

Disentangled Non-Local Neural Networks

Minghao Yin^{1*}, Zhuliang Yao^{1,2*}, Yue Cao², Xiu Li¹, Zheng Zhang², Stephen Lin², and Han Hu²

¹ Tsinghua University

{yinmh17,yzl17}@mails.tsinghua.edu.cn li.xiu@sz.tsinghua.edu.cn

² Microsoft Research Asia

{yuecao,zhez,stevelin,hanhу}@microsoft.com

Abstract. The non-local block is a popular module for strengthening the context modeling ability of a regular convolutional neural network. This paper first studies the non-local block in depth, where we find that its attention computation can be split into two terms, a whitened pairwise term accounting for the relationship between two pixels and a unary term representing the saliency of every pixel. We also observe that the two terms trained alone tend to model different visual clues, e.g. the whitened pairwise term learns within-region relationships while the unary term learns salient boundaries. However, the two terms are tightly coupled in the non-local block, which hinders the learning of each. Based on these findings, we present the disentangled non-local block, where the two terms are decoupled to facilitate learning for both terms. We demonstrate the effectiveness of the decoupled design on various tasks, such as semantic segmentation on Cityscapes, ADE20K and PASCAL Context, object detection on COCO, and action recognition on Kinetics.

1 Introduction

The non-local block [34], which models long-range dependency between pixels, has been widely used for numerous visual recognition tasks, such as object detection, semantic segmentation, and video action recognition. Towards better understanding the non-local block’s efficacy, we observe that it can be viewed as a self-attention mechanism for pixel-to-pixel modeling. This self-attention is modeled as the dot-product between the features of two pixels in the embedding space. At first glance, this dot-product formulation represents *pairwise* relationships. After further consideration, we find that it may encode *unary* information as well, in the sense that a pixel may have its own independent impact on all other pixels. Based on this perspective, we split the dot-product based attention into two terms: a whitened pairwise term that accounts for the impact of one pixel *specifically* on another pixel, and a unary term that represents the influence of one pixel *generally* over all the pixels.

* Equal contribution. This work is done when Minghao Yin and Zhuliang Yao are interns at MSRA.

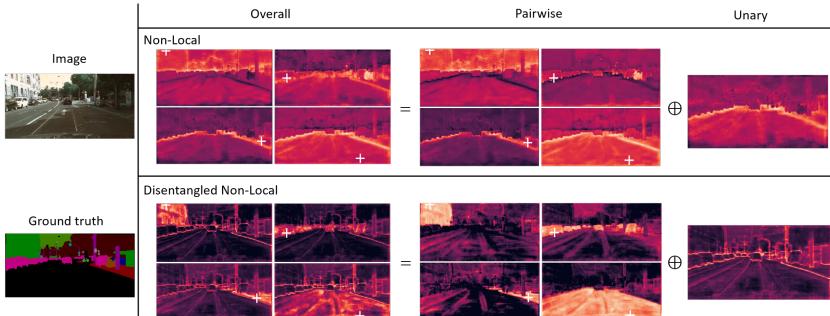


Fig. 1. Visualization of attention maps in the non-local block and our disentangled non-local block. With the disentanglement of our non-local block, the whitened pairwise term learns clear within-region clues while the unary term learns salient boundaries, which cannot be observed with the original non-local block

We investigate the visual properties of each term without interference from the other. Specifically, we train two individual networks, with either the whitened pairwise term or the unary term removed in the standard attention formula of the non-local block. It is found that the non-local variant using the whitened pairwise term alone generally learns within-region relationships (the 2nd row of Fig. 3), while the variant using the unary term alone tends to model salient boundaries (the 3rd row of Fig. 3). However, the two terms do not learn such clear visual clues when they are both present within a non-local block, as illustrated in the top row of Fig. 1. This observation is verified via statistical analysis on the whole validation set. Also, the standard non-local block combining both terms performs even worse than the variant that includes only the unary term (shown in Table 2). This indicates that coupling the two terms together may be detrimental to the learning of these visual clues, and consequently affects the learning of discriminative features.

To address this problem, we present the disentangled non-local (DNL) block, where the whitened pairwise and unary terms are cleanly decoupled by using independent *Softmax* functions and embedding matrices. With this disentangled design, the difficulty in joint learning of the whitened pairwise and unary terms is greatly diminished. As shown in second row of Fig. 1, the whitened pairwise term learns clear within-region clues while the unary term learns salient boundaries, even more clearly than what is learned when each term is trained alone.

The disentangled non-local block is validated through various vision tasks. On semantic segmentation benchmarks, by replacing the standard non-local block with the proposed DNL block with all other settings unchanged, significantly greater accuracy is achieved, with a 2.0% mIoU gain on the Cityscapes validation set, 1.3% mIoU gain on ADE20k, and 3.4% on PASCAL-Context using a ResNet-101 backbone. With few bells and whistles, our DNL obtains state-of-the-art performance on the challenging ADE20K dataset. Also, with a task-specific DNL block, noticeable accuracy improvements are observed on both COCO object detection and Kinetics action recognition.

2 Related Works

Non-local/self-attention. These terms may appear in different application domains, but they refer to the same modeling mechanism. This mechanism was first proposed and widely used in natural language processing [1,33] and physical system modeling [35,20,30]. The self-attention / relation module affects an individual element (e.g. a word in a sentence) by aggregating features from a set of elements (e.g. all the words in the sentence), where the aggregation weights are usually determined on embedded feature similarities among the elements. They are powerful in capturing long-range dependencies and contextual information.

In the computer vision, two pioneering works [21,34] first applied this kind of modeling mechanism to capture the relations between objects and pixels, respectively. Since then, such modeling methods have demonstrated great effectiveness in many vision tasks, such as image classification [22], object detection [21,15], semantic segmentation [39], video object detection [10,36,16,6] and tracking [37], and action recognition [34]. There are also works that propose improvements to self-attention modeling, e.g. an additional relative position term [21,22], an additional channel attention [13], simplification [2], and speed-up [23].

This paper also presents an improvement over the basic self-attention / non-local neural networks. However, our work goes beyond straightforward application or technical modification of non-local networks in that it also brings a new perspective for understanding this module.

Understanding non-local/self-attention mechanisms. Our work is also related to several approaches that analyze the non-local/self-attention mechanism in depth, including the performance of individual terms [21,32,46] on various tasks. Also, there are studies which seek to uncover what is actually learnt by the non-local/self-attention mechanism in different tasks [2].

This work also targets a deeper understanding of the non-local mechanism, in a new perspective. Beyond improved understanding, our paper presents a more effective module, the disentangled non-local block, that is developed from this new understanding and is shown to be effective on multiple vision tasks.

3 Non-local Networks in Depth

3.1 Dividing Non-local Block into Pairwise and Unary Terms

Non-local block [34] computes pairwise relations between features of two positions to capture long-range dependencies. With \mathbf{x}_i representing the input features at position i , the output features \mathbf{y}_i of a non-local block are computed as

$$\mathbf{y}_i = \sum_{j \in \Omega} \omega(\mathbf{x}_i, \mathbf{x}_j) g(\mathbf{x}_j), \quad (1)$$

where Ω denotes the set of all pixels on a feature map of size $H \times W$; $g(\cdot)$ is the value transformation function with parameter W_v ; $\omega(\mathbf{x}_i, \mathbf{x}_j)$ is the embedded

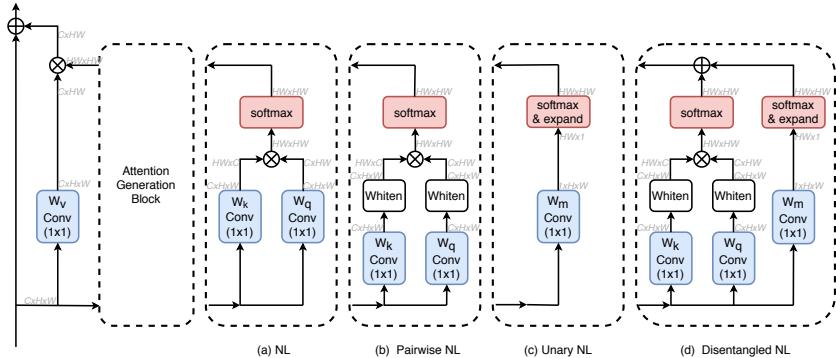


Fig. 2. Architectures of non-local block, disentangled non-local block, and other variants. The shapes of feature maps are indicated in gray, e.g., $C \times H \times W$. “ \otimes ” denotes matrix multiplication and “ \oplus ” denotes element-wise addition. Blue boxes represent 1×1 convolution. *Softmax* is performed on the first dimension of feature maps

similarity function from pixel j (referred to as a *key* pixel) to pixel i (referred to as a *query* pixel), typically instantiated by an Embedded Gaussian as

$$\omega(\mathbf{x}_i, \mathbf{x}_j) = \sigma\left(\mathbf{q}_i^T \mathbf{k}_j\right) = \frac{\exp(\mathbf{q}_i^T \mathbf{k}_j)}{\sum_{t \in \Omega} \exp(\mathbf{q}_i^T \mathbf{k}_t)}, \quad (2)$$

where $\mathbf{q}_i = W_q \mathbf{x}_i$ and $\mathbf{k}_j = W_k \mathbf{x}_j$ denote the *query* and *key* embedding of pixel i and j , respectively, and $\sigma(\cdot)$ denotes the softmax function.

