

Unifying Nonlocal Blocks for Neural Networks

Lei Zhu^{1,3,4} Qi She^{*,2} Duo Li⁷ Yanye Lu^{1,3} Xuejing Kang⁵ Jie Hu⁶ Changhu Wang²

zhulei@stu.pku.edu.cn, sheqi1991@gmail.com

¹Institute of Medical Technology, Peking University Health Science Center, Peking University

²Bytedance AI Lab ³Department of Biomedical Engineering, Peking University

⁴Institute of Biomedical Engineering, Peking University Shenzhen Graduate School

⁵Beijing University of Posts and Telecommunications

⁶SKLCS & University of Chinese Academy of Sciences

⁷Hong Kong University of Science and Technology

Abstract

The nonlocal-based blocks are designed for capturing long-range spatial-temporal dependencies in computer vision tasks. Although having shown excellent performance, they still lack the mechanism to encode the rich, structured information among elements in an image or video. In this paper, to theoretically analyze the property of these nonlocal-based blocks, we provide a new perspective to interpret them, where we view them as a set of graph filters generated on a fully-connected graph. Specifically, when choosing the Chebyshev graph filter, a unified formulation can be derived for explaining and analyzing the existing nonlocal-based blocks (e.g., nonlocal block, nonlocal stage, double attention block). Furthermore, by concerning the property of spectral, we propose an efficient and robust spectral nonlocal block, which can be more robust and flexible to catch long-range dependencies when inserted into deep neural networks than the existing nonlocal blocks. Experimental results demonstrate the clear-cut improvements and practical applicabilities of our method on image classification, action recognition, semantic segmentation, and person re-identification tasks. Code are available at https://github.com/zh460045050/SNL_ICCV2021.

1. Introduction

Capturing the long-range spatial-temporal dependencies between spatial pixels or temporal frames plays a crucial role in computer vision tasks. Convolutional neural networks (CNNs) are inherently limited by their convolution operators, which are devoted to concern local relations (e.g., a 7×7 region), and they are inefficient in modeling long-range dependencies. Deep CNNs model these dependencies, which commonly refers to enlarge receptive fields, via stacking mul-

multiple convolution operators. However, two unfavorable issues are raised in practice. Firstly, repeating convolutional operations comes with higher computational and memory costs as well as the risk of over-fitting [10]. Secondly, stacking more layers cannot always increase the effective receptive fields [23], which indicates the convolutional layers may still lack the mechanism to model these dependencies efficiently.

A common practice to tackle these challenges is to aggregate the feature in a non-local way with fewer learning weights. Thus the aggregation can act on not only the k-hop neighbors but also the long-range positions [1, 30, 4, 36, 35]. Typically, inspired by the self-attention strategy, the Nonlocal (NL) block [30] firstly creates a dense affinity matrix that contains the relation among every pairwise position, and then uses this matrix as an attention map to aggregate the features by weighted mean. Nonetheless, because the dense attention map concerns humongous amounts of feature pairs (e.g. the relations between background and background), the aggregation feature map contains too much noise.

To solve this problem, recent state-of-the-art methods focus on creating a more reasonable attention map for the NL block [2, 28, 34, 17, 28]. Chen *et al.* [2] propose the Double Attention (A²) block that firstly gathers the features in the entire space and then distributes them back to each location. Yue *et al.* [34] propose the Compact Generalized Nonlocal (CGNL) block to catch cross-channel clues, which also increases the noise of the attention map inevitably. Huang *et al.* [17] propose a lightweight nonlocal block called Criss-Cross Attention block (CC), which decomposes the position-wise attention of NL into continuously column-wise and row-wise attention. To enhance the stability of the NL block, Tao *et al.* [28] propose the Nonlocal Stage (NS) module that can follow the diffusion nature by using the Laplacian of the affinity matrix as the attention map.

In general, though the above works can generate a better

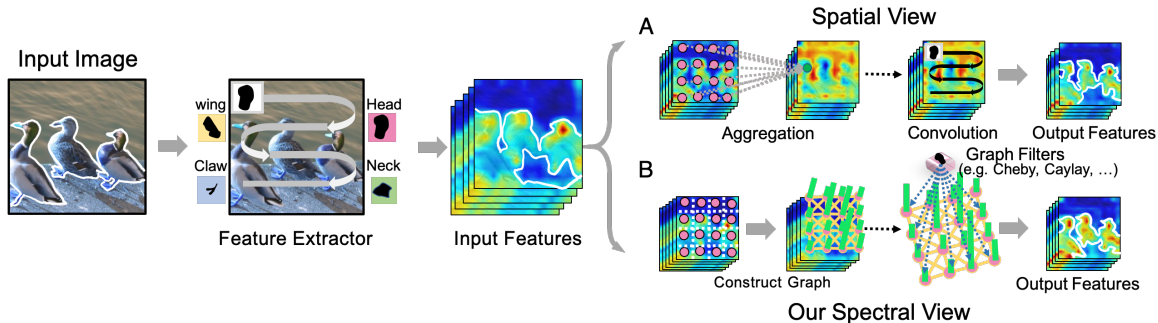


Figure 1: The spatial (A) and spectral (B) view of a nonlocal block. The pink dots indicate each patch in the feature map and the “Aggregation” means calculating the weighted mean as the numerator of Eq. (1). The dotted arrows mean “copy” and full arrows mean “feed forward”. The green bars are the node features and the length means their strength (best view in color).

aggregation feature map by improving the attention mechanism of the nonlocal block, their parameterized filter step still lacks the long-range dependence concerning where only localization filters (usually 1×1 convolutions) are used as shown in Fig. 1 A. To conquer the limitations, we aim at combining the nonlocal aggregation with the local filter step of nonlocal block via graph filters. We call this perspective “the spectral view of the nonlocal block”. Specifically, as shown in Fig. 1 B, a fully-connected graph is firstly constructed based on the input features to contain the similarity among different positions. Then the output is generated by directly filtering the features with learnable graph filters. Under this perspective, long-range cues are maintained by the affinity of graph structure, and the graph filters provide more learning flexibility when catching long-range dependencies.

Moreover, the proposed “spectral view” also provides a valid tool to theoretically analyze the nonlocal-based blocks [30, 28, 2, 17, 34], which are all experimentally designed and lacks theoretical supports to explain their effectiveness. With the proposed “spectral view”, a unified formulation of these nonlocal-based blocks can be deduced. It shows that they all utilize “an incomplete formation” Chebyshev graph filter, which limits their performance and robustness. Profited by our “spectral view”, these incomplete aspects can be easily solved by concerning the graph nature. Hence, a novel nonlocal block called the Spectral Nonlocal Block (SNL) with a stronger theoretical basis is proposed, which uses a more accurate approximation and can avoid complex eigenvalue. The proposed SNL is more flexible when inserted into the neural network and achieves superior performance over other nonlocal-based blocks. This shows the effectiveness of the proposed “spectral view” in defining novel nonlocal blocks on the basis of graph signal processing [26] or with the help of graph neural networks [5, 21, 31].

