ABSTRACT

Capturing long-range feature relationships is becoming a central issue with regard to convolutional neural networks (CNNs). In particular, several recent end-to-end trainable attention modules have attempted to model spatial-channel relationships within a given layer. In this work, we focus instead on modeling relationships among visual information captured in different layers and propose a novel module, referred to as a Parallel Recurrent Module with Inter-layer Attention (PI module). The PI module exhibits several unique characteristics, including the ability to memorize information from earlier layers and ameliorate gradient vanishing, both of which are issues not addressed by existing attention modules. Furthermore, due to its easy-to-adopt structure also incurring negligible computational overheads, the module successfully extends to not only CNNs on regular grids but also to graph convolution networks, and even other attention modules. We demonstrate by extensive experiments that the PI module is cost-efficient yet effectively provides additional performance gains on multiple benchmarks in classification, detection, and segmentation tasks in the image domain and a segmentation task in the point cloud domain.

INDEX TERMS

Deep Learning Network Engineering, Attention Mechanism, Image/Point Cloud Analysis.

I. INTRODUCTION

Convolutional neural networks (CNNs) have become the standard for solving vision tasks. Although recent works on transformer architecture [1] and its families [2], [3] triggered the transition of the CNNs into transformer-based networks, they are vulnerable to inductive bias and require a larger dataset to acquire the locality that CNNs have. Therefore, auxiliary attention-modules that support CNNs are still actively developed [4]–[6]. Several architectural design aspects of CNNs, such as depth, width, cardinality, and spatial-channel relationships, have been investigated for successful feature learning on vision. In particular, studies that use end-to-end trainable attention modules to model spatial-channel relationships inside a convolution block are on the rise [7]–[10]. However, existing attention modules referred to as intra-layer attention modules, only model the interdependencies of neurons inside a single layer. Therefore, they cannot structurally model relationships between neurons in different layers.

To encode inter-layer relationships into the model, we focus on an essential property of CNNs: the feed-forward structure. As CNNs model long-range interactions through the progressive propagation of visual signals by iterative local aggregations, we are motivated to offer a new interpretation of CNNs: concatenating spatially-pooled features in the depth direction forms a sequence of visual features arranged in increasing order of the receptive field size.

Inspired by the missing aspect in current attention modules and the structural characteristic of CNNs, we propose a novel attention module called a ‘Parallel Recurrent Module with Inter-layer Attention (PI module).’ The goal of this work is to achieve the explicit modeling of inter-layer relationships in CNNs. To do this, we design two concepts: a parallel recurrent module (PRM) and an inter-layer attention mechanism. Briefly, the PRM is a small parallel memory stream external to the main CNN that carries attention information along with the propagation of the CNN, and the inter-layer attention mechanism serves to calibrate the network based...
on the layer-wise importance distribution. Furthermore, when incorporating our module into recent advances in spatial-channel attention [7], [8], the PI module is recognized as a generalized attention module that considers the relationships among neurons both inside and outside of a layer.

We thoroughly investigate the impact of each module component and verify the high compatibility of the PI module to Graph Convolution Networks (GCNs) and even to other attention modules. For a fair evaluation, we evaluate our module on image classification, object detection, image semantic segmentation, and point cloud segmentation tasks on multiple benchmarks: CIFAR [11], ImageNet-1K [12], Microsoft COCO [13], Pascal VOC [14], and S3DIS [15].

In general, the PI module alone significantly boosts backbone networks, achieving comparable results to respective state-of-the-art while incurring a negligible increase in computational complexities (GFLOPS, parameters, and inference time). The PI module also improves on state-of-the-art when it is integrated with them.

The contribution of this work can be summarized as:

- We propose a novel attention module that resolves a unique problem that other attention modules do not handle, the inter-layer relationship.
- We demonstrate the performance-boosting effects and generality of the PI module through extensive experiments on multiple benchmarks.
- We show the scalability of the PI module by introducing a structural derivative: ‘PI-Intra,’ showing the compatibility of proposing concepts not only to CNNs but also to existing attention modules.
- We investigate various visualizations for aiding the understanding of the PI module.

II. RELATED WORKS

End-to-end Trainable Attention Module. Emergence of transformer architecture [1] boosted the computer vision literature to widely incorporate the attention mechanism in the work. Consequently, vision-oriented transformer-based approaches including ViT [2] and AANet [3] are proposed, leading to transition of traditional CNN to transformer-based architectures. While the complete transition of the architecture significantly increased the performances in the visual task, ViT and its family requires a larger dataset to train the model properly when compared to the CNNs, since they lack inductive bias. Few approaches [7], [8] introduce an attention-based module to support the main network, whereas the lightness of the module was out of consideration. We opt to design a lightweight simple attention-based module that fosters the power of the network.