At first glance, $\omega(\mathbf{x}_i, \mathbf{x}_j)$ (defined in Eq. 2) appears to represent only a *pairwise* relationship in the non-local block, through a dot product operation. However, we find that it may encode some *unary* meaning as well. Considering a special case where the query vector is a constant over all image pixels, a *key* pixel will have global impact on all *query* pixels. In [2], it was found that non-local blocks frequently degenerate into a pure *unary* term in several image recognition tasks where each *key* pixel in the image has the same similarity with all *query* pixels. These findings indicate that the *unary* term does exist in the non-local block formulation. It also raises a question of how to divide Eq. (2) into *pairwise* and *unary* terms, which account for the impact of one *key* pixel specifically on another *query* pixel and the influence of one *key* pixel generally over all the *query* pixels, respectively.

To answer this question, we first present a *whitened* dot product between *key* and *query* to represent the *pure pairwise* term: $(\mathbf{q}_i - \boldsymbol{\mu}_q)^T (\mathbf{k}_j - \boldsymbol{\mu}_k)$, where $\boldsymbol{\mu}_q = \frac{1}{|\Omega|} \sum_{i \in \Omega} \mathbf{q}_i$ and $\boldsymbol{\mu}_k = \frac{1}{|\Omega|} \sum_{j \in \Omega} \mathbf{k}_j$ are the averaged *query* and *key* embedding over all pixels, respectively. To remove the *unary/global* component of *key* pixels, the *whitened* dot product is determined by maximizing the normalized differences between *query* and *key* pixels. In following proposition, we show how this can be achieved via an optimization objective, which allows for the whitened dot product to be computed.

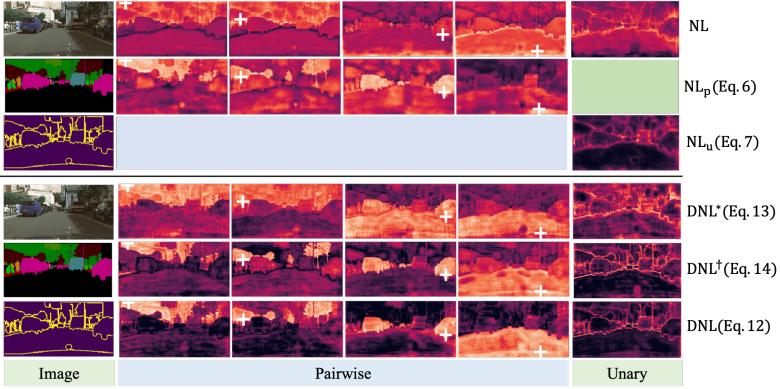


Fig. 3. Visualization of attention maps for all variants of the NL block mentioned in this paper. Column 1: image, ground truth and edges of ground truth. Columns 2–5: attention maps of pairwise terms. Column 6: attention maps of unary terms. As NL_u has no pairwise attention map, and NL_p has no unary attention map, we leave the corresponding spaces empty

Proposition 1: $\alpha^* = \frac{1}{|\Omega|} \sum_{i \in \Omega} \mathbf{q}_i$, $\beta^* = \frac{1}{|\Omega|} \sum_{m \in \Omega} \mathbf{k}_m$ is the optimal solution of the following optimization objective:

$$\begin{aligned} \arg \max_{\alpha, \beta} & \frac{\sum_{i, m, n \in \Omega} ((\mathbf{q}_i - \alpha)^T (\mathbf{k}_m - \beta) - (\mathbf{q}_i - \alpha)^T (\mathbf{k}_n - \beta))^2}{\sum_{i \in \Omega} ((\mathbf{q}_i - \alpha)^T (\mathbf{q}_i - \alpha)) \cdot \sum_{m, n \in \Omega} ((\mathbf{k}_m - \mathbf{k}_n)^T (\mathbf{k}_m - \mathbf{k}_n))} \\ & + \frac{\sum_{m, i, j \in \Omega} ((\mathbf{k}_m - \beta)^T (\mathbf{q}_i - \alpha) - (\mathbf{k}_m - \beta)^T (\mathbf{q}_j - \alpha))^2}{\sum_{m \in \Omega} ((\mathbf{k}_m - \beta)^T (\mathbf{k}_m - \beta)) \cdot \sum_{i, j \in \Omega} ((\mathbf{q}_i - \mathbf{q}_j)^T (\mathbf{q}_i - \mathbf{q}_j))} \end{aligned} \quad (3)$$

Proof sketch: The Hessian of the objective function O with respect to α and β is a non-positive definite matrix. The optimal α^* and β^* are thus the solutions of the following equations: $\frac{\partial O}{\partial \alpha} = 0$, $\frac{\partial O}{\partial \beta} = 0$. Solving this yields $\alpha^* = \frac{1}{|\Omega|} \sum_{i \in \Omega} \mathbf{q}_i$, $\beta^* = \frac{1}{|\Omega|} \sum_{m \in \Omega} \mathbf{k}_m$. Please see the appendix for a detailed proof.

By extracting the whitened dot product as the *pure* pairwise term, we can divide the dot product computation of the standard non-local block as

$$\mathbf{q}_i^T \mathbf{k}_j = (\mathbf{q}_i - \boldsymbol{\mu}_q)^T (\mathbf{k}_j - \boldsymbol{\mu}_k) + \boldsymbol{\mu}_q^T \mathbf{k}_j + \mathbf{q}_i^T \boldsymbol{\mu}_k + \boldsymbol{\mu}_q^T \boldsymbol{\mu}_k. \quad (4)$$

Note that the last two terms ($\mathbf{q}_i^T \boldsymbol{\mu}_k$ and $\boldsymbol{\mu}_q^T \boldsymbol{\mu}_k$) are factors that appear in both the numerator and denominator of Eq. (2). Hence, these two terms can be eliminated (see proof in the Appendix). After this elimination, we reach the following *pairwise* and *unary* split of a standard non-local block:

$$\omega(\mathbf{x}_i, \mathbf{x}_j) = \sigma(\mathbf{q}_i^T \mathbf{k}_j) = \sigma(\underbrace{(\mathbf{q}_i - \boldsymbol{\mu}_q)^T (\mathbf{k}_j - \boldsymbol{\mu}_k)}_{\text{pairwise}} + \underbrace{\boldsymbol{\mu}_q^T \mathbf{k}_j}_{\text{unary}}), \quad (5)$$

where the first *whitened* dot product term represents the *pure* pairwise relation between a *query* pixel i and a *key* pixel j , and the second term represents the *unary* relation where a *key* pixel j has the same impact on all *query* pixels i .

Table 1. Consistency statistics between attention maps of the non-local variants and the ground-truth within-category and boundary maps on the Cityscapes validation set

method	pair \cap within-category	pair \cap boundary	unary \cap boundary
random	0.259	0.132	0.135
pairwise NL (Eq. 6)	0.635	0.141	-
unary NL (Eq. 7)	-	-	0.460
NL (Eq. 2)	0.318	0.160	0.172
DNL* (Eq. 13)	0.446	0.146	0.305
DNL [†] (Eq. 14)	0.679	0.137	0.657
DNL (Eq. 12)	0.759	0.130	0.696

3.2 What Visual Clues are Expected to be Learnt by Pairwise and Unary Terms?

To study what visual clues are expected to be learnt by the pairwise and unary terms, respectively, we construct two variants of the non-local block by using either the pairwise or unary term alone, such that the influence of the other term is eliminated. The two variants use the following similarity computation functions instead of the one in Eq. (2):

$$\omega_p(\mathbf{x}_i, \mathbf{x}_j) = \sigma\left((\mathbf{q}_i - \boldsymbol{\mu}_q)^T (\mathbf{k}_j - \boldsymbol{\mu}_k)\right), \quad (6)$$

$$\omega_u(\mathbf{x}_i, \mathbf{x}_j) = \sigma(\boldsymbol{\mu}_q^T \mathbf{k}_j). \quad (7)$$

The two variants are denoted as “pairwise NL” and “unary NL”, and illustrated in Fig. 2(b) and 2(c), respectively. We apply these two variants of non-local block to the Cityscapes semantic segmentation [8] (see Section 5.1 for detailed settings), and visualize their learnt attention (similarity) maps on several randomly selected validation images in Cityscapes, as shown in Fig. 3 (please see more examples in the Appendix). It can be seen that the pairwise NL block tends to learn pixel relationships within the same category region, while the unary NL block tends to learn the impact from boundary pixels to all image pixels.

