero-centred range accelerates optimisation relative to sigmoid and naturally encodes both excitatory (positive) and inhibitory (negative) attention signals [26]. In contexts such as sequence models or attention-based gating where inhibition of distractors is as important as excitation of targets, scaled tanh provides balanced, contrast-sensitive weighting; further, linear scaling via b mitigates the vanishing-gradient problem while a adjusts overall attentional strength [26].

Softmax The softmax function

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}, \quad \text{for } i = 1, \ldots, K$$

K converts vectors into dense probability distributions over K elements, enforcing global competition among features [27]. In transformer-style self-attention, softmax normalisation is the driver for

computing attention weights across tokens, producing sharply peaked distributions when one descriptor is more prominent.

Sparsemax

$$\text{sparsemax}(\mathbf{z}) = \arg\min_{\mathbf{p}\in\Delta^{K-1}} \|\mathbf{p}-\mathbf{z}\|^2$$

As an alternative to softmax, sparsemax replaces softmax's dense probabilities with truly sparse outputs where many weights are exactly zero [21, **?**]. Sparsemax is particularly advantageous in multi-label classification and attention-based inference, where interpretability and computational efficiency derive from hard selection of a limited subset of features. Empirical studies on natural language inference and hierarchical attention networks demonstrate that sparsemax yields compact, highly interpretable attention maps without sacrificing accuracy.

These six gating nonlinearities entail trade-offs in focus versus diffusion, gradient flow versus saturation, and interpretability versus expressive flexibility. An experiment was therefore conducted to discern the optimal gating choice for a given attention-based model.

7.4 Comparative Analysis of Activation Functions in Attention Module

To identify the most effective activation function in both channel and spatial attention, we conduct separate experiments across six activation functions (softmax, scaled_tanh, parametric_sigmoid, swish, sparsemax, and sigmoid), then compare their performance in terms of top-1, top-5 and average precision. Using ResNet-18 on CIFAR-10 (50K train/10K test images, 10 classes), we compare activation functions under consistent training conditions (10 epochs, lr=0.0001, 224×224 input).

Channel Attention

PCAM uses a parametric sigmoid in its channel attention module to map pooling values to attention weights. It is chosen for its superior performance, achieving the lowest Top-1 error (8.87%) and highest AP (0.9685), surpassing standard Sigmoid and Swish. Notably, using softmax and Sparsemax does not drastically reduce their average precision, as shown in channel attention. Using softmax or sparsemax in channel attention led to high sparsity, meaning that only a few channels were activated. However, in spatial attention, the objective is to determine where in the image to focus. In this case, sparsity may not be as detrimental, as spatial attention is intended to highlight specific regions rather than distribute attention across numerous channels. Looking at the numbers: in spatial attention, softmax yields a Top-1 error of 11.41% and sparsemax 11.93%, both worse than the parametric sigmoid (8.87%), but not as severely degraded as in the case of channel attention, where softmax and sparsemax result in errors of 13.27% and 24.64%, respectively. This suggests that spatial attention is more tolerant of sparsity-inducing mechanisms.

Spatial Attention

PCAM uses a parametric sigmoid in its channel attention module to map pooling values to attention weights. It is chosen for its superior performance, achieving the lowest Top-1 error (8.87%) and highest AP (0.9685), surpassing standard Sigmoid and Swish. Notably, using softmax and Sparsemax does not drastically reduce their average precision, as shown in channel attention. Using softmax or sparsemax in channel attention led to high sparsity, meaning that only a few channels were activated. However, in spatial attention, the objective is to determine where in the image to focus. In this case, sparsity may not be as detrimental, as spatial attention is intended to highlight specific regions rather than distribute attention across numerous channels. Looking at the numbers: in spatial attention, softmax yields a Top-1 error of 11.41% and sparsemax 11.93%, both worse than the parametric sigmoid (8.87%), but not as severely degraded as in the case of channel attention, where softmax and sparsemax result in errors of 13.27% and 24.64%, respectively. This suggests that spatial attention is more tolerant of sparsity-inducing mechanisms.

7.5 Evaluating Design Choices in PCAM via Ablation Study

To evaluate the contributions of individual components in PCAM, we conducted an ablation study comprising 9 distinct experimental configurations. These configurations systematically varied critical design choices, including: (1) the inclusion or exclusion of pooling weights in channel attention, (2) the integration or omission of pooling weights in spatial attention, (3) the selection of either standard sigmoid or parametric sigmoid activation functions, and (4) comparisons against the baseline ResNet architecture. This structured analysis shows the synergistic impact of the parametric sigmoid's learnable temperature in modulating attention focus and the adaptive pooling fusion mechanism.