In a nutshell, our contributions are threefold:

- We provide a novel perspective for the model design of nonlocal-based blocks, which can help to catch long-

range dependencies more flexibly. In this context, we seamlessly make the connection between different variants of nonlocal methods and graph signal processing.

- We present, for the first time, that five well-known nonlocal blocks can be unified and interpreted under the proposed perspective with the help of Chebyshev approximation. This motivates our advocate of other potential nonlocal blocks embodied with discrimination capability and theoretical basis as an alternative.
- We propose a novel nonlocal block with a stronger theoretical basis that using an accurate approximated filter with the concern of the graph nature. The proposed block works universally well across a wide array of vision tasks, including image classification, action recognition, semantic segmentation, and person Re-identification, offering significantly better performance than other nonlocal-based counterparts.

2. Preliminary

In this paper, **bold** we use uppercase characters to denote the parameter matrix and *italic bold* uppercase characters to denote other matrices. Vectors are denoted with lowercase.

2.1. Nonlocal Block

The NL block calculates a weighted sum of pairings between the features of each position and all possible positions as shown in Fig. 1 A. The nonlocal operator is defined:

$$F(\mathbf{X}_{i,:}) = \frac{\sum_j \left[f(\mathbf{X}_{i,:}, \mathbf{X}_{j,:}) g(\mathbf{X}_{j,:}) \right]}{\sum_j f(\mathbf{X}_{i,:}, \mathbf{X}_{j,:})}, \quad (1)$$

where $\mathbf{X} \in \mathbb{R}^{N \times C_1}$ is the input feature map, i, j are the position indexes in the feature map, $f(\cdot)$ is the affinity kernel with a finite Frobenius norm. $g(\cdot)$ is a linear embedding that is defined as: $g(\mathbf{X}_{j,:}) = \mathbf{X}_{j,:} \mathbf{W}_Z$ with $\mathbf{W}_Z \in \mathbb{R}^{C_1 \times C_s}$.

Here N is the total positions of each feature. C_1 and C_s are the numbers of channels for the input and the transferred features, respectively.

When inserting the NL block into the network structure, a linear transformation with weight matrix $\mathbf{W} \in \mathbb{R}^{C_s \times C_1}$ and a residual connection are added:

$$\mathbf{Y}_{i,:} = \mathbf{X}_{i,:} + F(\mathbf{X}_{i,:})\mathbf{W}. \quad (2)$$

It is worth noting that, though NL block and the Vision Transformers (ViT) [7, 22, 19] both utilize the self-attention mechanism, the former is added on a certain stage of CNNs to perceive long-range dependencies rather than using to replace all convolutional operators in CNN as the latter.

2.2. Graph Fourier Transform & Graph Filter

Assuming that a graph \mathcal{G} contains N vertices, an arbitrary function (or signal) vector $\mathbf{f} = \{f_1, f_2, \dots, f_N\}$ can be defined, where the i^{th} component of the vector $\mathbf{f}(i)$ represents the function value at the i^{th} vertex of the graph \mathcal{G} . Then, the graph Fourier transformation [26] $\hat{\mathbf{f}}$ of \mathbf{f} on vertices of \mathcal{G} can be formulated as the expansion of \mathbf{f} in terms of the eigenvectors of the graph Laplacian:

$$\hat{\mathbf{f}}(\lambda_l) = \sum_{i=1}^N \mathbf{f}(i) \mathbf{u}_l^*(i), \quad (3)$$

where $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_l, \dots]$ and $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_l, \dots]$ are the eigenvalues and eigenvectors of the graph Laplacian. $\hat{\mathbf{f}}(\lambda_l)$ is the corresponding spectral coefficient under \mathbf{u}_l . \mathbf{u}_l^* is the l^{th} row vector of \mathbf{U}^\top . The inverse graph Fourier transformation is then given by $\mathbf{f}(i) = \sum_{l=1}^N \hat{\mathbf{f}}(\lambda_l) \mathbf{u}_l(i)$.

A graph filter is an operator that modifies the components of an input signal \mathbf{x} based on the eigenvalues \mathbf{U} , according to a frequency response \mathbf{g}_θ acting on $\boldsymbol{\lambda}$. Thus, based on the graph Fourier transformation, the output of filtering \mathbf{x} under graph \mathcal{G} can be defined as:

$$\mathbf{O}(i)_{\mathbf{x} *_{\mathcal{G}} \mathbf{g}_\theta} = \sum_{l=1}^N \hat{\mathbf{O}}(\lambda_l) \mathbf{u}_l(i) = \sum_{l=1}^N \hat{\mathbf{x}}(\lambda_l) \hat{\mathbf{g}}_\theta(\lambda_l) \mathbf{u}_l(i) \quad (4)$$

where $\hat{\mathbf{x}}$, $\hat{\mathbf{g}}_\theta$, $\hat{\mathbf{O}}$ are the graph Fourier transformation of input signal \mathbf{x} , filter \mathbf{g}_θ and the output signal $\mathbf{O}_{\mathbf{x} *_{\mathcal{G}} \mathbf{g}_\theta}$ respectively. Further, the formulation of the output signal can be also derived as (more details of this derivation and graph signal process refer to the survey [26]):

$$\mathbf{O}_{\mathbf{x} *_{\mathcal{G}} \mathbf{g}_\theta} = \mathbf{U} \text{diag}([\hat{\mathbf{g}}_\theta(\lambda_1), \dots, \hat{\mathbf{g}}_\theta(\lambda_l), \dots]) \mathbf{U}^\top \mathbf{x} \quad (5)$$

3. Approach

The nonlocal operator can be explained under the graph spectral domain, where it is the same as operating a set of graph filters on a fully connected weighted graph. This

process can be briefly divided into two steps: 1) generating a fully-connected graph to model the similarity among the position pairs, and 2) converting the input features into the graph domain and learning a graph filter. In this section, we firstly propose our framework which gives the definition of the spectral view for the nonlocal operator. Then, we unify five existing nonlocal-based operators from this spectral view. We further propose a novel nonlocal block based on the framework, which is more effective and robust.