An attention mechanism can be interpreted as a methodology by which to bias the allocation of available neurons to the most informative components of input signals [7]. One recent application of attention mechanisms in the area of computer vision is the integration of end-to-end trainable attention modules for deep CNNs. Attention modules can be divided into two categories: a spatial attention module and a channel-wise attention module. Spatial attention modules learn 2-D spatial masks on features for the purpose of regulating the activations of neurons [16], [17], and Channel-wise attention modules learn 1-D channel-wise attention vectors and then utilize it to refine features [7], [10]. Furthermore, several architectural designs that have inherited the virtues of both spatial and channel-wise attention have also appeared [8]– [10], [18], [19].

However, existing spatial and channel-wise attention modules, referred to as intra-layer attention modules, only handle inter-dependencies inside a single layer. In contrast, our module aims to model relationships between layers.

RNNs in Computer Vision. In the field of computer vision, Recurrent Neural Networks (RNNs) are used to (1) model tasks that require sequential decisions or to (2) model non-sequential tasks (e.g., single image classification) as sequential procedures. Numerous vision studies including the tasks of visual question answering [20], [21], image captioning [22], multi-object classification [23], [24], video description [25], and text recognition [26], [27] belong to the first type. They share common ground with regard to how they utilize RNNs; they extract visual features from images using a pre-trained CNN and then utilize a RNN to model the sequential procedures of each task. Among them, the text recognition applications [26], [27] utilize bidirectional RNN mechanism in order to capture the text dependencies in both direction. Studies involving the second type include those that model spatial relationships in features [28] and those that exploit the RNN itself as a methodology by which to understand images [29], [30].

The PI module is similar to previous works [29], [30] in that RNN units carry visual signals. However, our module does not make predictions for tasks directly but rather aids in the representation learning of CNNs by acting as an informative attention source. In terms of RNNs, the most different aspect of the PI module about concerning other studies is that the PI module recurs over the depth dimension of networks, not the spatial dimensions [28], [31]. Thereby, it models inter-layer relationships while leaving parallelized implementation of CNNs unaltered.

III. METHODOLOGY

The PI module is a powerful but lightweight unit that models inter-layer relationships, incurring a negligible increase in the model parameters and computational cost.

Figure 1 illustrates the concept of the PI module. When applied to CNNs, it memorizes the feature hierarchy of earlier layers and regulates the activation of the currently processing layer in the form of self-attention.

We insist that the PI module can be a new practical choice for the network engineering in that it (1) exhibits high extendibility to be adapted to generic architectures, including CNNs, GCNs, and even with other attention modules, and (2) aids the main network both in forward and backward pass. Given its ease of use, as the PI module is cost-efficient
and provides an orthogonal improvement to existing attention modules, it becomes a more favorable option in practice.

The PI module is an integration of two main concepts: the Parallel Recurrent Module and the inter-layer attention mechanism. Below, we explain each component and its benefits and then describe the workflow of the PI module.

A. PARALLEL RECURRENT MODULE

The Parallel recurrent module (PRM) is a small separate memory stream that carries information from earlier layers that are propagated along the depth direction of the main network.

Here, we compare attention inferring operation of recent self-attention modules in vision and the PRM as follows:

\[
x'_i = h_i(x_i), \\
x'_i = h_i(r_i), \\
\text{where } x_i = f_i(x_{i-1}), \quad r_i = g(x_i, r_{i-1}),
\]

and Eq. (1) and Eq. (2) corresponds to existing self-attention modules and PRM, respectively. Here, \( x \) and \( r \) denote convolution features and RNN memory units, respectively. \( f(\cdot) \) is convolution, \( h(\cdot) \) is arbitrary self-attention operation, and \( g(\cdot, \cdot) \) is RNN update function (layer index \( i \) is omitted for brevity). As implied by Eq (2), the PRM is an extension of self-attention that employs internal memory units for explicitly using previous features \( x_{1}, \ldots, x_{i-1} \). According to this concept, by recurrently memorizing co-propagating CNN features, we can extend the source of self-attention to the inter-layer level. The PRM can be used with any attention function \( h \), but in this paper, we focus on using the proposed inter-layer attention mechanism.²

In the following paragraphs, we describe how PRM assists with the feature learning of CNNs.

Memorizing Feature Hierarchy across Layers. CNNs model long-range interactions through the propagative progression of visual signals via iterative local feature aggregation.

Focusing on the feed-forward structure of CNNs, we hypothesize that spatially pooled features obtained through this iterative process are viewed as a sequence of visual features with small receptive fields to global features. To this end, we are motivated to encode that sequence in an RNN following the propagation of the CNN.

As RNNs carry inductive bias via their Markovian structure [32] for the locality in the sequence, the PRM encodes the causalities of layers. For example, the dependency of a zebra’s torso captured in the third layer of the CNN on the zebra’s stripes captured in the first layer can be embedded in PRM. This is enabled by the fact that the PRM recurs over the depth direction of the main network. Moreover, it is important to note that the PRM does not degrade the parallelizable capability of CNNs; it differs from previous works [28], [31] that encode spatial information by recurring over the spatial dimensions of the feature volume using an RNN.

