A UNIFIED FRAMEWORK TO ANALYZE AND DESIGN THE NONLOCAL BLOCKS FOR NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

The nonlocal-based blocks are designed for capturing long-range spatial-temporal dependencies in computer vision tasks. Although having shown excellent performances, they lack the mechanism to encode the rich, structured information among elements in an image. In this paper, to theoretically analyze the property of these nonlocal-based blocks, we provide a unified framework to interpret them, where we view them as a graph filter generated on a fully-connected graph. When choosing Chebyshev graph filter, a generalized formulation can be derived for explaining the existing nonlocal-based blocks ($e.g.$, nonlocal block, nonlocal stage, double attention block) and uses to analyze their irrationality. Furthermore, by removing the irrationality, we propose an efficient and robust Chebyshev spectral nonlocal block, which can be more flexibly inserted into deep neural networks than the existing nonlocal blocks. Experimental results demonstrate the clear-cut improvements and practical applicabilities of the proposed spectral nonlocal blocks on image classification (Cifar-10/100, ImageNet), fine-grained image classification (CUB-200), action recognition (UCF-101) tasks.

1 INTRODUCTION

Capturing the long-range spatial-temporal dependencies between spatial pixels or temporal frames plays a crucial role in the computer vision tasks. Convolutional neural networks (CNNs) are inherently limited by their convolution operators which are devoted to concern local features and relations, $e.g.$, a 7×7 region, and are inefficient in modeling long-range dependencies. Deep CNNs model these dependencies, which commonly refers to enlarge receptive fields, via stacking multiple convolution operators. However, two unfavorable issues are raised in practice. Firstly, repeating convolutional operations comes with higher computation and memory cost as well as the risk of over-fitting He & Sun (2015). Secondly, stacking more layers cannot always increase the effective receptive fields Luo et al. (2016), which indicates the convolutional layers may still lack the mechanism to efficiently model these dependencies.

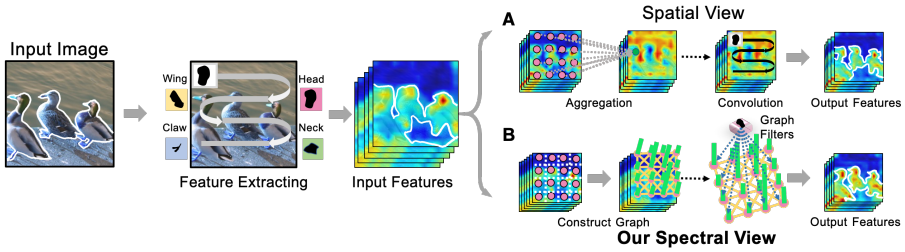

Figure 1: The spatial (A) and spectral (B) view of a nonlocal block. The pink dots indicate each patch in the feature map and the "Aggregation" means calculating the weighted mean as the numerator of Eq. (6). The dotted arrows mean "copy" and full arrows mean "feed forward". The green bars are the node features and the length means their strength (best view in color).

A common practice to tackle this challenge is to aggregate the feature in a non-local way with less number of learning weights. Thus the aggregation can act on not only the k-hop neighbors but also the long-range positions Chen et al. (2017); Wang et al. (2018); Dai et al. (2017); Zhu et al. (2019); Zhao

et al. (2017). Typically, inspired by the self-attention strategy, the Nonlocal (NL) block firstly create a dense affinity matrix that contains the relation between every pairwise positions by generating dot product on their feature maps and then uses this matrix as an attention map to aggregates the features by weighted mean. Nonetheless, because the dense attention map concerns humongous amount of useless feature pairs (e.g. the relations between background and background), the aggregation feature map contains too much noise and increase response on the background location. Thus, filtering discriminative features by the convolution operator on this noisy feature map is still hard, which limit the effectiveness of the NL block and make it unstable without elaborate arrangement (e.g. the number or the position of the added blocks).

Recently, many researchers focuses on creating a more reasonable attention map for the NL block to enhance its performance. Chen et al. (2018) proposes the Double Attention (A^2) block that firstly gather the feature in the entire space and then distribute them back to each location. It can be seen as using the soft-max operator to restrain the high response of the unimportant location and generate a more reasonable attention map for the NL Block. Yue et al. (2018) proposed the Compact Generalized Nonlocal (CGNL) block to catch cross-channel clues, which generalizes its attention map via further considering the correlations between pairwise channels. Due to the comprehensive channel concern, its has satisfactory performance on fine-grained classification. However, it also increases the noise of the attention map, which further weak its stability and robustness. To enhance the stability of the NL block, Tao et al. (2018) proposes the Nonlocal Stage (NS) module that can follow the diffusion nature by using the Laplacian of the affinity matrix as the attention map. Although the performance is just comparable as the NL, it can allow a more deeper connection without performance drop. Also, the NS module contains more than two nonlocal blocks, which increases computation complexity. To reduces the computation complexity, Huang et al. (2019) proposes lightweight nonlocal block called Criss-Cross Attention (CC) block, which decomposes the position-wise attention of NL into conterminously column-wise and row-wise attention.

In this paper, profited by the graph signal processing, we proposed a framework to increase the robustness and applicability of the nonlocal block in real-world applications by reformulating the nonlocal block based on the property of spectral graph. This framework combines the non-parameter feature aggregation step of nonlocal with the feature filter step to better concerns the position-wise affinity. As shown in Fig. 1, the input image is fed into the convolutional layers to extract discriminative features such as the wing, the head, the claw and the neck. These features can be seen as the input of the nonlocal block. Different from the spatial view which firstly aggregates the input features by weighted mean and then uses convolutional operator to filter as in Fig. 1 A, our spectral view constructs a fully-connected graph based on their similarity and then directly filters the input features in a global view profited by the graph filter shown in Fig. 1 B. Moreover, our framework can unify five existing nonlocal-based blocks ($e.g.$, nonlocal block, nonlocal stage, double attention block) to analyze their potential mechanism with the help of Chebyshev polynomials. Based on these analysis, we propose a novel nonlocal block called Chebyshev Spectral Nonlocal Block (ChebySNL) that concerns the rich, structured information in an image via encoding the graph structure. The ChabySNL guarantees the existence of the graph spectral domain and has more mathematical guarantees to improve the robustness and accuracy. In a nutshell, our contributions are threefold: (1) A generalized framework is proposed in this paper to theoretically analyze and design nonlocal-based blocks in the spectral graph view; (2) We unifying well-known nonlocal-based blocks such as NL, NS, A^2, CC, CGNL and anaylize them based on the proposed framework and analyzed their potential mechanism under graph spectral view; (3) We propose a novel nonlocal block with our pipeline, which achieve a clear-cut improvement over existing nonlocal-based blocks in four vision tasks.

2 APPROACHES

The nonlocal operator can be explained under the graph spectral domain. It can be briefly divided into two steps: generating a fully-connected graph to model the relation between the position pairs; converting the input features into the graph domain and learning a graph filter. In this section, we firstly propose our framework which gives definition of the spectral view for the nonlocal operator. Then, we unified five existing nonlocal-based operators from this spectral view. Finally, we propose a novel nonlocal blocks based on the framework, which is more effective and robust. In this paper, we use **bold** uppercase characters to denote the matrix-valued random variable and ***italic bold*** uppercase to denote the matrix. Vectors are denoted with lowercase.

2.1 THE SPECTRAL VIEW OF NONLOCAL-BASED BLOCKS

Firstly, we defines the "nonlocal operator in the spectral view" $\mathcal{F}(\boldsymbol{A}, \boldsymbol{Z})$ by merging the aggregation step and the filter matrix $\mathbf{W}$ and called it "nonlocal operator" in the following paper for simplicity:

$$\boldsymbol{Y} = \boldsymbol{X} + F(\boldsymbol{X})\mathbf{W} = \boldsymbol{X} + \mathcal{F}(\boldsymbol{A}, \boldsymbol{Z}). \tag{1}$$

where $F(\boldsymbol{X})$ is the original nonlocal whose details can be seen in Appendix. A. In the spectral view, $\mathcal{F}(\boldsymbol{A}, \boldsymbol{Z})$ can be seen as firstly computing the affinity matrix $\boldsymbol{A}$ that defines a graph spectral domain and then learns a filter for graph spectral features. Specifically, a fully-connected graph $\mathcal{G} = \{\mathbb{V}, \boldsymbol{A}, \boldsymbol{Z}\}$ is firstly constructed, in which $\mathbb{V}$ is the vertex set. Then, the node feature $\boldsymbol{Z}$ is transformed into the graph spectral domain by the graph Fourier transformation $\mathscr{F}(\boldsymbol{Z})$. Finally, a graph filter $\mathbf{g}_\theta$ is generated to enhance the feature discrimination. From this perspective, we interpret the nonlocal operator in the spectral view as below.

