
LatentGNN: Learning Efficient Non-local Relations for Visual Recognition

Songyang Zhang¹ Shipeng Yan¹ Xuming He¹

Abstract

Capturing long-range dependencies in feature representations is crucial for many visual recognition tasks. Despite recent successes of deep convolutional networks, it remains challenging to model non-local context relations between visual features. A promising strategy is to model the feature context by a fully-connected graph neural network (GNN), which augments traditional convolutional features with an estimated non-local context representation. However, most GNN-based approaches require computing a dense graph affinity matrix and hence have difficulty in scaling up to tackle complex real-world visual problems. In this work, we propose an efficient and yet flexible non-local relation representation based on a novel class of graph neural networks. Our key idea is to introduce a latent space to reduce the complexity of graph, which allows us to use a low-rank representation for the graph affinity matrix and to achieve a linear complexity in computation. Extensive experimental evaluations on three major visual recognition tasks show that our method outperforms the prior works with a large margin while maintaining a low computation cost. For facilitating the future research, code is available: <https://github.com/latentgnn/LatentGNN-V1-PyTorch>

1. Introduction

Modeling non-local relations in feature representations is a fundamental problem in visual recognition, which enables us to capture long-range dependencies between scene entities and handle feature ambiguity for high-level semantic understanding tasks (Strat & Fischler, 1991). In particular, tremendous progress has been made recently in computer

¹School of Information Science and Technology, ShanghaiTech University, Shanghai, China. Correspondence to: Xuming He <xmhe@shanghaitech.edu.cn>.

vision community by learning hierarchical feature representations of images with deep convolutional networks. Nevertheless, the dominating feature learning strategy in deep Convnets is through stacking many local operations of convolution and pooling, which are effectively limited to local receptive fields and short-range context (Luo et al., 2016; Chen et al., 2017).

There have been several attempts in the literature to tackle the problem of incorporating non-local context into representations of existing deep models. Early works tend to incorporate recurrent neural network (RNN) module to capture the long-range dependencies by constructing a directed acyclic graph model in convolutional feature space (Byeon et al., 2015; Shuai et al., 2018) or semantic label space (Zheng et al., 2015). These RNN-based models are typically defined on graphs with local neighborhood or specific edge functions, which are limited in modeling non-local relations and computationally expensive. An alternative strategy of context modeling is to exploit image structure and integrate a region-based representation with deep convolutional features (Li & Gupta, 2018). Regions and their relations (e.g., similarity) are typically modeled as an undirected graph, which can be learned jointly with convolutional features to encode long-range dependencies. The region generation (Gould et al., 2009), however, relies on bottom-up grouping that is sensitive to low-level image cues and may yield erroneous representations.

More recent efforts have been focused on introducing non-local operations into the deep Convnets (Wang et al., 2018; Yue et al., 2018). In particular, Wang et al. (2018) propose to augment the convolutional features by a context representation computed from the entire feature map at certain convolution layers. Such non-local operations can be viewed as a densely-connected graph neural network (GNN) module with attention mechanism (Velickovic et al., 2018), which estimates the feature context by a single iteration of message passing on the underlying graph. This feature augmentation method allows a flexible way to represent non-local relations between features and has led to significant improvements in several vision recognition tasks (Wang et al., 2018). However, despite its successes, the GNN-based non-local neural network suffers from two major limitations, especially in the context of visual recognition. First, the graph message passing requires computing a dense affinity matrix of the

graph and its product with the node messages, which are computationally prohibitive for large-sized feature maps. In addition, the graph edge affinities, which capture relations between two node features, are typically defined by a simple pairwise function (with possible normalization) and hence has a limited capacity in modeling complex relationships.

In this work, we propose an efficient and yet flexible non-local relation representation of convolutional features to address the aforementioned limitations. Our goal is to achieve scalable, fast message passing and low-cost context-aware feature augmentation for visual recognition tasks. To this end, we develop a novel graph neural network module, inspired by the latent variable graphical models (Koller et al., 2009), for encoding long-range dependency between features. Specifically, instead of propagating context information on a fully-connected graph, we introduce an augmented graph with a set of latent nodes that are connected to convolutional features and themselves. Typically the dimension of the latent space is much lower than the convolutional features, which enable us to design an efficient message passing procedure to compute non-local context information for feature nodes through the ‘global’ latent space. Figure. 1 shows an overview of our model structure and inference process.