Using ResNet-18 on CIFAR-10 (50K train/10K test images, 10 classes) and STL (50K train/10K test images, 10 classes), we compare activation functions under consistent training conditions (10 epochs, lr=0.0001, 224×224 input).

Channel Activation	Top-1 Error (%)	Top-5 Error (%)	Average Precision
Softmax	13.269	1.01	0.9317
Scaled Tanh	9.2199	0.3499	0.9668
Parametric Sigmoid	9.5	0.28	0.9663
Swish	9.090	**0.26**	0.96815
Sparsemax	24.64	1.989	0.80211
Sigmoid	**8.939**	0.29	**0.96956**

Table 4: Channel attention performance by activation. Sigmoid yields the best results, followed by Swish, Parametric Sigmoid, and Scaled Tanh. Softmax and Sparsemax underperform due to excessive sparsity (Gini 0.99; see Appendix 4).

Spatial Activation	Top-1 Error (%)	Top-5 Error (%)	Average Precision
Softmax	11.41	0.5699	0.9513
Scaled Tanh	9.54	0.34	0.9656
Parametric Sigmoid	**8.87**	**0.3199**	**0.9685**
Swish	9.45	0.34	0.9647
Sparsemax	11.93	0.53	0.9487
Sigmoid	9.38	0.29	0.9664

Table 5: Channel attention performance by activation. (a) Parametric Sigmoid yields the best results, followed by (b) Sigmoid, (c) Swish, and (d) Scaled Tanh. (e) Softmax and (f) Sparsemax underperform due to excessive sparsity (Gini 0.99; see Appendix 1.6).

Image Classification with STL

Configuration g [parametric spatial sigmoid, standard channel sigmoid, channel pooling on, spatial pooling off] achieves 95.46% Top-1, outpacing ResNet-18 (95.16%) and the best sigmoid-only spatial-pooling model (94.69%). All parametric variants hit AP 0.989, confirming the stability benefit of learnable temperature. By contrast, configurations d and h, where both channel and spatial pools are enabled, regress to 95.03% and 95.34% (which equals decreases by 0.43 and 0.12 points, respectively). This suggests that dual pooling potentially introduces conflicting signals that degrade accuracy.

Image Classification with CIFAR-10

On CIFAR-10, we observed analogous trends. The fixed sigmoid gate delivered 94.06% Top-1 accuracy, while the parametric sigmoid alone improved this to 94.16%. Pairing the parametric sigmoid with channel-weight pooling (configuration g) yielded the highest Top-1 score of 94.51%. Spatial pooling under the fixed gate produced the most pronounced Top-5 improvement (rising from 99.87% to 99.95%), suggesting that coarse spatial smoothing can benefit high-confidence retrieval. However, spatial pooling with a parametric gate slightly degraded Top-1 (94.05%), and dual pooling underperformed relative to the single-pool optimum. AP for parametric models was tightly grouped (0.985), whereas fixed-sigmoid AP ranged more widely, reaffirming that learnable gating principally enhances ranking stability rather than dramatically altering overall accuracy.

Aggregate Findings:

Our ablation study yields a clear design prescription for lightweight CNNs: (1) adopt a parametric spatial-attention nonlinearity to ensure consistent Top-1 and AP gains; (2) leverage channel-weight pooling with a learnable gating temperature, as it synergizes to yield the largest uplift; and (3) avoid combining both channel and spatial pooling, since dual aggregation potentially introduces redundant normalization that could negate individual benefits of these pools. Spatial pooling under a fixed gate may still be useful for Top-5 retrieval in contexts similar to CIFAR-10, but its utility is dataset-dependent.