3.1. The Spectral View of Nonlocal-based blocks

To define the spectral view of nonlocal, we start from taking the matrix form of the nonlocal operator into Eq. (2) and decompose the parameter matrix \mathbf{W} into \mathbf{W}_{s1} and \mathbf{W}_{s2} :

$$\mathbf{Y} = \mathbf{X} + F(\mathbf{X})\mathbf{W} = \mathbf{X} + \mathbf{A}\mathbf{Z}\mathbf{W}_{s1}\mathbf{W}_{s2}, \quad (6)$$

In this Eq. (6), $\mathbf{A} = \mathbf{D}_M^{-1}\mathbf{M}$ is the affinity matrix and $\mathbf{M} = [M_{ij}]$ is composed of pairwise similarities between pixels, i.e. $M_{ij} = f(\mathbf{X}_{i,:}, \mathbf{X}_{j,:})$ where $f(\cdot)$ is usually the dot product. \mathbf{D}_M is a diagonal matrix containing the degree of each vertex of \mathbf{M} . $\mathbf{Z} = \mathbf{X}\mathbf{W}_Z \in \mathbb{R}^{N \times C_s}$ is the transferred feature map that compresses the channels of \mathbf{X} by a linear transformation with $\mathbf{W}_Z \in \mathbb{R}^{C_1 \times C_s}$. $\mathbf{W}_{s1} \in \mathbb{R}^{C_s \times C_s}$ and $\mathbf{W}_{s2} \in \mathbb{R}^{C_s \times C_1}$ are two parameter-matrices that are used to filter discriminative features and restore the number of channels respectively. Then based on Eq. (6), the nonlocal block can be formulated in the spectral view by generalizing \mathbf{W}_{s1} into a set of graph filters $\mathbf{g}_\theta = \{\mathbf{g}_\theta^1, \dots, \mathbf{g}_\theta^i, \dots, \mathbf{g}_\theta^{C_s}\}$:

$$\mathbf{Y} = \mathbf{X} + \mathcal{F}(\mathbf{A}, \mathbf{Z}, \mathbf{W}_{s1})\mathbf{W}_{s2} = \mathbf{X} + \mathcal{F}(\mathbf{A}, \mathbf{Z}, \mathbf{g}_\theta)\mathbf{W}_{s2}, \quad (7)$$

where $\mathcal{F}(\mathbf{A}, \mathbf{Z}, \mathbf{g}_\theta)$ is the ‘‘nonlocal operator in the spectral view’’. For clarity, we omit \mathbf{W}_{s2} by assuming $C_1 = C_s$, abbreviate $\mathcal{F}(\mathbf{A}, \mathbf{Z}, \mathbf{g}_\theta)$ into $\mathcal{F}(\mathbf{A}, \mathbf{Z})$ and call it ‘‘nonlocal operator’’ in following paper.

In this view, the nonlocal operator firstly computes the affinity matrix \mathbf{A} that defines a graph spectral domain and then learns filters \mathbf{g}_θ for graph spectral features. Specifically, a fully-connected graph $\mathcal{G} = \{\mathbb{V}, \mathbf{A}, \mathbf{Z}\}$ is constructed, in which \mathbb{V} is the vertex set. Then, for each column vector $\mathbf{z}_i \in \mathbb{R}^{N \times 1}$ of \mathbf{Z} , a graph filter \mathbf{g}_θ^i is generated to enhance the feature discrimination. From this perspective, the nonlocal operator can be theoretically interpreted in the spectral view as below:

Theorem 1. *Given an affinity matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ and the signal $\mathbf{Z} \in \mathbb{R}^{N \times C}$, the nonlocal operator is the same as filtering the signal \mathbf{Z} by a set of graph filters $\{\mathbf{g}_\theta^1, \mathbf{g}_\theta^2, \dots, \mathbf{g}_\theta^C\}$ under the graph domain of a fully-connected graph \mathcal{G} :*

$$\mathcal{F}(\mathbf{A}, \mathbf{Z}) = [\mathbf{O}_{\mathbf{z}_1 *_{\mathcal{G}} \mathbf{g}_\theta^1}, \dots, \mathbf{O}_{\mathbf{z}_i *_{\mathcal{G}} \mathbf{g}_\theta^i}, \dots, \mathbf{O}_{\mathbf{z}_C *_{\mathcal{G}} \mathbf{g}_\theta^C}] \quad (8)$$

where the graph $\mathcal{G} = (\mathbb{V}, \mathbf{Z}, \mathbf{A})$ has the vertex set \mathbb{V} , node feature \mathbf{Z} and affinity matrix \mathbf{A} . $\mathbf{O}_{\mathbf{z}_i *_{\mathcal{G}} \mathbf{g}_\theta^i} \in \mathbb{R}^{N \times 1}$ is the output signal on \mathbf{z}_i .

Table 1: Summary of five existing nonlocal-based blocks in the spectral view.

Models	Vertex ($ \mathbb{V} $)	Edge ($ E $)	Affinity Matrix (\mathbf{A})	Node Feature (\mathbf{Z})	Formulation of $F(\mathbf{A}, \mathbf{Z})$
Chebyshev	-	-	-	-	$\mathbf{Z}\mathbf{W}_1 + \mathbf{A}\mathbf{Z}\mathbf{W}_2 + \sum_{k=2}^{K-1} \mathbf{A}^k \mathbf{Z}\mathbf{W}_k$
NL	N	$N \times N$	$\mathbf{D}_M^{-1} \mathbf{M}$	$\mathbf{X}\mathbf{W}_Z$	$\mathbf{A}\mathbf{Z}\mathbf{W}$
\mathbf{A}^2	N	$N \times N$	\mathbf{M}	$\mathbf{X}\mathbf{W}_Z$	$\mathbf{A}\mathbf{Z}\mathbf{W}$
CGNL	NC_s	$NC_s \times NC_s$	$\mathbf{D}_M^{-1} \mathbf{M}^f$	$\text{vec}(\mathbf{X}\mathbf{W}_Z)$	$\mathbf{A}\mathbf{Z}\mathbf{W}$
NS	N	$N \times N$	$\mathbf{D}_M^{-1} \mathbf{M}$	$\mathbf{X}\mathbf{W}_Z$	$-\mathbf{Z}\mathbf{W} + \mathbf{A}\mathbf{Z}\mathbf{W}$
CC	N	$N \times N$	$\mathbf{D}_{C \odot M}^{-1}(\mathbf{C} \odot \mathbf{M})$	\mathbf{X}	$\mathbf{A}\mathbf{Z}\mathbf{W}$

Remark 1. The main difference between Theorem. 1 and the original spatial view of nonlocal [30] is that the former learns a graph filter to obtain the feature under the spectral domain while the latter filters the feature by the convolutional operator without concerning the graph structure. Moreover, to confirm the existence of the graph spectral domain, Theorem.1 requires that the graph Laplacian \mathbf{L} should be diagonalized and not contain complex eigenvalues and eigenvectors. Thus the affinity matrix \mathbf{A} should be symmetric.

Specifically, a generalized implementation [6] of the output signal on each column vector \mathbf{z}_i can be used for Theorem. 1 by setting the graph spectral filter as a set of diagonal parameter matrix $\Omega^i \in \mathbb{R}^{N \times N}$:

$$\mathcal{O}_{\mathbf{z}_i * \mathbf{G}} \mathbf{g}_\theta^i = \mathbf{U} \Omega^i \mathbf{U}^\top \mathbf{z}_i \quad (9)$$

where $\Omega^i = \text{diag}([\omega_1, \omega_2, \dots, \omega_n])$ contains n parameters. In addition, new nonlocal operators can also be theoretically designed by using different types of graph filters to obtain output signal in Theorem. 1, such as Chebyshev filter [6, 20], graph wavelet filter [8], Cayley filter [21].