Our message passing algorithm consists of three steps in a latent graph network: each latent node first collect messages from convolutional feature nodes, which is followed by message propagation among latent nodes, and finally the messages from latent nodes are propagated back to the feature nodes. Mathematically, our algorithm is equivalent to representing the graph affinity matrix of the fully-connected feature graph as a mixture of low-rank kernel matrices defined on convolutional features. Such equivalence allows us to introduce a parametrized mixture of low-rank matrices to encode a rich set of non-local relations and an end-to-end task-driven training strategy to learn the relations and feature representation jointly. Moreover, we can easily stack multiple layers of the basic latent graph network modules in any deep ConvNet architecture.

We validate our model on two typical classes of feature graphs in vision tasks: the first is grid-like convolutional feature graphs that are widely adopted in object detection and instance segmentation; and the second is irregular-shaped feature graphs for point cloud semantic segmentation. Extensive experiment evaluations show that our model outperforms the baselines with a large margin while maintaining a low computation cost, indicating its broad applicability in a variety of vision tasks.

The main contributions of this work are summarized as the following:

- We propose an efficient feature augmentation framework to incorporate global context and long range de-

pendency through a novel graph neural network for visual recognition tasks.

- We develop a flexible representation for non-local relations based on mixture of low-rank kernel matrices for fast message passing, which can be learned jointly with convolutional features with task-aware losses.
- Our model has a modularized design, which can be easily incorporated into any layer in deep ConvNets, and stacked into multiple blocks with different computation cost constraints.

2. Related Work

Context-aware Representation Learning While deep convolution networks are capable of extracting a hierarchical feature representation from an image, the effective receptive fields of their neurons are typically limited to local regions and unable to capture long-range dependencies (Luo et al., 2016; Chen et al., 2017).

There have been a large body of works aiming to incorporate context information into existing deep ConvNet models. One type of strategies tends to aggregate context information through the image pyramid techniques. He et al. (2014) propose to use ‘spatial pyramid pooling’ to aggregates local features for object detection. Chen et al. (2018) introduce atrous spatial pyramid pooling with multi-scale dilation convolution for richer contextual information. Zhao et al. (2017) introduce pyramid pooling to capture feature context. However, these pyramid pooling-based methods typically reply on homogeneous contextual information and lack the capacity to capture pixel-wise relationship. Enlarging the receptive field explicitly is also a common strategy. Yu & Koltun (2016) explore dilated convolution for large receptive field, and Dai et al. (2017b) improve the standard convolution operator to handle the deformation and varying object scales. However, these methods are still unable to capture non-local relations as the receptive fields mostly cover local neighborhood and short-range contextual information.

Region-based methods exploit image regions for context reasoning and visual recognition. Gould et al. (2009) propose a hierarchical region-based approach to reason about pixels, regions and objects in a coherent probabilistic model. Russell et al. (2009); Li & Gupta (2018) introduce flat graph structures—with regions as vertices and the similarity between regions as edges for recognition and context modeling. However, region generation is challenging but essential in this paradigm, which requires careful tuning and extra efforts to design.

Graph Neural Networks Our work is related to learning deep networks on graph-structured data. The GNN was first proposed in (Gori et al., 2005; Scarselli et al., 2009), which is a trainable recurrent message passing process. Before

the GNN, graphical models can also model the long-range dependencies such as the conditional random field (He et al., 2004), and it are exploited as the post-processing in the context of deep Convnets (Chen et al., 2018; Zheng et al., 2015). Graph neural network are natural generalizations of convolutional neural networks to non-Euclidean graphs. Bruna et al. (2014) and Henaff et al. (2015) propose to learn smooth spectral multipliers of the graph Laplacian. Kipf & Welling (2017) introduce learning polynomials of the graph laplacian instead of compute eigenvectors to alleviate the computational bottleneck. Kearnes et al. (2016); Gilmer et al. (2017) generalize the GNN and also learn edge features from the current node hidden representations.

Latent Variable Models and Kernel Learning The idea of using latent variables to capture long-range dependency has been widely explored in the literature of probabilistic graphical model (Koller et al., 2009). Our graph structure resembles the Restricted Boltzmann Machines with lateral connections (Osindero & Hinton, 2008). However, we mainly use the latent graph for feature computation instead of defining a distribution over random variables.

