

Adaptively Connected Neural Networks

Guangrun Wang
Sun Yat-sen University
Guangzhou

wanggrun@mail2.sysu.edu.cn

Keze Wang
University of California, Los Angeles
Los Angeles

kezewang@gmail.com

Liang Lin
Sun Yat-sen University
Guangzhou

linliang@ieee.org

Abstract

This paper presents a novel adaptively connected neural network (ACNet) to improve the traditional convolutional neural networks (CNNs) in two aspects. First, ACNet employs a flexible way to switch global and local inference in processing the internal feature representations by adaptively determining the connection status among the feature nodes (e.g., pixels of the feature maps)¹. We can show that existing CNNs, the classical multilayer perceptron (MLP), and the recently proposed non-local network (NLN) [48] are all special cases of ACNet. Second, ACNet is also capable of handling non-Euclidean data. Extensive experimental analyses on a variety of benchmarks (i.e., ImageNet-1k classification, COCO 2017 detection and segmentation, CUHK03 person re-identification, CIFAR analysis, and Cora document categorization) demonstrate that ACNet cannot only achieve state-of-the-art performance but also overcome the limitation of the conventional MLP and CNN². The code is available at <https://github.com/wanggrun/Adaptively-Connected-Neural-Networks>.

1. Introduction

Artificial neural networks have been extensively studied and applied over the past three decades, achieving remarkable accomplishments in artificial intelligence and computer vision. Among such networks, two types of neural networks have had a large impact on the research community. The first type is the multi-layer perceptron (MLP), which first became popular and effective via the development of the back-propagation training algorithm [34, 17]. However, since each neuron of the hidden layer in MLP is assigned with a private weight, the network parameters of MLP usually have a huge number and can be easily overfitted during the training phase. Moreover, MLP has difficulty in representing the spatial structure of 2D data (e.g., images). The second type

¹In a computer vision domain, a node refers to a pixel of a feature map, while in the graph domain, a node denotes a graph node.

²Corresponding author: Liang Lin (linliang@ieee.org)

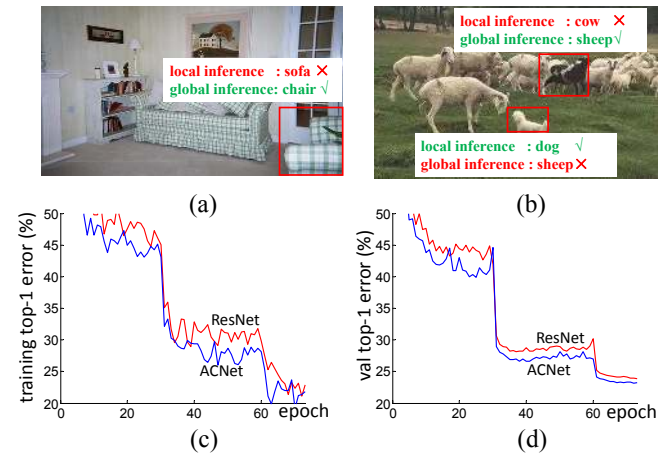


Figure 1: Some pixels prefer global dependencies, while others prefer local inference. For example, without global inference we cannot recognize the *chair* in (a). While in (b), the representation capacity of the *dog* is weakened by global information. Thanks to the adaptively determining the global/local inference, our ACNet achieves lower top-1 training/validation error than ResNet on ImageNet-1k shown in (c) and (d).

is convolutional neural networks (CNNs) [18]. Motivated by the biological visual cortex model, CNNs propose to group adjacent neurons to share identical weights and represent 2D data by capturing the local pattern (i.e., receptive field) of each neuron.

Although CNNs have been proven to be significantly superior over MLP, they have two drawbacks, as highlighted by [36]. On one hand, due to only abstracting information from local neighborhood pixels, the convolution operation inside each layer of CNNs does not have the ability of global inference. Consequently, convolution operations have difficulties in recognizing objects with similar appearances. For example, a convolution operation cannot distinguish the difference between the chair and the sofa in Fig.1 (a) which share the same appearance. In practice, CNN captures the global dependencies by stacking a number of local convolution operations, which still have several limitations, such as computational inefficiency, optimization difficulty, and

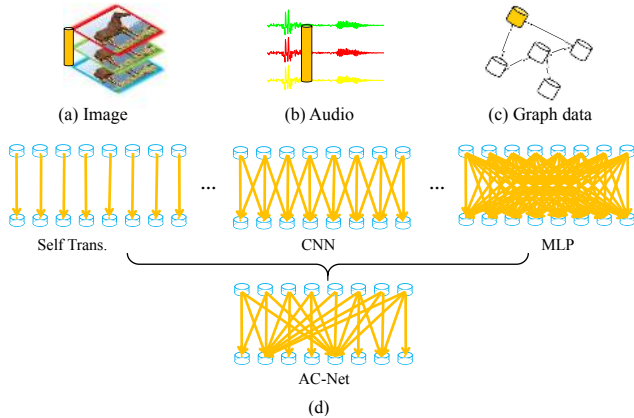


Figure 2: “Nodes” are presented in form of orange cylinder in (a) an image, (b) an audio, and (c) a general graph. (d) ACNet can be considered as a generalization of MLP and CNN on these “nodes”.

message passing inefficiency [48]. On the other hand, unlike MLP, conventional CNNs cannot be directly applied for non-Euclidean data (e.g., graph data), which are quite common in the area of machine learning.

To tackle the locality problem in CNNs, the recently proposed non-local network [48] (denoted as fully non-local networks) imposes global dependencies to all the feature nodes. However, empirically we observe degradations in fully non-local networks: as the non-locality of the network increases, both the training and validation accuracies degrade for the ImageNet-1k classification. We conjecture the degradation due to over-globalization. Specifically, the *dog* in Fig.1 (b) is easy to recognize if we only perform the local inference, while it can be misclassified as a cow when only performing the global inference. Intuitively, although quite challenging, it is necessary to jointly consider the global and local inference from image-aware (Fig.1(a)) or even node-aware (pixel-aware) (Fig.1(b)) perspective.