Theorem 1. *Given an affinity matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ and the signal $\boldsymbol{Z} \in \mathbb{R}^{N\times C_s}$, the nonlocal operator is the same as filtering the signal $\boldsymbol{Z}$ in the graph domain of a fully-connected weighted graph $\mathcal{G}$:*

$$\begin{aligned} &\mathcal{F}(\boldsymbol{A}, \boldsymbol{Z}) = \boldsymbol{Z} *_{\mathcal{G}} \mathbf{g}_\theta = \boldsymbol{U}\mathbf{g}(\boldsymbol{\Lambda})\boldsymbol{U}^\top \boldsymbol{Z} \\ &\text{with} \quad \boldsymbol{L} = \boldsymbol{D}_L - \boldsymbol{A} = \boldsymbol{U}^\top \boldsymbol{\Lambda} \boldsymbol{U}, \end{aligned} \tag{2}$$

where the fully-connected graph $\mathcal{G} = (\mathbb{V}, \boldsymbol{Z}, \boldsymbol{A})$ has the vertex set $\mathbb{V}$, node feature $\boldsymbol{Z}$ and affinity matrix $\boldsymbol{A}$. $\boldsymbol{\Lambda} = \operatorname{diag}\big(\{\lambda_1, \lambda_2, \cdots, \lambda_N\}\big)$ and $\boldsymbol{U} = \{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_N\}$ are the eigenvalues and eigenvectors of the graph Laplacian $\boldsymbol{L}$, respectively. $\boldsymbol{D}_L$ is the degree matrix of $\boldsymbol{L}$. For generality, the graph spectral filter function $\mathbf{g}(\boldsymbol{\Lambda})$ can be set as a diagonal parameter matrix $\boldsymbol{\Omega} \in \mathbb{R}^{N\times N}$, i.e., $\boldsymbol{\Omega} = \operatorname{diag}(\omega)$, $\omega = (\omega_1, \omega_2, \cdots, \omega_n)$.

Property 1. *Theorem.1 requires the graph Laplacian $\boldsymbol{L}$ has non-singular eigenvalues and eigenvectors. Thus, the affinity matrix $\boldsymbol{A}$ should be non-negative and symmetric.*

Remark 1. *Based on Theorem. 1, new nonlocal operators can be theoretically designed by using different types of graph filter such as the Chebyshev filter Defferrard et al. (2016a;b), the graph wavelet filter Hammond et al. (2011) and the Cayley filter Levie et al. (2018).*

The main difference between Theorem. 1 and the original spatial view of nonlocal Wang et al. (2018) is that the former learns a graph filter to aggregate the feature under the spectral domain while the latter aggregate the features in a non-parameter manner which cannot sufficiently suppress the noise. Moreover, Theorem. 1 enable us to theoretically analyze existing nonlocal blocks and design novel nonlocal blocks.

2.2 UNIFYING EXISTING NONLOCAL-BASED BLOCKS

To unifying other nonlocal-based block, here we use the Chebyshev filter to express the graph filter, i.e. using Chebyshev polynomials Defferrard et al. (2016a) to reduce the N parameters in $\boldsymbol{\Omega}$ into k (k is the order of polynomials, and $k \ll N$). For simplicity, we assume that the input $\boldsymbol{Z}$ has one channel. Then the graph filter approximated by k_{th}-order Chebyshev polynomials is formulated as:

$$\mathcal{F}(\boldsymbol{A}, \boldsymbol{Z}) = \sum_{k=0}^{K-1} \hat{\theta}_k T_k(\widetilde{\boldsymbol{L}})\boldsymbol{Z} \tag{3}$$

where $T_k(\widetilde{\boldsymbol{L}}) = 2\widetilde{\boldsymbol{L}}T_{k-1}(\widetilde{\boldsymbol{L}}) - T_{k-2}(\widetilde{\boldsymbol{L}})$ with $T_0(\widetilde{\boldsymbol{L}}) = \boldsymbol{I}_N$, $T_1(\widetilde{\boldsymbol{L}}) = \widetilde{\boldsymbol{L}}$. $\hat{\theta}_k$ is the coefficient of the k_{th} term which can be learned via SGD. $\widetilde{\boldsymbol{L}} = 2\boldsymbol{L}/\lambda_{\max} - \boldsymbol{I}_N$. Since $\boldsymbol{L}$ is a normalized graph Laplacican, the maximum eigenvalue $\lambda_{\max} = 2$, which makes $\widetilde{\boldsymbol{L}} = -\boldsymbol{A}$. Extending Eq. (3) into multiple channels, we can get a generalized formulation of the nonlocal operator with Chebyshev filter:

$$\mathcal{F}(\boldsymbol{A}, \boldsymbol{Z}) = \boldsymbol{Z}\mathbf{W}_1 + \boldsymbol{A}\boldsymbol{Z}\mathbf{W}_2 + \sum_{k=2}^{K-1} \boldsymbol{A}^k \boldsymbol{Z}\mathbf{W}_{k+1}, \tag{4}$$

where $F(\boldsymbol{A}, \boldsymbol{Z})$ is the nonlocal operator, $\mathbf{W}_k \in \mathbb{R}^{C_s\times C_1}$.

Property 2. *Eq. (4) requires that the graph Laplacican $\boldsymbol{L}$ is a normalized Laplacican whose maximum eigenvalue satisfies $\lambda_{\max} = 2$. Thus, $\widetilde{\boldsymbol{L}} = 2\boldsymbol{L}/\lambda_{\max} - \boldsymbol{I}_N = -\boldsymbol{A}$.*

Eq. (4) gives the connection between spatial view and spectral view of the nonlocal operator, in which the graph filter is expressed by the aggregation between the k_{th} neighbor nodes, i.e., all nodes for nonlocal. Thus, existing nonlocal-based structures can be theoretically analyzed by Eq. (4) in the spectral view. Here, we elaborate 5 types of existing nonlocal-based blocks that can be unified under certain graph structure and assumption summarized in Table. 1. More details of the proofs can be also found in the Appendix. B.

1) Original Nonlocal Block Wang et al. (2018): The Nonlocal (NL) Block in the spectral view is the same as defining the graph $\mathcal{G} = (\mathbb{V}, \boldsymbol{D}_M^{-1}\boldsymbol{M}, \boldsymbol{Z})$ and then using the second term of the Chebyshev polynomials to approximate the generalized graph filter, where $\boldsymbol{M}_{ij} = f(\boldsymbol{X}_{i,:}, \boldsymbol{X}_{j,:})$ is the dense attention map.

2) Nonlocal Stage Tao et al. (2018): The Nonlocal Stage (NS) in the spectral view is the same as defining the graph $\mathcal{G} = (\mathbb{V}, \boldsymbol{D}_M^{-1}\boldsymbol{M}, \boldsymbol{Z})$ and then using the 1_{st}-order Chebyshev polynomials to approximate the graph filter with the condition $\mathbf{W}_1 = \mathbf{W}_2 = -\mathbf{W}$.

3) Double Attention Block Chen et al. (2018): The Double Attention Block in the spectral view is the same as defining the graph $\mathcal{G} = (\mathbb{V}, \overline{\boldsymbol{M}}, \boldsymbol{Z})$ and then using the second term of the Chebyshev polynomial to approximate the graph filter, i.e $F(\boldsymbol{A}, \boldsymbol{Z}) = \overline{\boldsymbol{M}}\boldsymbol{Z}\mathbf{W}$, where $\overline{\boldsymbol{M}} = \sigma(\boldsymbol{X}\mathbf{W}_\phi)\sigma(\boldsymbol{X}\mathbf{W}_\psi)$.

4) Compact Generalized Nonlocal Block Yue et al. (2018): When grouping all channels into one group, the CGNL in the spectral view is the same as defining a channel-awareness graph $\mathcal{G} = (\mathbb{V}^f, \boldsymbol{M}^f, \text{vec}(\boldsymbol{Z}))$ and then using the second term of the Chebyshev Polynomail to approximate the graph filter, i.e $F(\boldsymbol{A}, \boldsymbol{Z}) = \boldsymbol{D}_{M^f}^{-1}\boldsymbol{M}^f\text{vec}(\boldsymbol{Z})\mathbf{W}$, where $\boldsymbol{M}_{ij}^f = f(vec(\boldsymbol{X})_i, vec(\boldsymbol{X})_j)$.

5) Criss-Cross Attention Block Huang et al. (2019): The Criss-Cross Attention Block in the spectral view is the same as defining a graph $\mathcal{G} = (\mathbb{V}, \boldsymbol{D}_{\boldsymbol{C}\odot\boldsymbol{M}}^{-1}\boldsymbol{C}\odot\boldsymbol{M}, \boldsymbol{X})$ with edge mask $\boldsymbol{C}$ and then using the second term of the Chebyshev Polynomail to approximate the graph filter with node feature $\boldsymbol{X}$.

