Multi-Scale Unrectified Push-Pull with Channel Attention for Enhanced Corruption Robustness

Robin Narsingh Ranabhat[1], Longwei Wang[1], Xiao Qin[2], Yang Zhou[2], and KC Santosh[1]

[1]AI Research Lab, Department of Computer Science, University of South Dakota, USA
[2]Department of Computer Science and Software Engineering, Auburn University, USA
robin.ranabhat@coyotes.usd.edu, longwei.wang@usd.edu, xqin@auburn.edu, yangzhou@auburn.edu, kc.santosh@usd.edu

Abstract

Convolutional Neural Networks (CNNs) have achieved remarkable success in computer vision tasks; however, they often experience substantial performance degradation when confronted with real-world corruptions such as noise, compression artifacts, and lighting variations. The original push–pull CNN (PP-CNN) architecture addresses this challenge by employing a biologically inspired mechanism that contrasts local excitatory (push) and broader inhibitory (pull) responses to suppress noise. In this work, we enhance the robustness of PP-CNN through three key modifications: (1) removing the half-wave rectification constraint to enable more expressive interactions between push and pull signals, allowing for richer linear feature enhancement; (2) introducing a dynamic channel attention mechanism that adaptively recalibrates feature responses by amplifying discriminative signals and suppressing noise-dominated channels; and (3) designing a multi-scale push–pull (MSPP) framework that searches for pattern consistency across multiple spatial resolutions, reinforcing the model's ability to generalize under corruptions at varying scales. Our proposed enhancements introduce a stronger inductive bias toward learning scale-consistent features—a fundamental property of natural images that remains stable even under corruption—without requiring corruption-specific data augmentation. Comprehensive evaluations on the CIFAR-10-C benchmark demonstrate that the enhanced PP-CNN achieves improvements in robustness across diverse corruption types while maintaining competitive accuracy on clean data. Notably, the multi-scale variant delivers the best trade-off between robustness and clean data performance, demonstrating the effectiveness of exploiting multi-scale feature consistency for generalization to unseen common image corruptions.

Introduction

Convolutional Neural Networks (CNNs) have demonstrated remarkable success across a broad range of computer vision tasks, including image classification, object detection, and semantic segmentation. However, their performance deteriorates significantly when exposed to real-world input corruptions such as noise, compression artifacts, and lighting variations (Hendrycks and Dietterich 2019). In practical settings, natural images are frequently affected by environmental factors like low lighting, motion blur, and lossy compression, which introduce subtle distortions that can mislead CNN-based models. Despite the fact that natural images often exhibit consistent patterns such as edges and textures across different spatial scales, CNNs struggle to leverage this intrinsic property to separate meaningful signals from local noise. Geirhos et al. (2017) experimentally demonstrated that humans consistently outperform state-of-the-art CNNs in classifying corrupted images, highlighting the limitations of existing CNN models in handling natural image corruptions.

Traditional CNNs are designed to extract hierarchical feature representations using local receptive fields and spatial pooling. While effective on clean data, these operations are highly sensitive to localized noise and distortions because they lack an explicit mechanism to distinguish between genuine patterns and noise-based activations. Existing regularization techniques such as data augmentation and adversarial training attempt to mitigate this sensitivity by exposing models to synthetic noise and transformations. However, these approaches are computationally expensive and may not generalize well to unseen corruption types or real-world noise patterns.

The push–pull CNN (PP-CNN) architecture (Strisciuglio, Lopez-Antequera, and Petkov 2020) was introduced to address these challenges by drawing inspiration from biological vision systems. Human visual processing is known to employ a mechanism where local excitatory (push) responses are balanced by broader inhibitory (pull) responses to enhance contrast and suppress noise. The original PP-CNN models this process using two convolutional operations with kernels of different sizes and opposite polarities. The push kernel extracts local high-frequency patterns, while the pull kernel captures broader contextual information to counteract noise and background activations. The output is computed by combining these two responses after applying half-wave rectification.

While the baseline push–pull mechanism demonstrated improved robustness to noise and distortions, it has several limitations. First, applying ReLU-based rectification to the push and pull responses before combining them limits the model's ability to fully exploit the contrast between local and surround signals. Second, the original PP-CNN processes spatial features independently within each channel, ignoring the contextual relationships between different feature channels. Third, the push–pull mechanism operates at a fixed spatial scale, limiting its ability to handle corruptions that occur at different scales.

In this work, we propose three enhancements to the original PP-CNN to improve robustness against common corruptions without compromising clean data performance.

First, we remove half-wave rectification from push–pull responses, converting the mechanism into a linear filter that preserves both positive and negative contrasts. This sharpening effect amplifies structured patterns while suppressing noise.

Second, we incorporate a Squeeze-and-Excitation-inspired channel attention module to adaptively reweight push responses. By emphasizing informative channels and suppressing noise, this improves feature selectivity under corruption.