There have been many other recent attempts to address the aforementioned issues raised by CNNs and have achieved promising results [36, 12]. However, all of these methods are either over-localized or over-globalized. In contrast, this work focuses on developing a simple and general adaptively-connected neural network (ACNet) to adaptively capture the global and local dependencies, which inherits the strengths of both MLP and CNNs and overcomes their drawbacks. Thanks to the adaptively determining the global/local inference, our ACNet achieves lower top-1 training/validation error than ResNet on ImageNet-1k (see Fig. 1(c) and (d)).

ACNet first defines a simple yet basic unit named “node”, which is a unit of vectors in meta-data. As depicted in Fig.2, a node may be seen as a pixel of an image (Fig. 2(a)), a sampling of an audio (Fig. 2(b)), and a node of a general graph (Fig. 2(c)). Given the input data, ACNet adaptively is trained to search the optimal connection for each node, i.e., the connection \subseteq connecting {the node itself, its neighbor nodes,

all possible nodes}. Keep in mind that different nodes are connected adaptively, i.e., some nodes may be conjectured to themselves, some nodes may relate to its neighborhood, while other nodes have the global vision. Therefore, our ACNet can be considered as a generalization of CNN and MLP (Fig. 2 (d)). Note that, searching the optimal connections is differential by learning the importance degrees for different kinds of connections, which can be optimized via back-propagation.

The **main** contributions of this paper are summarized as follows. Firstly, we propose a conceptually general yet powerful network, which learns to switch global and local inference for general data (i.e. both Euclidean and non-Euclidean data) in a flexible parameter saving manner. Secondly, to the best of our knowledge, our proposed ACNet is the first one who is capable of inheriting the strength of both MLP and CNN while overcoming their drawbacks on a variety of computer vision and machine learning tasks, i.e., image classification on ImageNet-1k/CIFARs, object detection and segmentation on COCO 2017, person re-identification on CUHK03, and document categorization on Cora.

2. Related Work

Although significant progress has been achieved in the architecture design of CNNs from LeNet [19] to more recent deep and powerful networks (e.g., ResNet [10]), evolving the structure of CNNs to overcome their drawbacks is also quite crucial and a long-standing problem in machine learning (e.g. [21]). This issue motivates many researchers to extend CNNs to obtain different receptive fields [5]. Specifically, Dai *et al.* [5] proposed to enhance the transformation modeling capability of CNNs by introducing learnable offsets to augment the spatial sampling locations within the feature map. Chen *et al.* [2] revisited atrous convolution, a powerful tool to explicitly adjust filter’s field-of-view as well as control the resolution of feature responses computed by DNNs, in the application of semantic image segmentation. Peng *et al.* [31] proposed to use the large kernel filter and effective receptive field for semantic segmentation. Sabour *et al.* [36] proposed employing a group of neurons named a capsule to represent the instantiation parameters of a specific type of entity, such as an object and an object part. Building upon the work of [36], Hinton *et al.* [12] further presented a new type of capsule that has a logistic unit to represent the presence of an entity and a 4×4 pose matrix to represent the pose of that entity. Motivated by the self-attention mechanism [40], Wang *et al.* [48] incorporated non-local operations into CNNs as a generic family of building blocks for capturing long-range dependencies. Similarly, PSANet [52] is built upon NLN by introducing a position encoding to each pixel; GloRe [3] improves on NLN in a way of using a graph-CNN to capture the global dependencies. Although these methods achieved promising results, their performances are

limited due to the over-localization or over-globalization of the internal feature representation.

Moreover, several limited attempts [15, 38, 8, 59] have been made to extend CNNs for handling graph data. For instance, Kipf *et al.* [15] presented a layer-wise propagation rule for CNNs to operate directly on graph-structured data. Such *et al.* [38] defined filters as polynomials of functions of the graph adjacency matrix for unstructured graph data. However, these variants of CNNs pay close attention to bridge the gap between the graph structure of network inputs and the general graph data. The global inference inside the internal representation are ignored.

Our work is also related to the fully-connected neural networks (i.e. multilayer perceptron, or MLP), the densely connected neural networks [13], and the skip-connection neural networks (e.g. UNet [33], ResNet [11]), sharing the goal of finding an effective connection for the neural networks. However, the connections in our ACNet are automatically learned and adaptative to the data, while the connections in existing methods are fixed and handcrafted.

3. Adaptive-Connected Neural Networks

In this section, we first present the formulation of our proposed ACNet. Then, we discuss the relations between our ACNet and three most representative prior works, i.e., MLP, CNN, and NLN. Actually, they are special cases of our ANN. Moreover, we have also generalized our ANN for non-Euclidean data. Finally, we present the details of training, testing, and implementing our ACNet.

3.1. Formulation

Suppose x denotes the input signal (e.g., images, voices, graph matrices or their features). We propose to obtain the corresponding output signal as follows:

$$\mathbf{y}_i = \alpha_i \sum_{j=i} \mathbf{x}_j \mathbf{u}_{ij} + \beta_i \sum_{j \subseteq N(i)} \mathbf{x}_j \mathbf{v}_{ij} + \gamma_i \sum_{\forall j} \mathbf{x}_j \mathbf{w}_{ij}, \quad (1)$$

where y_i implies the i -th output node (e.g., the i -th pixel of a feature map) of the output signal, and j is the index of some possible nodes related to the i -th node. Actually, the j -th node belongs to three different sets, including {the i -th node itself}, {the neighborhood $N(i)$ of the i -th node}, and {all possible nodes}. These three sets indicate three different modes of inference: self transformation, local inference, and global inference, respectively. Moreover, \mathbf{u}_{ij} , \mathbf{v}_{ij} and \mathbf{w}_{ij} refer to the learnable weights between the i -th and j -th nodes for the three different sets, respectively. Note that the biases are omitted for notation simplification.