7.6 Activation Functions Probing Details

To probe how different activation functions translate pooling weights into attention scores, we experimented with multiple nonlinearities: scaled tanh, sigmoid, parametric sigmoid, sparsemax, softmax, and swish, within both channel and spatial attention modules. Each function modulates attention differently: sigmoid and tanh tend to produce smooth, bounded outputs; parametric sigmoid may

Configuration	Spatial Attention	Channel Attention	Channel Weight Pool	Spatial Weight Pool	Top-1 Accuracy (%)	Top-5 Accuracy (%)	Average Precision
a	Sigmoid	Sigmoid	no	no	95.0125	99.83	0.988
b	Sigmoid	Sigmoid	no	yes	94.6875	99.81	0.985
c	Sigmoid	Sigmoid	yes	no	94.875	99.85	0.987
d	Sigmoid	Sigmoid	yes	yes	95.025	99.89	0.988
e	Parametric Sigmoid	Sigmoid	no	no	95.275	99.84	0.988
f	Parametric Sigmoid	Sigmoid	no	yes	95.3375	**99.9**	0.989
g	Parametric Sigmoid	Sigmoid	yes	no	**95.4625**	99.875	**0.989**
h	Parametric Sigmoid	Sigmoid	yes	yes	95.3375	99.8875	0.989
baseline	n/a	n/a	n/a	n/a	95.1625	99.8375	0.989

Table 6: STL classification. Parametric sigmoid with channel pooling (g) yields the highest Top-1 accuracy (95.46%), while parametric sigmoid variants (f–h) achieve the highest average precision (0.989).

Configuration	Spatial Attention	Channel Attention	Channel Weight Pool	Spatial Weight Pool	Top-1 Accuracy (%)	Top-5 Accuracy (%)	Average Precision
a	Sigmoid	Sigmoid	no	no	94.06	99.87	0.98455
b	Sigmoid	Sigmoid	no	yes	94.43	**99.95**	**0.98665**
c	Sigmoid	Sigmoid	yes	no	94.04	99.93	0.98375
d	Sigmoid	Sigmoid	yes	yes	94.22	99.88	0.9832
e	Parametric Sigmoid	Sigmoid	no	no	94.16	99.86	0.985
f	Parametric Sigmoid	Sigmoid	no	yes	94.05	99.82	0.9842
g	Parametric Sigmoid	Sigmoid	yes	no	**94.51**	99.89	0.9856
h	Parametric Sigmoid	Sigmoid	yes	yes	94.22	99.86	0.9835
baseline	n/a	n/a	n/a	n/a	94.39	99.85	0.98452

Table 7: CIFAR-10 classification. Sigmoid with spatial pooling (g) yields the best Top-1 accuracy (94.51%). Parametric sigmoid with pooling (b) achieves the highest average precision (0.98665), though performance differences remain small.

allow for fine control of how much attention each part of the input should receive [29]. Swish, being self-gated, amplifies stronger signals while preserving gradient propagation. In contrast, sparsemax [21, **?**] and softmax force competition, so only the most salient region gets a high weight. This might help in tasks where a single dominant region is key, but could suppress other important areas. Empirically, sparsemax consistently yielded more semantically coherent saliency in spatial maps, while parametric sigmoid improved feature discrimination in channel attention with negligible parameter overhead. To isolate the effect of activation functions on channel attention, we fix the spatial attention module to use the standard sigmoid. We then vary the activation function solely within the channel attention module across six potential functions: scaled tanh, sigmoid, parametric sigmoid, sparsemax, softmax and swish. For interpretability, we construct channel index vs. attention weight plots, where the x-axis denotes the channel index and the y-axis shows the average attention weight assigned to each channel over a validation set. This setup allows us to directly compare how each activation function distributes emphasis across feature channels, revealing patterns of selectivity or diffuseness in channel-wise modulation. This probing method is motivated by the hypothesis that certain activation functions induce selective channel emphasis(e.g., sparsemax, softmax), while others may distribute attention more evenly (e.g., swish, scaled tanh). We found that softmax and sparsemax enforce extreme sparsity (Gini > 0.99), often emphasising a few dominant channels per class. In contrast, swish and scaled tanh distribute attention more diffusely ($0.7 >$ Gini > 0.4).

Activation	**Average Attention Score**	**Focus Areas (%)**	**Gini**
Sigmoid	0.626	62.3046	0.3738
Softmax	0.00195	0.9765	0.9954
Sparsemax	0.00195	0.1953	0.9980
Scaled Tanh	0.5986	59.57	0.4013
Parametric Sigmoid	0.6242	61.719	0.3729
Swish	2.1928	29.2968	0.7983

Table 8: Comparison of activation functions for channel attention. (a) Parametric sigmoid and (b) swish produce the highest average attention scores, with (b) showing broad activation and low focus. (c) Sigmoid and (d) scaled tanh provide smooth, balanced attention distributions. (e) Softmax and (f) sparsemax result in highly sparse attention (Gini > 0.99).