Third, we extend the architecture to a multi-scale push–pull framework by introducing pull kernels of varying sizes. These multi-scale features, concatenated and refined via channel attention and a 1×1 convolution, enhance robustness across spatial resolutions while maintaining efficiency.

Related Works

CNNs have become the foundation of modern computer vision, enabling state-of-the-art performance in image classification, object detection, semantic segmentation, and other vision tasks. Their hierarchical feature extraction enables learning of spatial and semantic patterns directly from raw data. However, they remain susceptible to corruptions such as noise, blur, occlusions, and contrast variations, which hinder generalization and degrade real-world performance.

Robustness of CNNs

CNNs have demonstrated remarkable success on large-scale datasets such as ImageNet (Deng et al. 2009) and CIFAR (Krizhevsky, Hinton et al. 2009), but their vulnerability to input perturbations has remained a major challenge. Hendrycks and Dietterich (2019) showed that CNNs exhibit significant performance degradation when evaluated on corrupted versions of ImageNet and CIFAR-10. Their study introduced the CIFAR-10-C and ImageNet-C datasets, which include common corruptions such as Gaussian noise, motion blur, and pixelation, to evaluate the robustness of CNN models under real-world perturbations. The results highlighted that even high-performing models, such as ResNet (He et al. 2016) and DenseNet (Huang et al. 2017), experience a sharp drop in accuracy under these perturbations, indicating that standard CNN architectures lack intrinsic robustness to environmental variability.

Geirhos et al. (2017) demonstrated that CNNs tend to overfit to texture information rather than shape, which makes them highly sensitive to style changes and noise. This texture bias reduces generalization capacity, particularly when the model encounters data that deviates from the clean training distribution. Further, Ilyas et al. (2019) argued that adversarial vulnerability stems from the presence of "non-robust" but predictive features in the training data, which CNNs exploit to maximize accuracy. These findings suggest that the fundamental learning strategy of CNNs prioritizes texture-based patterns over more meaningful shape-based features, making them inherently fragile under perturbations.

Several strategies have been proposed to improve CNN robustness:

Data Augmentation Data augmentation is one of the earliest and most widely used methods to improve CNN generalization. Techniques such as random cropping, flipping, rotation, color jittering, and cutout augmentation increase the diversity of the training set, helping the model learn invariance to minor perturbations (Krizhevsky, Sutskever, and Hinton 2012)(Wang et al. 2021a). More sophisticated augmentation strategies, such as AutoAugment (Cubuk et al. 2019)(Wang et al. 2021b) and RandAugment (Cubuk et al. 2020), automate the search for optimal augmentation policies, leading to further improvements in generalization.

Adversarial Training Adversarial training involves generating adversarial examples by perturbing the input data and training the model to resist these attacks. Goodfellow et al. introduced the Fast Gradient Sign Method (FGSM) to create adversarial examples by adding small, structured perturbations to the input image (Goodfellow, Shlens, and Szegedy 2015). Madry et al. proposed a more robust adversarial training method using projected gradient descent (PGD), which creates stronger adversarial examples (Madry et al. 2018)(Wang, Li, and Zhang 2024). Adversarial training improves robustness against specific attacks but often leads to a drop in clean accuracy and increased computational cost.

Regularization and Normalization Batch normalization (Ioffe and Szegedy 2015) and dropout (Srivastava et al. 2014) have been shown to improve generalization by reducing overfitting and stabilizing the learning process. Label smoothing (Szegedy et al. 2016) reduces overconfidence in predictions by encouraging the model to produce more calibrated probabilities. Weight decay and mixup (Zhang et al. 2017) also help improve robustness by encouraging smoother decision boundaries.

Push-Pull Design

Biological vision systems process visual stimuli through a balance of excitatory and inhibitory responses, a mechanism known as **push–pull inhibition**. Push–pull inhibition enhances the signal-to-noise ratio by reinforcing relevant patterns through excitatory responses and suppressing irrelevant background information through inhibitory feedback. Petkov and Westenberg (2003) introduced a computational model of push–pull inhibition for contour detection, demonstrating that this mechanism improves robustness to noise and cluttered backgrounds. Inspired by this work, Strisciuglio et al. (2020) integrated push–pull inhibition into CNNs by introducing a push–pull layer composed of two convolutional kernels. The push kernel models excitatory responses, while the pull kernel models inhibitory responses. The network response is computed as the difference between these two kernels.