ACNet switches among different inference modes by adaptively learning α , β and γ in Eqn.1, which are importance degrees used to weighted average the modes. Note that, α , β and γ can be simple scalar variables, which are shared across all channels. We force $\alpha + \beta + \gamma = 1$, and

$\alpha, \beta, \gamma \in [0, 1]$, and define

$$\alpha = \frac{e^{\lambda_\alpha}}{e^{\lambda_\alpha} + e^{\lambda_\beta} + e^{\lambda_\gamma}}. \quad (2)$$

Here α is computed by using a softmax function with λ_α as the control parameter, which can be learned by the standard back-propagation (BP). Similarly, β and γ are defined by using another parameters λ_β and λ_γ , respectively. Note that the third term $\sum_{\forall j} \mathbf{x}_j \mathbf{w}_{ij}$ in Eqn.1 is quite computational consuming, because it equals to a fully-connected layer with large feature maps as input, leading to potential overfitting. To overcome this shortcoming, the \mathbf{x} is first transformed by an average pooling operation for downsampling in practice before being fed to calculate $\sum_{\forall j} \mathbf{x}_j \mathbf{w}_{ij}$. Finally, the obtained \mathbf{y} in Eqn.1 can be activated by a non-linear function $f(\cdot)$, such as BatchNorm+ReLU.

Actually, if α, β, γ are formulated as scalar variables, the connection for adaptively determining the global/local inference is an average connection over the whole dataset. To enable node-aware connection for each node (e.g., a pixel), α, β, γ can be also formulated as sample-dependent ones:

$$\gamma_i = \gamma_i(\mathbf{x}) = \mathbf{w}_{\gamma_i, 2} f(\mathbf{w}_{\gamma_i, 1} \left[\sum_{j=i} \mathbf{x}_j \mathbf{u}_{ij}; \sum_{j \subseteq N(i)} \mathbf{x}_j \mathbf{v}_{ij}; \sum_{\forall j} \mathbf{x}_j \mathbf{w}_{ij} \right]), \quad (3)$$

where $[\cdot; \cdot]$ denotes a concatenation operation and \mathbf{w}_{γ_i} denotes a linear transformation. α and β are defined in the similar way, which are omitted here. In the experimental section we will show that the above two kinds of formulation have the similar performance.

3.2. Relation to Prior Works

CNN. We take CNN as an illustrative example. For notation simplification, we omit the non-linear activation f , which does not affect the derivation process of the formulation. Let \mathbf{x} be the input data represented by a 3D tensor (C, H, W) . Let \mathbf{x}_i and \mathbf{y}_i be a node (pixel) of the input data and the output respectively, where $i, j \in [1, H \times W]$. Then a general 3×3 convolution can be formulated as

$$\mathbf{y}_i = \sum_{j \subseteq S} \mathbf{x}_j \mathbf{v}_{ij} \quad (4)$$

where S is the set that containing the nodes which have interactions with the given i -th node. Specifically, S denotes the set of eight neighbors for the i -th node, in addition to the i -th node itself, i.e., $S = \{i - W - 1, i - W, i - W + 1, i - 1, i, i + 1, i + W - 1, i + W, i + W + 1\}$.

MLP. MLP shares the formulation of Eqn.4, but it uses different sets of nodes to perform the linear combination. In other words, MLP enables more nodes to interact with the given i -th node, performing a global inference. For MLP, $S = \{1, 2, 3, \dots, H \times W\}$.

In summary, ACNet can be seen as a pure data-driven combination of CNN and MLP, fully exploiting the advantage of these two kinds of basic neural networks. For instance, let $\alpha = 0, \beta = 1, \gamma = 0$ in Eqn.1, ACNet degrades

into CNN; let $\alpha = 0, \beta = 0, \gamma = 1$ in Eqn.1, ACNet degrades into MLP. More importantly, ACNet dynamically switches between them by learning α, β and γ , providing more reasonable inferences. This allows us to build a richer hierarchy that combines both global and local information adaptively.

NLN. NLN also shares the formulation of Eqn.4, with $S = \{1, 2, 3, \dots, H \times W\}$, which is similar to MLP. However, there is a limitation in NLN. The \mathbf{v}_{ij} in NLN is obtained by computing the similarity of the i -th and the j -th nodes, which is very computation-consuming and easy to overfit. Therefore, NLN is rarely employed for image classification tasks. Instead of directly computing v_{ij} , our proposed ACNet absorbs the advantage of MLP (i.e., regarding v_{ij} as a learnable weight) and tackles its heavy computation problem by employing downsampling operation to perform the global inference. The relations between ACNet and prior works have been summarized in Fig. 2 (d).

3.3. Generalization to Non-Euclidean Data

We present the difference between Euclidean and non-Euclidean data, and then give a general definition of ACNet to handle both Euclidean and non-Euclidean data.

Euclidean data include the image, audio, and video, while non-Euclidean data contains graph and manifold. The difference is that Euclidean data are structured and non-Euclidean data are unstructured. Mathematically, for Euclidean data, we can denote the neighborhood of the i -th node in Eqn. 1 as $N(i) = \{i - W - 1, \dots, i + W + 1\}$, representing the {upper left, ..., low right} neighbors. But for non-Euclidean data we have difficulties. Besides, each node in Euclidean data has a fixed number of neighbors, while the number of neighbors is flexibly adapted to non-Euclidean data. Consequently, there is a gap in using Eqn.1 between Euclidean and non-Euclidean data. For Euclidean data \mathbf{v}_{ij} has different values at different j . But for non-Euclidean data \mathbf{v}_{ij} is shared among different j in Eqn.1. This weakens the representation capacity for non-Euclidean data due to the lack of position encoding. The similar phenomenon also occurs in \mathbf{w}_{ij} .