Our work is also inspired from low-rank approximation in kernel methods (Schölkopf et al., 2002). One limitation of kernel methods is their high computation cost, which is at least quadratic in the number of training examples, due to the kernel matrix calculation. To avoid computing kernel matrix, one common approach is to based on approximate kernel learning, which generates a vector representation of data that approximates kernel similarity between any two data points. The most well known approaches in this category are *random Fourier features* (Rahimi & Recht, 2008; 2009) and *the Nyström method* (Williams & Seeger, 2001; Drineas & Mahoney, 2005). Randon Fourier features is an approach to scaling up kernel methods for shift-invariant kernels. Nyström method that approximate the kernel matrix by a low rank matrix.

Visual Recognition Tasks Visual recognition community has witnessed tremendous development with deep convolutional neural network. Object detection, instance segmentation, point cloud semantic segmentation and many other challenging visual recognition tasks have been actively studied in plenty of recent works.

For image object detection, a common branch is R-CNN (Girshick et al., 2014) based methods, which is detecting objects based on region proposals. Fast R-CNN (Girshick, 2015) was proposed to reduce the computation cost of R-CNN by extracting the whole image feature onceRoI Align layer (He et al., 2017) is designed to address the coarse spatial quantization for improving RoI pooling further. Faster R-CNN (Ren et al., 2015) introduce a region proposal network(RPN) that share the same backbone with the detection

to build the first unified end-to-end efficient framework for object detection.

Deep learning with 3D data is a growing filed of research recently. There a series of works propose several effective architectures to process point cloud directly. PointNet (Qi et al., 2017a) is a pioneering effort to handler the unordered point data, which applies deep learning to point clouds by point-wise encoding and aggregation through global max pooling. To capture more local geometric details, PointNet++ (Qi et al., 2017b) introduce a hierarchical neural network to learn multi scale feature descriptor for the point cloud. PointCNN (Li et al., 2018) propose \mathcal{X} -conv layer instead of MLP layer to exploit canonical ordering of points. We conduct experiments both on grid-like data irregular-shaped data to validate our method is generic and effective. Superpoint Graphs (Landrieu & Simonovsky, 2018)(SPG) first partitions point cloud into superpoints and encode every super point with PointNet. The semantic labels of superpoints are predicted from the PointNet embedding of current superpoint and spatially neighboring superpoints.

3. Our Approach

Our goal is to learn a context-aware feature representation for a visual input (e.g., image or point cloud) that captures long-range dependencies between convolutional features. To this end, we adopt a graph neural network (GNN) framework to model the non-local relations of conv-features, and develop an efficient message passing algorithm to estimate non-local context representation of each conv-feature. Unlike previous non-local relation networks (Wang et al., 2018; Yue et al., 2018), our strategy proposes a novel graph structure for fast message updates and a flexible graph affinity representation for learning the non-local relations in a task-driven manner.

Below we start with a brief introduction to the GNN framework (Wang et al., 2018; Velickovic et al., 2018) for estimating non-local context information and motivate our network design in Sec. 3.1. We then present our network structure and message passing algorithm in Sec. 3.2, followed by a low-rank kernel learning formulation in Sec. 3.3. Finally, we describe feature augmentation strategies that combine our GNN module with deep ConvNets for visual recognition tasks in Sec. 3.4.

3.1. Non-local Graph Neural Networks

We adopt the formulation in (Wang et al., 2018) for the non-local context feature estimation and first introduce the fully-connected graph neural network for this task. Formally, let $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]^T$ be a set of convolutional features, where $\mathbf{x}_i \in \mathbb{R}^c$ is a c -channel feature vector and i indexes the spatial location of the feature vector. Taking image