Attention Mechanisms in CNNs

Attention mechanisms have revolutionized deep learning by enabling models to selectively focus on task-relevant features while ignoring irrelevant patterns. Attention was first introduced in natural language processing with the Transformer model (Vaswani et al. 2017), which used self-attention to model long-range dependencies in sequential data. The success of Transformers motivated the adaptation of attention to CNNs for computer vision tasks. Hu et al. (2018) introduced SE networks, which compute channel-wise attention through global average pooling and a fully connected gating mechanism. SE networks improved classification accuracy on ImageNet with minimal computational cost. Woo et al. (2018) proposed CBAM, which combines spatial and channel-wise attention to refine feature maps. CBAM improves both object detection and classification but increases complexity due to additional convolutional layers. Wang et al. (2020) proposed ECA to reduce the computational cost of attention by using a 1D convolution for channel attention. ECA achieves strong performance while maintaining efficiency.

Methodology

We propose three key modifications to the push–pull CNN (PP-CNN) architecture to improve robustness against image common image corruptions without relying on targeted data augmentation for each type of distortion. These modifications include: (1) removing the activation constraint to enable more expressive interactions between push and pull signals, (2) introducing a channel attention mechanism to dynamically recalibrate feature maps based on global channel dependencies, and (3) extending the architecture to incorporate pull activations at multiple scales. These enhancements reinforce the architectural bias of push–pull layers, improving the network's ability to handle common corruptions in a more generalizable manner.

Revisiting the Push–Pull CNN

A push–pull CNN layer employs two sets of convolutional kernels: a set of push kernels $\boldsymbol{K}_p$ and a corresponding set of pull kernels $\boldsymbol{K}_q$ derived from the push kernels. The pull kernels are obtained by negating the push kernels and applying bilinear upsampling to increase their spatial extent, thereby creating a complementary pair of feature extraction operations.

Given an input image $\boldsymbol{X} \in \mathbb{R}^{H' \times W' \times C'}$, the push response is computed as: $\boldsymbol{U}_p = \boldsymbol{X} \circledast \boldsymbol{K}_p$, where $\circledast$ denotes the convolution operator, and $\boldsymbol{U}_p \in \mathbb{R}^{H \times W \times C}$ represents the feature map produced by the push kernels. For a specific channel c, the push response is computed as: $\boldsymbol{u}_p^c = \boldsymbol{X} \circledast \boldsymbol{k}_p^c$, where $\boldsymbol{k}_p^c$ is the c-th 2D push kernel, and $\boldsymbol{u}_p^c \in \mathbb{R}^{H \times W}$ represents the corresponding response map. Similarly, the pull response is computed using the pull kernels $\boldsymbol{K}_q$ as $\boldsymbol{U}_q = \boldsymbol{X} \circledast \boldsymbol{K}_q$.

The baseline push–pull layer applied ReLU activation to both push and pull responses, limiting the model's ability to capture fine-grained variations. We propose removing this rectification step and computing the final output with a raw **unrectified subtraction**:

$$\boldsymbol{U} = \boldsymbol{U}_p - \alpha \cdot \boldsymbol{U}_q \tag{1}$$

where α is a scaling factor that can be fixed or learned during training. This modification enhances the contrast enhancement effect by reinforcing consistent patterns that are repeated across spatial regions. When a pattern is detected by both push and pull kernels (with opposite polarity), the subtraction operation amplifies the response, effectively sharpening the feature map. The absence of rectification allows more nuanced responses, enabling the network to preserve fine details and subtle variations that would otherwise be suppressed by ReLU.

Dynamic Channel Attention for Cross-Channel Interaction

To improve feature selectivity and aid with noisy feature suppression, we introduce a channel attention mechanism that recalibrates the push response $\boldsymbol{U}_p$ based on global cross-channel interaction. Inspired by the SE network, the attention mechanism allows the network to adaptively weight different channels based on their contribution to the final output.

Squeeze Operation We compute a channel-wise descriptor by averaging each push response map: $z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \boldsymbol{u}_p^c(i, j)$, where z_c represents the global average value of the c-th push response. This generates a channel descriptor $\boldsymbol{z} \in \mathbb{R}^C$ that captures the overall activation strength across all feature channels.

Excitation Operation The channel descriptor is passed through a fully connected gating mechanism