To fill the gap, Eqn. 1 is rewritten into a general form:

$$\mathbf{y}_i = \alpha_i \sum_{j=i} \mathbf{x}_j \mathbf{u} + \beta_i \sum_{j \subseteq N(i)} p_{ij}(\mathbf{x}_j \mathbf{v}) + \gamma_i \sum_{\forall j} q_{ij}(\mathbf{x}_j \mathbf{w}). \quad (5)$$

where \mathbf{u}, \mathbf{v} , and \mathbf{w} are shared among all kinds of j , which may be considered as 1×1 convolution in computer vision. Note that here α, β, γ is defined by using Eqn. 3. In compensation for the information loss in local structure, another two position encoding functions, i.e., p_{ij} and q_{ij} , are proposed to encode the index. These functions are just simple linear transformations using constant Gaussian noise. Specifically,

$$p_{ij}(\mathbf{x}_j \mathbf{v}) = \mathbf{x}_j \mathbf{v} \zeta_{ij}, \quad q_{ij}(\mathbf{x}_j \mathbf{w}) = \mathbf{x}_j \mathbf{w} \xi_{ij} \quad (6)$$

where ζ_{ij} and ξ_{ij} are constant variables sampled from a Gaussian noise.

Remark 1. Let ζ_{ij} and ξ_{ij} in Eqn.6 be learnable parameters instead of constant variables, then Eqn.6 turns out to be Eqn.1.

Remark 1 reveals that Eqn.5 is a lightweight version of Eqn.1, because a number of parameters are represented as constant variables in Eqn.5, exception the 1×1 convolution kernels \mathbf{u}, \mathbf{v} , and \mathbf{w} . In the experimental section we will show that compared to the state-of-the-art CNNs that use large kernels, ACNet with considerably fewer parameters can also achieve their strengths in feature learning, by only exploiting highly efficient 1×1 convolution operations.

3.4. Training, Inference, and Implementation

Training & Inference. Let Θ be a set of network parameters (e.g. convolution filters and fully-connected weights) and Φ be a set of control parameters that control the network architecture. In ACNet, we have $\Phi = \{\lambda_\alpha, \lambda_\beta, \lambda_\gamma\}$. Training an ACNet network is to minimize a loss function $\mathcal{L}(\Theta, \Phi)$, where Θ and Φ can be optimized jointly by back-propagation (BP). ACNet is tested in the same way as standard networks such as CNN and MLP.

Compatibility with CNN Tricks and Techniques. Our proposed ACNet is quite compatible with most existing tricks and techniques for CNNs. For instance, through embedding a batch normalization [14] layer into every non-linear mapping function $f(\cdot)$, our ACNet can support a large learning rate for high learning efficiency. Meanwhile, we can also exploit the residual connection strategy [10] to create a short-cut connection for each layer inside our ACNet.

Implementation. ACNet can be easily implemented by using the existing software such as TensorFlow and PyTorch. The backward computation of ACNet can be obtained by automatic differentiation techniques (AD) in these software. Without AD, ACNet can also be implemented by regarding $\Phi = \{\lambda_\alpha, \lambda_\beta, \lambda_\gamma\}$ as learnable parameters.

4. Experiments

This section presents the main results of ACNet in multiple challenging problems and benchmarks, such as ImageNet-1k classification [35], COCO 2017 detection and segmentation [27], CUHK03 person re-identification [20], CIFAR [16] classification, and Cora document categorization [37], where the effectiveness of ACNet is demonstrated by comparing with the existing state-of-the-art CNNs/NLNs.

4.1. ImageNet-1k Classification

We first compare our ACNet with the most representative CNNs/NLNs on the ImageNet classification dataset of 1k categories. All the models are trained on the 1.28M training images and evaluated on the 50k validation images. Our baseline model is the representative ResNet50. We examine top-1 accuracy on the 224×224 single/center-crop-single-scale images. Note that the top-1 accuracies of the baseline

Table 1: Comparison of ImageNet *val* top-1 accuracies and parameter numbers on ResNet50. **ACNet \ddagger** : pixel-aware ACNet using Eqn. 3; **ACNet**: dataset-aware ACNet with α, β, γ being scalar variables;

	top-1 accuracies	#params
CNN-ResNet50	76.4 $^{\uparrow 0.0}$	25.56M $^{\times 1.00}$
ACNet-ResNet50	77.5$^{\uparrow 1.1}$	29.38M $^{\times 1.15}$
ACNet\ddagger-ResNet50	77.5$^{\uparrow 1.1}$	31.85M $^{\times 1.25}$
generalized ACNet	76.2$^{\downarrow 0.2}$	19.80M$^{\times 0.77}$

Table 2: Comparison between ACNet-Resnet50 and CNN-ResNet60 in terms of ImageNet *val* top-1 accuracies and parameter numbers.

	top-1 accuracies (%)	#params
CNN-ResNet60	76.7 $^{\uparrow 0.0}$	30.03M $^{\times 1.00}$
ACNet-ResNet50	77.5$^{\uparrow 0.8}$	29.38M$^{\times 0.98}$

approximately equals to the official results and the model zoo³ (Caffe; Tensorflow; Pytorch). CNN-ResNet50 is exactly the original ResNet50. For ACNet-ResNet50, all the 3×3 convolution in CNN-ResNet50 are replaced with ACNet layers. And for NLN-ResNet50, the non-local operations are attached to every 3×3 convolution in CNN-ResNet50.

Classification accuracies. The comparison results of top-1 validation accuracies are illustrated in Table 1. As depicted, our ACNet-ResNet50 performs approximately 1.1% better than the compared CNN-ResNet50 (77.5% vs 76.4%). The training and validation curves in Fig. 1 (c) and (d) also show the sustainable competitive advantage of our ACNet-ResNet50 over CNN-ResNet50. This improvement is quite significant due to the challenge of ImageNet-1k.

The superior performance of our ACNet is attributed to two reasons. First, ACNet adaptively performs global and local inference for different pixels of internal feature maps from each layer, leading to a flexible discriminative representation learning fashion, which contributes to capturing the local and global dependencies for improving classification accuracy. Second, the mechanism of ACNet may implicitly act as comprehensive data-driven ensembling, which aggregates the advantage of both global and local information.