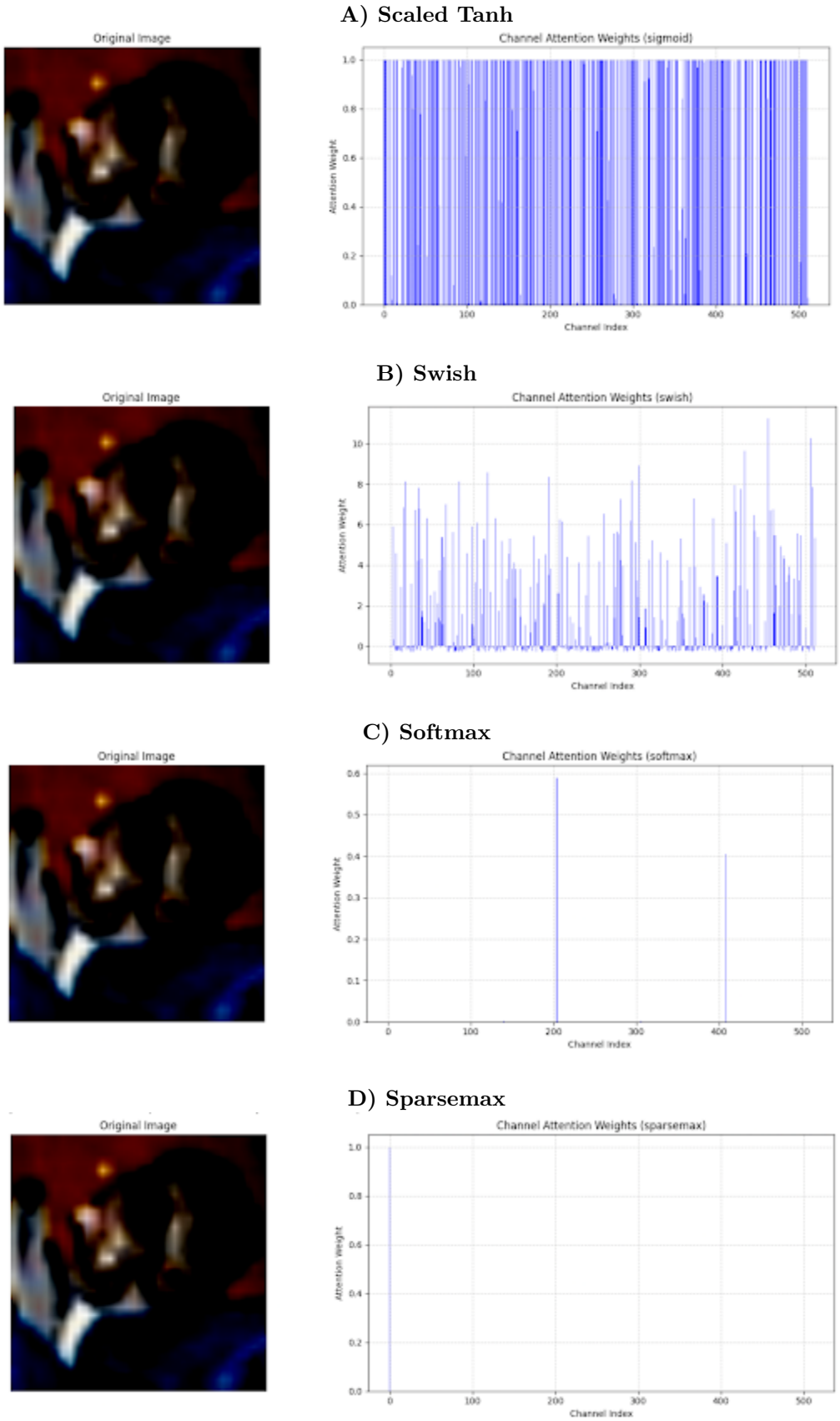
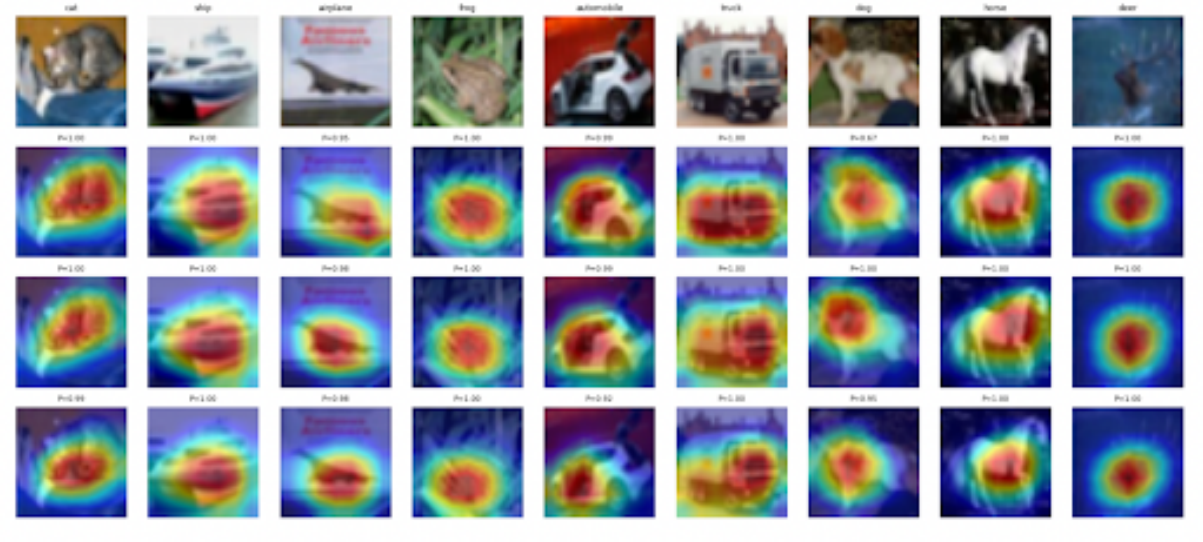