Providing Light-weight Information Highway. In practice, memorization in PRM is effectively strengthened by the exploitation of LSTM, as this enables the regulation of the information of visual features captured at earlier convolutional layers using the forget gate. In this sense, the concept of gating the information flow employing LSTM is reminiscent of a Highway Network (HN) [33], as Highway Network models the transformation between input and output by adopting the non-linear transformations, denoted as ‘transform gate’ and ‘carry gate’. However, the PI module is a more practical choice deviating from the HN in two ways: layer-wise connections and a computational advantage.

First, the PRM operates as an extra path for information flows. During a forward pass, the PRM provides crucial inductive bias, whereby the attention values (transform gate in an HN) are inferred using the memory units. Interestingly, the extra path provided by the PRM is also effective during a backward pass, ameliorating the vanishing gradient problem. This is the characteristic that distinguishes the PRM from the HN and also from existing intra-layer attention modules.

Second, the proposed PRM is more efficient than highway networks, which suffer from the computational costs...
associated with what is termed “expressive gating.” This refers to the use of identically sized convolution filters for the transform gate and the plain layer so as successfully to train the Highway Networks. While larger kernels for the transform gate result insufficiently expressive transformations, increasing the kernel size increases the computational costs. On the other hand, the PRM does not require large and expensive filters for gating, as it operates on spatially pooled 1-D features. In general, the use of 1-D global pooled features and weight sharing makes the PRM significantly light in terms of computational complexities. Please refer to Table 1 in Section IV and Appendix D, E, F. Furthermore, the weight sharing scheme can also be viewed as a type of regularization to support generalization.

In short, the PRM is a new type of learned information highway. Importantly, because the PRM is an architectural unit that only requires a multi-layer structure to be applied, it is highly extendable to arbitrary networks.

B. INTER-LAYER ATTENTION MECHANISM

The inter-layer attention mechanism serves to regulate features by multiplication between the output of convolutional transformation and the correspondingly inferred layer importance value (scalar). By calibrating the network using a layer-wise importance distribution, the inter-layer attention mechanism explicitly models the inter-dependencies between neurons of different layers. In the PI module, we adopt the inter-layer attention mechanism to the feature from the CNNs and the layer attention value inferred from the internal memory units. By introducing the inter-layer attention mechanism, we can empower the CNNs to regulate layer activations selectively by emphasizing layers that catch salient visual information or reducing layers that acquire redundant visual information. This differs from intra-layer attention mechanisms [7], [8] in that those mechanisms calibrate a particular layer using an inferred spatial-channel importance distribution. Figure 2 (Top) depicts the process of the inter-layer attention mechanism along with the feed-forward structure of the CNNs.

C. WORKFLOW

The PI module recurs by \( L \) processing steps, becoming connected immediately after the arbitrary convolutional transformations. Here, the decision of \( L \) depends on how the PRM connects to the CNNs. A graphical representation of the PI module is illustrated in Figure 2 (Top) and the following equations represent the workflow:

\[
\begin{align*}
    s_l &= \varphi_l(f_l), \\
    r_l &= RNN(s_l, r_{l-1}), \\
    \alpha_l &= \phi_l(r_l) = \sigma(W_l^2 \delta(W_l^1 r_l)), \\
    f_l^* &= \alpha_l \ast f_l, \\
    f_{l+1} &= Conv(f_l^*), \\
    r_0 &= 0, \quad t > 0.
\end{align*}
\]

At each time step \( l \), the PI module obtains an RNN input \( s_l \in \mathbb{R}^N \) by passing a conveyed feature from the CNNs \( f_l \in \mathbb{R}^{H_l \times W_l \times C} \) to a summarizing function \( \varphi_l \) and then updates memory units \( r_l \in \mathbb{R}^M \) using a previous \( r_{l-1} \) and \( s_l \). The internal memory \( r_l \) infers the layer attention value \( \alpha_l \in [0, 1] \) using an inter-layer inferring function \( \phi_l \), after which the layer attention value \( \alpha_l \) serves as a scalar coefficient for calibrating a feature \( f_l \) with a Hadamard product \( \ast \). Lastly, the scaled feature \( f_l^* \) is converted by a subsequent convolutional transformation \( Conv \) and fed back to the RNN’s next time step.

The PI module maintains this recurrence until CNN propagation ends, updating the RNN internal memory by utilizing the summarized layer statistic as the RNN input for each time step (convolutional transformation) and inferring the inter-layer attention value from the updated RNN internal memory. For model learning, the parameters of the PI module are jointly optimized with the CNN parameters. Detailed designs of Eq. (4) and (6) are discussed below.

Feature Summarization Function. \( \varphi_l \) refers to the combination of the feature summarizing operation and the consequent down-sampling (or up-sampling) operation. A parametric study led to the decision to adopt the ‘global average pooling (GAP)’ operation. Please refer to Appendix C. After feature summarization, the channel size must be adaptively adjusted, as RNNs require a fixed input size. To acquire the fixed-size inputs, we opt for max-pooling or interpolation as these include no learnable parameters.
Inter-layer Attention Inferring Function. \( \phi_l \) consists of two fully-connected layers with \( W^1_l \in \mathbb{R}^{M \times M} \) and \( W^2_l \in \mathbb{R}^{1 \times \frac{M}{l}} \), \( \delta \) and \( r \) indicate ReLU [34] and sigmoid operations, respectively. Reduction ratio \( r \) denotes the compressing ratio of the first fully-connected layer. The inter-layer attention inferring function can be viewed as an arbitrary function of \( s_l \) and \( r_{l-1} \), and we use a multi-layer perceptron with two hidden layers to approximate this function. Note that we found that the sharing of an inter-layer attention inferring function over layers causes a slight degradation of the performance in our early experiment; consequently, we assign a separate function for layers. This only incurs a minor increase in the number of parameters.