3.2. Unifying Existing Nonlocal-based Blocks

The proposed ‘‘spectral view’’ provides a valid tool to analyze the experimental designed nonlocal block on basis of graph signal processing. To unify other nonlocal-based blocks based on Theorem. 1, here we use the Chebyshev filter for illustration (Cayley filter is also presented in appendix 2), *i.e.* using Chebyshev polynomials [6] to reduce the n parameters in Ω^i into K (K is the order of polynomials, and $K \ll N$). For simplicity, we firstly assume that both the input and output signals have one channel, *i.e.* $\mathbf{Z} = \mathbf{z}_1$ and $\mathcal{F}(\mathbf{A}, \mathbf{Z}) = \mathbf{U} \Omega^1 \mathbf{U}^\top \mathbf{Z}$. Then the parameter matrix Ω^1 of the graph filter approximated by K^{th} -order Chebyshev polynomials is formulated as:

$$\mathcal{F}(\mathbf{A}, \mathbf{Z}) = \sum_{k=0}^{K-1} \hat{\theta}_k T_k(\tilde{\mathbf{L}}) \mathbf{Z}, \quad (10)$$

$$\text{s.t. } T_k(\tilde{\mathbf{L}}) = 2\tilde{\mathbf{L}}T_{k-1}(\tilde{\mathbf{L}}) - T_{k-2}(\tilde{\mathbf{L}}),$$

where $\tilde{\mathbf{L}} = 2\mathbf{L}/\lambda_{\max} - \mathbf{I}_n$, $T_0(\tilde{\mathbf{L}}) = \mathbf{I}_N$, $T_1(\tilde{\mathbf{L}}) = \tilde{\mathbf{L}}$, and $\hat{\theta}_k$ is the coefficient.

Note that the affinity matrix \mathbf{A} is affected by the input feature \mathbf{X} rather than using a fixed graph structure. Thus, an upper-bound exists for the maximum eigenvalue on all possible affinity matrices, *i.e.* $\lambda_{\max} = 2$, when all their graph Laplacian \mathbf{L} are normalized graph Laplacian [26]. With this assumption, we can get $\tilde{\mathbf{L}} = -\mathbf{A}$ and take it into Eq. (10):

$$\mathcal{F}(\mathbf{A}, \mathbf{Z}) = \theta_0 \mathbf{Z} + \theta_1 \mathbf{A}\mathbf{Z} + \sum_{k=2}^{K-1} \theta_k \mathbf{A}^k \mathbf{Z}, \quad (11)$$

where θ_k can be learned via SGD. Then, extending Eq. (11) into multiple channels, we can get a generalized formulation of the nonlocal operator with Chebyshev filter:

$$\mathcal{F}(\mathbf{A}, \mathbf{Z}) = \mathbf{Z}\mathbf{W}_1 + \mathbf{A}\mathbf{Z}\mathbf{W}_2 + \sum_{k=2}^{K-1} \mathbf{A}^k \mathbf{Z}\mathbf{W}_{k+1}, \quad (12)$$

where $F(\mathbf{A}, \mathbf{Z})$ is the nonlocal operator, $\mathbf{W}_k \in \mathbb{R}^{C_s \times C_s}$ is a parameter matrix. Note that, when $C_s \neq C_1$, it is straightforward to merge \mathbf{W}_{s2} with \mathbf{W}_k , which makes $\mathbf{W}_k = \mathbf{W}_k * \mathbf{W}_{s2} \in \mathbb{R}^{C_s \times C_1}$.

Eq. (12) gives the connection between spatial view and spectral view of the nonlocal operator, in which the graph filter is expressed by the aggregation among the k^{th} neighbor nodes. Thus, existing nonlocal-based structures can be theoretically analyzed by Eq. (12) in the spectral view. Here we elaborate 5 types of existing nonlocal-based blocks that can be unified. They can be interpreted under certain graph structures and assumptions as shown in Table 1. More derivation details can be found in appendix 1.

(1) NL block [30]: The NL block in the spectral view is the same as defining the graph as $\mathcal{G} = (\mathbb{V}, \mathbf{D}^{-1} \mathbf{M}, \mathbf{Z})$ and then using the second term of the Chebyshev polynomials to approximate the generalized graph filter.

(2) NS module [28]: The NS module in the spectral view can be considered as the graph in the form of $\mathcal{G} = (\mathbb{V}, \mathbf{D}_M^{-1} \mathbf{M}, \mathbf{Z})$. The 1st-order Chebyshev polynomials is utilized to approximate the graph filter with the condition $\mathbf{W}_1 = -\mathbf{W}_2 = -\mathbf{W}$.

(3) \mathbf{A}^2 block [2]: The Double Attention Block can be viewed as the graph $\mathcal{G} = (\mathbb{V}, \mathbf{M}, \mathbf{Z})$ and then we can use the second term of the Chebyshev polynomial to approximate the graph filter, *i.e.* $F(\mathbf{A}, \mathbf{Z}) = \mathbf{M}\mathbf{Z}\mathbf{W}$.

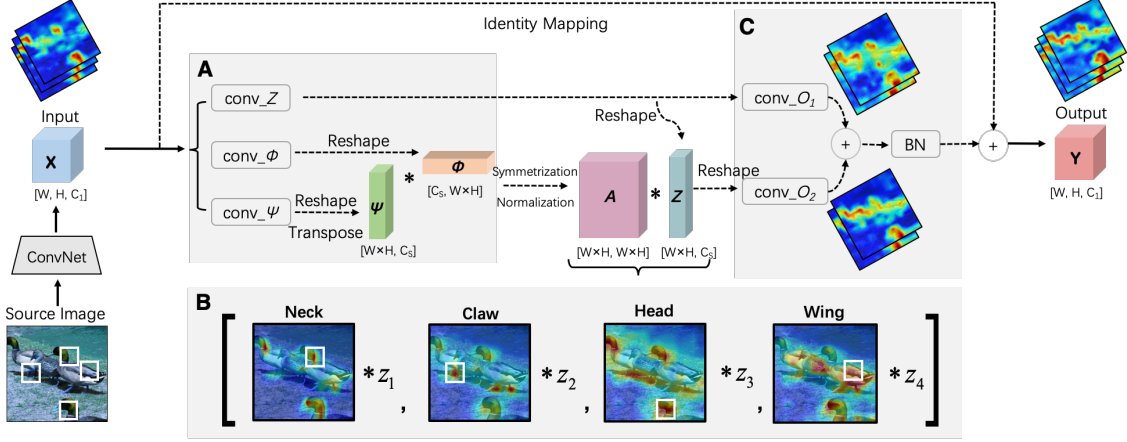


Figure 2: The implementation of our SNL. **A**. Three feature maps ϕ , ψ , Z are generated by feeding the \mathbf{X} into three 1×1 convolutions. **B**. The second term of Eq. (13) is calculated with Z and a normalized symmetrization affinity matrix A . Each row of A contains a N -dimension spatial attention map (heat maps) and $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n$ are the column vectors of Z (for simplicity, here we pick $n = 4$ where the white squares are the central positions we visualize). **C**. The graph filter is approximated by respectively feeding the 0^{th} -order and 1^{st} -order term into convolutions to obtain the output \mathbf{Y} .