This observation is further verified by quantitative analysis using the ground-truth region and boundary annotations in Cityscapes. Denote $P^{(i)} = \{\omega_p(\mathbf{x}_i, \mathbf{x}_j) | j \in \Omega\} \in \mathbb{R}^{H \times W}$ as the attention map of pixel i according to the pairwise term of Eq. (6), $U = \{\omega_u(\mathbf{x}_i, \mathbf{x}_j) | j \in \Omega\} \in \mathbb{R}^{H \times W}$ as the attention map for all query pixels according to the unary term of Eq. (7), $C^{(i)} \in \mathbb{R}^{H \times W}$ as the binary within-category region map of pixel i , and $E \in \mathbb{R}^{H \times W}$ as the binary boundary map indicating pixels with distance to ground truth contour of less than 5 pixels.

We evaluate the consistency between attention maps $A \in \{P^{(i)}, U\}$ and ground-truth boundary/same-category region $G \in \{C^{(i)}, E\}$ by their overlaps:

$$A \cap G = \sum_{j \in \Omega} A_j \odot G_j, \quad (8)$$

where A_j, G_j are the element values of the corresponding attention map and binary map at pixel j , respectively.

Table 1 shows the averaged consistency measures of the attention maps in Eq. (6) and Eq. (7) to ground-truth region maps (denoted as pairwise NL and

unary NL) using all 500 validation images in the Cityscapes datasets. We also report the consistency measures by a random attention map for reference (denoted as random). The following can be seen:

- The attention map by the pairwise NL block of Eq. (6) has significantly larger overlap with the ground-truth same-category region than the random attention map (0.635 vs. 0.259), but has similar overlap with the ground-truth boundary region (0.141 vs. 0.132), indicating that *the pure pairwise term tends to learn relationship between pixels within same-category regions.*
- The attention map by the unary NL block of Eq. (7) has significantly larger overlap with the ground-truth boundary region than the random attention map (0.460 vs. 0.135), indicating that *the unary term tends to learn the impact of boundary pixels on all image pixels.* This is likely because the image boundary area provides the most informative cues when considering the general effect on all pixels.

3.3 Does the Non-local Block Learn Visual Clues Well?

We then study the learnt pairwise and unary terms by the non-local block. We follow Eq. (5) to split the standard similarity computation into the pairwise and unary terms, and normalize them by a softmax operation separately. After splitting and normalization, we can compute their overlaps with the ground-truth within-category region map and boundary region map, as shown in Table 1.

It can be seen that the pairwise term in the standard NL block which is jointly learnt with the unary term has significantly smaller overlap with the ground-truth within-category region than in the pairwise NL block where the pairwise term is learnt alone (0.318 vs. 0.635). It can be also seen that the unary term in the standard NL block which is jointly learnt with the pairwise term has significantly smaller overlap with the ground-truth boundary region than in the unary NL block where the unary term is learnt alone (0.172 vs. 0.460). These results indicate that neither of the pairwise and unary terms learn the visual clues of within-category regions and boundaries well, as also demonstrated in Fig. 1 (top).

3.4 Why the Non-Local Block Does Not Learn Visual Clues Well?

To understand why the non-local block does not learn the two visual clues well, while the two terms alone can clearly learn them, we rewrite Eq. (5) as:

$$\begin{aligned}\sigma(\mathbf{q}_i \cdot \mathbf{k}_j) &= \sigma\left((\mathbf{q}_i - \boldsymbol{\mu}_q)^T (\mathbf{k}_j - \boldsymbol{\mu}_k) + \boldsymbol{\mu}_q^T \mathbf{k}_j\right) \\ &= \frac{1}{\lambda_i} \sigma\left((\mathbf{q}_i - \boldsymbol{\mu}_q)^T (\mathbf{k}_j - \boldsymbol{\mu}_k)\right) \cdot \sigma(\boldsymbol{\mu}_q^T \mathbf{k}_j) = \frac{1}{\lambda_i} \omega_p(\mathbf{x}_i, \mathbf{x}_j) \cdot \omega_u(\mathbf{x}_i, \mathbf{x}_j),\end{aligned}\quad (9)$$

where λ_i is a normalization scalar such that the sum of attention map values over Ω is 1.

Consider the back-propagation of loss L to the pairwise and unary terms:

$$\frac{\partial L}{\partial \sigma(\omega_p)} = \frac{\partial L}{\partial \sigma(\omega)} \cdot \frac{\partial \sigma(\omega)}{\partial \sigma(\omega_p)} = \frac{\partial L}{\partial \sigma(\omega)} \cdot \sigma(\omega_u),$$

$$\frac{\partial L}{\partial \sigma(\omega_u)} = \frac{\partial L}{\partial \sigma(\omega)} \cdot \frac{\partial \sigma(\omega)}{\partial \sigma(\omega_u)} = \frac{\partial L}{\partial \sigma(\omega)} \cdot \sigma(\omega_p).$$

It can be seen that both gradients are determined by the value of the other term. When the value of the other term becomes very small (close to 0), the gradient of this term will be also very small, thus inhibiting the learning of this term. For example, if we learn the unary term to well represent the boundary area, the unary attention weights on the non-boundary area will be close to 0 and the pairwise term at the non-boundary area would thus be hard to learn well due to the vanishing gradient issue. On the other hand, if we learn the pairwise term to well represent the within-category area, the unary attention weights on the boundary area will be close to 0 and the pairwise term at the non-boundary area would also be hard to learn well due to the same vanishing gradient issue.

Another problem is the *shared* key transformation W_k used in both the pairwise and unary terms, causing the computation of the two terms to be coupled. Such coupling may introduce additional difficulties in learning the two terms.

4 Disentangled Non-local Neural Networks

In this section, we present a new non-local block, named disentangled non-local (DNL) block, which effectively disentangles the learning of pairwise and unary terms. In the following sections, we first describe how we modify the standard non-local (NL) block into a disentangled non-local (NL) block, such that the two visual clues described above can be learnt well. Then we analyze its actual behavior in learning visual clues using the method in Section 3.2.

4.1 Formulation

Our first modification is to change the *multiplication* in Eq. (9) to *addition*:

$$\omega(\mathbf{x}_i, \mathbf{x}_j) = \omega_p(\mathbf{x}_i, \mathbf{x}_j) \cdot \omega_u(\mathbf{x}_i, \mathbf{x}_j) \Rightarrow \omega(\mathbf{x}_i, \mathbf{x}_j) = \omega_p(\mathbf{x}_i, \mathbf{x}_j) + \omega_u(\mathbf{x}_i, \mathbf{x}_j). \quad (10)$$

The gradients of these two terms are

$$\frac{\partial L}{\partial \sigma(\omega_p)} = \frac{\partial L}{\partial \sigma(\omega)}, \frac{\partial L}{\partial \sigma(\omega_u)} = \frac{\partial L}{\partial \sigma(\omega)}.$$

So the gradients of each term will not be impacted by the other.

The second modification is to change the transformation W_k in unary term to be an independent linear transformation W_m with output dimension of 1:

$$\boldsymbol{\mu}_q^T \mathbf{k}_j = \boldsymbol{\mu}_q^T W_k \mathbf{x}_j \Rightarrow m_j = W_m \mathbf{x}_j. \quad (11)$$

After this modification, the pairwise and unary terms will no longer share the W_k transformation, which further decouples them.