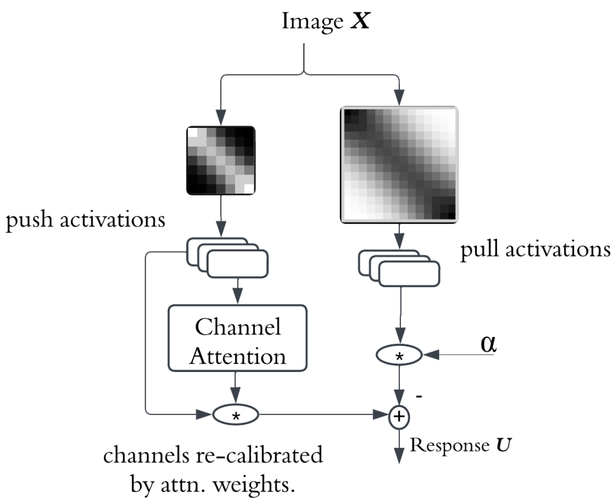

Figure 1: PP-CNN layer with dynamic attention

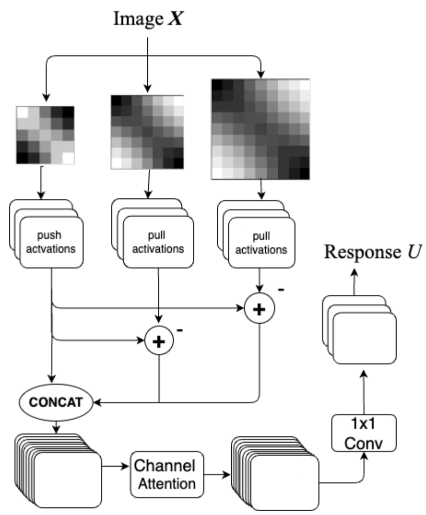

Figure 2: PP-CNN layer utilizing pull kernels derived from push kernels at multiple scales

to compute the importance weights of the channel. Specifically, we use a two-layer network with a bottleneck dimensionality reduction to reduce computational complexity and improve generalization as : $\boldsymbol{w} = \sigma(\boldsymbol{W}_2(\delta(\boldsymbol{W}_1\boldsymbol{z})))$, where $\boldsymbol{W}_1 \in \mathbb{R}^{C/r\times C}$ and $\boldsymbol{W}_2 \in \mathbb{R}^{C\times C/r}$ are the weight matrices, r is the reduction ratio, δ denotes the ReLU activation, and σ represents the sigmoid function. The final push-pull response with channel attention is computed as:

$$\boldsymbol{U} = \boldsymbol{U}'_p - \alpha \cdot \boldsymbol{U}_q, \quad \boldsymbol{U}'_p = \boldsymbol{U}_p \odot \boldsymbol{w}, \tag{2}$$

where $\odot$ represents channel-wise multiplication of $\boldsymbol{u}_p^c$ and attention-weights $\boldsymbol{w}$.

This attention mechanism naturally enhances robustness to corruptions by dynamically prioritizing channels that contain stable, semantically relevant features. When input images are corrupted by noise or other distortions, different channels are affected to varying degrees. Channels that capture fundamental structural information tend to maintain consistent activation patterns even under corruption, while channels sensitive to fine details or textures become less reliable.

Utilizing Multi-Scale Pull Activations

To enhance scale invariance, we extend the push–pull mechanism to incorporate pull activations at multiple spatial scales. Given a push kernel $\boldsymbol{K}_p$, we generate a set of pull kernels $\{\boldsymbol{K}_q^1, \boldsymbol{K}_q^2, \ldots, \boldsymbol{K}_q^S\}$ at S different scales by bilinear upsampling:

$$\boldsymbol{U}_q^s = \boldsymbol{X} \circledast (-\boldsymbol{K}_q^s) \tag{3}$$

where s denotes the scale index. For each scale, the push–pull response is computed as:

$$\boldsymbol{V}^s = \boldsymbol{U}_p - \alpha \cdot \boldsymbol{U}_q^s \tag{4}$$

The multi-scale features are concatenated along the channel dimension : $\boldsymbol{U}_{concat} = [\boldsymbol{U}_p, \boldsymbol{V}^1, \boldsymbol{V}^2, \ldots, \boldsymbol{V}^S]$, and then re-calibrated with channel attention similarly as in Eq. (2) : $\boldsymbol{U}'_{concat} = \boldsymbol{U}_{concat} \odot \boldsymbol{w}$. To reduce dimensionality and limit computational complexity, the concatenated feature map is processed through a 1×1 convolution layer:

$$\boldsymbol{U}_{final} = \mathrm{Conv}_{1\times1}(\boldsymbol{U}'_{concat}) \tag{5}$$

Fact that most of these channels have redundant information, also makes it a natural choice. This multi-scale approach allows the network to account for corruptions occurring across different receptive field sizes.

Experimental Results and Discussion

The baseline architectures used in our experiments are based on ResNet-20 and DenseNet-40 (with a growth rate of 12). To evaluate the impact of the proposed modifications, we replaced approximately one-third of the top CNN layers (*Block 1*) with push–pull layers in both architectures. The rationale behind this design choice is discussed in detail in *Ablation studies* section below. All models were trained on non-corrupted CIFAR-10 images using standard data augmentation techniques, including random cropping and horizontal flipping. Training was conducted for 120 epochs using a batch size of 2048. The optimization was performed using stochastic gradient descent (SGD) with momentum. We evaluated the baseline and our proposed enhanced configurations on both clean CIFAR-10 and CIFAR-10-C. CIFAR-10-C introduces 19 different corruption types, including noise, blur, weather effects, and digital distortions, each applied at five levels of severity. The evaluation aimed to measure the model's accuracy and robustness across different corruption types and to quantify the trade-off between robustness and clean data performance.