Pixel-aware Connection. As is mentioned in Section 3.1, different pixels can have different pixel-aware connection by using Eqn. 3. We report the accuracies of pixel-aware and dataset-aware connection in Table 1 respectively. For the pixel-aware connection, we let $\alpha = 0, \beta = 1$ and only learn γ to save parameters and memory. The results show that these two kinds of connection have the same top-1 accuracy. While the pixel-aware connection has more parameters (31.85M vs 29.38M).

³<https://github.com/Cadene/pretrained-models.pytorch>

Table 3: Computational complexity analysis on the ImageNet-1k.

Networks	CNN-ResNet50	NLN-ResNet50	ACNet-ResNet50
Speed $^{\text{images/sec}}$	198 $^{\times 1.00}$	nan	144$^{\times 0.77}$
Memory $^{\text{GB}}$	8.579 $^{\times 1}$	out of memory	8.580$^{\times 1}$

Extra Parameters. In fact, ACNet has introduced extra parameters by $0.15 \times$ (29.4M vs 25.6M, Table 1). The extra parameters are from the global inference (i.e. $\sum_{\forall j} \mathbf{x}_j \mathbf{w}_{ij}$ in Eqn.1). Thanks to downsampling operation, the extra parameters only introduce negligible computation time and memory usage, which will be examined later. To eliminate the confounding factor of extra parameters and justify the gain of ACNet, we present more comparisons:

1. We compare ACNet-ResNet50 with CNN-ResNet60, which has the same level of parameters. The result in Table 2 shows that ACNet-ResNet50 obtains a slightly higher accuracy (77.5% vs 76.7%) than CNN-ResNet60, demonstrating the superiority of ACNet over CNN with the nearly same number of parameters.
2. The general form of ACNet is also compared with CNN. As is discussed in Sect. 3.3, ACNet can be rewritten to a general form for supporting both Euclidean and non-Euclidean data. Remark 1 in Sect. 3.3 reveals that the general form is much more ($0.77 \times$) lightweight. The experimental result in Table 2 confirms this remark, and further shows that ACNet with considerably fewer parameters can also achieve their strengths in feature learning (76.2% vs 76.4%), by only exploiting highly efficient 1×1 convolution operations.

Computation Complexity. Table 3 reports the computation complexity of ACNet, CNN and NLN. For a fair comparison, all methods are trained in the same desktop with 8 Titan Xp GPUs. We observe that ACNet and CNN have similar computational costs. Specifically, the memory consumption of both ACNet and CNN are the same, i.e. 8.6GB. But the speed of ACNet is slightly slower than CNN (144 vs 198 images/second/GPU). As a comparison, NLN is intractable because NLN requires a vast amount of memory to calculate the similarity between any two pixels of a feature map. The memory required is beyond the testing desktop can provide. Actually, NLN performs significantly slower than ACNet and CNN according to our observation.

Visualization of importance degrees. The importance degrees in each ACNet layer are visualized in Fig. 3, from which we have two observations. First, the importance degrees differ from pixel to pixel. This is due to the global and local inference are pixel-aware, i.e. different pixels have different inference modes. Second, the importance degrees also differs from layer to layer – there is much more global inference in lower-level layers than in higher-level layers.

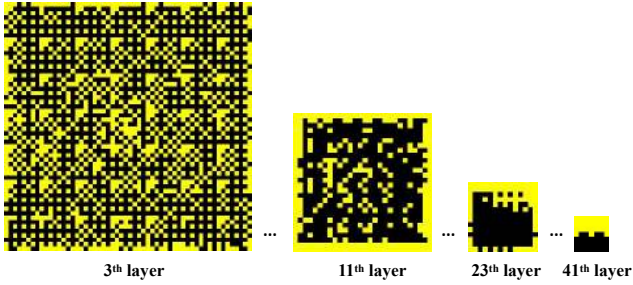


Figure 3: Visualization of the nodes with different types of inference generated by our ACNet, which is trained on ImageNet. One node painted by the yellow color indicates its the output of the global inference from the preceding layer (i.e., it connects to all nodes in the preceding layer), while the opposite black nodes indicate the outputs of the local inference from the preceding layer.

Table 4: Ablation studies on CIFAR10.

Method	Error (%)
Standard ACNet	6.0 ^{±0.0}
w/o global inference	7.1 ^{±1.1}
w/o local inference	24.0 ^{±18}
Fixed global+local	6.8 ^{±0.8}

Although CNN somewhat can capture a few global dependencies in high-level layers by stacking a number of local convolutional layer, it has difficulties in local inference in lower-level layers, as shown in Fig. 3. Fortunately, our ACNet provides compensatory global inference for these lower-level layers. Overall, examining the necessity of global inference in lower layer discloses interesting characteristics and impacts in DNNs, and sheds light on model design in many research fields.

4.2. Analysis on CIFAR10

As the ImageNet-1k dataset is quite large and the training from scratch is extremely time-consuming, we conduct more ablation studies on CIFAR10 [16] classification benchmark to deeply analyze ACNet. CIFAR-10 consists of 50k training images and 10k testing image in 10 classes. The presented experiments are trained on the training set and evaluated on the testing set as [11]. Our focus is to analyze the components of ACNet instead of achieving the state-of-the-art results, therefore we use the representative ResNet32 proposed in [11]. All the implementation details and experiment settings are the same as [11, 46, 47].

The role of global inference. We first evaluate the effectiveness of global inference by constructing two different networks, i.e. with and without the third term $\sum_{\forall j} q_{ij}(\mathbf{x}_j \mathbf{w})$ in Eqn.5. As shown in Table 8, without global inference, ACNet has a performance degradation of 1.1%. As we know, ACNet without global inference equals to CNN. This com-

parison verifies the superiority of ACNet over CNN.

The role of local inference. Next, we investigate the necessity of local inference. In Table 8, we compare two operations, i.e. with and without local inference. Table 8 shows that equipped with local inference, ACNet has a significant performance gain of 18%, verifying the contribution of local inference. This is natural in the image domain. The lack of local inference leads to neglecting some critical information. Intuitively, we can easily represent an image as an adjacent matrix. But we can never recover the original image from the adjacent matrix, demonstrating an information loss by discarding the local inference.