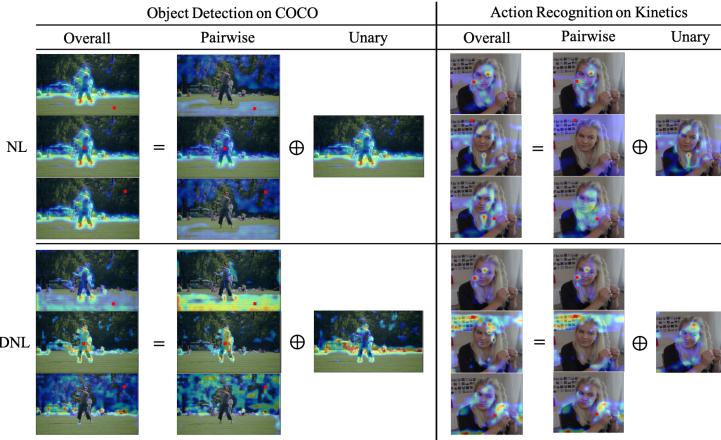


Fig. 4. Visualization of attention maps in NL and our DNL block on COCO object detection and Kinetics action recognition. The query points are marked in red. Please refer to appendix for more examples

DNL formulation. With these two modifications, we obtain the following similarity computation for the disentangled non-local (DNL) block:

$$\omega^D(\mathbf{x}_i, \mathbf{x}_j) = \sigma\left((\mathbf{q}_i - \boldsymbol{\mu}_q)^T (\mathbf{k}_j - \boldsymbol{\mu}_k)\right) + \sigma(m_j). \quad (12)$$

The resulting DNL block is illustrated in Fig. 2 (d). Note that we adopt a single *value* transform for both pairwise and unary terms, which is similarly effective on benchmarks as using independent value transform but with reduced complexity.

Complexity. For an input feature map of $C \times H \times W$, we follow [34] by using $C/2$ dimensional *key* and *query* vectors. The space and time complexities are $\mathcal{O}^D(\text{space}) = (2C+1)C$ and $\mathcal{O}^D(\text{time}) = ((2C+1)C + (\frac{3}{2}C+2)HW)HW$, respectively. For reference, the space and time complexity of a standard non-local block are $\mathcal{O}(\text{space}) = 2C^2$ and $\mathcal{O}(\text{time}) = (2C^2 + (\frac{3}{2}C+1)HW)HW$, respectively. The additional space and computational overhead of the disentangled non-local block over a standard non-local block is marginal, specifically 0.1% and 0.15% for $C = 512$ in our semantic segmentation experiments.

DNL variants for diagnostic purposes. To diagnose the effects of the two decoupling modifications alone, we consider the following two variants:

$$\omega^{D*}(\mathbf{x}_i, \mathbf{x}_j) = \sigma\left((\mathbf{q}_i - \boldsymbol{\mu}_q)^T (\mathbf{k}_j - \boldsymbol{\mu}_k) + m_j\right), \quad (13)$$

$$\omega^{D\dagger}(\mathbf{x}_i, \mathbf{x}_j) = \sigma\left((\mathbf{q}_i - \boldsymbol{\mu}_q)^T (\mathbf{k}_j - \boldsymbol{\mu}_k)\right) + \sigma(\boldsymbol{\mu}_q^T \mathbf{k}_j), \quad (14)$$

which each involves only one of the two modifications.

4.2 Behavior of DNL on Learning Visual Clues

We compute the overlaps of the pairwise and unary attention maps in DNL (Eq. 12) with the ground-truth within-category region map and boundary region map, as shown in Table 1.

It can be seen that the pairwise term in DNL has significantly larger overlap with the ground-truth within-category region than the one in the standard NL block (0.759 vs. 0.318), and the unary term has significantly larger overlap with the boundary region than that in the standard NL block (0.696 vs. 0.172). These results indicate better learning of the two visual clues by the DNL block in comparison to the standard NL block.

Compared with the blocks which learn the pairwise or unary terms alone (see the “pairwise NL” and “unary NL” rows), such measures are surprisingly 0.124 and 0.236 higher with DNL. We hypothesize that when one term is learned alone, it may encode some portion of the other clue, as it is also useful for inference. By explicitly learning both terms, our disentangling design can separate one from the other, allowing it to better extract these visual clues.

We then verify the effects of each disentangling modification by these measures. By incorporating the “disentangled transformation” modification alone (ω^*) as in Eq. (13), it achieves 0.446 and 0.305 on within-category modeling and boundary modeling, respectively, which is marginally better than the standard non-local block. By incorporating the “multiplication to addition” modification alone (ω^\dagger) as in Eq. (14), it achieves 0.679 and 0.657 on within-category modeling and boundary modeling, respectively.

The results indicate that the two modifications both benefit the learning of two visual clues and work better if combined together. The improvements in visual clue modeling by two disentangling strategies are also illustrated in Fig. 3.

Note such disentangling strategies also effect on other tasks beyond semantic segmentation. In object detection and action recognition tasks, we also observe clearer learnt visual clues by the DNL block than by the standard NL. As shown in Fig. 4, while in NL the pairwise term is almost hindered by the unary term (also observed by [2]), the pairwise term in DNL shows clear within-region meaning and appears significant in the final overall attention maps. The unary term in DNL also shows more focus to salient regions (not limited to boundaries which is different from that observed in the semantic segmentation task) than the one in an NL block. More examples will be shown in appendix.

5 Experiments

We evaluate the proposed DNL method on the recognition tasks of semantic segmentation, object detection/instance segmentation, and action recognition.

5.1 Semantic Segmentation

Datasets. We use three benchmarks for semantic segmentation evaluation.

Cityscapes [8] focuses on semantic understanding of urban street scenes. It provides a total of 5,000 finely annotated images, which is divided into 2,975/500/1,525 images for training, validation and testing. Additional 20,000 coarsely annotated images are also provided. The dataset contains annotations for over 30 classes, of which 19 classes are used in evaluation.

Table 2. Ablation study on the validation set of Cityscapes

(a) Decoupling strategy				(b) Pairwise and unary terms			
	mul → add	non-shared W_k	mIoU		pairwise term	unary term	mIoU
Baseline	-	-	75.8	Baseline	-	-	75.8
NL	×	×	78.5	NL	✓	✓	78.5
DNL [†] (14)	✓	×	79.2	NL _p	✓	✗	77.5
DNL*(13)	×	✓	79.0	NL _u	✗	✓	79.3
DNL	✓	✓	80.5	DNL	✓	✓	80.5

ADE20K [45] was used in the ImageNet Scene Parsing Challenge 2016 and covers a wide range of scenes and object categories. It contains 20K images for training, 2K images for validation, and another 3K images for testing. It includes 150 semantic categories for evaluation.

PASCAL-Context [28] is a set of additional annotations for PASCAL VOC 2010, which label more than 400 categories of 4,998 images for training and 5,105 images for validation. For semantic segmentation, 59 semantic classes and 1 background class are used in training and validation.

Architecture. We follow recent practice [23] by using dilated FCN [27] and a ResNet101 [19] backbone for our major segmentation experiments. The strides and dilations of 3×3 convolutions are set to 1 and 2 for stage4, and 1 and 4 for stage5. The baseline model uses a segmentation head consisting of a 3×3 convolution layer to reduce the channels to 512 and a subsequent classifier to predict the final segmentation results. For experiments with a non-local or a disentangled non-local block, the block is inserted right before the final classifier.

Training and Inference. The implementation and hyper-parameters mostly follow [23]. The SGD optimizer with poly learning rate policy $(1 - (\frac{iter}{iter_{max}})^0.9)$ is employed. For Cityscapes, the networks are trained on 4 GPUs with 2 images per GPU for 60K iterations. The initial learning rate is 0.01, the weight decay is 0.0005. Input images are cropped to 769×769 . For ADE20K, the networks are trained on 8 GPUs with 2 images per GPU for 150K iterations. The initial learning rate is 0.02, and the weight decay is 0.0001. Input images are cropped to 520×520 . For PASCAL-Context, the network is trained on 4 GPUs with 4 images per GPU for 30K iterations. The initial learning rate is 0.001, and the weight decay is 0.0001. Input images are cropped to 520×520 . For all datasets, the data is augmented with random horizontal flipping, random scaling within $[0.5, 2.0]$, and random brightness jittering of $[-10, 10]$. Following [39], online hard example mining (OHEM) and an auxiliary loss on the output of conv4 with a weight of 0.5 are employed for Cityscapes and ADE20K, only auxiliary loss is employed for PASCAL-Context.

We mostly follow [23] in inference. For Cityscapes, we sample 769×769 windows for inference and their results are fused to generate the prediction of an entire image. For other datasets, we resize the image resolution to be the same as in training and a multi-scale test is adopted.