Noise Type	ResNet-20					DenseNet-40				
	PP ReLU (baseline)	PP + GELU (new baseline)	PP attn (ours)	PP attn + GELU (ours)	Mult. Scale PP attn + GELU (ours)	PP ReLU (baseline)	PP + GELU (new baseline)	PP attn (ours)	PP attn + GELU (ours)	Mult. Scale PP attn + GELU (ours)
Clean CIFAR-10	82.05	82.63	79.51	80.00	**83.66**	**79.52**	78.62	74.23	75.2	79.04
Gaussian Noise	57.06	62.9	**65.90**	63.28	56.85	52.15	56.36	62.26	**64.12**	56.98
Shot Noise	62.83	67.82	**69.58**	67.48	63.90	57.86	61.54	66.02	**67.32**	62.87
Impulse Noise	57.99	**62.92**	59.65	57.02	57.61	56.38	57.15	57.28	**58.41**	58.33
Speckle Noise	63.84	68.33	**69.97**	67.74	64.96	58.90	61.91	65.86	**67.41**	63.30
Avg Noise Acc	60.43	65.49	**66.27**	65.43	60.54	56.32	59.24	62.85	**64.35**	60.92
Gaussian Blur	67.75	69.03	75.21	**76.05**	73.32	62.29	62.70	69.48	**70.08**	63.63
Defocus Blur	72.62	73.53	77.06	77.86	**76.82**	67.84	67.72	71.38	**71.92**	68.50
Glass Blur	72.04	72.53	75.77	**76.66**	74.55	65.54	67.49	71.37	**72.05**	69.66
Motion Blur	67.99	68.61	72.99	**73.43**	71.71	61.97	62.69	67.56	**68.22**	63.74
Zoom Blur	69.10	70.86	75.92	**76.42**	74.48	63.58	63.91	69.68	**70.51**	65.24
Avg Blur Acc	69.9	70.91	**75.39**	71.34	74.17	64.66	64.9	69.89	**70.55**	66.15
Snow	72.77	72.25	73.15	74.08	**75.34**	67.47	67.79	67.44	67.48	**69.23**
Frost	70.73	70.45	71.18	72.19	**73.95**	64.43	65.26	64.46	63.60	**66.12**
Fog	66.65	64.88	62.06	63.91	**68.16**	**62.09**	57.77	55.08	55.91	60.17
Brightness	79.41	78.89	76.24	76.37	**79.83**	**75.42**	74.05	70.45	69.83	74.19
Contrast	49.93	49.74	46.88	49.17	**51.32**	**46.49**	43.62	42.17	43.28	44.17
Elastic Transform	75.03	75.4	75.87	76.62	**77.49**	70.77	70.09	70.83	**71.63**	71.13
Pixelate	80.6	80.45	78.82	79.65	**81.81**	74.71	75.59	73.69	74.61	**77.05**
JPEG Compression	75.07	76.79	73.35	73.49	**76.08**	71.68	**72.65**	69.29	69.37	72.39
Spatter	76.80	77.11	75.09	76.11	**78.41**	73.60	72.57	70.48	70.96	**74.31**
Saturate	72.39	73.45	68.94	68.78	**72.96**	69.20	**66.96**	61.10	61.45	66.76
Avg Corruption Acc	68.98	70.31	70.72	70.86	**70.97**	64.40	64.62	65.57	**66.20**	65.67

Table 1: Evaluating different push-pull layers on CIFAR-10 and CIFAR-10-C for ResNet-20 and DenseNet-40. For readability purpose, the best results are in **bold** and the second best results are underlined.

Model Comparison for Noise Robustness

Table 1 shows the detailed classification accuracy for each corruption type on the CIFAR-10-C dataset. The proposed variants consistently outperformed the baseline PP-CNN (*PP ReLU*) across most corruption types. The evaluated variants include: push–pull CNN with unrectified operations and attention (*PP attn*), PP-CNN with unrectified operations and attention combined with GELU activation (*PP attn + GELU*), and MSPP CNN with unrectified operations and attention (*Mult. Scale PP attn + GELU*).

A key modification in all attention-based variants is the direct linear subtraction between push and pull responses without rectification. Unlike the original PP-CNN, which applies ReLU to both responses, this approach retains both positive and negative signals, allowing more effective contrast sharpening and improved feature extraction. For a fair comparison, the activation functions in all models were replaced with GELU (marked by + *GELU*), as it demonstrated slightly better performance than ReLU in our preliminary experiments.

Incorporating channel attention consistently improved performance across the majority of corruption types. The proposed models showed particularly strong performance on *Noise* and *Blur* corruptions, where attention-based feature recalibration helped to suppress irrelevant patterns and amplify discriminative features. However, for certain corruptions, such as *Saturate*, *Fog*, and *JPEG Compression*, the baseline push–pull configuration outperformed the enhanced variants, especially in the DenseNet-40 architecture. This suggests that some types of image distortions may benefit more from the baseline push–pull's suppression mechanism.