(4) CGNL block [34]: When grouping all channels into one group, the CGNL can be viewed as the graph $\mathcal{G} = (\mathbb{V}^f, \mathbf{D}_{M^f}^{-1} \mathbf{M}^f, \text{vec}(\mathbf{Z}))$, where \mathbb{V}^f contain both spatial position and feature channel. Again we can also use the second term of the Chebyshev Polynomial to approximate the graph filter, i.e $F(\mathbf{A}, \mathbf{Z}) = \mathbf{D}_{M^f}^{-1} \mathbf{M}^f \text{vec}(\mathbf{Z}) \mathbf{W}$.

(5) CC block [17]: In the CC block, $\mathcal{G} = (\mathbb{V}, \mathbf{D}_{C \odot M}^{-1} \mathbf{C} \odot \mathbf{M}, \mathbf{X})$ with mask \mathbf{C} and then the second term of the Chebyshev is used to approximate filters with node feature \mathbf{X} .

3.3. Designing Spectral Nonlocal Blocks

Except for unifying the existing nonlocal blocks, our proposed “spectral view” can also help to design novel nonlocal-based blocks with theoretical guarantees. As an example, we elaborate a more rational nonlocal block called Spectral Nonlocal Block (SNL), which is still based on the **Chebyshev filter** but has a stronger theoretical basis and robustness.

Based on the above section, existing nonlocal-based operators use the random walk normalized (NL, NS, CGNL, CC) or the non-normalized affinity matrix (A^2). The symmetry property is not guaranteed and the performance depends on the affinity kernel. This leads to the non-existence of the graph spectral domain and causes large amounts of complex eigenvalues for the graph Laplacian as discussed in Sec.3.1. This problem is also magnified under the fully-connected graph model of the nonlocal-based blocks.

Moreover, existing nonlocal blocks only use the second term (NL, A^2 , CGNL, CC) or the 1^{st} -order approximation with sharing weight (NS) rather than the complete form of the 1^{st} -order approximation, which also hinders their performance. Thus considering these factors, the proposed SNL block applies a symmetry affinity matrix with a more

complete approximation as:

$$\mathbf{Y} = \mathbf{X} + \mathcal{F}_s(\mathbf{A}, \mathbf{Z}) = \mathbf{X} + \mathbf{Z} \mathbf{W}_1 + \mathbf{A} \mathbf{Z} \mathbf{W}_2, \quad (13)$$

$$\text{s.t. } \mathbf{A} = \mathbf{D}_M^{-\frac{1}{2}} \hat{\mathbf{M}} \mathbf{D}_M^{-\frac{1}{2}}, \quad \hat{\mathbf{M}} = (\mathbf{M} + \mathbf{M}^\top) / 2$$

where $\mathcal{F}_s(\mathbf{A}, \mathbf{Z})$ is the SNL operator, $\mathbf{W}_1, \mathbf{W}_2$ are two parameter matrices.

Remark 2. The proposed SNL uses Chebyshev filter as the generalized graph filter but has a more thorough formulation analyzed by our proposed spectral view. Specifically, it uses a symmetric affinity matrix $\mathbf{A} = \mathbf{D}_M^{-\frac{1}{2}} \hat{\mathbf{M}} \mathbf{D}_M^{-\frac{1}{2}}$ to ensure the existence of the real eigenvalue. This makes it more stable when inserted into the deep neural networks. Moreover, the proposed SNL also uses the complete form of 1^{st} -order Chebyshev Approximation that is a more accurate approximation of the graph filter. Thus, it can give the parameters a liberal learning space with only one more parameter matrix.

The implementation details of the SNL block are shown in Fig. 2 (A, B, C). The input feature map $\mathbf{X} \in \mathbb{R}^{W \times H \times C_1}$ is firstly fed into three 1×1 convolutions with the weight kernels: $\mathbf{W}_{\phi, \psi, g} \in \mathbb{R}^{C_1 \times C_s}$ to subtract the number of channels and then be reshaped into $\mathbb{R}^{W \times H \times C_s}$. One of the output $\mathbf{Z} \in \mathbb{R}^{W \times H \times C_s}$ is used as the transferred feature map to reduce the calculation complexity, while the other two outputs $\Phi, \Psi \in \mathbb{R}^{W \times H \times C_s}$ are used to get the affinity matrix \mathbf{A} with the affinity kernel function $f(\cdot)$. Then, \mathbf{A} is made to be symmetric and normalized as in Eq. (13). Finally, with the affinity matrix \mathbf{A} and the transferred feature map \mathbf{Z} , the output of the nonlocal block can be obtained by the Eq. (13). Specifically, the two weight matrices $\mathbf{W}_{1,2} \in \mathbb{R}^{C_s \times C_1}$ are yielded by two 1×1 convolutions.

Table 2: The Performance of Nonlocal-based Blocks with Different Number of Transferred Channels on CIFAR-100

Non-Reduction			Half-Reduction			Quarter-Reduction		
Models	Top-1 (%)	Top-5 (%)	Models	Top-1 (%)	Top-5 (%)	Models	Top-1 (%)	Top-5 (%)
PreResNet56	75.33 ^{↑0.00}	93.97 ^{↑0.00}	PreResNet56	75.33 ^{↑0.00}	93.97 ^{↑0.00}	PreResNet56	75.33 ^{↑0.00}	93.97 ^{↑0.00}
+ NL	75.29 ^{↓0.04}	94.07 ^{↑0.10}	+ NL	75.31 ^{↓0.02}	92.84 ^{↓1.13}	+ NL	75.50 ^{↑0.17}	93.75 ^{↓0.22}
+ NS	75.39 ^{↑0.06}	93.00 ^{↓0.97}	+ NS	75.83 ^{↑0.50}	93.87 ^{↓0.10}	+ NS	75.61 ^{↑0.28}	93.66 ^{↓0.31}
+ A ²	75.51 ^{↑0.18}	92.90 ^{↓1.07}	+ A ²	75.58 ^{↑0.25}	94.27 ^{↑0.30}	+ A ²	75.61 ^{↑0.28}	93.61 ^{↓0.36}
+ CGNL	74.71 ^{↓0.62}	93.60 ^{↓0.37}	+ CGNL	75.75 ^{↑0.42}	93.74 ^{↓0.23}	+ CGNL	75.27 ^{↓0.06}	93.05 ^{↓0.92}
+ Ours	76.34^{↑1.01}	94.48^{↑0.51}	+ Ours	76.41^{↑1.08}	94.38^{↑0.41}	+ Ours	76.02^{↑0.69}	94.08^{↑0.11}