Table 3. Comparisons with state-of-the-art approaches on the Cityscapes test set

Method	Backbone	ASPP	Coarse	mIoU (%)
PSANet [44]	ResNet-101			80.1
DANet [13]	ResNet-101			81.5
HRNet [31]	HRNetV2-W48			81.9
SeENet [29]	ResNet-101			81.2
SPGNet [7]	ResNet-101			81.1
CCNet [23]	ResNet-101			81.4
ANN [47]	ResNet-101			81.3
DenseASPP [38]	DenseNet-161	✓		80.6
OCNet [39]	ResNet-101	✓		81.7
ACFNet [40]	ResNet-101	✓		81.8
PSPNet [43]	ResNet-101		✓	81.2
PSANet [44]	ResNet-101		✓	81.4
DeepLabv3 [5]	ResNet-101	✓	✓	81.3
NL	ResNet-101		✓	80.8
DNL (ours)	ResNet-101		✓	82.0
NL	HRNetV2-W48		✓	82.5
DNL (ours)	HRNetV2-W48		✓	83.0

Ablation Study We ablate several design components in the proposed disentangled non-local block on the Cityscapes validation set. A ResNet-101 backbone is adopted for all ablations.

DNL variants. The disentangled non-local block has two decoupling modifications on the standard non-local block: multiplication to addition, and separate *key* transformations. In addition to comparing the full DNL model with the standard non-local model, we also conduct experiments for these two variants which include only one of the decoupling modifications.

The results are shown in Table 2(a). While the standard non-local model brings 2.7% mIoU gains over a plain ResNet-101 model (78.5% vs. 75.8%), by replacing the standard non-local block by our disentangled non-local block, we achieve an additional 2.0% mIoU gain over the standard non-local block (80.5% vs. 78.5%), with almost no complexity increase.

The variants that use each decoupling strategy alone achieve 0.5% and 0.7% mIoU gains over the standard non-local block (79.0 vs. 78.5 and 79.2 vs. 78.5), showing that both strategies are beneficial alone. They are also both crucial, as combining them leads to significantly better performance than using each alone.

Effects of pairwise and unary term alone. Table 2(b) compares the methods using the pairwise term or unary term alone. Using the pairwise term alone achieves 77.5% mIoU, which is 1.7% better than the baseline plain network without it. Using the unary term alone achieves 79.3% mIoU, which is 3.5% better than the baseline plain network and even 0.8% mIoU better than the standard non-local network which models both pairwise and unary terms. These results indicate that the standard non-local block hinders the effect of the unary term, probably due to the coupling of two kinds of relationships. Our disentangled non-local networks effectively disentangle the two terms, and thus can better exploit their effects to achieve a higher accuracy of 80.5% mIoU.

Complexities. As discussed in Section 4.1, the time and space complexity of the DNL model over the NL model is tiny. Table 5 show the FLOPs and actual

Table 4. Comparisons with state-of-the-art approaches on the validation set and test set of ADE20K, and test set of PASCAL-Context

Method	Backbone	ADE20K		PASCAL-Context test mIoU (%)
		val mIoU (%)	test mIoU (%)	
PSANet [44]	ResNet-101	43.77	55.46	-
CCNet [23]	ResNet-101	45.22	-	-
OCNet [39]	ResNet-101	45.45	-	-
SVCNet [11]	ResNet-101	-	-	53.2
EMANet [25]	ResNet-101	-	-	53.1
HRNetV2 [31]	HRNetV2-W48	42.99	-	54.0
EncNet [41]	ResNet-101	44.65	55.67	52.6
DANet [13]	ResNet-101	45.22	-	52.6
CFNet [42]	ResNet-101	44.89	-	54.0
ANN [47]	ResNet-101	45.24	-	52.8
DMNet [17]	ResNet-101	45.50	-	54.4
ACNet [14]	ResNet-101	45.90	55.84	54.1
NL	ResNet-101	44.67	55.58	50.6
DNL (ours)	ResNet-101	45.97	56.23	54.8
NL	HRNetV2-W48	44.82	55.60	54.2
DNL (ours)	HRNetV2-W48	45.82	55.98	55.3

latency (single-scale inference using a single GPU) on semantic segmentation, using a ResNet-101 backbone and input resolution of 769×769 .

Comparison with other methods.

Results on Cityscapes. Table 3 shows comparison results for the proposed disentangled non-local network on the Cityscapes test set. Using a ResNet-101 backbone, the disentangled non-local network achieves 82.0% mIoU, 1.2% better than that of a standard non-local network. On a stronger backbone of HRNetV2-W48, the disentangled non-local network achieves 0.5% better accuracy than a standard non-local network. Considering that the standard non-local network has 0.6% mIoU improvement over a plain HRNetV2-W48 network, such additional gains are significant.

Results on ADE20K. Table 4 shows comparison results of the proposed disentangled non-local network on the ADE20k benchmark. Using a ResNet-101 backbone, the disentangled non-local block achieves 45.97% and 56.23% on the validation and test sets, respectively, which are 1.30% and 0.65% better than the counterpart networks using a standard non-local block. Our result reveals a new SOTA on this benchmark. On a HRNetV2-W48 backbone, the DNL block is 1.0% and 0.38% better than a standard non-local block. Note on ADE20K, HRNetV2-W48 backbone does not perform better than a ResNet-101 backbone, which is different with the other datasets.

Table 5. Complexity comparisons

	#param(M)	FLOPs(G)	latency(s/img)
baseline	70.960	691.06	0.177
NL	71.484	765.07	0.192
DNL	71.485	765.16	0.194

Table 6. Results based on Mask R-CNN, using R50 as backbone with FPN, for object detection and instance segmentation on COCO 2017 validation set

	AP ^{bbox}	AP ₅₀ ^{bbox}	AP ₇₅ ^{bbox}	AP ^{mask}	AP ₅₀ ^{mask}	AP ₇₅ ^{mask}
baseline	38.8	59.3	42.5	35.1	56.2	37.9
NL	39.6	60.3	43.2	35.8	57.1	38.5
NL _p	39.8	60.4	43.7	35.9	57.3	38.4
NL _u	40.1	60.9	43.8	36.1	57.6	38.7
DNL	40.3	61.2	44.1	36.4	58.0	39.1

Table 7. Results based on Slow-only baseline using R50 as backbone on Kinetics validation set

	Top-1 Acc	Top-5 Acc
baseline	74.94	91.90
NL	75.95	92.29
NL _p	76.01	92.28
NL _u	75.76	92.44
DNL	76.31	92.69

Results on PASCAL-Context. Table 3 shows comparison results of the proposed disentangled non-local network on the PASCAL-Context test set. On ResNet-101, our method improves the standard non-local method significantly, by 3.4% mIoU (53.7 vs. 50.3). On HRNetV2-W48, our DNL method is 1.1% mIoU better, which is significant considering that the NL method has 0.2% improvements over the plain counterpart.

5.2 Object Detection/Segmentation and Action Recognition

Object Detection and Instance Segmentation on COCO. We adopt the open source mmdetection [4] codebase for experiments. Following [34], the non-local variants are inserted right before the last residual block of c4.

We use the standard configuration of Mask R-CNN [18] with FPN and ResNet-50 as the backbone architecture, and report the mean average-precision scores at different boxes and the mask IoUs on the COCO2017 validation set. The input images are resized such that their shorter side is 800 pixels [26]. We trained on 4 GPUs with 4 images per GPU (effective mini batch size of 16). The backbones of all models are pretrained on ImageNet classification [9], then all layers except for stage1 and stage2 are jointly fine-tuned. In training, synchronized BatchNorm is adopted, and the learning rate scheduler follows the 1× settings of 12 epochs in [18] where the initial learning rate is 0.02 and decayed by a factor of 10 at the 8th and 11th epoch. The weight decay is 0.0001 and momentum is 0.9.

Table 6 shows comparisons of different methods. While the standard non-local block outperforms the baseline counterpart by 0.8% bbox mAP and 0.7% mask mAP, the proposed disentangled non-local block brings an additional 0.7% bbox mAP and 0.6% mask mAP in gains. Please also see Appendix for experiments when stacking more non-local or disentangled non-local blocks.

Action Recognition on Kinetics. We adopt the Kinetics [24] dataset for experiments, which includes ∼240k training videos and 20k validation videos in 400 human action categories. All models are trained on the training set, and we report the top-1 (%) and top-5 (%) accuracy on the validation set. We adopt the slow-only baseline in [12], the best single model to date that can utilize weights inflated [3] from the ImageNet pretrained model. All the experiment settings follow the slow-only baseline in [12], where 8 frames (8×8) are used as

input, and 30-clip validation is adopted. Following [34], we insert (disentangled) non-local blocks after every two residual blocks.

Table 7 shows the comparison of different blocks. It can be seen that the disentangled design performs 0.36% better than using standard non-local block. Noting the gains of the standard non-local block over the baseline is 1.0, the relative gains of disentangled non-local block over a standard NL block is 36%.