Integrating attention improved robustness to corrupted inputs but introduced a slight reduction in accuracy on clean CIFAR-10 images in most configurations. Nevertheless, the *Mult. Scale PP attn + GELU* configuration consistently achieved either comparable or superior performance in almost all corruption types while maintaining competitive accuracy on clean CIFAR-10. This balance between robustness and accuracy is examined further in the next section.

Multi-Scale Push–Pull (MSPP)

Our modifications involving unrectified push–pull operation and channel attention strengthen the model's inherent bias toward feature sharpening. This sharpen-

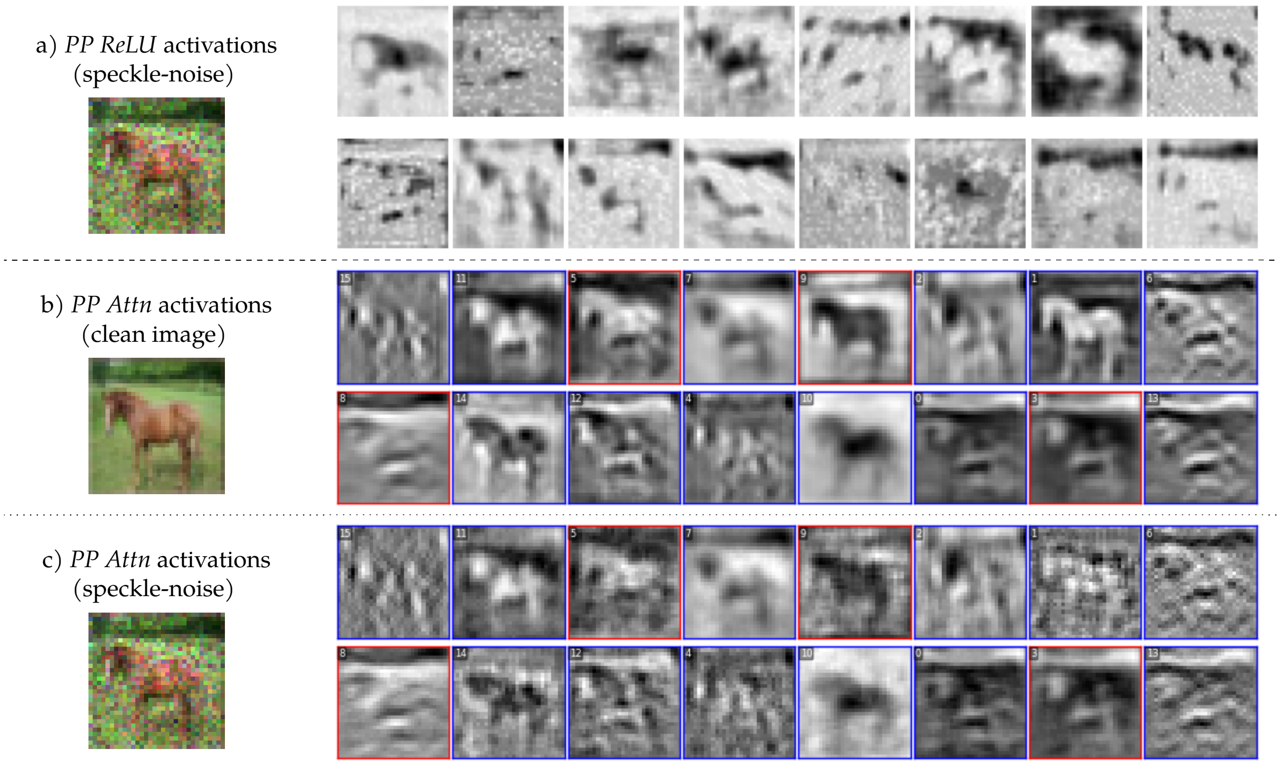

Figure 3: First layer activation visualizations for *PP ReLU* (*Baseline*) and *PP attn + GELU* on ResNet-20. (a) *PP ReLU* (*Baseline*) activations on speckle-noise corrupted horse image (misclassified as deer). (b) *PP attn + GELU* activations on clean horse image. (c) *PP attn + GELU* activations on speckle-noise corrupted horse image (correctly classified). Blue and red borders indicate channels with decreased and increased attention scores respectively under noise corruption.

ing effect enhances the model's ability to handle noise and blur corruptions by reinforcing consistent patterns across scales, making it easier to distinguish meaningful signals from noise. However, this same sharpening tendency can reduce performance on clean images and certain corruption types, where excessive sharpening may amplify irrelevant details and noise artifacts. To mitigate this trade-off, we propose a Multi-Scale Push–Pull (MSPP) mechanism that preserves information from both the original and enhanced feature maps. MSPP concatenates the original activation map (push) with a set of sharpened activations generated using pull kernels at multiple scales, as shown in Fig. 2. This enables the push–pull layer to consider for corruptions that can occur at different spatial scales while also preserving fine-grained details of original push kernel response necessary for accurate classification on clean images.