Adaptively global+local vs fixed global+local. Next, we investigate the necessity of adaptively switching between global and local inference. We fixed the importance degrees α , β and γ as constant variables, forming the *fixed global+local* version of ACNet. We have an **interesting** observation in Table 8: imposing global information to every pixel has poorer performance than adaptively adding global information (6.0% vs 6.8%). In other words, the global inference is unessential for every pixel, because it may hurt the training. This implies the superiority of adaptively connected neural networks over the fully non-local networks.

4.3. COCO Object Detection and Segmentation

We have demonstrated the adaptive inference capacity of ACNet in ImageNet classification task, whose receptive field is quite large due to 5 times of subsampling and a global pooling. Next, we investigate an inevitable smaller receptive field task, i.e. COCO 2017 detection & segmentation task [27]. These computer vision tasks in general benefit from higher-resolution input and output. Therefore, the global pooling and some subsampling are removed from the backbone of ResNet50, leading to a smaller receptive field. As a result, the adaptively global and local inference is in desire.

We finetune the models trained on ImageNet [35] for transferring to detection and segmentation. The batch normalization parameters are frozen during the finetuning.

We conduct experiments on the Mask RCNN baselines [9] using a ResNet50-FPN backbone. We replace CNN layers with ACNet layers. The models are trained in the COCO train2017 set and evaluated in the COCO val2017 set. We use the standard training setting following the COCO model zoo. We report the standard COCO metrics of Average Precision (AP) for bounding box detection (AP^{bbox}) and instance segmentation (AP^{mask}).

Table 5 shows the comparison of ACNet vs NLN vs CNN. ACNet improves over CNN by 1.5% box AP and 0.6% mask AP. This may be contributed to the fact that CNN lacks adaptive inference capacity. We have also found NLN is 0.5% box AP worse than ACNet. In summary, although NLN is also suitable global inference, its representational power is slightly weaker than ACNet according to our current

Table 5: Detection and segmentation ablation studies on COCO2017 using Mask RCNN.

backbone	AP ^{box}	AP ^{mask}
CNN	38.0 ^{↑0.0}	34.6 ^{↑0.0}
NLN	39.0 ^{↑1.0}	35.5 ^{↑0.9}
ACNet	39.5^{↑1.5}	35.2^{↑0.6}

evaluation. The inferiority of NLN is attributed to the over-globalization. Specifically, the redundant global context may hurt but NOT help the model learning. This phenomenon has also been observed experimentally by [58] and theoretically by [28], confirming that the over-globalization is a shortcoming of NLN.

4.4. CUHK03 Person Re-identification

To demonstrate the good generalization performance of our proposed ACNet on the other recognition tasks, we have conducted the extensive experiments on the person re-identification challenge, which refers to the problem of re-identifying individuals across cameras. Though quite challenging, person re-identification is fundamental and beneficial from many applications in video surveillance for keeping the security of safety of the whole society [43, 60, 42, 24, 44, 6, 45, 26, 23, 22].

Dataset. We conduct experiments on the CUHK03 dataset [20], which is one of the largest databases for person re-identification. This database contains 14,096 images of 1,467 pedestrians. Each person is observed by two disjoint camera views and is shown in 4.8 images on average in each view. We follow the **new** standard setting of using CUHK03 [56], where 767 individuals are regarded as the training set and another 700 individuals are considered as the testing set without sharing the same individuals.

Evaluation metric. For the evaluation, the testing set is further divided into a gallery set of images and a probe set. We use the standard rank-1 as the evaluation metric.

Result Analysis. In Table 6, we compare with the current best models. A total of 11 representative state-of-the-art methods, BOW+XQDA [53], PUL [7], LOMO+XQDA [25], IDE [54], IDE+DaF [51], IDE+XQ.+Re-ranking [55], PAN, DPFL [4], and the newly proposed methods SVDNet [39], TriNet + Era. [56], and TriNet + Era. + Reranking [56], are used as the competing methods. All the settings of the above methods are consistent with the common training settings as [56]. ACNet has achieved a new state-of-the-art performance. Specifically, ACNet achieves a rank-1 accuracy of 64.8%. We can also observe that ACNet surpasses its baseline by a clear margin (3.6%, Table 6). This verifies the effectiveness of ACNet on the person re-identification task.

Table 6: Comparison on a Person Re-identification task (CUHK03, where ‘bs’ denotes batch size.)

	Rank-1
BOW+XQDA [53]	6.4
PUL [7]	9.1
LOMO+XQDA [25]	12.8
IDE [54]	21.3
IDE+DaF [51]	26.4
IDE+XQ.+Re-ranking [55]	34.7
PAN	36.3
DPFL [4]	40.7
SVDNet [39]	41.5
TriNet + Era. [56]	55.5
TriNet + Era.(Our reproduction)	62.0 ^{↑0.0}
TriNet + Era. + ACNet	64.3^{↑2.3}
TriNet + Era. + reranking(bs = 32)	61.2 ^{↑0.0}
TriNet + Era. + reranking + ACNet(bs = 32)	64.8^{↑3.6}

4.5. Analysis on Cora : a Non-Euclidean Domain

A common form of graph-structured data is a network of documents. For example, scientific documents in a database are related to each other through citations and references. Administrators of such large networks may desire to automatically label documents according to their relationships to the remainder of the literature. To demonstrate the compatibility of ACNet for non-Euclidean data, we adapt our proposed ACNet to tackle such a vertex classification task on the Cora benchmark [37], which is a large network of scientific publications connected through citations. The vertex features, in this case, are binary word vectors that indicate the presence of a word from a dictionary of 1,433 unique words. There are 2708 publications classified under 7 different categories - case-based, genetic algorithms, neural networks, probabilistic methods, reinforcement learning, rule learning, and theory. There is an edge connection from a cited article to a citing article and another edge connection from a citing article to a cited article. These edge features are also binary representations. We use a quite simple architecture following [15], which only contains two graph convolutional layers. The first layer is used for feature learning, and the second layer is used for classifier learning. We replace the first graph convolutional layer in [15] with our ACNet layer. Note that in our ACNet α, β, γ is defined by using Eqn. 3. Considering the Cora dataset is quite small-scale, we let $\alpha = 0, \beta = 1$ and only learn γ to avoid overfitting. We perform 10-fold cross validations to form the training and test set for a fair comparison as the majority of methods [1, 15] did.