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

Models	Vertex ($\lvert\mathbb{V}\rvert$)	Edge ($\lvert E\rvert$)	Affinity Matrix ($\boldsymbol{A}$)	Node Feature ($\boldsymbol{Z}$)	Filter ($F(\boldsymbol{A}, \boldsymbol{Z})$)
Unify	-	-	-	-	$\sum_{k=0}^{K-1}\boldsymbol{A}^k\boldsymbol{Z}\mathbf{W}_k$
NL	N	$N \times N$	$\boldsymbol{D}_M^{-1}\boldsymbol{M}$	$\boldsymbol{X}\mathbf{W}_Z$	$\boldsymbol{A}\boldsymbol{Z}\mathbf{W}$
A^2	N	$N \times N$	$\boldsymbol{M}$	$\boldsymbol{X}\mathbf{W}_Z$	$\boldsymbol{A}\boldsymbol{Z}\mathbf{W}$
CGNL	NC_s	$NC_s \times NC_s$	$\boldsymbol{M}^f$	$\text{vec}(\boldsymbol{X}\mathbf{W}_Z)$	$\boldsymbol{A}\boldsymbol{Z}\mathbf{W}$
NS	N	$N \times N$	$\boldsymbol{D}_M^{-1}\boldsymbol{M}$	$\boldsymbol{X}\mathbf{W}_Z$	$-\boldsymbol{Z}\mathbf{W} + \boldsymbol{A}\boldsymbol{Z}\mathbf{W}$
CC	N	$N \times N$	$\boldsymbol{D}_{\boldsymbol{C}\odot\boldsymbol{M}}^{-1}(\boldsymbol{C}\odot\boldsymbol{M})$	$\boldsymbol{X}$	$\boldsymbol{A}\boldsymbol{Z}\mathbf{W}$

2.3 Proposing Novel Nonlocal Blocks

Except for unifying existing nonlocal-based blocks, the proposed spectral view can also help to theoretically define novel nonlocal-based block.

Based on the above section, we can that existing nonlocal-based operators use the random walk normalized (NL, NS, CC) or the non-normalized affinity matrix (A^2, CGNL) whose symmetry is not guaranteed and depends on the affinity kernel. This makes the affinity matrix against Property. 1 and leads to the non-existence of the graph spectral domain. Thus, their robustness and flexibility are weakened.

Moreover, all existing nonlocal-based operators 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 hinders their performance.

Thus, still based on the Chebyshev filter, we firstly propose a more rational nonlocal-based block called the Chebyshev Spectral Nonlocal Block (ChebySNL) that uses a symmetry affinity matrix with a more complete approximation:

$$\begin{aligned} &\boldsymbol{Y} = \boldsymbol{X} + \mathcal{F}_s(\boldsymbol{A}, \boldsymbol{Z}) = \boldsymbol{X} + \boldsymbol{Z}\mathbf{W}_1 + \boldsymbol{A}\boldsymbol{Z}\mathbf{W}_2, \\ &\text{s.t.} \quad \boldsymbol{A} = \boldsymbol{D}_{\hat{M}}^{-\frac{1}{2}}\hat{\boldsymbol{M}}\boldsymbol{D}_{\hat{M}}^{-\frac{1}{2}}, \quad \hat{\boldsymbol{M}} = (\boldsymbol{M} + \boldsymbol{M}^\top)/2 \end{aligned} \tag{5}$$

where $\mathcal{F}_s(\boldsymbol{A}, \boldsymbol{Z})$ is the SNL operator, $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{C_s \times C_1}$, $\mathbf{W}_2$ are two parameter matrixes.

Remark 2. *The proposed ChebySNL following other nonlocal-based blocks that uses Chebyshev filter as the generalized graph filter but remove their irrationalities analyzed by our spectral view.*

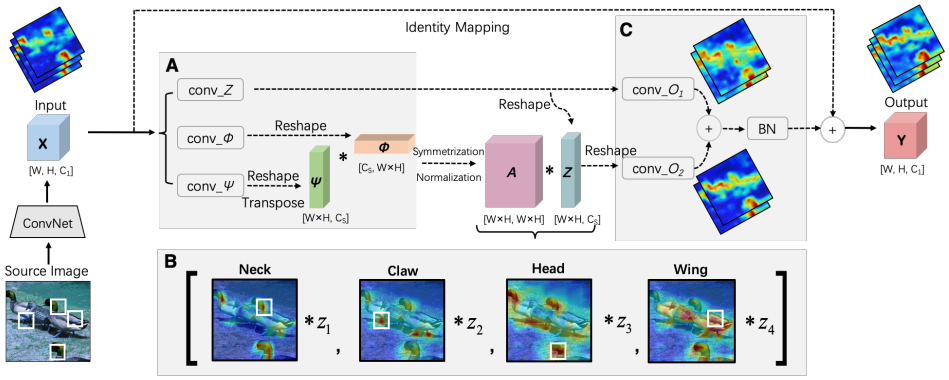

Figure 2: The implementation of our SNL. A. Three feature maps ϕ, ψ, $\boldsymbol{Z}$ are generated by feeding the $\mathbf{X}$ into three 1×1 convolutions. Then, the correlation matrix $\boldsymbol{M}$ is obtained by generating affinity kernel on ϕ and ψ. B The second term of Eq. (5) is calculated with $\boldsymbol{Z}$ and the a normalized symmetrization affinity matrix $\boldsymbol{A}$. Each row of $\boldsymbol{A}$ contains a N-dimension spatial attention map (heat maps) and $\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_n$ are the column vectors of $\boldsymbol{Z}$ (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 1^{st}-order term and the 0^{th}-order term: $\boldsymbol{Z}$ into two convolutions. Finally, batch normalization is conducted and added with $\mathbf{X}$ to obtain the output $\mathbf{Y}$.

Remark 3. *The proposed ChebySNL uses a symmetric affinity matrix $\boldsymbol{A} = \boldsymbol{D}_{\hat{M}}^{-\frac{1}{2}} \hat{\boldsymbol{M}} \boldsymbol{D}_{\hat{M}}^{-\frac{1}{2}}$ to ensure the existence of the real eigenvalue, which definitely satisfies the precondition of defining a graph filter. Thus, our ChebySNL is more stable when inserted into the deep neural networks.*

Remark 4. *The proposed ChebySNL uses the complete form of 1_{st}-order Chebyshev Approximation which is a more accurate approximation of the graph filter. Thus, our ChebySNL can give the parameters a liberal learning space with only one more parameter matrix.*

The implementation details of the ChebySNL block is shown in Fig. 2. 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 reshaped into $\mathbb{R}^{WH \times C_s}$. One of the output $\boldsymbol{Z} \in \mathbb{R}^{WH \times C_s}$ is used as the transferred feature map to reduce the calculation complexity, while the other two outputs $\boldsymbol{\Phi}, \boldsymbol{\Psi} \in \mathbb{R}^{WH \times C_s}$ are used to get the affinity matrix $\boldsymbol{A}$ with the affinity kernel function $f(\cdot)$. Then, $\boldsymbol{A}$ is made to be symmetric and normalized as in Eq. (5). Finally, with the affinity matrix $\boldsymbol{A}$ and the transferred feature map $\boldsymbol{Z}$, the output of the nonlocal block can be obtained by the Eq. (5). Specifically, the two weight matrices $\mathbf{W}_{1,2} \in \mathbb{R}^{C_s \times C_1}$ are implemented by two 1×1 convolutions.

3 Experiments

In this section, we design the ablation experiments to test the robustness of nonlocal-based blocks with different numbers, different positions, and different channels when inserted into deep models. Then, we show performance of the proposed SNL in 4 vision tasks, including image classification (Cifar-10/100, ImageNet), fine-grained image classification (CUB-200), action recognition (UCF-101). More experimental results on the person re-identification tested with ILID-SVID Wang et al. (2014), Mars Spr (2016), and Prid-2011 Hirzer et al. (2011) datasets are given in Appendix. C. All the methods are implemented using PyTorch Paszke et al. (2019) toolbox with an Intel Core i9 CPU and 2 Nvidia RTX 2080 Ti GPUs.

3.1 Ablation experiments on CIFAR-100

Experimental Setup The ablation experiments are conducted on CIFAR-100 dataset which contains $60,000$ images of 100 classes. We use $50,000$ images as the training set and $10,000$ images as the testing set. PreResNet56 He et al. (2016a) 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 ($res1$). The initial learning rate 0.1 is used with the weight decay 10^{-4} and momentum 0.9. The learning rate is divided by 10 at 150 and 250 epochs. All the models are trained for 300 epochs. We choose the evaluation criterion of the classification accuracy: Top1 and Top5 accuracy,

which means the model prediction (the one with the highest probability) is exactly the expected label and 5 highest probability predictions contains the expected label.