The multi-scale features are processed through a 1×1 convolution layer to reduce dimensionality and control computational complexity. This dimensionality reduction is particularly effective because the multi-scale features often contain redundant information, as they are derived from the same push kernel at different scales. By consolidating this information, the network can focus on the most relevant patterns while minimizing the computational burden.

Our approach is conceptually similar to the Inception architecture (Szegedy et al. 2016), which utilizes feature extraction at multiple scales. However, it differs in a critical aspect: instead of extracting different types of features at varying scales, we enhance the same set of features using inverse kernels (pull responses).

By combining multi-scale processing with channel attention and unrectified push–pull operations, model more resilient to real-world corruptions while preserving competitive accuracy on clean data.

Attention Visualization and Analysis

Figure 3 presents response map visualizations from the first Push–Pull layer in ResNet-20. The baseline Push–Pull model (Figure 3a) misclassifies a noise-corrupted horse image as a deer, whereas the Push–Pull-Attention model successfully identifies it. This improvement stems from the attention mechanism's ability to dynamically adjust channel activations based on their discriminative importance, thereby enhancing the network's ability to focus on relevant patterns while suppressing noise.

Figure 4 illustrates the channel-wise attention score differences between the noisy and clean versions of the same horse image, corrupted by speckle noise. Chan-

Noise Type	ResNet-20				DenseNet-40			
	PP attn (*ALL*)	PP attn (*Block 1*)	PP attn + GELU (*ALL*)	PP attn + GELU (*Block 1*)	PP attn (*ALL*)	PP attn (*Block 1*)	PP attn + GELU (*ALL*)	PP attn + GELU (*Block 1*)
Clean CIFAR-10	76.48	**79.51**	77.64	**80.00**	**76.15**	74.23	**76.96**	75.2
Avg Noise Acc	**68.86**	66.27	64.84	**65.43**	**66.55**	62.85	53.12	**64.35**
Avg Blur Acc	73.07	**75.39**	**73.9**	71.34	**72.15**	69.89	**72.46**	70.55
Avg Corruption Acc	68.96	**70.72**	68.80	**70.86**	65.54	**65.57**	**67.97**	66.20

Table 2: Effect of swapping CNNs with PP-CNN on CIFAR-10 and CIFAR-10-C

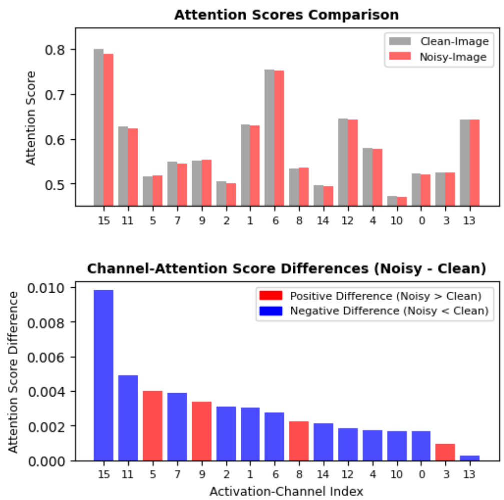

Figure 4: Channel-wise attention score differences between noisy and clean image activations. Blue bars indicate channels with decreased attention in the noisy image compared to the clean image, while red bars show channels with increased attention.

nels are sorted by the magnitude of the attention score difference, with blue bars indicating reduced attention in the noisy condition and red bars indicating increased attention. Notably, channels that receive reduced attention in the noisy image (e.g., channels 1, 6, and 10) exhibit more blurred and less discriminative activations, as seen in Figures 3b and 3c.

This analysis suggests that the attention mechanism dynamically reallocates focus toward channels that preserve stronger and more stable feature representations, even under noise corruption. The mechanism appears to prioritize channels with visually pronounced activations while suppressing those where noise has weakened the signal quality. This adaptive channel reweighting enables the network to retain robust performance even under significant image distortions, effectively enhancing the model's resilience to real-world input variations.

Ablation Study

To assess the impact of replacing standard convolutional layers with push–pull layers, we conducted an

Model Variants (PP attn)	# of params	Clean CIFAR-10 Acc	Avg Noise Acc	Avg Blur Acc	CIFAR-10-C Acc
ResNet-20					
SE Attn.	273510	80.00	65.43	71.34	70.86
SimAM	272474	80.54	71.00	72.92	70.5
DenseNet-40					
SE Attn.	176962	75.2	64.35	70.55	66.2
SimAM	176122	74.47	61.95	69.95	65.35

Table 3: Effect of different attention mechanisms (SE and SimAM) on CIFAR-10 and CIFAR-10-C, evaluated on ResNet-20 and DenseNet-40

ablation study by progressively substituting layers in ResNet-20 and DenseNet-40. As shown in Table 2, ResNet-20 with push–pull layers integrated into the top layers (marked as *Block 1*) demonstrated a significant improvement in performance. In contrast, replacing all convolutional layers with push–pull layers (marked as *ALL*) resulted in only marginal gains in DenseNet-40.