D. EXTENDABILITY OF PI MODULE: SKIP CONNECTION, GRAPH CONVOLUTION, AND ATTENTION MODULES.

In this section, we extend the PI module to other architectural units beyond normal CNNs for demonstrating that it is a practical choice for general application.

PI Module with Residual Architectures. It is intuitive to make links for connecting the PI modules to non-residual architectures. However, when applying the PI module to residual architectures, the following facts are required to be clarified: the locations of connections between the PI module and the CNN, and the logic behind the selections of such locations. Among possible candidates, we opt for linking the PI module to residuals, which is the same strategy as intra-layer attention modules [7], [8] adopt. Inter-layer attention mechanism with residual architectures solves a preceding problem in an elegant way: adjusting residuals by inferred layer importance.

PI Module with Graph Convolution Networks. We enhance our scope in demonstrating the generality of PI modules, beyond architectures defined on the regular 2-D grids to graph convolution networks in the irregular domain. Note, several intra-layer attention modules that handle complex spatial-wise statistics [31], [35] do not apply to irregular point cloud or graph domains because of their permutation variance.

Integration to Attention Modules. By virtues of the weight sharing of PRM and inter-layer attention mechanism, PI module achieves an extreme lightness. In ResNet50 case, the PI module requires \( \times 10, \times 23, \times 31 \) smaller additional parameters than SE [7]/CBAM [8], GE [10], \( A^2 \)-Net [9] respectively, while adding negligible FLOPs.

This lightness of the PI module leads us to integrate it with intra-layer attention modules as well as the backbone CNNs, thereby we link the PI module after the calibrated features with the intra-layer attention mechanism. Figure 2 (Bottom) illustrates the graphical representation of PI-Intra.

IV. EXPERIMENTS

In this section, to demonstrate the effectiveness and high extendability of the PI module, we report extensive experimental results on various tasks and datasets including CIFAR-10/100 and ImageNet-1K for image classification, Microsoft COCO for object detection, Pascal VOC for image segmentation, and S3DIS for point cloud segmentation. For thorough evaluation of performance on PI module, we also reproduce experiments on backbone networks, those with SE [7] and CBAM [8] except a few asterisked instances borrowed from previous papers. To ensure a fair comparison, we apply the same optimization and augmentation scheme for model instances that are compared; our implementations mostly follow official GitHub repositories of previous studies [8], [36], [37] and all details are available in Supplementary Material.

We validate the PI module on image classification task using ImageNet-1K 2012 dataset [12]. To verify the validity of our designs, extensive experiments are performed on a variety of conditions: architectures, datasets, and depth.

Model Comparison. To demonstrate the extent of the additional performance gains and the computational cost provided by the PI and PI-Intra modules compared to recent attention modules, Table 1 provides a comparison with the outcomes of several recent attention studies. Here, for PI-Intra, a SE block is adopted, denoted as ‘PI-SE’ in the table.

Performance Gains. The PI module significantly improves the performance of backbone architectures, showing similar performance to the SE block. Furthermore, notably, PI-SE-ResNet50 demonstrates better performance than the PI module and SE block alone, showing an additional reduction of 0.41% of the top-1 error compared to SE-ResNet50. Considering that GE [10] and \( A^2 \)-Net [9] present marginal improvements compared to SE-block with more parameters (+5.6M and +7.2M, respectively; for, PI-: +0.2M, PI-SE: +2.8M), the margin is quite significant. This successful integration of ‘inter-intra attention’ provides evidence that the PI module operates with other intra-layer attention modules in an orthogonal manner.

Computational Advantages. Importantly, the PI module only incurs ‘negligible’ computational overheads. Practically, except for SE, recent attention modules sacrifice their lightness to ensure better performance. In terms of the inference time, most attention networks, except for PI and SE, are \( \times 2 \sim 3 \) slower than the original ResNet. Additionally, the PI module only adds a \( \approx 1\% \) increase in the model parameters and a negligible increase in the number of GFLOPs, distinctly surpassing the others with regard to each metric.

On the other hand, PI-SE surpasses GE and \( A^2 \)-Net in all metrics and provides performance gains very similar to CBAM, while being nearly \( \times 2 \) faster.

In short, this comparison supports our contention that the PI module efficiently uses these parameters and exhibits high extendability to a wide range of applications due to its lightness.