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 the computation complexity. If C_s is too large, the feature map will contain redundant information which introduces the noise when calculating the affinity matrix $\boldsymbol{A}$. However, if C_s is too small, it is hard to reconstruct the output feature map 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 setting: "No Reduction" ($C_s = C_1$), "Reduction by Two Times" ($C_s = C_1/2$), "Reduction by Four Times " ($C_s = C_1/4$). 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 performances of the CGNL steeply drops when the number of the transferred channels increases. This is because the CGNL block concerns the relations between channels. When the number of the transferred channels increases, the relations between the redundant channels seriously interfere with its effects. Overall, our SNL block is the most robust for the large number of transferred channels, which generates nearly $\times 5$ improvement than other nonlocals (our SNL rises 1.01% in Top1 while the best of others rises 0.18% over the backbone).

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

	No Reduction		Reduction by Two Times		Reduction by Four Times	
Models	Top1 (%)	Top5 (%)	Top1 (%)	Top5 (%)	Top1 (%)	Top5 (%)
PreResNet56	$75.33^{\uparrow 0.00}$	$93.97^{\uparrow 0.00}$	$75.33^{\uparrow 0.00}$	$93.97^{\uparrow 0.00}$	$75.33^{\uparrow 0.00}$	$93.97^{\uparrow 0.00}$
+ NL	$75.29^{\downarrow 0.04}$	$94.07^{\uparrow 0.10}$	$75.31^{\downarrow 0.02}$	$92.84^{\downarrow 1.13}$	$75.50^{\uparrow 0.17}$	$93.75^{\downarrow 0.22}$
+ NS	$75.39^{\uparrow 0.06}$	$93.00^{\downarrow 0.97}$	$75.83^{\uparrow 0.50}$	$93.87^{\downarrow 0.10}$	$75.61^{\uparrow 0.28}$	$93.66^{\downarrow 0.31}$
+ A^2	$75.51^{\uparrow 0.18}$	$92.90^{\downarrow 1.07}$	$75.58^{\uparrow 0.25}$	$94.27^{\uparrow 0.30}$	$75.61^{\uparrow 0.28}$	$93.61^{\downarrow 0.36}$
+ CGNL	$74.71^{\downarrow 0.62}$	$93.60^{\downarrow 0.37}$	$75.75^{\uparrow 0.42}$	$93.74^{\downarrow 0.23}$	$75.27^{\downarrow 0.06}$	$93.05^{\downarrow 0.92}$
+ Ours	$\underline{\mathbf{76.34}}^{\uparrow \mathbf{1.01}}$	$\underline{\mathbf{94.48}}^{\uparrow \mathbf{0.51}}$	$\underline{\mathbf{76.41}}^{\uparrow \mathbf{1.08}}$	$\underline{\mathbf{94.38}}^{\uparrow \mathbf{0.41}}$	$\underline{\mathbf{76.02}}^{\uparrow \mathbf{0.69}}$	$\underline{\mathbf{94.08}}^{\uparrow \mathbf{0.11}}$

The stage/position for adding the nonlocal-based blocks The nonlocal-based blocks can be added into the different stage of the preResNet to form the Nonlocal Network. In Tao et al. (2018), the nonlocal-based blocks are added into the early stage of the preResNet to catch the long-range relations. Here we show the performances of adding different types of nonlocal-based blocks into the 3 stages (the first, the second and the third stage of the preResNet). The experimental results are shown in Table 3. We can see that the performances of the NL block is lower than the backbones when added into the early stage. However, our proposed SNL block has averagely 1.08% improvement over the backbone when added into the early stage, which is more than $\times 2$ improvement over other types of nonlocal-based blocks (0.42% for the best case).

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

	Stage 1		Stage 2		Stage 3	
Models	Top1 (%)	Top5 (%)	Top1 (%)	Top5 (%)	Top1 (%)	Top5 (%)
PreResNet56	$75.33^{\uparrow 0.00}$	$93.97^{\uparrow 0.00}$	$75.33^{\uparrow 0.00}$	$93.97^{\uparrow 0.00}$	$75.33^{\uparrow 0.00}$	$\underline{\mathbf{93.97}}^{\uparrow \mathbf{0.00}}$
+ NL	$75.31^{\downarrow 0.02}$	$92.84^{\downarrow 1.13}$	$75.64^{\uparrow 0.31}$	$93.79^{\downarrow 0.18}$	$75.28^{\downarrow 0.05}$	$93.93^{\downarrow 0.04}$
+ NS	$75.83^{\uparrow 0.50}$	$93.87^{\downarrow 0.10}$	$75.74^{\uparrow 0.41}$	$94.02^{\uparrow 0.05}$	$75.44^{\uparrow 0.11}$	$93.86^{\downarrow 0.11}$
+ A^2	$75.58^{\uparrow 0.25}$	$94.27^{\uparrow 0.30}$	$75.60^{\uparrow 0.27}$	$93.82^{\downarrow 0.15}$	$75.21^{\downarrow 0.12}$	$93.65^{\downarrow 0.32}$
+ CGNL	$75.75^{\uparrow 0.42}$	$93.74^{\downarrow 0.23}$	$74.54^{\downarrow 0.79}$	$92.65^{\downarrow 1.32}$	$74.90^{\downarrow 0.43}$	$92.46^{\downarrow 1.51}$
+ Ours	$\underline{\mathbf{76.41}}^{\uparrow \mathbf{1.08}}$	$\underline{\mathbf{94.38}}^{\uparrow \mathbf{0.41}}$	$\underline{\mathbf{76.29}}^{\uparrow \mathbf{0.96}}$	$\underline{\mathbf{94.27}}^{\uparrow \mathbf{0.30}}$	$\underline{\mathbf{75.68}}^{\uparrow \mathbf{0.35}}$	$93.90^{\downarrow 0.07}$

The number of the nonlocal-based blocks We test the robustness for adding nonlocal-based blocks into the backbone. The results are shown in Table 4. "$\times 3$" means three blocks are added into the stage $1, 2$, and 3 respectively, and the accuracy in the brackets represent 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 when adding NS and NL into the early stage, these two models cannot well aggregate the low-level features and interfere with the following blocks.

Table 4: Experiments for Adding Different Number of Blocks into PreResNet56 on CIFAR-100

Models	Top1 (%)	Top5 (%)
PreResNet56	$75.33^{\uparrow 0.00}$	$93.97^{\uparrow 0.00}$
+ NL ($\times$3†)	$75.31^{\downarrow 0.02}$ ($74.34^{\downarrow 0.99}$)	$92.84^{\downarrow 1.13}$ ($93.11^{\downarrow 0.86}$)
+ NS ($\times$3)	$75.83^{\uparrow 0.50}$ ($75.00^{\downarrow 0.33}$)	$93.87^{\downarrow 0.10}$ ($93.57^{\downarrow 0.40}$)
+ A^2 ($\times$3)	$75.58^{\uparrow 0.25}$ ($75.63^{\uparrow 0.33}$)	$94.27^{\uparrow 0.30}$ (**94.12**$^{\uparrow \mathbf{0.15}}$)
+ CGNL ($\times$3)	$75.75^{\uparrow 0.42}$ ($75.96^{\uparrow 0.63}$)	$93.74^{\downarrow 0.23}$ ($93.10^{\downarrow 0.87}$)
+ Ours ($\times$3)	**76.41**$^{\uparrow \mathbf{1.08}}$ (**76.70**$^{\uparrow \mathbf{1.37}}$)	**94.38**$^{\uparrow \mathbf{0.41}}$ ($93.94^{\downarrow 0.03}$)

† *The number in the bracket means the performance when adding* 3 *this kind of nonlocal-based blocks.*

3.2 Applications on Computer Vision Tasks

Image Classification CIFAR-10/100 and ImageNet are tested where our SNL outperforms other types of the nonlocal-based blocks on these standard benchmarks. We use the ResNet50 He et al. (2016b) as the backbone and insert the SNL block right before the last residual block of *res4* for fair comparison. Other settings for the CIFAR-10/100 are the same as our ablation experiments (Sec. 3.1). For the ImageNet, the initial learning rate 0.01 is used with the weight decay 10^{-4} and momentum 0.9. The learning rate is divided by 31 at 61 and 81 epochs. All the models are trained for 110 epochs. The "Floating-point operations per second" (Flops) and the "Model size" (Size) are used to compare the computation complexity and memory consumption.