6 Conclusion

In this paper, we first study the non-local block in depth, where we find that its attention computation can be split into two terms, a whitened pairwise term and a unary term. Via both intuitive and statistical analysis, we find that the two terms are tightly coupled in the non-local block, which hinders the learning of each. Based on these findings, we present the disentangled non-local block, where the two terms are decoupled to facilitate learning for both terms. We demonstrate the effectiveness of the decoupled design for learning visual clues on various vision tasks, such as semantic segmentation, object detection and action recognition.

References

1. Britz, D., Goldie, A., Luong, M.T., Le, Q.: Massive exploration of neural machine translation architectures. arXiv preprint arXiv:1703.03906 (2017)
2. Cao, Y., Xu, J., Lin, S., Wei, F., Hu, H.: Gcnet: Non-local networks meet squeeze-excitation networks and beyond. arXiv preprint arXiv:1904.11492 (2019)
3. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6299–6308 (2017)
4. Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., et al.: Mmdetection: Open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155 (2019)
5. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
6. Chen, Y., Cao, Y., Hu, H., Wang, L.: Memory enhanced global-local aggregation for video object detection. In: The Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
7. Cheng, B., Chen, L.C., Wei, Y., Zhu, Y., Huang, Z., Xiong, J., Huang, T.S., Hwu, W.M., Shi, H.: Spgnnet: Semantic prediction guidance for scene parsing. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5218–5228 (2019)
8. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3223 (2016)
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)

10. Deng, J., Pan, Y., Yao, T., Zhou, W., Li, H., Mei, T.: Relation distillation networks for video object detection. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019)
11. Ding, H., Jiang, X., Shuai, B., Liu, A.Q., Wang, G.: Semantic correlation promoted shape-variant context for segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8885–8894 (2019)
12. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. arXiv preprint arXiv:1812.03982 (2018)
13. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3146–3154 (2019)
14. Fu, J., Liu, J., Wang, Y., Li, Y., Bao, Y., Tang, J., Lu, H.: Adaptive context network for scene parsing. In: Proceedings of the IEEE international conference on computer vision. pp. 6748–6757 (2019)
15. Gu, J., Hu, H., Wang, L., Wei, Y., Dai, J.: Learning region features for object detection. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 381–395 (2018)
16. Guo, C., Fan, B., Gu, J., Zhang, Q., Xiang, S., Prinet, V., Pan, C.: Progressive sparse local attention for video object detection. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019)
17. He, J., Deng, Z., Qiao, Y.: Dynamic multi-scale filters for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3562–3572 (2019)
18. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
19. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
20. Hoshen, Y.: Vain: Attentional multi-agent predictive modeling. In: Advances in Neural Information Processing Systems. pp. 2701–2711 (2017)
21. Hu, H., Gu, J., Zhang, Z., Dai, J., Wei, Y.: Relation networks for object detection (2017)
22. Hu, H., Zhang, Z., Xie, Z., Lin, S.: Local relation networks for image recognition (2019)
23. Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W.: Ccnet: Criss-cross attention for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 603–612 (2019)
24. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
25. Li, X., Zhong, Z., Wu, J., Yang, Y., Lin, Z., Liu, H.: Expectation-maximization attention networks for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 9167–9176 (2019)
26. Lin, T.Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
27. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015)

28. Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.: The role of context for object detection and semantic segmentation in the wild. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
29. Pang, Y., Li, Y., Shen, J., Shao, L.: Towards bridging semantic gap to improve semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4230–4239 (2019)
30. Santoro, A., Raposo, D., Barrett, D.G., Malinowski, M., Pascanu, R., Battaglia, P., Lillicrap, T.: A simple neural network module for relational reasoning. In: Advances in neural information processing systems. pp. 4967–4976 (2017)
31. Sun, K., Zhao, Y., Jiang, B., Cheng, T., Xiao, B., Liu, D., Mu, Y., Wang, X., Liu, W., Wang, J.: High-resolution representations for labeling pixels and regions. arXiv preprint arXiv:1904.04514 (2019)
32. Tang, G., Sennrich, R., Nivre, J.: An analysis of attention mechanisms: The case of word sense disambiguation in neural machine translation. arXiv preprint arXiv:1810.07595 (2018)
33. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
34. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7794–7803 (2018)
35. Watters, N., Zoran, D., Weber, T., Battaglia, P., Pascanu, R., Tacchetti, A.: Visual interaction networks: Learning a physics simulator from video. In: Advances in neural information processing systems. pp. 4539–4547 (2017)
36. Wu, H., Chen, Y., Wang, N., Zhang, Z.: Sequence level semantics aggregation for video object detection. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019)
37. Xu, J., Cao, Y., Zhang, Z., Hu, H.: Spatial-temporal relation networks for multi-object tracking. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019)
38. Yang, M., Yu, K., Zhang, C., Li, Z., Yang, K.: Denseaspp for semantic segmentation in street scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3684–3692 (2018)
39. Yuan, Y., Wang, J.: Ocnet: Object context network for scene parsing. arXiv preprint arXiv:1809.00916 (2018)
40. Zhang, F., Chen, Y., Li, Z., Hong, Z., Liu, J., Ma, F., Han, J., Ding, E.: Acfnet: Attentional class feature network for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6798–6807 (2019)
41. Zhang, H., Dana, K., Shi, J., Zhang, Z., Wang, X., Tyagi, A., Agrawal, A.: Context encoding for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7151–7160 (2018)
42. Zhang, H., Zhang, H., Wang, C., Xie, J.: Co-occurrent features in semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 548–557 (2019)
43. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2881–2890 (2017)
44. Zhao, H., Zhang, Y., Liu, S., Shi, J., Change Loy, C., Lin, D., Jia, J.: Psanet: Pointwise spatial attention network for scene parsing. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 267–283 (2018)

45. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 633–641 (2017)
46. Zhu, X., Cheng, D., Zhang, Z., Lin, S., Dai, J.: An empirical study of spatial attention mechanisms in deep networks. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019)
47. Zhu, Z., Xu, M., Bai, S., Huang, T., Bai, X.: Asymmetric non-local neural networks for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 593–602 (2019)

Table 8. Results with more NL and DNL blocks based on Mask R-CNN, using R50 as backbone with FPN, for object detection and instance segmentation on COCO 2017 validation set

	AP ^{bbox}	AP ₅₀ ^{bbox}	AP ₇₅ ^{bbox}	AP ^{mask}	AP ₅₀ ^{mask}	AP ₇₅ ^{mask}
baseline	38.8	59.3	42.5	35.1	56.2	37.9
NL (c4 one)	39.6	60.3	43.2	35.8	57.1	38.5
NL (c5 all)	40.0	62.1	43.5	36.1	58.6	38.6
NL (c4c5 all)	40.1	62.3	43.5	36.0	58.9	38.3
DNL (c4 one)	40.3	61.2	44.1	36.4	58.0	39.1
DNL (c5 all)	41.2	62.7	44.7	37.0	59.5	39.5
DNL (c4c5 all)	41.4	63.2	45.3	37.3	59.8	39.8

A More NL/DNL blocks for COCO Object Detection

In section 5.2 of the main paper, we follow the settings in [34] where 1 non-local (NL) or disentangled non-local (DNL) block is inserted right before the last residual block of c4. In this section, we investigate the performance of NL and DNL when more attention blocks are inserted into the backbone, as shown in Table 8.

While the proposed DNL method outperforms NL method by 0.7% bbox mAP and 0.6% mask mAP when 1 attention block is inserted into the backbone (denoted as “c4 one”), the gains brought by the proposed DNL method over the NL method are enlarged to 1.2% bbox mAP and 0.9% mask mAP, respectively, when every residual block of stage c5 is followed by 1 attention block (denoted as “c4 all”). The gains are further enlarged to 1.3% bbox mAP and 1.3% mask mAP when additionally every residual block of stage c4 is followed by 1 attention block (denoted as “c4 c5 all”). These results indicate that the DNL method can benefit more from increasing block number than the NL method.