Figure 6: Channel attention distributions across activation functions. Each row shows the original image (left) and the corresponding channel attention weights (right) using a different activation function, with spatial attention fixed to sigmoid. (a) Scaled Tanh yields diffuse, uniform weights. (b) Swish produces broad, high-variance responses. (c) Softmax highlights a few dominant channels. (d) Sparsemax enforces extreme sparsity.

Grad-CAM Comparisons

A) Grad-CAM comparisons of the three models on the CIFAR-10 image dataset.



B) Grad-CAM comparisons of the three models on the STL-10 image dataset.

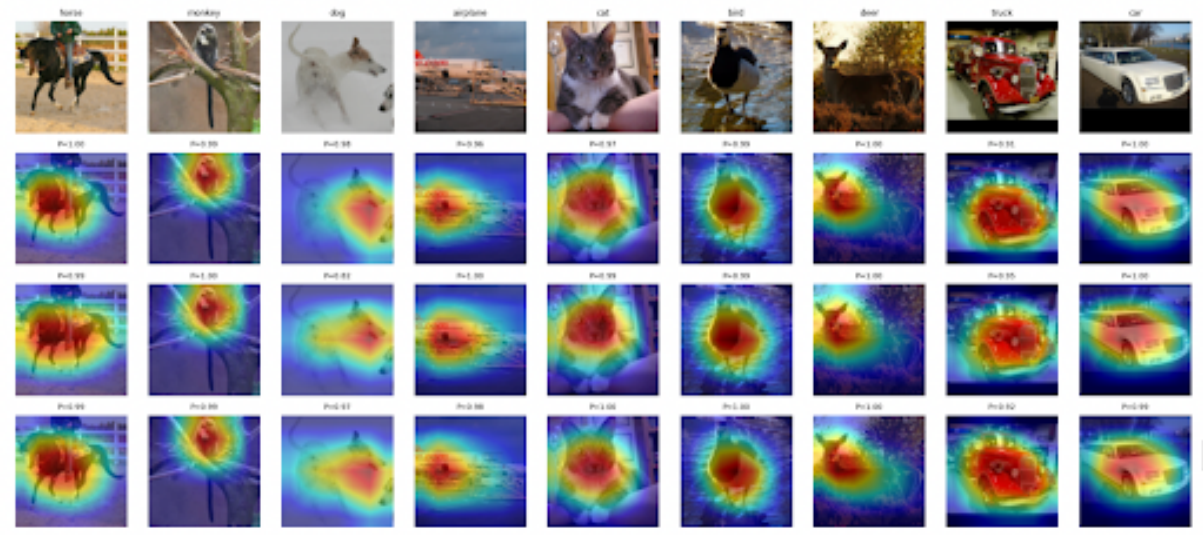


Figure 7: Grad-CAM comparisons of (a) ResNet18, (b) ResNet18+CBAM, and (c) ResNet18+PCAM (ours), showing final-layer activations. Ground-truth labels and prediction scores (P) demonstrate PCAM's superior focus on key features (e.g., dog's head) versus alternatives.