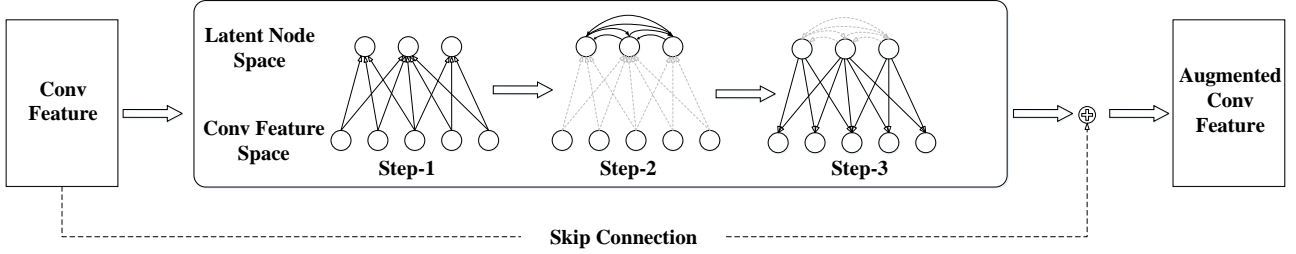


Figure 1. Illustration of the latent graph network. Step-1 shows the message passing from convolutional feature space the latent node space, and message passing between pairwise nodes in latent node space is step 2. Step-3 indicates propagated message back to the convolutional feature nodes.

features as an example, we have a conv-feature map defined on a 2D grid with the size of $c \times h \times w$, where h , w and c are the height, width and number of channels, respectively. Here $N = w \cdot h$ and for \mathbf{x}_i , i indexes the spatial location on the feature map.

We then build a fully-connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with N nodes $v_i \in \mathcal{V}$, and edges $(v_i, v_j) \in \mathcal{E}, \forall i < j$, to represent the non-local relations between features. The graph neural network assigns \mathbf{x}_i as the input to the node v_i and computes the feature representation of each node through an iterative message passing process. Each iteration updates the node features as follows,

$$\tilde{\mathbf{x}}_i = h \left(\frac{1}{Z_i(\mathbf{X})} \sum_{j=1}^N g(\mathbf{x}_i, \mathbf{x}_j) \mathbf{W}^T \mathbf{x}_j \right) \quad (1)$$

where $\tilde{\mathbf{x}}_i$ represents the updated feature representation at node i , h is an element-wise activation function (e.g., ReLU). g is a kernel function encoding relations between two feature vectors \mathbf{x}_i and \mathbf{x}_j , and $Z_i(\mathbf{X})$ is a normalization factor for node i . $\mathbf{W} \in \mathbb{R}^{c \times c}$ is the weight matrix defining a linear mapping to encode the message from node i . Note that we can write the updating equation Eq (1) in a matrix form:

$$\tilde{\mathbf{X}} = h(\mathbf{A}(\mathbf{X})\mathbf{X}\mathbf{W}) \quad (2)$$

where $\mathbf{A}(\mathbf{X}) \in \mathbb{R}^{N \times N}$ denotes the affinity matrix of the graph in which $\mathbf{A}_{i,j} = \frac{1}{Z_i(\mathbf{X})} g(\mathbf{x}_i, \mathbf{x}_j)$. Moreover, we can easily extend this updating procedure to multiple iterations by unrolling the message passing into a multi-layer network (Gilmer et al., 2017). We will focus the single iteration setting in the remaining of this section for notation clarity.

As the updated $\tilde{\mathbf{x}}_i$ integrates context information from the entire feature set \mathbf{X} through message passing, we can use it as an estimate of the non-local context for feature \mathbf{x}_i . For visual recognition, the common strategy for defining the graph affinity matrix is based on similarity of neighboring node features (denoted as \mathbf{A}_{sim}) or the random walk graph Laplacian of node features (denoted as \mathbf{A}_{lap}). Specifically,

for \mathbf{A}_{sim} , we define $g(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$ and $Z_i(\mathbf{X}) = 1$, while for \mathbf{A}_{lap} , we have the same g function but set $Z_i(\mathbf{X}) = \sum_j g(\mathbf{x}_i, \mathbf{x}_j)$. These two graph affinity matrices can be written compactly as follows,

$$\mathbf{A}_{\text{sim}} = \mathbf{M} = \mathbf{X}\mathbf{X}^T \quad (3)$$

$$\mathbf{A}_{\text{lap}} = \mathbf{D}^{-1}\mathbf{M} \quad (4)$$

where $\mathbf{M} \in \mathbb{R}^{N \times N}$, and \mathbf{D} is the diagonal degree matrix in which $\mathbf{D}_{ii} = \sum_j \mathbf{M}_{ij}$.