B Detailed Proof of Proposition 1

The object function $O(\alpha, \beta)$ in Eq. (3) of the main paper can be rewritten as

$$O(\alpha, \beta) = \frac{\sum_{i \in \Omega} (\mathbf{q}_i - \alpha)^T A (\mathbf{q}_i - \alpha)}{\sum_{i \in \Omega} ((\mathbf{q}_i - \alpha)^T (\mathbf{q}_i - \alpha))} + \frac{\sum_{m \in \Omega} (\mathbf{k}_m - \beta)^T B (\mathbf{k}_m - \beta)}{\sum_{m \in \Omega} ((\mathbf{k}_m - \beta)^T (\mathbf{k}_m - \beta))} \quad (15)$$

where

$$A = \frac{\sum_{m,n \in \Omega} (\mathbf{k}_m - \mathbf{k}_n)(\mathbf{k}_m - \mathbf{k}_n)^T}{\sum_{m,n \in \Omega} (\mathbf{k}_m - \mathbf{k}_n)^T (\mathbf{k}_m - \mathbf{k}_n)} \quad B = \frac{\sum_{i,j \in \Omega} (\mathbf{q}_i - \mathbf{q}_j)(\mathbf{q}_i - \mathbf{q}_j)^T}{\sum_{i,j \in \Omega} (\mathbf{q}_i - \mathbf{q}_j)^T (\mathbf{q}_i - \mathbf{q}_j)} \quad (16)$$

We first prove that all eigenvalues of matrix A and B are smaller or equal than 1. Denote the eigenvalues of matrix A as $\lambda_1, \dots, \lambda_d$. According to CauchySchwarz

inequality, we have

$$\begin{aligned}
\sum_{1 \leq i \leq d} \lambda_i^2 &= \text{tr}(A^T A) \\
&= \text{tr} \left(\frac{\sum_{m,n \in \Omega} (\mathbf{k}_m - \mathbf{k}_n)(\mathbf{k}_m - \mathbf{k}_n)^T \cdot \sum_{s,t \in \Omega} (\mathbf{k}_s - \mathbf{k}_t)(\mathbf{k}_s - \mathbf{k}_t)^T}{\sum_{m,n \in \Omega} (\mathbf{k}_m - \mathbf{k}_n)^T(\mathbf{k}_m - \mathbf{k}_n) \cdot \sum_{s,t \in \Omega} (\mathbf{k}_s - \mathbf{k}_t^T)(\mathbf{k}_s - \mathbf{k}_t)} \right) \\
&= \frac{\sum_{m,n,s,t \in \Omega} (\mathbf{k}_m - \mathbf{k}_n)^T(\mathbf{k}_s - \mathbf{k}_t) \cdot \text{tr}((\mathbf{k}_m - \mathbf{k}_n)(\mathbf{k}_s - \mathbf{k}_t)^T)}{\left(\sum_{m,n \in \Omega} (\mathbf{k}_m - \mathbf{k}_n)^T(\mathbf{k}_m - \mathbf{k}_n) \right)^2} \\
&= \frac{\sum_{m,n,s,t \in \Omega} ((\mathbf{k}_m - \mathbf{k}_n)^T(\mathbf{k}_s - \mathbf{k}_t))^2}{\left(\sum_{m,n \in \Omega} (\mathbf{k}_m - \mathbf{k}_n)^T(\mathbf{k}_m - \mathbf{k}_n) \right)^2} \leq 1
\end{aligned} \tag{17}$$

Given Eq. (17), we have: $\forall 1 \leq i \leq d$, $\lambda_i \leq 1$. Similarly, we can prove all eigenvalues of matrix B are smaller or equal than 1. The hessian matrix of Eq. (1) with respect to α and β are non-positive definite matrix. The optimal α^* and β^* are thus the solutions of the following equations: $\frac{\partial O}{\partial \alpha} = 0$, $\frac{\partial O}{\partial \beta} = 0$.

For α^* , we have

$$\begin{aligned}
\frac{\partial O}{\partial \alpha^*} &= \sum_{i=1}^{N_p} 2 \left(\frac{\sum_{m,n} (\mathbf{k}_m - \mathbf{k}_n)(\mathbf{k}_m - \mathbf{k}_n)^T}{\sum_{m,n} (\mathbf{k}_m - \mathbf{k}_n)^T(\mathbf{k}_m - \mathbf{k}_n)} - 1 \right) (\mathbf{q}_i - \alpha^*) = 0, \\
\Leftrightarrow & \left(\frac{\sum_{m,n} (\mathbf{k}_m - \mathbf{k}_n)(\mathbf{k}_m - \mathbf{k}_n)^T}{\sum_{m,n} (\mathbf{k}_m - \mathbf{k}_n)^T(\mathbf{k}_m - \mathbf{k}_n)} - 1 \right) \sum_{i=1}^{N_p} 2(\mathbf{q}_i - \alpha^*) = 0.
\end{aligned} \tag{18}$$

To satisfy Eqn. 18, we have:

$$\sum_{i=1}^{N_p} (\mathbf{q}_i - \alpha^*) = 0. \tag{19}$$

The optimal α^* is thus

$$\alpha^* = \frac{1}{N_p} \sum_{i=1}^{N_p} \mathbf{q}_i. \tag{20}$$

Similarly, the optimal β^* is computed as

$$\beta^* = \frac{1}{N_p} \sum_{i=1}^{N_p} \mathbf{k}_i. \tag{21}$$

C Proof for Eqn. 4 in the main paper

Here, we provide a proof for Eqn. 4 in Section 3.1. The dot product of query \mathbf{q}_i and key \mathbf{k}_j can be split into several terms by introducing a whitening operation on the key and query:

$$\mathbf{q}_i^T \mathbf{k}_j = (\mathbf{q}_i - \boldsymbol{\mu}_q)^T (\mathbf{k}_j - \boldsymbol{\mu}_k) + \boldsymbol{\mu}_q^T \mathbf{k}_j + \mathbf{q}_i^T \boldsymbol{\mu}_k + \boldsymbol{\mu}_q^T \boldsymbol{\mu}_k, \tag{22}$$

where $\boldsymbol{\mu}_q$ and $\boldsymbol{\mu}_k$ denote $\frac{1}{N_p} \sum_{i=1}^{N_p} \mathbf{q}_i$ and $\frac{1}{N_p} \sum_{i=1}^{N_p} \mathbf{k}_j$, respectively.

Note that the last two terms ($\mathbf{q}_i^T \boldsymbol{\mu}_k$ and $\boldsymbol{\mu}_q^T \boldsymbol{\mu}_k$) are factors in common with both the numerator and denominator of the correlation function f and the normalization factor \mathcal{C} , so they can be eliminated as follows:

$$\begin{aligned}
& \frac{\exp(\mathbf{q}_i^T \mathbf{k}_j)}{\sum_{t=1}^{N_p} \exp(\mathbf{q}_t^T \mathbf{k}_t)} \\
&= \frac{\exp((\mathbf{q}_i - \boldsymbol{\mu}_q)^T (\mathbf{k}_j - \boldsymbol{\mu}_k) + \boldsymbol{\mu}_q^T \mathbf{k}_j + \mathbf{q}_i^T \boldsymbol{\mu}_k + \boldsymbol{\mu}_q^T \boldsymbol{\mu}_k)}{\sum_{t=1}^{N_p} \exp((\mathbf{q}_i - \boldsymbol{\mu}_q)^T (\mathbf{k}_t - \boldsymbol{\mu}_k) + \boldsymbol{\mu}_q^T \mathbf{k}_t + \mathbf{q}_i^T \boldsymbol{\mu}_k + \boldsymbol{\mu}_q^T \boldsymbol{\mu}_k)} \\
&= \frac{\exp((\mathbf{q}_i - \boldsymbol{\mu}_q)^T (\mathbf{k}_j - \boldsymbol{\mu}_k) + \boldsymbol{\mu}_q^T \mathbf{k}_j) \exp(\mathbf{q}_i^T \boldsymbol{\mu}_k + \boldsymbol{\mu}_q^T \boldsymbol{\mu}_k)}{\sum_{t=1}^{N_p} \exp((\mathbf{q}_i - \boldsymbol{\mu}_q)^T (\mathbf{k}_t - \boldsymbol{\mu}_k) + \boldsymbol{\mu}_q^T \mathbf{k}_t) \exp(\mathbf{q}_i^T \boldsymbol{\mu}_k + \boldsymbol{\mu}_q^T \boldsymbol{\mu}_k)} \\
&= \frac{\exp((\mathbf{q}_i - \boldsymbol{\mu}_q)^T (\mathbf{k}_j - \boldsymbol{\mu}_k) + \boldsymbol{\mu}_q^T \mathbf{k}_j)}{\sum_{t=1}^{N_p} \exp((\mathbf{q}_i - \boldsymbol{\mu}_q)^T (\mathbf{k}_t - \boldsymbol{\mu}_k) + \boldsymbol{\mu}_q^T \mathbf{k}_t)}. \tag{23}
\end{aligned}$$

Finally, we obtain

$$\sigma(\mathbf{q}_i^T \mathbf{k}_j) = \sigma(\underbrace{(\mathbf{q}_i - \boldsymbol{\mu}_q)^T (\mathbf{k}_j - \boldsymbol{\mu}_k)}_{pairwise} + \underbrace{\boldsymbol{\mu}_q^T \mathbf{k}_j}_{unary}). \tag{24}$$

D More Examples of Learnt Attention Maps by NL/DNL Methods

In this section, we show more examples of the learnt attention maps by the NL/DNL methods on the Cityscapes semantic segmentation, COCO object detection/instance segmentation and Kinetics action recognition tasks.