Table 3: The Performance of Nonlocal-based Blocks Inserted into Different Position of Deep Networks on CIFAR-100

Stage 1			Stage 2			Stage 3		
Models	Top-1 (%)	Top-5 (%)	Models	Top-1 (%)	Top-5 (%)	Models	Top-1 (%)	Top-5 (%)
PreResNet56	75.33 ^{↑0.00}	93.97 ^{↑0.00}	PreResNet56	75.33 ^{↑0.00}	93.97 ^{↑0.00}	PreResNet56	75.33 ^{↑0.00}	93.97^{↑0.00}
+ NL	75.31 ^{↓0.02}	92.84 ^{↓1.13}	+ NL	75.64 ^{↑0.31}	93.79 ^{↓0.18}	+ NL	75.28 ^{↓0.05}	93.93 ^{↓0.04}
+ NS	75.83 ^{↑0.50}	93.87 ^{↓0.10}	+ NS	75.74 ^{↑0.41}	94.02 ^{↑0.05}	+ NS	75.44 ^{↑0.11}	93.86 ^{↓0.11}
+ A ²	75.58 ^{↑0.25}	94.27 ^{↑0.30}	+ A ²	75.60 ^{↑0.27}	93.82 ^{↓0.15}	+ A ²	75.21 ^{↓0.12}	93.65 ^{↓0.32}
+ CGNL	75.75 ^{↑0.42}	93.74 ^{↓0.23}	+ CGNL	74.54 ^{↓0.79}	92.65 ^{↓1.32}	+ CGNL	74.90 ^{↓0.43}	92.46 ^{↓1.51}
+ Ours	76.41^{↑1.08}	94.38^{↑0.41}	+ Ours	76.29^{↑0.96}	94.27^{↑0.30}	+ Ours	75.68^{↑0.35}	93.90 ^{↓0.07}

4. Experiments

In this section, we validate the robustness of nonlocal-based blocks with varying numbers, channels, and positions. Then, we show the performance of the proposed SNL in image classification tasks on Cifar-10/100, ImageNet, action recognition tasks on UCF-101 dataset, and semantic segmentation on Cityscapes dataset. The experimental results of fine-graining classification on CUB-200 datasets and person re-identification task on ILID-SVID [29], Mars [27], and Prid-2011 [14] datasets are given in the appendix 3. All the methods are implemented using PyTorch [24] toolbox with an Intel Core i9 CPU and 8 Nvidia RTX 2080 Ti GPUs.

4.1. Ablation Studies

Experimental Setup. Following Tao *et al.* [28], the robustness testings are conducted on CIFAR-100 dataset containing 50k training images and 10k test images of 100 classes. PreResNet56 [12] is used as the backbone network. Unless otherwise specified, we set $C_s = C_1/2$ and add 1 nonlocal-based block right after the second residual block in the early stage (stage 1). The SGD optimizer is used with the weight decay 10^{-4} and momentum 0.9. The initial learning rate is 0.1, which is divided by 10 at 150 and 250 epochs. All the models are trained for 300 epochs.

The number of channels in transferred feature space. The nonlocal-based block firstly reduces the channels of original feature map C_1 into the transferred feature space C_s to reduce computation. The larger C_s is, the more redundant information tends to be contained. This introduces the noise when calculating the affinity matrix \mathbf{A} . However, if C_s is too

small, the output feature map is hard to be reconstructed due to inadequate features. To test the robustness for the value of the C_s , we generate three types of models with different C_s settings: *Non-Reduction* ($C_s = C_1$), *Half-Reduction* ($C_s = C_1/2$), and *Quarter Reduction* ($C_s = C_1/4$). Table 2 shows the experimental results of the 3 types of models with different nonlocal-based blocks. Our SNL block outperforms other models profited by the flexibility for learning.

Moreover, from Table 2, we can see that the performance of the CGNL steeply drops when adopts large transferred channels. The reason is that the CGNL block concerns the relations among channels. When the number of the transferred channels increases, the relations among the redundant channels seriously interfere with its effects. Overall, our SNL block is the most robust for a large number of transferred channels (our model rises 1.01% in Top-1 while the best of others only rise 0.18% over the backbone).

The stage/position for adding the nonlocal-based blocks. The nonlocal-based blocks can be added into the different stages of the PreResNet to form the Nonlocal Network. Tao *et al.* [28] add them into the early stage of the PreResNet to catch the long-range relations. Here we show the performance of adding different types of nonlocal-based blocks into the 3 stages (the first, the second, and the third stage of the PreResNet) in Table 3. We can see that the results of the NL block are lower than the backbones when added into the early stage. However, our proposed SNL block has an averagely 1.08% improvement over the backbone when being added into the early stage, which is more than two times over other types of blocks (0.42% for the best case).

The number of the nonlocal-based blocks. We test the ro-

Table 4: Experiments for Adding Different Types of Nonocal-based Blocks into PreResnet56 and ResNet50 on CIFAR-10/100

CIFAR-10			CIFAR-100					
Models	Top-1 (%)	Top-5 (%)	Models	Top-1 (%)	Top-5 (%)	Models	Top-1 (%)	Top-5 (%)
ResNet50	94.94 ^{+0.00}	99.87 ^{+0.00}	PreResnet56	75.33 ^{+0.00}	93.97 ^{+0.00}	ResNet50	76.50 ^{+0.00}	93.14 ^{+0.00}
+ NL	94.01 ^{-0.93}	99.82 ^{-0.05}	+ NL	75.31 ^{-0.02}	92.84 ^{-1.33}	+ NL	76.77 ^{+0.27}	93.55 ^{+0.41}
+ NS	95.15 ^{+0.21}	99.88 ^{+0.01}	+ NS	75.83 ^{+0.50}	93.87 ^{-0.10}	+ NS	77.90 ^{+1.40}	94.34 ^{+1.20}
+ A ²	94.41 ^{-0.53}	99.83 ^{-0.05}	+ A ²	75.58 ^{+0.25}	94.27 ^{+0.30}	+ A ²	77.30 ^{+0.80}	93.40 ^{+0.26}
+ CGNL	94.49 ^{-0.45}	99.92 ^{+0.05}	+ CGNL	75.75 ^{+0.42}	93.74 ^{-0.23}	+ CGNL	74.88 ^{-1.62}	92.56 ^{-0.58}
+ Ours	95.32 ^{+0.38}	99.94 ^{+0.07}	+ Ours	76.41 ^{+1.08}	94.38 ^{+0.39}	+ Ours	78.17 ^{+1.67}	94.17 ^{+1.03}

business for adding different numbers of the nonlocal-based blocks into the backbone. The results are shown in Table 5. “×3” means three blocks are added into stage 1, 2, and 3, respectively, and the accuracy in the brackets represents their results. We can see that adding three proposed SNL operators into different stages of the backbone generates a larger improvement (1.37%) than the NS operator and NL operator. This is because these two models cannot well aggregate the low-level features and interfere with the following blocks when adding NS and NL into the early stage.