Table 5 shows the experimental results on the CIFAR10 dataset. When adding one proposed block, the Top1 classification accuracy rises about 0.38%, which is nearly twice over other types of nonlocal-based blocks (the best is 0.21%). As the experiments on CIFAR100 shown in Table 5, using our proposed block brings significant improvements about 1.67% with ResNet50. While using a more simple backbone PreResnet56 as shown in Table 5, our model can still generate 1.08% improvement which is not marginal.

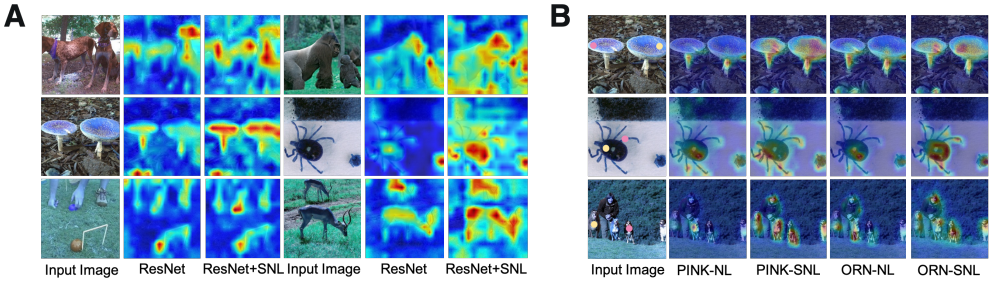

Figure 3: A. The visualization of the feature maps when adding SNL into the ResNet50 backbone. B. The visualization of the attention maps for two positions ("Pink" and "Orange" dots). The heatmaps show the strength of similarity between them and other positions.

The results of ImageNet are shown in Table 6. Note that other baselines are reported with the scores in their paper. We can see that compared with the nonlocal-based blocks, our SNL achieves a clear-cut improvement (1.96%) with a minor increment in complexity (0.51G Flops and 2.62M of Size compared with original 4.14G Flops and 25.56M). Moreover, our SNL is also superior to other types of blocks such as SE block Hu et al. (2018b), CGD block He et al. (2019), GE block Hu et al. (2018a) (0.11% higher in Top1 and 2.02M lower in size than the GE block).

Table 5: Experiments for Adding Different Types of Nonlocal-based Blocks into PreResnet56 and ResNet50 on CIFAR-10/100 Dataset

	CIFAR-10 / Resnet50		CIFAR-100 / Resnet50		CIFAR-100 / PreResnet56	
Models	Top1 (%)	Top5 (%)	Top1 (%)	Top5 (%)	Top1 (%)	Top5 (%)
Backbone	$94.94^{\uparrow 0.00}$	$99.87^{\uparrow 0.00}$	$76.50^{\uparrow 0.00}$	$93.14^{\uparrow 0.00}$	$75.33^{\uparrow 0.00}$	$93.97^{\uparrow 0.00}$
+ NL	$94.01^{\downarrow 0.93}$	$99.82^{\downarrow 0.05}$	$76.77^{\uparrow 0.27}$	$93.55^{\uparrow 0.41}$	$75.31^{\downarrow 0.02}$	$92.84^{\downarrow 1.33}$
+ NS	$95.15^{\uparrow 0.21}$	$99.88^{\uparrow 0.01}$	$77.90^{\uparrow 1.40}$	**94.34**$^{\uparrow \mathbf{1.20}}$	$75.83^{\uparrow 0.50}$	$93.87^{\downarrow 0.10}$
+ A^2	$94.41^{\downarrow 0.53}$	$99.83^{\downarrow 0.05}$	$77.30^{\uparrow 0.80}$	$93.40^{\uparrow 0.26}$	$75.58^{\uparrow 0.25}$	$94.27^{\uparrow 0.30}$
+ CGNL	$94.49^{\downarrow 0.45}$	$99.92^{\uparrow 0.05}$	$74.88^{\downarrow 1.62}$	$92.56^{\downarrow 0.58}$	$75.75^{\uparrow 0.42}$	$93.74^{\downarrow 0.23}$
+ Ours	**95.32**$^{\uparrow \mathbf{0.38}}$	**99.94**$^{\uparrow \mathbf{0.07}}$	**78.17**$^{\uparrow \mathbf{1.67}}$	$94.17^{\uparrow 1.03}$	**76.41**$^{\uparrow \mathbf{1.08}}$	**94.38**$^{\uparrow \mathbf{0.39}}$

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 between long-range spatial positions are enhanced as shown in the two mushroom, 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. 2.3.

Fine-grained Image Classification The experiments for the fine-grained classification are generated on the Birds-200-2011 (CUB-200) dataset which contains $11,788$ images of 200 categories of different birds. We use $5,994$ images as the training set and $5,794$ images as the testing set as Yue et al. (2018). We use the ResNet50 model pre-trained on ImageNet as the backbone and train the models for total 110 epochs with the initial learning rate 0.1 which is subsequently divided by 10 at 31, 61, 81 epochs. Table 7 (CUB-200) shows that our model can generate (0.59%) improvement. Compared with the CGNL block concerning channel-wise relations, our SNL is just a bit lower in Top1 (0.12%). That is because the dependencies between channels play an important role in the fine-grained classification. However, these channel dependencies of CGNL can impede the practical implementations, which needs elaborate preparations for the number of channels per block, the number of blocks and their positions as shown in Table 2, 3, 4. Compared with the other nonlocal block with non-channel concerned, our SNL has improvements with a large margin.

Action Recognition Experiments are conducted on the UCF-101 dataset, which contains $9,537$ videos for 101 different human actions. We use $7,912$ videos as the training set and $1,625$ videos as the testing set. Our SNL block are tested on the UCF-101 dataset for capturing the dependence for the temporal frames. We follow the I3D structure Hara et al. (2018) which uses $k \times k \times k$ kernels to replace the convolution operator in the residual block for learning seamless spatial-temporal feature extractors. The weights are initialized by the pre-trained I3D model on Kinetics dataset Kay et al. (2017). Inserting nonlocal-based blocks into the I3D can helps to capture the relations between frame pairs with long distance and improves the feature representation. We train the models with the initial learning rate of 0.01 which is subsequently divided by 10 each 40 epochs. The training stops at the 100 epochs. Other hyper-parameters of the experimental setup are the same as in Sec. 3.1.

Table 7 (UCF-101) shows the results on the action recognition. The network with our proposed block can generate significant improvements (2.82%) than the I3D and outperforms all other nonlocal-based models on the UCF-101 dataset. This shows that our proposed SNL is also effective for catching the long-range dependencies between the temporal frames. We also conduct the experiments on UCF-101 dataset with other state-of-the-art action recognition models in Appendix. C, which also shows that the effectiveness of our proposed ChebySNL.

Table 6: Experiments for adding different types of nonocal-based blocks into Resnet50 on ImageNet

Models	Top1 (%)	Flops (G)	Size (M)
ResNet50	$76.15^{\uparrow 0.00}$	4.14	25.56
+ CGD	$76.90^{\uparrow 0.75}$	$+0.01$	$+0.02$
+ SE	$77.72^{\uparrow 1.57}$	$+0.10$	$+2.62$
+ GE	$78.00^{\uparrow 1.85}$	$+0.10$	$+5.64$
+ NL	$76.70^{\uparrow 0.55}$	$+0.41$	$+2.09$
+ A^2	$77.00^{\uparrow 0.85}$	$+0.41$	$+2.62$
+ CGNL	$77.32^{\uparrow 1.17}$	$+0.41$	$+2.09$
+ Ours	$\underline{\mathbf{78.11}}^{\uparrow \mathbf{1.96}}$	$+0.51$	$+2.62$

Table 7: Experiments for adding NLs on CUB-200 and UCF-101 Datasets

	CUB-200	UCF-101
Models	Top1 (%)	Top1 (%)
Backbone†	$85.43^{\uparrow 0.00}$	$81.57^{\uparrow 0.00}$
+ NL	$85.34^{\downarrow 0.09}$	$82.88^{\uparrow 1.31}$
+ NS	$85.54^{\uparrow 0.11}$	$82.50^{\uparrow 0.93}$
+ A^2	$85.91^{\uparrow 0.48}$	$82.68^{\uparrow 1.11}$
+ CGNL	$\underline{\mathbf{86.14}}^{\uparrow \mathbf{0.71}}$	$83.38^{\uparrow 1.81}$
+ Ours	$\underline{\mathbf{86.02}}^{\uparrow \mathbf{0.59}}$	$\underline{\mathbf{84.39}}^{\uparrow \mathbf{2.82}}$

† The ResNet50 is used for CUB-200 as the backbone and I3D is used for UCF-101 dataset.

4 CONCLUSION

In this paper, we propose a framework to design nonlocal which can captures the long-range dependencies between spatial pixels (image classification) or temporal frames (video classification). Inspired by our framework, we can interpret the existing nonlocals in the graph view, and design the ChebySNL block, which is more robust and effective. Experimental results demonstrate the clear-cut improvements across four vision tasks.