While the non-local graph neural networks are capable of effectively capturing long-range dependency in conv-features, they suffer from two inherent limitations in the context of visual recognition. First, the message passing in Eqn (2) involves computing the matrix product $\mathbf{A}(\mathbf{X})\mathbf{X}$, which has a quadratic complexity in terms of size of the graph ($\mathcal{O}(N^2 \cdot c)$). Hence this inference process is computationally prohibitive for large-sized feature graphs, which restricts its usage in real-world vision tasks. Moreover, the graph edge affinities, which capture relations between two node features, are typically defined by a simple pairwise function (with possible normalization) and hence has a limited capacity in modeling complex relationships. To address those issues, we propose a new graph neural network model in the following subsections.

3.2. Latent GNN for Non-local Feature Context

Instead of relying on fully-connected graph defined on input features, we now introduce a new graph structure to efficiently encode the long-range dependencies. To achieve this, we first augment the original feature nodes with a set of additional latent nodes and then connect those nodes in a structural manner. Our design is inspired by the undirected graph models with latent representations, such as the Restricted Boltzmann Machine (RBMs) (Nair & Hinton, 2010). The newly-designed neural network is capable of encoding non-local relations through a shared latent space and admits efficient message passing due to its sparse connections.

Specifically, we introduce a set of latent features $\mathbf{Z} = [\mathbf{z}_1, \dots, \mathbf{z}_d]^\top$ where $\mathbf{z}_i \in \mathbb{R}^c$ is a c -channel feature vector and the number of latent features $d \ll N$. We then build an augmented graph $\mathcal{G}_L = (\mathcal{V}_L, \mathcal{E}_L)$ with $N + d$ nodes, $\mathcal{V}_L = \mathcal{V} \cup \mathcal{V}_h$. Here \mathcal{V} corresponds to the input conv-features and \mathcal{V}_h is associated with the latent features. We also define the graph connectivity by two subsets of graph edges, $\mathcal{E}_L = \mathcal{E}_v \cup \mathcal{E}_h$, where \mathcal{E}_v denote the connections between \mathcal{V} and \mathcal{V}_h while \mathcal{E}_h are the graph edges within \mathcal{V}_h .

Our latent graph neural network takes the conv-features \mathbf{X} as inputs to the nodes in \mathcal{V} and computes context-aware representations of each node through an iterative message passing with a specific scheduling scheme. Concretely, each iteration consists of three steps for updating the node features (see Figure.1 for the illustration):

Step 1. *Visible-to-latent propagation*. We update the latent features by propagating messages from the conv-feature nodes to the latent nodes:

$$\mathbf{z}_k = \sum_{j=1}^N \psi(\mathbf{x}_j, \theta_k) \mathbf{W}^\top \mathbf{x}_j, \quad 1 \leq k \leq d, \quad (5)$$

where $\psi(\mathbf{x}_j, \theta_k)$ encodes the affinity between the feature node \mathbf{x}_j and the latent node \mathbf{z}_k , and θ_k is the parameter of the affinity function.

Step 2. *Latent-to-latent propagation*. We refine the latent features by propagating messages in the fully-connected latent subgraph $(\mathcal{V}_h, \mathcal{E}_h)$:

$$\tilde{\mathbf{z}}_k = \sum_{j=1}^d f(\phi_k, \phi_j, \mathbf{X}) \mathbf{z}_j, \quad 1 \leq k \leq d, \quad (6)$$

where $f(\phi_k, \phi_j, \mathbf{X})$ represents the data-dependent pairwise relations between two latent nodes, and ϕ_k, ϕ_j are the parameters for node k and j respectively.

Step 3. *Latent-to-visible propagation*. We update the conv-features by propagating the messages from the latent nodes back to the conv-feature nodes:

$$\tilde{\mathbf{x}}_i = h \left(\sum_{k=1}^d \psi(\mathbf{x}_i, \theta_k) \tilde{\mathbf{z}}_k \right), \quad 1 \leq i \leq N, \quad (7)$$

where h is an element-wise activation function. And the output of the final step is used as an estimate of the non-local context for feature \mathbf{x}_i .