Table 5: Experiments for Adding Different Number of Nonlocal-based Blocks into PreResNet56 on CIFAR-100

Models	Top-1 (%)	Top-5 (%)
PreResNet56	75.33 ^{+0.00}	93.97 ^{+0.00}
+ NL (×3)	75.31 ^{-0.02} (74.34 ^{-0.99})	92.84 ^{-1.13} (93.11 ^{-0.86})
+ NS (×3)	75.83 ^{+0.50} (75.00 ^{-0.33})	93.87 ^{-0.10} (93.57 ^{-0.40})
+ A ² (×3)	75.58 ^{+0.25} (75.63 ^{+0.33})	94.27 ^{+0.30} (94.12 ^{+0.15})
+ CGNL (×3)	75.75 ^{+0.42} (75.96 ^{+0.63})	93.74 ^{-0.23} (93.10 ^{-0.87})
+ Ours (×3)	76.41 ^{+1.08} (76.70 ^{+1.37})	94.38 ^{+0.41} (93.94 ^{-0.03})

4.2. Main Results

Image Classification. We use the ResNet50 [11] as the backbone and insert the SNL block right before the last residual block of *res4* for a fair comparison. Other settings for the CIFAR-10/100 are the same as the setting discussed in Sec. 4.1. For the ImageNet, the SGD optimizer is used with the weight decay 10^{-4} and momentum 0.9. The initial learning rate is set to be 0.01, which is divided by 10 at 31, 61, and 81 epochs. All the models are trained for 110 epoch.

Table 4 shows the results on the CIFAR datasets. When adding SNL, it improves the Top-1 accuracy by 0.38% absolutely, which is nearly two times over other nonlocal-based blocks (the best is 0.21%). For CIFAR100, using SNL brings significant improvements about 1.67% with ResNet50. While using a more simple backbone PreResnet56, our model can still generate 1.08% improvement which is not marginal compared with previous works [30, 28, 34].

The results of ImageNet are shown in Table 6. Note that we exhibit other results from their original paper. Our

Table 6: Results on ImageNet Dataset

Models	Top-1 (%)	FLOPs (G)	Size (M)
ResNet50	76.15 ^{+0.00}	4.14	25.56
+ CGD	76.90 ^{+0.75}	+0.01	+0.02
+ SE	77.72 ^{+1.57}	+0.10	+2.62
+ GE	78.00 ^{+1.85}	+0.10	+5.64
+ NL	76.70 ^{+0.55}	+0.41	+2.09
+ A ²	77.00 ^{+0.85}	+0.41	+2.62
+ simpNL[33]	77.28 ^{+1.13}	-	+1.05
+ CGNL	77.32 ^{+1.17}	+0.41	+2.09
+ Ours	78.11 ^{+1.96}	+0.51	+2.62

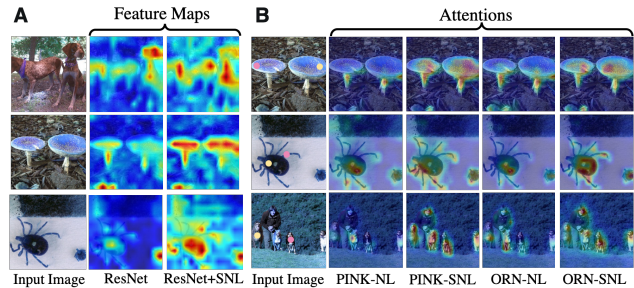


Figure 3: A. The visualization of the feature maps when adding SNL into the the backbone. B. The visualization of the attention maps for two positions (“Pink” and “Orange” dots). The heatmaps show the strength of similarity.

SNL achieves a clear-cut improvement (1.96%) with a minor increment in complexity (12% and 10% higher in FLOPs and Size respectively) compared with the nonlocal-based blocks. Moreover, our SNL is also superior to other types of blocks such as SE [16], CGD [13] and GE [15] (0.11% higher in Top-1 and 2.02M lower in size than the GE block).

We also visualize the output feature maps of the ResNet50 with SNL and the original ResNet50 in Fig. 3 A. Benefited from the rich and structured information considered in SNL, the response of the similar features among long-range spatial positions are enhanced as shown in the two mushrooms, balls, and those animals. Moreover, Fig. 3 B shows the attention maps produced by our SNL and the original NL

block where the ‘‘Pink’’ and ‘‘Orange’’ dots are the central positions and the heatmaps represent the similarity between the central position and other positions. Compared with the original NL block, SNL can pay more attention to the crucial parts than the original NL block profited by the better approximation formation as discussed in Sec. 3.3.

Action Recognition. Experiments are conducted on the UCF-101 dataset, which contains 7912 training videos and 1625 test videos of 101 different human actions. Our SNL block is tested on the UCF-101 dataset for capturing the dependence for the temporal frames. We follow the I3D structure [9] which uses $k \times k \times k$ kernels to replace the convolution operator in the residual block to learn seamless spatial-temporal feature extractors. The weights are initialized by the pre-trained I3D model on Kinetics dataset [18]. Inserting nonlocal-based blocks into the I3D can help to capture the relations among frame pairs with long distance and improve the feature representation. In the training process, each input video clip is cropped into $124 * 124 * 16$ to train our model. An initial learning rate of 0.01 is adopted, which is subsequently divided by 10 each 40 epoch. The training stops at the 100 epoch. Other hyper-parameters of the experimental setup are the same in Sec. 4.1.

Table 7: Results on UCF-101 Datasets

Models	Top-1 (%)	Top-5 (%)	FLOPs (G)	Size (M)
I3D	81.57 ^{↑0.00}	95.40 ^{↑0.00}	10.10	47.02
+ NL	82.88 ^{↑1.31}	95.74 ^{↑0.34}	+ 0.21	+ 2.10
+ NS	82.50 ^{↑0.93}	95.84 ^{↑0.44}	+ 0.26	+ 2.10
+ A ²	82.68 ^{↑1.11}	95.85 ^{↑0.45}	+ 0.21	+ 2.10
+ CGNL	83.38 ^{↑1.81}	95.42 ^{↑0.02}	+ 0.21	+ 2.10
+ Ours	84.39^{↑2.82}	97.66^{↑2.26}	+ 0.26	+ 2.62

Table 7 (UCF-101) shows the clip-level Top1 and Top5 metrics on the action recognition. The network with our SNL generates significant improvements (2.82%) than the backbone and outperforms all other nonlocal-based models. It shows that our proposed SNL is also effective to catch the long-range dependencies among the temporal frames. We also conduct the experiments on UCF-101 dataset with other state-of-the-art action recognition models in appendix 3.