ResNet and MobileNet Backbone. Table 2, 5 summarize self-investigation results on PI and PI-Intra for ResNet50 and MobileNet [40]. We observe that PI networks record similar performance with smaller computational costs. Also,
TABLE 1: Comparison table of PI and PI-Intra to recent attention modules: CBAM [8], GE [10], $A^2$-Net [9] on ImageNet-1K classification using backbone ResNet50 (R50). Inference time, parameter number, GFLOPs, and delta Top-1% single-crop error compared to SE-ResNet50 are reported, where the inference time is measured by a processing time of one minibatch in test phase using a Titan Xp.

| Models | GFLOPs | MParams (△%) | Top-1% | Top-5% |
|--------|--------|---------------|--------|--------|
| R50    | 3.86   | 25.6          | 24.84  | 7.55   |
| SE-R50 | 3.86   | 28.1 (9.9%)   | 23.64  | 7.03   |
| PI-R50 | 3.86   | 25.8 (1.1%)   | 29.67  | 7.05   |
| PI-SE-R50 | 3.86 | 28.4 (10.6%) | 23.23  | 6.72   |

TABLE 2: ImageNet-1K classification with ResNet50 (R50)

|    | GFLOPs | MParams (△%) | C10 | C100 |
|----|--------|---------------|-----|------|
| R20 | 0.041  | 0.275         | 8.75| 32.21|
| SE20| 0.041  | 0.278 (0.86%) | 7.96| 31.08|
| PI20| 0.041  | 0.276 (0.30%) | 7.48| 31.77|
| R32 | 0.070  | 0.471         | 7.51| 31.90|
| SE32| 0.070  | 0.475 (0.84%) | 7.32| 30.34|
| PI32| 0.070  | 0.471 (0.20%) | 6.83| 29.65|
| R44 | 0.099  | 0.667         | 7.17| 31.13|
| SE44| 0.099  | 0.672 (0.83%) | 6.64| 29.54|
| PI44| 0.099  | 0.668 (0.17%) | 6.68| 28.75|
| R56 | 0.127  | 0.858         | 6.97| 29.19|
| SE56| 0.127  | 0.871 (1.51%) | 6.30| 28.08|
| PI56| 0.127  | 0.859 (0.14%) | 6.40| 27.15|
| R110| 0.256  | 1.735         | 6.61| 28.95|
| SE110| 0.256  | 1.755 (1.15%) | 6.04| 27.01|
| PI110| 0.256 | 1.737 (0.11%) | 5.92| 26.40|

TABLE 3: Network depth analysis on CIFAR-10/100 classification. Top-1% errors are reported.

Table 3, 4 summarizes results. Notably, for CIFAR experiments, we find out that PI networks distinctily surpass ResNets in performance and most of them also outperform SE networks; PI-ResNet44 outperforms ResNet10 at approximately one-third of model parameters. For ImageNet-1K experiments, we verify that the PI module improves backbones to a similar level to SE networks while being lighter.

TABLE 4: Network depth analysis on ImageNet-1K classification.

|    | GFLOPs | MParams (△%) | Top-1% | Top-5% |
|----|--------|---------------|--------|--------|
| R18 | 1.82   | 11.7          | 29.61  | 10.47  |
| SE18*| 1.82  | 11.8 (0.76%)  | 29.41  | 10.22  |
| PI18| 1.82   | 11.7 (0.33%)  | 29.32  | 10.30  |
| R101| 7.61   | 44.5          | 22.99  | 6.51   |
| SE101*| 7.62  | 49.3 (10.72%) | 22.38  | 6.07   |
| PI101| 7.64   | 44.8 (0.64%)  | 22.27  | 6.37   |

TABLE 5: ImageNet-1K classification with MobileNet (M). VOC 2012 [14] dataset. For backbone networks, we use (PI) ResNet50 with the output stride of 16. Table 6 shows that the PI module significantly boosts performance of ResNet50-16s instance (+7.7 mIoU).

B. POINT CLOUD SEGMENTATION

To demonstrate the extendibility of the PI module to GCNs and point cloud domain, we conduct segmentation experiments using the S3DIS [15] dataset. For backbone networks, we opt for DeepGCNs [36]. To extend the PI module in point cloud networks, we replace the global average pooling over spatial dimension to point dimension. Table 7 shows that the PI module effectively boosts performance of ResGCN instances; +1.7% mIoU for PI-ResGCN-7, +0.9% mIoU for PI-ResGCN-28. Figure 3 displays a training curve for showing that PI-networks gets consistently lower training loss along with the learning schedule.

C. OBJECT DETECTION

We conduct object detection experiments using Microsoft COCO 2014 [13] dataset. We adopt Faster R-CNN [41] as a detection method and replace backbone networks from ResNet to PI-ResNet. As results, PI-ResNet101 provides additional performance gains (+1.1% mAP@.5, +0.6% mAP@[.5, 95]) compared to ResNet101.

V. ANALYSIS
A. DISENTANGLING PI MODULE.

To investigate the PI module fully, we disentangle the impacts of the PRM and the inter-layer attention mechanism. Furthermore, we decouple how the co-propagating recurrent network affects the feature learning of CNNs.