Comparisons with the state-of-the-art methods. We first compare with the current best models. A total of 11 representative state-of-the-art methods, i.e., ManiReg [1], SemiEmb [49], LP [57], DeepWalk [32], ICA [29], Planetoid [50], the newly proposed methods Graph-CNN [15],

Table 7: Comparison with state-of-the-art on Cora document classification dataset.

Method	Accuracy (%)
ManiReg [1]	59.5
SemiEmb [49]	59.0
LP [57]	68.0
DeepWalk [32]	67.2
ICA [29]	75.1
Planetoid* [50]	75.7
Graph-CNN [15]	81.5
MoNet [30]	81.7
GAT [41]	83.0
LGCN [8]	83.3
Dual GCN [59]	83.5
ACNet	83.5

Table 8: Ablation studies on Cora document classification dataset.

Method	Accuracy (%)
Standard ACNet	83.5^{±0.0}
w/o global inference	82.1 ^{±1.4}
w/o local inference	76.3 ^{±7.2}
w/o position encoding	83.0 ^{±0.5}
Fixed global+local	82.7 ^{±0.8}

MoNet [30], GAT [41], LGCN [8], and Dual GCN [59] are used as competing methods. Table 7 shows that ACNet achieves comparable performance to the best of all competitive methods, e.g., Dual GCN [59] (83.5%). This comparison once again verifies the generalization performance of ACNet. Next, we investigate which component of ACNet contributes to the non-Euclidean data to shed light on future researches.

The role of global inference. We first evaluate the effectiveness of global inference by constructing two different networks, *i.e.* with and without the third term $\sum_{\forall j} q_{ij}(\mathbf{x}_j \mathbf{w})$ in Eqn.5. As shown in Table 8, without global inference, ACNet has a performance degradation of 1.4%. This is reasonable for a document categorization problem like Cora. A document categorization problem is slightly different from a conventional image classification one because each article is not isolated. All the articles are connected with each other in the form of citations. In this sense, a document categorization problem is more like a semantic image segmentation problem in computer vision. Therefore global inference in ACNet is essential for Cora.

The role of local inference. Next, we investigate the necessity of local inference. In Table 8, we compare two operations, *i.e.* with and without local inference. Table 8 shows that equipped with local inference, ACNet obtains a gain of 7.2%, verifying the contribution of local inference. The lack of local inference leads to neglecting some critical information. Specifically, each article in Cora cites several other articles, as well as being cited by other articles. Actu-

ally, the citing articles and the cited articles may share the same category with the given article. Without local inference, ACNet cannot capture the citation information. The performance degradation of “w/o local inference” may be due to ignoring this knowledge.

Adaptively global+local vs fixed global+local. We fixed the importance degrees α , β and γ as constant variable, forming the *fixed global+local* version of ACNet. Similar to the CIFAR10 case, the results in Table 8 confirms the effectiveness of adaptively global and local inference, with a gain of 0.8%. The reason can be attributed to the property of the document. Some article can be easier to categorize when considered in local range than in wide range. For example, at first, we can easily categorize the reinforcement-learning-based article into the “reinforcement learning” area. But after reading more and more article, we may confuse it with “neural networks” area with the emergence of deep reinforcement learning.

The role of position encoding. At last, we investigate the impact of position encoding. We remove the position encoding in Eqn.5 to obtain the counterpart. Table 8 shows that without the position encoding, ACNet suffers a performance drop of 0.5%. This is because the non-Euclidean data is unstructured compared with the Euclidean data. Without a position encoding, the non-Euclidean data is with too many degrees of freedom (*i.e.*, the same graph data may have different representations because theoretically, a graph has endless isomorphic graphs). This freedom leads to lower learning efficiency. By introducing the position encoding the training inefficiency has been alleviated

5. Conclusion

This paper presented a concise ACNet to be a promising substitute for overcoming the limitations of widely used deep CNNs without losing their strengths in feature learning. Specifically, ACNet advances in adaptively switching between global and local inference in a flexible and pure data-driven manner. We further applied our proposed ACNet for the recognition tasks of both Euclidean data and non-Euclidean data. Extensive experimental analyses from a variety of aspects justify the superiority of ACNet. In the future, we will extend our work to be suitable for more general tasks to demonstrate its superiority.

Acknowledgement

This work was supported in part by the National Key Research and Development Program of China under Grant No. 2018YFC0830103 and 2016YFB1001004, in part by National High Level Talents Special Support Plan (Ten Thousand Talents Program), and in part by National Natural Science Foundation of China (NSFC) under Grant No. 61622214, 61836012, and 61876224.