REFERENCES

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.

Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. Aˆ 2-nets: Double attention networks. In *Neural Information Processing Systems (NeurIPS)*, pp. 352–361, 2018.

Nieves Crasto, Philippe Weinzaepfel, Karteek Alahari, and Cordelia Schmid. MARS: Motion-Augmented RGB Stream for Action Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. pp. 764–773, 2017.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Neural Information Processing Systems (NeurIPS)*, pp. 3844–3852, 2016a.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Neural Information Processing Systems (NeurIPS)*, pp. 3844–3852, 2016b.

Jiyang Gao and Ram Nevatia. Revisiting temporal modeling for video-based person reid. *arXiv preprint arXiv:1805.02104*, 2018.

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.

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)*, pp. 6546–6555, 2018.

Kaiming He and Jian Sun. Convolutional neural networks at constrained time cost. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5353–5360, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision (ECCV)*, pp. 630–645. Springer, 2016a.

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)*, pp. 770–778, 2016b.

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.

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.

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)*, 2018a.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018b.

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)*, pp. 603–612, 2019.

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.

Alexander Kozlov, Vadim Andronov, and Yana Gritsenko. Lightweight network architecture for real-time action recognition. *arXiv preprint arXiv:1905.08711*, 2019.

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.

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)*, pp. 4898–4906, 2016.

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)*, pp. 8024–8035, 2019.

Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 5533–5541, 2017.

MARS: A Video Benchmark for Large-Scale Person Re-identification, 2016. Springer.

Yunzhe Tao, Qi Sun, Qiang Du, and Wei Liu. Nonlocal neural networks, nonlocal diffusion and nonlocal modeling. In *Neural Information Processing Systems (NeurIPS)*, pp. 496–506, 2018.

Taiqing Wang, Shaogang Gong, Xiatian Zhu, and Shengjin Wang. Person re-identification by video ranking. In *European Conference on Computer Vision (ECCV)*, pp. 688–703. Springer, 2014.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7794–7803, 2018.

Kaiyu Yue, Ming Sun, Yuchen Yuan, Feng Zhou, Errui Ding, and Fuxin Xu. Compact generalized non-local network. In *Neural Information Processing Systems (NeurIPS)*, pp. 6510–6519, 2018.

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)*, pp. 2881–2890, 2017.

Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. pp. 9308–9316, 2019.

A BACKGROUND

A.1 NONLOCAL BLOCK

The Nonlocal Block (NL) follows the nonlocal operator that calculates a weighted mean between the features of each position and all possible positions as shown in Fig. 1 A. The nonlocal operator is defined as:

$$F(\boldsymbol{X}_{i,:}) = \sum_j \Big[f(\boldsymbol{X}_{i,:}, \boldsymbol{X}_{j,:}) g(\boldsymbol{X}_{j,:}) \Big] / \sum_j f(\boldsymbol{X}_{i,:}, \boldsymbol{X}_{j,:}) \tag{6}$$

where $\boldsymbol{X} \in \mathbb{R}^{N \times C_1}$ is the input feature map, i, j are the position indexes in the feature map, $f(\cdot)$ is the affinity kernel which can adopt the "Dot Product", "Traditional Gaussian", "Embedded Gaussian" or other kernel metrics with a finite Frobenius norm. $g(\cdot)$ is a linear embedding that is defined as: $g(\boldsymbol{X}_{j,:}) = \boldsymbol{X}_{j,:} \mathbf{W}_Z$ with $\mathbf{W}_Z \in \mathbb{R}^{C_1 \times C_s}$. Here N is the total positions of each features and C_1, C_s are the number of channels for the input and the transferred features.

When inserting the NL block into the network structure, a linear transformation and a residual connection are added:

$$\boldsymbol{Y}_{i,:} = \boldsymbol{X}_{i,:} + F(\boldsymbol{X}_{i,:}) \mathbf{W}, \tag{7}$$

where $\mathbf{W} \in \mathbb{R}^{C_s \times C_1}$ is the weight matrix.

A.2 GRAPH FOURIER TRANSFORM & GRAPH FILTER

The graph Fourier transform $\hat{\mathscr{F}}$ of any function $\mathscr{F} \in \mathbb{R}^N$ on the vertices of a graph $\mathcal{G}$ is defined as the expansion of $\mathscr{F}$ in terms of the eigenvectors of the graph Laplacian:

$$\hat{\mathscr{F}}(\lambda_l) := \langle \mathscr{F}, u_l \rangle = \sum_{i=1}^{N} \mathscr{F}(i) u_l^*(i) \tag{8}$$

where $\lambda = \{\lambda_1, \lambda_2, ..., \lambda_l, ...\}$ and $\boldsymbol{U} = \{\mathbf{u}_1, \mathbf{u}_2, ...\mathbf{u}_l, ...\}$ are the eigenvalue and eigenvector of the graph Laplacian, $\mathbf{u}_l^*$ is the l_{th} column vector of $\boldsymbol{U}^\top$.

The inverse graph Fourier transform $\hat{\mathscr{F}}^{-1}$ then given by

$$\hat{\mathscr{F}}^{-1}(\lambda_l) = \sum_{i=1}^{N} \mathscr{F}(i) u_l(i) \tag{9}$$

Based on the graph Fourier transform, the graph convolution of the input signal $\mathbf{x}$ with a filter $\mathbf{g}_\theta$ can be defined as

$$\mathbf{x} *_{\mathcal{G}} \mathbf{g}_\theta = \mathscr{F}^{-1}(\mathscr{F}(\mathbf{x}) \odot \mathscr{F}(\mathbf{g})) \tag{10}$$

where $\odot$ the Hadamard product.

B DETAILS OF THE PROVING FOR THE RELATIONS

We give the details of the relationship between other nonlocal operators in the spectral view discussed in our paper as shown in Table 1. In the following proving, we assume that $X \in \mathbb{R}^{\mathbb{N} \times \mathbb{C}}$, $\boldsymbol{Z} = g(\boldsymbol{X}) = \boldsymbol{X}\mathbf{W}_Z$, $\boldsymbol{M}_{ij} = f(\boldsymbol{X}_i, \boldsymbol{X}_j)$. All the normalized term uses the inverse of the degree $1/d_i$ where $d_i = \sum_j f(\boldsymbol{X}_i, \boldsymbol{X}_j)$. We also merge the output of the operators with the weight kernel $\mathbf{W} \in \mathbb{R}^{N \times C}$ and defines it as $\boldsymbol{O}$ for consistency. Thus the target formulations in this section are a bit different with the definition in their own papers.

B.1 NONLOCAL BLOCK

The Nonlocal (NL) Block in the spectral view is the same as defining the graph $\mathcal{G} = (\mathbb{V}, \boldsymbol{D}^{-1}\boldsymbol{M}, \boldsymbol{Z})$ and then using the second term of the Chebyshev Polynomail to approximate the graph filter.

Proof. The NL operator defined in Wang et al. (2018) can be formulated as:

$$\boldsymbol{O}_{i,:} = \frac{\sum_j \left[f(\boldsymbol{X}_{i,:}, \boldsymbol{X}_{j,:}) g(\boldsymbol{X}_{j,:}) \right]}{\sum_j f(\boldsymbol{X}_{i,:}, \boldsymbol{X}_{j,:})} \mathbf{W} \tag{11}$$

To unify it by our spectral view, we firstly define the graph $\mathcal{G} = (\mathbb{V}, \boldsymbol{A}, \boldsymbol{Z})$ to represent the graph structure of the NL operator, where the affinity matrix A is calculated by:

$$\boldsymbol{A} = \boldsymbol{D}_M^{-1} \boldsymbol{M}, \quad \boldsymbol{M} = f(\boldsymbol{X}_{i,:}, \boldsymbol{X}_{j,:}) \tag{12}$$

Thus, each element of the affinity matrix $\boldsymbol{A}$ is:

$$\boldsymbol{A}_{ij} = (\boldsymbol{D}_M^{-1} \boldsymbol{M})_{ij} = \frac{f(\boldsymbol{X}_{i,:}, \boldsymbol{X}_{j,:})}{\sum_j f(\boldsymbol{X}_{i,:}, \boldsymbol{X}_{j,:})} \tag{13}$$

Based on Theorem. 1, when using Chebyshev polynomail to approximate the generalized graph filter $\boldsymbol{\Omega}$ and only choosing the second term, it becomes :

$$F(\boldsymbol{A}, \boldsymbol{Z}) = \boldsymbol{A}\boldsymbol{Z}\mathbf{W} \tag{14}$$

Then taking Eq. (13) into this equation, we can get the formulation of the NL operator:

$$F_{i,:}(\boldsymbol{A}, \boldsymbol{Z}) = \frac{\sum_j \left[f(\boldsymbol{X}_{i,:}, \boldsymbol{X}_{j,:}) g(\boldsymbol{X}_{j,:}) \right]}{\sum_j f(\boldsymbol{X}_{i,:}, \boldsymbol{X}_{j,:})} \mathbf{W} \tag{15}$$

□

B.2 NONLOCAL STAGE

The Nonlocal Stage (NS) in the spectral view is the same as defining the graph $\mathcal{G} = (\mathbb{V}, \boldsymbol{D}_M^{-1}\boldsymbol{M}, \boldsymbol{Z})$ and then using the 1_{st}-order Chebyshev Polynomail to approximate the graph filter with the condition $\mathbf{W}_1 = \mathbf{W}_2 = -\mathbf{W}$.