Fig. 5 show more examples of the learnt attention maps by NL/DNL on Cityscapes. With DNL block, the whitened pairwise term learns clear within-region clues while the unary term learns salient boundaries, which cannot be observed in that of the original NL block.

Fig. 6 show more examples of the learnt attention maps by NL/DNL on COCO object detection/instance segmentation. It can be seen that the attention maps of NL block are mainly dominated by the unary term that different query points (marked in red) have similar overall attention maps. In DNL, the pairwise term in DNL shows clear within-region meaning and appears significant in the final overall attention maps, that different query points have different overall attention maps. DNL also shows more focus to salient regions than the one in an NL block.

Fig. 7 show more examples of the learnt attention maps by NL/DNL on Kinetics action recognition task. 4 frames in a video clip are visualized. The unary term of DNL shows better focus to salient regions than the one in an NL

block. The pairwise term in DNL also shows clearer within-region meaning than that in an NL block.

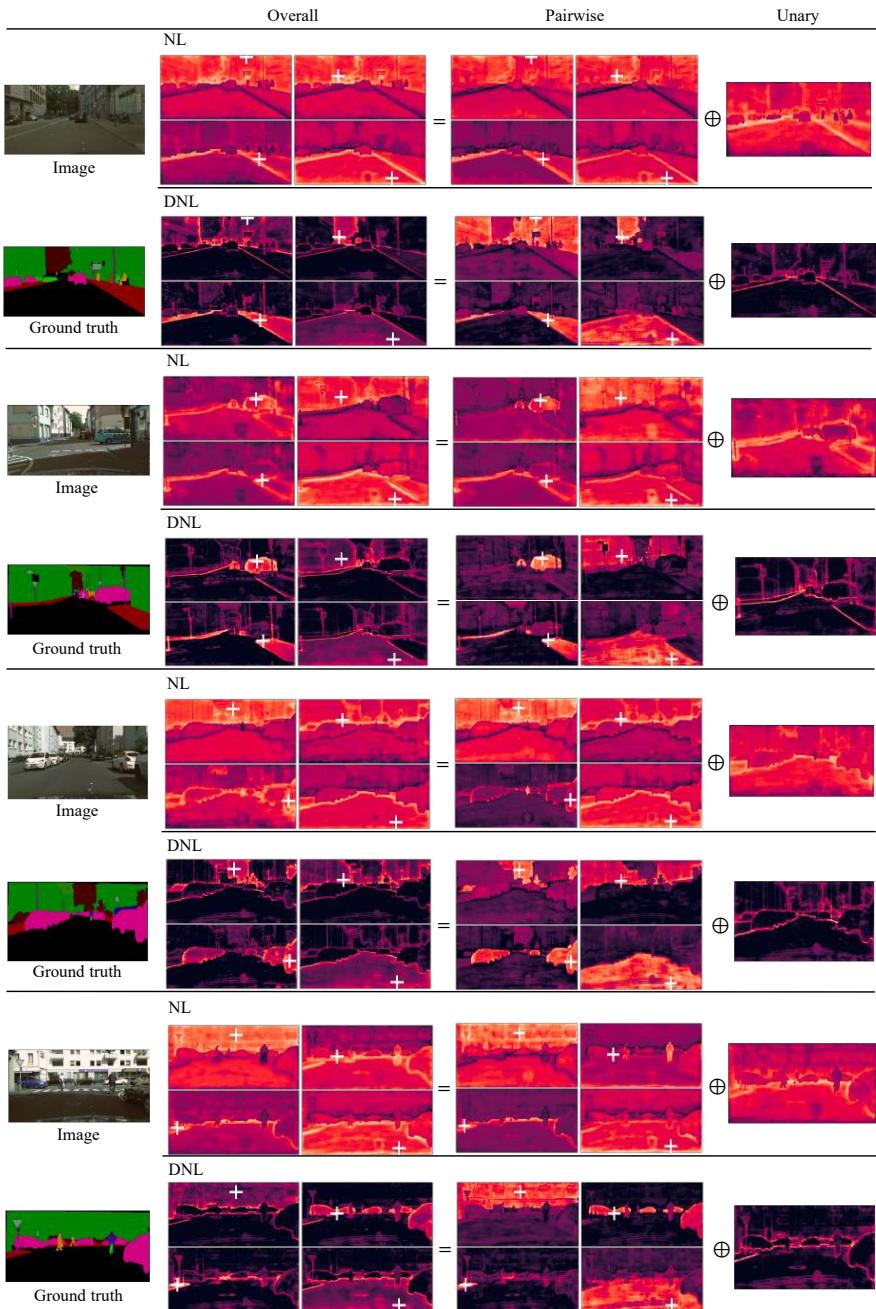


Fig. 5. Visualization of attention maps of NL block and our DNL block on Cityscapes Dataset. The query points are marked in white cross



Fig. 6. Visualization of attention maps of NL block and our DNL block on COCO object detection task. The query points are marked in red.

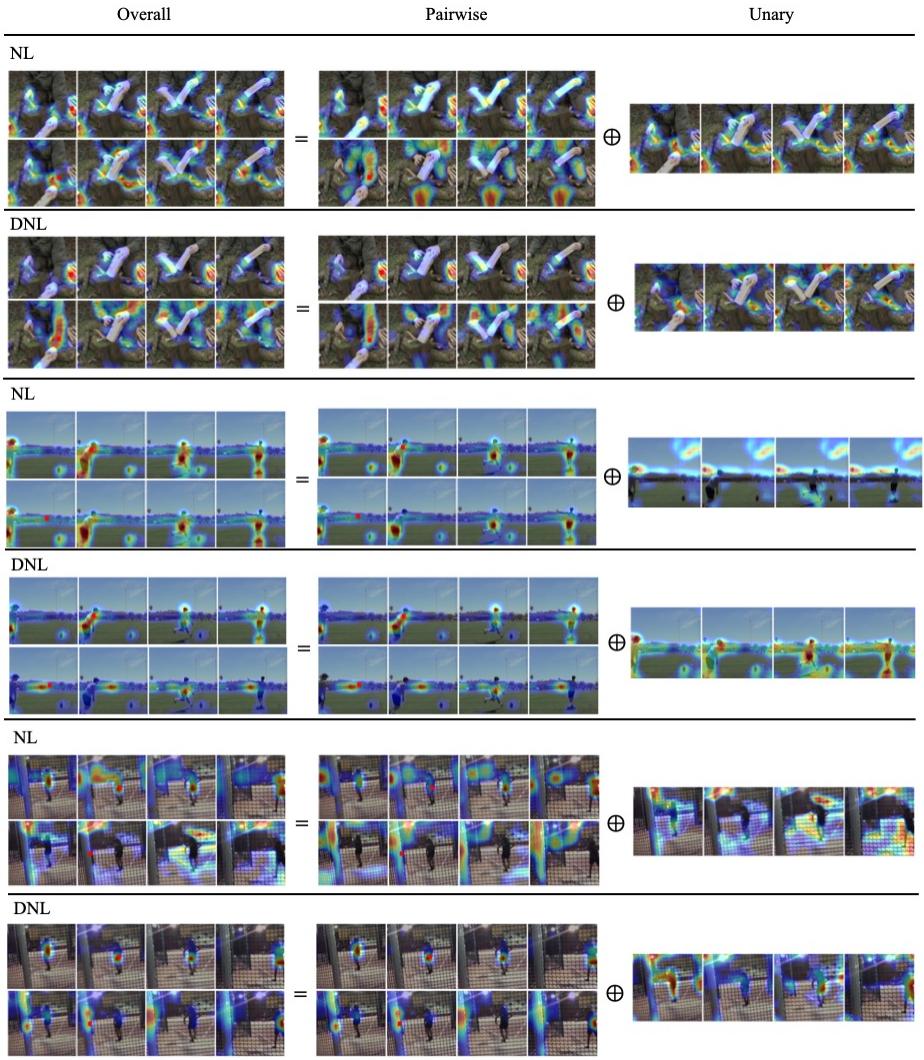


Fig. 7. Visualization of attention maps of NL block and our DNL block on Kinetics action recognition. 4 frames of a video clip are visualized. For each sample of each block, two different queries are chosen as the top and bottom rows. The query points are marked in red