**Ablate PRM from PI module.** For a better understanding of the PI module, we isolate the impact of the PRM by ablating it. We replace the RNN update function (Eq (5)) with linear projection followed by tanh non-linearity (i.e., no inter-layer connection). Table 8 indicates that the performance is downgraded by a distinguishable margin after removing the PRM from the PI module. This finding suggests the validity of recurrent networks for PRM.

**PI module in Backward Pass.** We claim that LSTM adopted for the PI module eases the feature learning of CNNs by providing an extra information highway through which gradients can be directly backpropagated. To investigate the extent to which the PRM ameliorates vanishing gradients, we compare the respective gradients of ResNet50 and PI-ResNet50 during training.

Figure 4 describes the behaviors of the mean absolute gradient value upon the input of each network layer. At the beginning of training (Epoch 0), we observe that P150 does not preserve the gradient flow compared to R50. This is the intended phenomenon for relieving the PI module of the need for a careful parameter initialization scheme, a known issue with a Highway Network [42]. Notably, we do not exploit any gradient-preserving strategies such as a negative bias initialization for the LSTM forget gate or the initialization of the inter-layer attention value to 1.

As the training progresses (Epoch 1), the mean absolute gradients of P150 become larger than those of R50 at most of the layers, thereby inducing faster learning [43]. More interestingly, as training continues to progress (Epochs 2 and 100), we find that the PI module distributes the gradients of the last layer, leaking them from the backbone CNN to the module and providing them to earlier layers. In this sense, the PI module shows that it aids the feature learning of CNN during a backward pass.

### Table 7: Scene semantic segmentation results on the area 5 of S3DIS [39].

| Architecture   | mIoU |
|----------------|------|
| ResNet50-16s   | 64.5 |
| PI-ResNet50-16s| 72.2 |

**B. MEAN INTER-LAYER ATTENTION CURVE**

Figure 5 presents the mean layer attention curves across the layers of PI networks. We found that the investigations of the layer attention value curves enable a deeper understanding of the residual architecture, described as follows.

**Alternative for Interpreting Residual Architectures.** The layer attention values are learned such that they are high in the early layers and relatively low in the intermediate layers. We claim that this observation could be another means of gaining a deeper understanding of residual architectures. The intuition behind residual learning is to let the layers learn the perturbations regarding an identity function, with this intuition supported by displaying the standard deviation of the batch normalization layer outputs, with the contention that they insist the smaller standard deviation represents the lower ‘response strength’ of the residual function.

Compared to this method, we note that our plots provide similar but more straightforward information. Given the observations of the decreased layer attention values from the early layers to the intermediate layers, we could support an intuition of residual learning, as the deep layers learn less important perturbations. In short, according to the essence of the attention mechanism, by allocating available neurons to more informative components, the PI module supports recent investigations that interpret residual architectures.

**Increasing variance of layer attention values as going deeper.** The variance of the layer attention values gradually increases along the depth direction of the network until the last residual block is reached, and the layers in the last block have the most significant layer attention value close to 1. Following the generic understanding of the interpretation of CNNs, i.e., learning simple and complex features in earlier and deeper layers respectively [44], it is not surprising that PI networks allocate large layer attention values for the last layers of CNNs, which are semantically salient, and that the layer attention values drastically vary in deep layers that contain abstract visual information.

### C. GRAD-CAM VISUALIZATION

We hypothesized that PI networks learn better representations by considering repeated appearances of visual information. Examining our hypothesis and semantically understanding how PI networks allocate neurons, we apply Grad-CAM [45] across the intermediate layers (Last 1x1 convolu-
E. Kim et al.: Preparation of Papers for IEEE TRANSACTIONS and JOURNALS

2.5×10^4
5×10^4
7.5×10^4
10^3
1.25×10^3
1.5×10^3
1.75×10^3
Mean absolute grad.

FIGURE 4: Mean absolute gradient magnitude of PI50’s each layer input at Epoch 0, 1, 2, 100, training on ImageNet-1K classification. ±1 stdev. are also illustrated (X: layer, Y: magnitude).

Mean Layer Attention

FIGURE 5: Mean layer attention curves of PI-ResNet18/50/101 (Left/Middle/Right). Data points signify mean layer attention values on ImageNet-1K validation dataset. The vertical range on each point denotes an ±1 sample standard deviation. For visibility, we mark a first layer of bottleneck blocks as red (X: layer, Y: magnitude).

ResNet-18(Top), PI-ResNet-18 (Bottom)

FIGURE 6: Grad-CAM visualizations on intermediate layers of R18/PI18 (Left) and the corresponding inter-layer attention curves (Right). For Grad-CAMs, maps are arranged in increasing order of layers. Red borders signify the layer of highest layer attention value and yellow borders indicate the layer that captures class-agnostic visual patterns. For inter-layer attention curves, the attention values are layer-wisely normalized by sample mean/stdev. More visualizations are available on Appendix L.