Proof. The NS operator given defined in Tao et al. (2018) can be formulated as:

$$\boldsymbol{O}_{i,:} = \frac{\sum_j \left[f(\boldsymbol{X}_{i,:}, \boldsymbol{X}_{j,:}) (\boldsymbol{Z}_{j,:} - \boldsymbol{Z}_{i,:}) \right]}{\sum_j f(\boldsymbol{X}_{i,:}, \boldsymbol{X}_{j,:})} \mathbf{W} \tag{16}$$

Similar with the proof of NL, we can get each element of the affinity matrix $\boldsymbol{A}$ as:

$$\boldsymbol{A}_{ij} = (\boldsymbol{D}_M^{-1} \boldsymbol{M})_{ij} = \frac{f(\boldsymbol{X}_{i,:}, \boldsymbol{X}_{j,:})}{\sum_j f(\boldsymbol{X}_{i,:}, \boldsymbol{X}_{j,:})} \tag{17}$$

The graph filter $\boldsymbol{\Omega}$ on $\mathcal{G}$ is approximated by the Chebyshev polynomail. When using the 1_{st}-order Chebyshev Approximation, it becomes:

$$F(\boldsymbol{A}, \boldsymbol{Z}) = \boldsymbol{Z}\mathbf{W}_1 - \boldsymbol{A}\boldsymbol{Z}\mathbf{W}_2 \tag{18}$$

When sharing the weight for $\mathbf{W}_1$ and $\mathbf{W}_2$, i.e $\mathbf{W}_1 = \mathbf{W}_2 = -\mathbf{W}$, we get:

$$F(\boldsymbol{A}, \boldsymbol{Z}) = \boldsymbol{A}\boldsymbol{Z}\mathbf{W} - \boldsymbol{Z}\mathbf{W} \tag{19}$$

Then, taking it $\boldsymbol{Z} = g(\boldsymbol{X}) = \boldsymbol{X}\mathbf{W}_Z$ and Eq. (17) into this equation, it becomes:

$$F_{i,:}(\boldsymbol{A}, \boldsymbol{Z}) = \frac{\sum_j \left[f(\boldsymbol{X}_{i,:}, \boldsymbol{X}_{j,:}) \boldsymbol{Z}_{j,:}\mathbf{W} \right]}{\sum_j f(\boldsymbol{X}_{i,:}, \boldsymbol{X}_{j,:})} - \boldsymbol{Z}_i\mathbf{W} \tag{20}$$

Due to the fact that $\frac{\sum_j f(\boldsymbol{X}_{i,:},\boldsymbol{X}_{j,:})}{\sum_j f(\boldsymbol{X}_{i,:},\boldsymbol{X}_{j,:})} = 1$, we can get the formulation of the NS operator:

$$\begin{aligned} F_i(\boldsymbol{A},\boldsymbol{Z}) &= \frac{\sum_j \left[f(\boldsymbol{X}_{i,:},\boldsymbol{X}_{j,:})\boldsymbol{Z}_{j,:}\mathbf{W}\right]}{\sum_j f(\boldsymbol{X}_{i,:},\boldsymbol{X}_{j,:})} \\ &\qquad - \frac{\sum_j f(\boldsymbol{X}_{i,:},\boldsymbol{X}_{j,:})}{\sum_{j,:} f(\boldsymbol{X}_{i,:},\boldsymbol{X}_{j,:})}\boldsymbol{Z}_{i,:}\mathbf{W} \\ &= \frac{\sum_j \left[f(\boldsymbol{X}_{i,:},\boldsymbol{X}_{j,:})(\boldsymbol{Z}_{j,:} - \boldsymbol{Z}_{i,:})\right]}{\sum_j f(\boldsymbol{X}_{i,:},\boldsymbol{X}_{j,:})}\mathbf{W} \end{aligned} \tag{21}$$

□

B.3 Double Attention Block

The Double Attention Block in the spectral view is the same as defining the graph $\mathcal{G} = (\mathbb{V}, \overline{\boldsymbol{M}}, \boldsymbol{Z})$ and then using the second term of the Chebyshev Polynomail to approximate the graph filter, i.e $F(\boldsymbol{A},\boldsymbol{Z}) = \overline{\boldsymbol{M}}\boldsymbol{Z}\mathbf{W}$:

Proof. The A^2 operator defined in Chen et al. (2018) can be formulated as:

$$\boldsymbol{O} = \sigma(\theta(\boldsymbol{X}))\sigma(\phi(\boldsymbol{X})^T)g(\boldsymbol{X}) = f^a(\boldsymbol{X}_{i,:},\boldsymbol{X}_{j,:})\boldsymbol{X}\mathbf{W} \tag{22}$$

The difference between the double A^2 operator and the NL operator is only the kernel function that calculating the affinity matrix. Thus we can use the similar proving strategy to reformulate the A^2 operator into the spectral only by change the affinity matrix as:

$$\boldsymbol{A} = \overline{\boldsymbol{M}} = \sigma(\boldsymbol{X}\mathbf{W}_\phi)\sigma(\boldsymbol{X}\mathbf{W}_\psi) \tag{23}$$

□

B.4 Compact Generalized Nonlocal Block

When grouping all channels into one group, the Compact Generalized Nonlocal Block in the spectral view is the same as defining the graph $\mathcal{G} = (\mathbb{V}^f, \boldsymbol{D}_{M^f}^{-1}\boldsymbol{M}^f, \mathrm{vec}(\boldsymbol{Z}))$ and then using the second term of the Chebyshev Polynomail to approximate the graph filter, i.e $F(\boldsymbol{A},\boldsymbol{Z}) = \boldsymbol{D}_{M^f}^{-1}\boldsymbol{M}^f\mathrm{vec}(\boldsymbol{Z})W$. Note that due to the dimension of the input feature $\mathrm{vec}(\boldsymbol{Z}) \in \mathbb{R}^{NC\times 1}$ which is different with other nonlocal operators, here we uses $\boldsymbol{M}^f, \boldsymbol{A}^f \in \mathbb{R}^{NC\times NC}$ for clearity.

Proof. The CGNL operator defined in Yue et al. (2018) can be formulated as:

$$\mathrm{vec}(\boldsymbol{O}) = f(\mathrm{vec}(\boldsymbol{X}),\mathrm{vec}(\boldsymbol{X}))\mathrm{vec}(\boldsymbol{Z})\mathbf{W} \tag{24}$$

For simplicity, we use $\boldsymbol{x}$ to represent $\mathrm{vec}(\boldsymbol{X})$, thus the target becomes:

$$\boldsymbol{o} = f(\boldsymbol{x},\boldsymbol{x})\boldsymbol{z}\boldsymbol{W} \tag{25}$$

Then, we define the graph $\mathcal{G} = (\mathbb{V}^f, \boldsymbol{A}^f, \boldsymbol{z})$, where the set $\mathbb{V}^f$ contains each index (including position and channel) of the vector $\boldsymbol{x}$. The affinity matrix $\boldsymbol{A}$ is calculated by:

$$\boldsymbol{A}^f = \boldsymbol{M}^f, \quad \boldsymbol{M}^f = f(\boldsymbol{x},\boldsymbol{x}) \tag{26}$$

The graph filter $\mathbf{\Omega}$ on $\mathcal{G}$ is approximated by the Chebyshev polynomials. When only choosing the second term, we can get the formulation of the CGNL operator:

$$F(\boldsymbol{A}^f, \boldsymbol{z}) = \boldsymbol{A}^f \boldsymbol{z}\mathbf{W} = f(\boldsymbol{x}, \boldsymbol{x})\boldsymbol{z}\mathbf{W} \tag{27}$$

□

B.5 CRISS-CROSS ATTENTION BLOCK

The Criss-Cross Attention Block in the spectral view is the same as defining the graph $\mathcal{G} = (\mathbb{V}, \boldsymbol{D}^{-1}_{\boldsymbol{C}\odot\boldsymbol{M}}\boldsymbol{C}\odot\boldsymbol{M}, \boldsymbol{X})$ and then using the second term of the Chebyshev Polynomail to approximate the graph filter with node feature $\boldsymbol{X}$:

Proof. The criss-cross attention operator defined in Huang et al. (2019) can be formulated as:

$$\boldsymbol{O}_{i,:} = \sum_{j\in\mathbb{V}^i} \boldsymbol{A}_{ij}\boldsymbol{\Phi}_{j,:} = \frac{\sum_{j\in\mathbb{V}^i} f(\boldsymbol{X}_{i,:}, \boldsymbol{X}_{j,:})\boldsymbol{X}_{j,:}}{\sum_{j\in\mathbb{V}^i} f(\boldsymbol{X}_{i,:}, \boldsymbol{X}_{j,:})}\mathbf{W}$$

where the set $\mathbb{V}^i$ is collection of feature vector in $\mathbb{V}$ which are in the same row or column with position u.