Semantic Segmentation. NL-based blocks are widely used in semantic segmentation due to the requirement of large respective fields. Thus, both experiments and ablation studies are conducted on Cityscapes dataset, which contains 2975 train images and 500 validation images of the urban scene. For a fair comparison, the ‘‘mmsegmentation toolbox’’ [3] is used with the recommendation hyper-parameter for NL-based blocks. Specifically, the ‘‘ResNet50 with two FCN decoders’’ [25] were used as the baseline model, where the first FCN was replaced by NL-based blocks with setting $C_s = C_1/4$. SGD optimizer with momentum 0.9, weight

Table 8: Experiments for Nonocal-based Blocks Added into ResNet50-FCN on Cityscapes Dataset

Models	mIoU(%)	mAcc(%)	FLOPs(G)	Size(M)
Backbone	69.19 ^{↑0.00}	76.60 ^{↑0.00}	395.76	47.13
NL	74.15 ^{↑4.96}	81.83 ^{↑5.23}	+4.31	+0.53
NS	75.44 ^{↑6.25}	83.36 ^{↑6.76}	+5.37	+0.53
CC	75.34 ^{↑6.15}	82.75 ^{↑6.15}	+5.37	+0.33
DNL	76.19 ^{↑7.00}	84.61 ^{↑8.01}	+4.31	+0.53
SNL _{a1}	75.76 ^{↑6.57}	83.58 ^{↑6.98}	+4.31	+0.53
SNL _{a2}	75.94 ^{↑6.75}	83.85 ^{↑7.25}	+5.37	+0.66
SNL	76.87^{↑7.68}	84.75^{↑8.15}	+5.37	+0.66

decay $5e-4$, and initialization learning rate 0.01 are used to train all the NL-based networks for 40k iterators, where poly strategy with power 0.9 is used to adjust the learning rate.

Table 8 shows the experimental results of NL-based networks on Cityscapes dataset. To evaluate the effectiveness of the two modifications of the proposed SNL, two new blocks SNL_{a1} and SNL_{a2} are conducted to consider the symmetrization affinity or ‘‘the 1st-order’’ Chebyshev term respectively. It can be seen that compared with the original NL block, our (SNL_{a1}) improves 1.61 in mIoU and 1.75 in mAcc with no complexity increasing benefited from confirming the existence of the graph spectral domain. As for the effect of Chebyshev term, using ‘‘the 1st-order’’ (SNL_{a2}) is the best, and the ‘‘sharing weight 1st-order’’ (NS) is better than using ‘‘only the 1st-order term’’ (NL). Besides, when combining these two factors, the performance of our SNL is 0.68 higher in mIoU and 0.14 higher in mAcc with only 0.13M higher model size than the state-of-the-art DNL block [32].

5. Conclusion

This paper provides a novel perspective for the model design of nonlocal-based blocks. In this context, we make the connection between different variants of nonlocal methods and graph signal processing. Five well-known nonlocal blocks are unified and interpreted under the perspective of graph filters. A novel nonlocal block called SNL with stronger theoretical basis is proposed. It works universally well across a wide array of vision tasks and offers better performance than other nonlocal-based counterparts. Future work will focus on designing novel nonlocal blocks based on our spectral view and extending our spectral view on other self-attention based modules such as the vision transformer.

6. Acknowledgements

This work was supported by the Shenzhen Science and Technology Program (1210318663); the National Biomedical Imaging Facility Grant; the Shenzhen Nanshan Innovation and Business Development Grant; the NSFC Grants (62072449, 61632003)

References

- [1] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(4):834–848, 2017.
- [2] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. A²-nets: Double attention networks. In *Neural Information Processing Systems (NeurIPS)*, pages 352–361, 2018.
- [3] MMSegmentation Contributors. Mmsegmentation, an open source semantic segmentation toolbox. <https://github.com/open-mmlab/msegmentation>, 2020.
- [4] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. pages 764–773, 2017.
- [5] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Neural Information Processing Systems (NeurIPS)*, pages 3844–3852, 2016.
- [6] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Neural Information Processing Systems (NeurIPS)*, pages 3844–3852, 2016.
- [7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [8] David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis (ACHA)*, 30(2):129–150, 2011.
- [9] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6546–6555, 2018.
- [10] Kaiming He and Jian Sun. Convolutional neural networks at constrained time cost. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5353–5360, 2015.
- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision (ECCV)*, pages 630–645. Springer, 2016.
- [13] Xiangyu He, Ke Cheng, Qiang Chen, Qinghao Hu, Peisong Wang, and Jian Cheng. Compact global descriptor for neural networks. *arXiv preprint arXiv:1907.09665*, 2019.
- [14] Martin Hirzer, Csaba Beleznai, Peter M. Roth, and Horst Bischof. Person Re-Identification by Descriptive and Discriminative Classification. In *Proc. Scandinavian Conference on Image Analysis (SCIA)*, 2011.
- [15] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. In *Neural Information Processing Systems (NeurIPS)*, 2018.
- [16] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [17] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 603–612, 2019.
- [18] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.
- [19] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *arXiv preprint arXiv:2101.01169*, 2021.
- [20] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- [21] Ron Levie, Federico Monti, Xavier Bresson, and Michael M Bronstein. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing (TSP)*, 67(1):97–109, 2018.
- [22] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.
- [23] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Neural Information Processing Systems (NeurIPS)*, pages 4898–4906, 2016.
- [24] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Neural Information Processing Systems (NeurIPS)*, pages 8024–8035, 2019.
- [25] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):640–651, 2017.
- [26] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, 2013.
- [27] Springer. *MARS: A Video Benchmark for Large-Scale Person Re-identification*, 2016.
- [28] Yunzhe Tao, Qi Sun, Qiang Du, and Wei Liu. Nonlocal neural networks, nonlocal diffusion and nonlocal modeling. In *Neural Information Processing Systems (NeurIPS)*, pages 496–506, 2018.

- [29] Taiqing Wang, Shaogang Gong, Xiatian Zhu, and Shengjin Wang. Person re-identification by video ranking. In *European Conference on Computer Vision (ECCV)*, pages 688–703. Springer, 2014.
- [30] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7794–7803, 2018.
- [31] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596*, 2019.
- [32] Minghao Yin, Zhuliang Yao, Yue Cao, Xiu Li, Zheng Zhang, Steve Lin, and Han Hu. Disentangled non-local neural networks. In *European Conference on Computer Vision (ECCV)*, August 2020.
- [33] Cao Yue, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *IEEE International Conference on Computer Vision (ICCV) Workshops*, 2019.
- [34] Kaiyu Yue, Ming Sun, Yuchen Yuan, Feng Zhou, Errui Ding, and Fuxin Xu. Compact generalized non-local network. In *Neural Information Processing Systems (NeurIPS)*, pages 6510–6519, 2018.
- [35] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890, 2017.
- [36] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. pages 9308–9316, 2019.