8 Statement about individual contributions

spark12800 - 45773

Amethyst456 - 41124

Santosgentilini07 - 41131

Neelb25 - 40388

Implementation

Implementation of CBAM Module - amethyst456

Implementation of Training Pipeline - amethyst456

Implementation of Evaluation Pipeline - amethyst456, spark12800

Implementation of PCAM Module - spark12800, amethyst456

Pretrained Model Set-Ups - santosgentilini07

Implementation of Explainability for Activation Functions - spark12800, neelb25

Implementation of Grad-Cam - neelb25, spark12800

Design

Design and Conduct Experiment - spark12800, amethyst456

Training

Training Model – santosgentilini07, amethyst456, neelb25, spark12800

References

[1] Squeeze-and-excitation networks | IEEE conference publication | IEEE xplore.

[2] Maan Alhazmi and Abdulrahman Altahhan. Best fit activation functions for attention mechanism: Comparison and enhancement. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

[3] Megat Syahirul Amin Megat Ali, Azlee Zabidi, Nooritawati Md Tahir, Ihsan Mohd Yassin, Farzad Eskandari, Azlinda Saadon, Mohd Nasir Taib, and Abdul Rahim Ridzuan. Short-term gini coefficient estimation using nonlinear autoregressive multilayer perceptron model. 10(4):e26438.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate.

[5] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V. Le. Attention augmented convolutional networks.

[6] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V. Le. Attention augmented convolutional networks.

[7] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic ReLU.

[8] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 215–223, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.

[9] Shikha Dubey, Farrukh Olimov, Muhammad Aasim Rafique, and Moongu Jeon. Improving small objects detection using transformer. 89:103620.

[10] Fartash Faghri, David Duvenaud, David J. Fleet, and Jimmy Ba. A study of gradient variance in deep learning.

[11] Jianli Feng and Shengnan Lu. Performance analysis of various activation functions in artificial neural networks. 1237(2):022030. Publisher: IOP Publishing.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Sur-

passing human-level performance on ImageNet classification.

[14] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks.

[15] Sarthak Jain and Byron C. Wallace. Attention is not explanation. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556. Association for Computational Linguistics.

[16] Alex Krizhevsky. Learning multiple layers of features from tiny images.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

[18] Aitor Lewkowycz and Guy Gur-Ari. On the training dynamics of deep networks with l_2 regularization.

[19] Hengyi Li, Xuebin Yue, and Lin Meng. Enhanced mechanisms of pooling and channel attention for deep learning feature maps. 8:e1161.

[20] Ningning Ma, Xiangyu Zhang, Ming Liu, and Jian Sun. Activate or not: Learning customized activation.

[21] André F. T. Martins and Ramón Fernandez Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. *CoRR*, abs/1602.02068, 2016. arXiv:1602.02068.

[22] André Martins, António Farinhas, Marcos Treviso, Vlad Niculae, Pedro Aguiar, and Mario Figueiredo. Sparse and continuous attention mechanisms. In *Advances in Neural Information Processing Systems*, volume 33, pages 20989–21001. Curran Associates, Inc.

[23] Diganta Misra. Mish: A self regularized non-monotonic neural activation function.

[24] Fateme Mostafaie, Zahra Nabizadeh, Nader Karimi, and Shadrokh Samavi. A general framework for saliency detection methods.

[25] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Swish: a self-gated activation function. version: 1.

[26] Swalpa Kumar Roy, Suvojit Manna, Shiv Ram Dubey, and Bidyut Baran Chaudhuri. LiSHT: Non-parametric linearly scaled hyperbolic tangent activation function for neural networks.

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

[28] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. CBAM: Convolutional block attention module.

[29] Yao Ying, Nengbo Zhang, Peng Shan, Ligang Miao, Peng Sun, and Silong Peng. PSigmoid: Improving squeeze-and-excitation block with parametric sigmoid. 51(10):7427–7439.