Then, we define the graph $\mathcal{G} = (\mathbb{V}, \widetilde{\boldsymbol{A}}, \boldsymbol{X})$ to represennt the criss-cross attention operator in the spectral view. The affinity matrix $\widetilde{\boldsymbol{A}}$ is calculated by:

$$\begin{aligned}
&\widetilde{\boldsymbol{A}} = \boldsymbol{D}^{-1}_{\boldsymbol{C}\odot\boldsymbol{M}}\boldsymbol{C}\odot\boldsymbol{M}, \quad \boldsymbol{M} = f(\boldsymbol{X}_i, \boldsymbol{X}_j) \\
&\boldsymbol{C}_{ij} = \begin{cases} 1 & j \in \mathbb{V}^i \\ 0 & \text{else} \end{cases},
\end{aligned}$$

We use $\widetilde{\boldsymbol{M}}$ to represent $\boldsymbol{C}\odot\boldsymbol{M}$, i.e. $\widetilde{\boldsymbol{M}} = \boldsymbol{C}\odot\boldsymbol{M}$. Thus, each element of the affinity matrix $\widetilde{\boldsymbol{M}}$ is:

$$\widetilde{\boldsymbol{M}}_{ij} = \begin{cases} \boldsymbol{M}_{ij} & j \in \mathbb{V}^i \\ 0 & \text{else} \end{cases},$$

Thus, we can get the definition of each element in the affinity matrix $\widetilde{\boldsymbol{A}}$:

$$\widetilde{\boldsymbol{A}}_{ij} = \begin{cases} \frac{f(\boldsymbol{X}_i, \boldsymbol{X}_j)}{\sum_{j\in\mathbb{V}^i} f(\boldsymbol{X}_i, \boldsymbol{X}_j)}, & j \in \mathbb{V}^i \\ 0, & \text{else} \end{cases},$$

When using the Chebyshev polynomials to approximate the generalized graph filter $\mathbf{\Omega}$ on $\mathcal{G}$ and choose the second term, it becomes:

$$F(\widetilde{\boldsymbol{A}}, \boldsymbol{X}) = \widetilde{\boldsymbol{A}}\boldsymbol{X}\mathbf{W} \tag{28}$$

When taking Eq.28 into this formulation, we can get the formulation of CC operator:

$$\begin{aligned}
F_{i,:}(\widetilde{\boldsymbol{A}}, \boldsymbol{X}) &= \Big(\frac{\sum_{j\in\mathbb{V}^i} f(\boldsymbol{X}_{i,:}, \boldsymbol{X}_{j,:})\boldsymbol{X}_{j,:}}{\sum_{j\in\mathbb{V}^i} f(\boldsymbol{X}_{i,:}, \boldsymbol{X}_{j,:})} + \sum_{j\notin\mathbb{V}^i} 0\boldsymbol{X}_{j,:}\Big)\mathbf{W} \\
&= \frac{\sum_{j\in\mathbb{V}^i} f(\boldsymbol{X}_{i,:}, \boldsymbol{X}_{j,:})\boldsymbol{X}_{j,:}}{\sum_{j\in\mathbb{V}^i} f(\boldsymbol{X}_{i,:}, \boldsymbol{X}_{j,:})}\mathbf{W}
\end{aligned} \tag{29}$$

□

C EXTERNAL EXPERIMENT

C.1 EXTERNAL EXPERIMENT ON ACTION RECOGNIZATION

We also conduct the experiments on UCF-101 dataset with other state-of-the-art action recognition models in our supplementary materials including the P3D Qiu et al. (2017), the MARS Crasto et al. (2019), and the VTN Kozlov et al. (2019). For Pseudo 3D Convolutional Network (P3D) and Motion-augmented RGB Stream (MARS), our SNL block are inserted into the P3D right before the last residual layer of the $res3$. For the Video Transformer Network (VTN), we replace its multi-head self-attention blocks (paralleled-connected NL blocks) into our SNL blocks. We use the model pre-trained on Kinetic dataset and fine-tuning on the UCF-101 dataset. Other setting such as the learning rate and training epochs are the same as the experiment on I3D in our paper. We can see that all the performance are improved when adding our proposed SNL model especially when training end-to-end on the small-scale dataset. In sum, our SNL blocks have shown superior results across three SOTAs (the VTN and MARS) in the action recognition tasks (0.30% improvement with VTN, 0.50% improvement with MARS).

Table 8: Experiments with state-of-the-art backbone

Models	Top1(%)
P3D Qiu et al. (2017)	$81.23^{\uparrow 0.00}$
P3D + **Ours**	$82.65^{\uparrow 1.42}$
VTN Kozlov et al. (2019)	$90.06^{\uparrow 0.00}$
VTN + **Ours**	$90.34^{\uparrow 0.30}$
MARS Crasto et al. (2019)	$92.29^{\uparrow 0.00}$
MARS + **Ours**	$92.79^{\uparrow 0.50}$

C.2 EXPERIMENT ON VIDEO PERSON RE-IDENTIFICATION

Table 9: Experiments on Video-Person Reidentification

	Mars		ILID-SVID		PRID-2011	
Models	Rank1(%)	mAP(%)	Rank1(%)	mAP(%)	Rank1(%)	mAP(%)
ResNet50tp	$82.30^{\uparrow 0.00}$	$75.70^{\uparrow 0.00}$	$74.70^{\uparrow 0.00}$	$81.60^{\uparrow 0.00}$	$86.50^{\uparrow 0.00}$	$90.50^{\uparrow 0.00}$
+ NL	$83.21^{\uparrow 0.91}$	$76.54^{\uparrow 0.84}$	$75.30^{\uparrow 0.60}$	$83.00^{\uparrow 1.40}$	$85.40^{\downarrow 1.10}$	$89.70^{\downarrow 0.80}$
+ **SNL**	$83.40^{\uparrow 1.10}$	$76.80^{\uparrow 1.10}$	$76.30^{\uparrow 1.60}$	$84.80^{\uparrow 3.20}$	$88.80^{\uparrow 2.30}$	$92.40^{\uparrow 1.90}$

For the backbone, we follow the strategy of Gao & Nevatia (2018) that use the pooling (RTMtp) to fuse the spatial-temporal features. (Note that the models are totally trained on ilidsvid and prid2011 rather than fintuning the pretrained model on Mars.) We only insert the SNL block into the ResNet50 (right before the last residual block of *res4*) in RTMtp. Other block setting are the same as the setting for fine-grained image classification on CUB in our paper. For all those datasets we train the model with Adam with the initial learning rate $3e-4$, the weight decay $5e-4$. The learning rate is divided by 200 at 400 epochs. All the models are trained for 500 epochs with the Cross Entropy and Triplet Loss($margin = 0.3$). We use rank-1 accuracy (Rank1) and men average precision (mAP) to evaluate the performance for the models.

For the large scale dataset MarsSpr (2016) which contains 1261 pedestrians captured by at least 2 cameras. The bounding boxes are generated by the detection algorithm DPM and tracking algorithm GMMCP, which forms 20715 person sequences. From Table. 9 (Mars), we can see that our SNL can generate 1.10% improvement both on Rank1 and mAP than the backbone, which are both higher than the original nonlocal block ($0.91\% on Rank1$, 0.84% on mAP).

We also generate experiments on two relatively small datasets: ILID-SVID datasets which contains 300 pedestrians captured by two cameras with 600 tracklets; PRID-2011 dataset which contains 200 pedestrians captured by two cameras with 400 tracklets. From Table. 9 (ILID-SVID), we can see that our model can generate 1.60% and 3.20% improvement on the Rank1 and mAP respectively for the

ILID-SVID dataset. Moreover, on PRID-2011, we get a more higher improvement (2.30% on Rank1, 1.90% on mAP) as shown in Table. 9 (PRID-2011).