Raising the activation of class-specific layers. PI networks learn to enhance the layer attention values of layers that capture class-specific visual features for each target class. The layers marked with a red border indicate those that achieve the highest layer attention value. For example, the first layer, which captures simple patterns or small parts of a racer class, achieves the highest value, and the fifth
layer, focusing on catching the head of an ostrich class, also obtains the highest value. According to these observations, we find that PI networks emphasize the layers that catch class-specific visual information for each target class. Learning class-agnostic features in deeper layers. Deep layers of PI networks tend to learn class-agnostic features. These layers, marked with a yellow border, indicate a typical example of focusing less on class-specific visual information in images. Using an example of an ostrich class, the second to last layer of PI networks allocate neurons to learn class-agnostic visual information, such as a background. Compared to the ResNet, which allocates more class-specific neurons up to the deep layers, these observations are quite impressive. Also, our findings are convincing because they are in good agreement with a recent analysis of the class selectivity [46] of Gather-Excite networks [10]; intra-layer attention networks have lower-class selectivity in deep layers compared to the backbone network. Please refer to Appendix J for the same analysis on PI-ResNet.

VI. CONCLUSION

In this work, we propose a novel attention module that is lightweight and highly compatible with general CNNs, graph CNNs, and even with other existing attention modules. The PI module sheds light on self-attention studies in vision by focusing on inter-layer relationships, an architectural aspect unexamined by other studies. We demonstrate the effectiveness and generality of the PI module through extensive analysis, experiments, and visualization. We expect that our perspective to model the inter-layer relationships will be helpful to the wide range of computer vision studies.

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is all you need,” in Proc. NeurIPS, vol. 30, 2017.

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhang, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” in Proc. ICLR. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[3] H. Xu and J. Zhang, “Aanet: Adaptive aggregation network for efficient stereo matching,” in Proc. CVPR, 2020, pp. 1959–1968.

[4] D. Misra, T. Nalamada, A. U. Arasanipal, and Q. Hou, “Rotate to attend: Convolutional triplet attention module,” in Proc. WACV, 2021, pp. 3139–3148.

[5] L. Yang, R.-Y. Zhang, L. Li, and X. Xie, “Siman: A simple, parameter-free attention module for convolutional neural networks,” in Proc. ICML, 2021, pp. 11 863–11 874.

[6] Q.-L. Zhang and Y.-B. Yang, “Sa-net: Shuffle attention for deep convolutional neural networks,” in Proc. ICASSP, 2021, pp. 2235–2239.

[7] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in Proc. CVPR, 2018, pp. 7132–7141.

[8] S.-W. Woo, J. Park, J.-Y. Lee, and I. So Kweon, “Chan: Convolutional block attention module,” in Proc. ECCV, 2018, pp. 3–19.

[9] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng, “q2-nets: Double attention networks,” in Proc. NeurIPS, vol. 31, 2018.

[10] J. Hu, L. Shen, S. Albanie, G. Sun, and A. Vedaldi, “Gather-excite: Exploiting feature context in convolutional neural networks,” in Proc. NeurIPS, 2018.

[11] A. Krizhevsky, “Learning multiple layers of features from tiny images,” 2009.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in Proc. CVPR, 2009, pp. 248–255.

[13] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in Proc. ECCV, 2014, pp. 740–755.

[14] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” Int. J. of Computer Vision (IJCV), vol. 88, no. 2, pp. 303–338, 2010.

[15] I. Arsham, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, “3d semantic parsing of large-scale indoor spaces,” in Proc. CVPR, 2016, pp. 1534–1543.

[16] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, “Residual attention network for image classification,” in Proc. CVPR, 2017, pp. 3156–3164.

[17] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu, “Residual non-local attention networks for image restoration,” in Proc. ICLR, 2019. [Online]. Available: https://openreview.net/forum?id=HkGhoASFX

[18] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua, “Sca-cnns: Spatial and channel-wise attention in convolutional networks for image captioning,” in Proc. CVPR, 2017, pp. 5659–5667.

[19] Z. Zhang, C. Lan, W. Zeng, X. Jin, and Z. Chen, “Relation-aware global attention for person re-identification,” Proc. CVPR, pp. 3186–3195, 2020.

[20] M. Malinowski, M. Rohrbach, and M. Fritz, “Ask your neurons: A neural-based approach to answering questions about images,” in Proc. ICCV, 2015, pp. 1–9.

[21] M. Ren, R. Kiros, and R. Zemel, “Exploring models and data for image question answering,” in Proc. NeurIPS, vol. 28, 2015.

[22] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, S. Lakshminarayanan, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” in Proc. ICML, 2015, pp. 2048–2057.

[23] S.-F. Chen, Y.-C. Chen, C.-K. Yeh, and Y.-C. F. Wang, “Order-free nn with visual attention for multi-label classification,” in Proc. AAAI, pp. 6714–6721.

[24] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, “Cnn-rrm: A unified framework for multi-label image classification,” in Proc. CVPR, 2016, pp. 2285–2294.

[25] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venu-gopalan, K. Saenko, and T. Darrell, “Long-term recurrent convolutional networks for visual recognition and description,” in Proc. ICCV, 2015, pp. 2625–2634.