References

- [1] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7(Nov):2399–2434, 2006.
- [2] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.
- [3] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Shuicheng Yan, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. *arXiv preprint arXiv:1811.12814*, 2018.
- [4] Yanbei Chen, Xiatian Zhu, and Shaogang Gong. Person re-identification by deep learning multi-scale representations. In *CVPR*, pages 2590–2600, 2017.
- [5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017.
- [6] Shengyong Ding, Liang Lin, Guangrun Wang, and Hongyang Chao. Deep feature learning with relative distance comparison for person re-identification. *Pattern Recognition*, 48(10):2993–3003, 2015.
- [7] Hehe Fan, Liang Zheng, and Yi Yang. Unsupervised person re-identification: Clustering and fine-tuning. *arXiv preprint arXiv:1705.10444*, 2017.
- [8] Hongyang Gao, Zhengyang Wang, and Shuiwang Ji. Large-scale learnable graph convolutional networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1416–1424. ACM, 2018.
- [9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [12] Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with em routing. In *ICLR*, 2018.
- [13] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. In *CVPR*, volume 1, page 3, 2017.
- [14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456, 2015.
- [15] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *ICLR*, 2017.
- [16] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.
- [17] Yann Lecun. *PhD thesis: Modeles connexionnistes de l'apprentissage (connectionist learning models)*. Universite P. et M. Curie (Paris 6), 6 1987.
- [18] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.
- [19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 11(6):2278–2324, 1998.
- [20] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 152–159, 2014.
- [21] Xilai Li, Tianfu Wu, Xi Song, and Hamid Krim. Aognets: Deep and-or grammar networks for visual recognition. *arXiv preprint arXiv:1711.05847*, 2017.
- [22] Ya Li, Guangrun Wang, Liang Lin, and Huiyou Chang. A deep joint learning approach for age invariant face verification. In *CCF Chinese Conference on Computer Vision*, pages 296–305. Springer, 2015.
- [23] Ya Li, Guangrun Wang, Lin Nie, Qing Wang, and Wenwei Tan. Distance metric optimization driven convolutional neural network for age invariant face recognition. *Pattern Recognition*, 75:51–62, 2018.
- [24] Wenqi Liang, Guangcong Wang, Jianhuang Lai, and Junyong Zhu. M2m-gan: Many-to-many generative adversarial transfer learning for person re-identification. *arXiv preprint arXiv:1811.03768*, 2018.
- [25] Shengcai Liao, Yang Hu, Xiangyu Zhu, and Stan Z Li. Person re-identification by local maximal occurrence representation and metric learning. In *CVPR*, pages 2197–2206, 2015.
- [26] Liang Lin, Guangrun Wang, Wangmeng Zuo, Xiangchu Feng, and Lei Zhang. Cross-domain visual matching via generalized similarity measure and feature learning. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1089–1102, 2017.
- [27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [28] David Lopez-Paz, Robert Nishihara, Soumith Chintala, Bernhard Scholkopf, and Léon Bottou. Discovering causal signals in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6979–6987, 2017.
- [29] Qing Lu and Lise Getoor. Link-based classification. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 496–503, 2003.
- [30] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proc. CVPR*, volume 1, page 3, 2017.
- [31] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large kernel matters—improve semantic segmentation by global convolutional network. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 1743–1751. IEEE, 2017.
- [32] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of*

- the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.
- [33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [34] D. Rumelhart, G. Hinton, and R. Williams. Learning internal representations by backpropagating errors. *Parallel distributed processing: Explorations in the microstructure of cognition*, 1986.
- [35] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.
- [36] Sara Sabour, Nicholas Frosst, and Geoffrey Hinton. Dynamic routing between capsules. In *NIPS*, 2017.
- [37] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93, 2008.
- [38] Felipe Petroski Such, Shagan Sah, Miguel Domínguez, Suhas Pillai, Chao Zhang, Andrew Michael, Nathan D. Cahill, and Raymond W. Ptucha. Robust spatial filtering with graph convolutional neural networks. *J. Sel. Topics Signal Processing*, 11(6):884–896, 2017.
- [39] Yifan Sun, Liang Zheng, Weijian Deng, and Shengjin Wang. Svdnet for pedestrian retrieval. *arXiv preprint arXiv:1703.05693*, 2017.
- [40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, 2017.
- [41] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 1(2), 2017.
- [42] Guangcong Wang, Jianhuang Lai, Peigen Huang, and Xiaohua Xie. Spatial-temporal person re-identification. *arXiv preprint arXiv:1812.03282*, 2018.
- [43] Guangcong Wang, Jianhuang Lai, and Xiaohua Xie. P2snet: Can an image match a video for person re-identification in an end-to-end way? *IEEE Transactions on Circuits and Systems for Video Technology*, 28(10):2777–2787, 2018.
- [44] Guangcong Wang, Jianhuang Lai, Zhenyu Xie, and Xiaohua Xie. Discovering underlying person structure pattern with relative local distance for person re-identification. *arXiv preprint arXiv:1901.10100*, 2019.
- [45] Guangrun Wang, Liang Lin, Shengyong Ding, Ya Li, and Qing Wang. Dari: Distance metric and representation integration for person verification. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [46] Guangrun Wang, Ping Luo, Xinjiang Wang, Liang Lin, et al. Kalman normalization: Normalizing internal representations across network layers. In *Advances in Neural Information Processing Systems*, pages 21–31, 2018.
- [47] Guangrun Wang, Jiefeng Peng, Ping Luo, Xinjiang Wang, and Liang Lin. Batch kalman normalization: Towards training deep neural networks with micro-batches. *arXiv preprint arXiv:1802.03133*, 2018.
- [48] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *arXiv:1711.07971 [cs.LG]*, 2017.
- [49] Jason Weston, Frédéric Ratle, Hossein Mobahi, and Ronan Collobert. Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade*, pages 639–655. Springer, 2012.
- [50] Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. *arXiv preprint arXiv:1603.08861*, 2016.
- [51] Rui Yu, Zhichao Zhou, Song Bai, and Xiang Bai. Divide and fuse: A re-ranking approach for person re-identification. *arXiv preprint arXiv:1708.04169*, 2017.
- [52] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. Psanet: Point-wise spatial attention network for scene parsing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 267–283, 2018.
- [53] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *ICCV*, 2015.
- [54] Liang Zheng, Yi Yang, and Alexander G Hauptmann. Person re-identification: Past, present and future. *arXiv preprint arXiv:1610.02984*, 2016.
- [55] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. Re-ranking person re-identification with k-reciprocal encoding. *arXiv preprint arXiv:1701.08398*, 2017.
- [56] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. *arXiv preprint arXiv:1708.04896*, 2017.
- [57] Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, pages 912–919, 2003.
- [58] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. *arXiv preprint arXiv:1811.11168*, 2018.
- [59] Chenyi Zhuang and Qiang Ma. Dual graph convolutional networks for graph-based semi-supervised classification. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 499–508. International World Wide Web Conferences Steering Committee, 2018.
- [60] Jiaxuan Zhuo, Zeyu Chen, Jianhuang Lai, and Guangcong Wang. Occluded person re-identification. In *2018 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2018.