These findings align with observations from Strisciuglio et al. (2020) that the initial layers of CNNs are most vulnerable to corruptions in input image. This makes them ideal candidates for the push–pull mechanism's sharpening effect which enhances contrast. Conversely, deeper layers process high-level, abstract features that are less affected by noise, which explains the diminishing returns observed when replacing deeper layers with push–pull convolutions. Additionally, each push–pull layer linearly increases computational and memory costs for the computation of pull activations and channel attention. This informed our design choice to restrict push–pull layers to the top layers, striking a balance between robustness gains and computational efficiency.

We also examined the impact of different attention mechanisms by comparing SE attention with parameter-free attention (SimAM). Both attention mechanisms were evaluated on ResNet-20 and DenseNet-40 architectures. As shown in Table 3, SE attention consistently outperformed the parameter-free variant across most metrics, including clean accuracy, noise robustness, and blur robustness. These results are consistent with our discussion on SE attention, confirming that learnable channel recalibration plays a critical role in improving feature discriminability and overall robustness.

Conclusion

In this work, we introduced three key enhancements to the PP-CNN architecture that significantly improved its robustness to common image corruptions. First, by removing the half-wave rectification from the push–pull mechanism, we transformed it into a linear feature enhancement filter that more effectively sharpens scale-consistent patterns. Second, the introduction of a dynamic channel attention mechanism enabled adaptive cross-channel interactions, allowing the network to emphasize the most discriminative features even under challenging corruption conditions. Third, the multi-scale framework leveraged pull responses at different scales, enhancing the model's ability to distinguish between consistent visual patterns and noise-induced activations across varying spatial resolutions. Experimental results on the CIFAR-10-C dataset demonstrated that the enhanced architecture substantially improved performance across a wide range of corruption types, particularly for noise and blur. On the whole, the multi-scale approach, in particular, addressesed the typical robustness–accuracy trade-off by improving performance on both clean and corrupted data.

References

Cubuk, E.; Zoph, B.; Shlens, J.; and Le, Q. R. 2020. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 3008–3017.

Cubuk, E. D.; Zoph, B.; Mane, D.; Vasudevan, V.; and Le, Q. V. 2019. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 113–123.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 248–255.

Geirhos, R.; Janssen, D. H.; Schütt, H. H.; Rauber, J.; Bethge, M.; and Wichmann, F. A. 2017. Comparing deep neural networks against humans: object recognition when the signal gets weaker. *arXiv preprint arXiv:1706.06969*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and harnessing adversarial examples. In *Proceedings of the 3rd International Conference on Learning Representations*. San Diego.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778. Las Vegas, NV, USA.

Hendrycks, D.; and Dietterich, T. 2019. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*.

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4700–4708. Honolulu, HI, USA.

Ilyas, A.; et al. 2019. Adversarial examples are not bugs, they are features. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 125–136.

Ioffe, S.; and Szegedy, C. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, 448–456.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 1106–1114.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.

Petkov, N.; and Westenberg, M. A. 2003. Suppression of contour perception by band-limited noise and its relation to non-classical receptive field inhibition. *Biological Cybernetics*, 88: 236–246.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1): 1929–1958.

Strisciuglio, N.; Lopez-Antequera, M.; and Petkov, N. 2020. Enhanced robustness of convolutional networks with a push–pull inhibition layer. *Neural Computing and Applications*.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2818–2826. Las Vegas, NV, USA.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, ; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

Wang, L.; Li, X.; and Zhang, Z. 2024. Dense cross-connected ensemble convolutional neural networks for enhanced model robustness. *arXiv preprint arXiv:2412.07022*.

Wang, L.; Wang, C.; Li, Y.; and Wang, R. 2021a. Explaining the behavior of neuron activations in deep neural networks. *Ad Hoc Networks*, 111: 102346.

Wang, L.; Wang, C.; Li, Y.; and Wang, R. 2021b. Improving robustness of deep neural networks via large-difference transformation. *Neurocomputing*, 450: 411–419.

Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; and Hu, Q. 2020. ECA-net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11531–11539.

Woo, S.; Park, J.; Lee, J.-Y.; and Kweon, I. S. 2018. CBAM: Convolutional block attention module. In Ferrari, V.; Hebert, M.; Sminchisescu, C.; and Weiss, Y., eds., *Computer Vision — ECCV 2018*, volume 11211 of *Lecture Notes in Computer Science*, 3–19. Cham: Springer.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.