[26] Z. Lin, Z. He, B. Tan, Y. Shen, P. Wang, and T. Liu, “An attention-based network for serial number recognition on banknotes,” Signal Processing: Image Communication, vol. 106, p. 116732, 2022.

[27] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai, “Aster: An attentional scene text recognizer with flexible rectification,” IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI), vol. 41, no. 9, pp. 2035–2048, 2018.

[28] S. Bell, C. Lawrence Zitnick, K. Bala, and R. Girshick, “Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks,” in Proc. CVPR, 2016, pp. 2874–2883.

[29] J. Ba, V. Mnih, and K. Kavukcuoglu, “Multiple object recognition with visual attention,” in Proc. CVPR, 2016, pp. 2874–2883.

[30] V. Mnih, N. Heess, A. Graves et al., “Recurrent models of visual attention,” in Proc. NeurIPS, vol. 27, 2014.

[31] S. Khandelwal and L. Sigal, “Attentionnnn: A structured spatial attention mechanism,” in Proc. ICCV, 2019, pp. 3425–3434.

[32] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner et al., “Relational inductive biases, deep learning, and graph networks,” in arXiv preprint arXiv:1412.7755, 2014.

[33] R. K. Srivastava, K. Greff, and J. Schmidhuber, “Highway networks,” in arXiv preprint arXiv:1505.00387, 2015.
[34] V. Nair and G. E. Hinton, “Rectified linear units improve restricted
boltzmann machines,” in Proc. ICML, 2010, pp. 807–814. [Online].
Available: https://icml.cc/Conferences/2010/papers/4632.pdf
[35] T. Zhao and X. Wu, “Pyramid feature attention network for saliency
detection,” in Proc. CVPR, 2019, pp. 3085–3094.
[36] G. Li, M. Mueller, A. Thabet, and B. Ghanem, “Deeplens: Can gens go as
deep as cnns?” in Proc. ICCV, 2019, pp. 9266–9275.
[37] X. Li, Z. Zhong, J. Wu, Y. Yang, Z. Lin, and H. Liu, “Expectation-
maximization attention networks for semantic segmentation,” in Proc.
ICCV, 2019, pp. 9167–9176.
[38] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and
A. Zisserman, “The pascal visual object classes challenge: A retrospec-
tive,” Int. J. of Computer Vision (IJCV), vol. 111, no. 1, pp. 98–136, 2015.
[39] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese, “Joint 2d-3d-semantic
data for indoor scene understanding,” arXiv preprint arXiv:1702.01105,
2017.
[40] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand,
M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional
neural networks for mobile vision applications,” in arXiv preprint
arXiv:1704.04861, 2017.
[41] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time
object detection with region proposal networks,” in Proc. NeurIPS, vol. 28,
2015.
[42] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual
networks,” in Proc. ECCV, 2016, pp. 630–645.
[43] A. E. Orhan and X. Pitkow, “Skip connections eliminate singularities,” in Proc.
ICLR, 2018. [Online]. Available: https://openreview.net/forum?id=HkwBEMWCZ
[44] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional
networks,” in Proc. ECCV, 2014, pp. 818–833.
[45] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Ba-
tra, “Grad-cam: Visual explanations from deep networks via gradient-
based localization,” in Proc. ICCV, 2017, pp. 618–626.
[46] A. S. Morcos, D. G. Barrett, N. C. Rabinowitz, and M. Botvinick, “On
the importance of single directions for generalization,” arXiv preprint
arXiv:1803.06959, 2018.

EUNSEOK KIM received his B.S degree in Department of Industrial & Systems Engineering from the Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 2018. He worked a Research Officer at the Agency for Defense Development (ADD), Daejeon, South Korea. He is studying for a Master degree in dentistry in school of dentist in Chonnam National University, Gwangju, Korea.

JIHWAN BAE received his B.S degree in Electrical Engineering and Computer Science (EECS) from the Gwangju Institute of Science and Technology (GIST) in Gwangju, South Korea, in 2020. Since 2020, he has been a researcher at the Ground Technology Research Institute, Agency for Defense Development in Daejeon, Republic of Korea. His research interest includes the theoretical machine learning and deep learning, high-level computer vision, and studies of perception for autonomous vehicles. He’s awards and honors include the Caltech Summer Undergraduate Research Fellow (SURF) Award, in 2019, and the Best Paper Award from the GIST EECS department, in 2019.

INWOOK SHIM received his B.S degree in Computer Science from Hanyang University in 2009, his M.S degree in Robotics Program, E.E. from KAIST in 2011, and his Ph.D degree from the Division of Future Vehicles, E.E., from KAIST in 2017. From 2017 to 2022, He worked a research scientist at the Agency for Defense Development, Korea, and is now an Assistant Professor, Department of Smart Mobility Engineering, INHA University, Incheon, Korea. He was a member of "Team KAIST", which took first place at the DARPA Robotics Challenge Finals in 2015. He was a recipient of the Qualcomm Innovation Award, a NI finalist at the NI Student Design Showcase, and he received the KAIST achievement award of Robotics, and the Creativity and Challenge award from KAIST. His research interests include 3D vision for autonomous systems and deep learning.