3.3. An Equivalent Low-rank Kernel Representation

To gain more insight on the latent GNN model, we provide an alternative interpretation of our message passing algorithm in this subsection. We will show that our non-local GNN model is equivalent to using a low-rank representation for the affinity matrix in the fully-connected

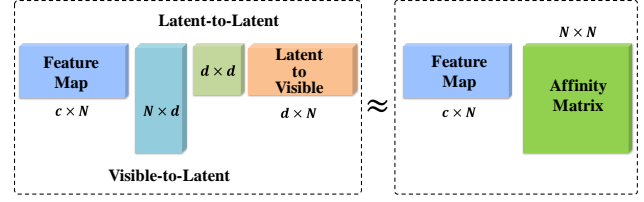


Figure 2. Computation diagram of the feature augmentation.

GNN model. To see this, we first represent our message passing algorithm in a matrix form. Specifically, denote $\psi(\mathbf{x}_i) = [\psi(\mathbf{x}_i, \theta_1), \dots, \psi(\mathbf{x}_i, \theta_d)]^\top$, we can write Eqn (5) in its matrix form as follows,

$$\mathbf{Z} = \Psi(\mathbf{X})^\top \mathbf{X} \mathbf{W} \quad (8)$$

$$\Psi(\mathbf{X}) = [\psi(\mathbf{x}_1), \dots, \psi(\mathbf{x}_N)]^\top \in \mathbb{R}^{N \times d}. \quad (9)$$

Similarly, let $\mathbf{F}_\mathbf{X} = [f(\phi_i, \phi_j, \mathbf{X})]_{d \times d}$, and the remaining two steps of the message passing can be written as:

$$\tilde{\mathbf{Z}} = \mathbf{F}_\mathbf{X} \mathbf{Z}, \quad (10)$$

$$\tilde{\mathbf{X}} = h \left(\Psi(\mathbf{X}) \tilde{\mathbf{Z}} \right). \quad (11)$$

Putting them together, we have the following matrix form for our message passing updating equations:

$$\tilde{\mathbf{X}} = h \left(\Psi(\mathbf{X}) \mathbf{F}_\mathbf{X} \Psi(\mathbf{X})^\top \mathbf{X} \mathbf{W} \right) \quad (12)$$

Comparing the above equation with Eqn (2), we can see that our latent GNN can be viewed as a fully-connected graph neural network where

$$\mathbf{A}(\mathbf{X}) = \Psi(\mathbf{X}) \mathbf{F}_\mathbf{X} \Psi(\mathbf{X})^\top, \quad (13)$$

which has a low-rank of d instead of its full size N .

Fast Message Passing One advantage of this low-rank representation is that it allows us to compute the matrix multiplications in Eqn (12) from right to left as in the message passing algorithm. This process does not require explicitly maintaining the full affinity matrix and significantly reduces the computation cost. Our method only has a linear complexity in terms of graph size, i.e., $\mathcal{O}(N \cdot c \cdot d)$ and achieves N/d times speed-up compared to the non-local graph neural networks.

Learning Low-rank Adjacency Matrix While the low-rank affinity matrix leads to significant improvements in computation efficiency, it has a limited representation power to model non-local relations between features. Inspired by the multiple kernel learning, we propose to extend the basic latent graph neural network by introducing a learnable mixture of low-rank matrices for capturing more complex non-local relations.

Formally, we define a set of local affinity functions $\{\Psi^{(m)}(\mathbf{X}), \mathbf{F}_{\mathbf{X}}^{(m)}\}$, which are used to construct the full affinity matrix as follows,

$$\mathbf{A}(\mathbf{X}) = \sum_m w_m \Psi^{(m)}(\mathbf{X}) \mathbf{F}_{\mathbf{X}}^{(m)} \Psi^{(m)}(\mathbf{X})^\top, \quad (14)$$

where w_m are the weights for each low-rank matrix. It is straightforward to extend the message passing algorithm to the case of mixture of low-rank kernels, and we can also jointly learn the mixture weights with the graph network.

3.4. Feature Augmentation for Visual Recognition

Given the context feature $\tilde{\mathbf{X}}$, we adopt a simple feature augmentation strategy to integrate the context representation with the original conv-features. Specifically, we use a weighted summation of two features as in the Residual Network (He et al., 2016):

$$\mathbf{X}_{\text{aug}} = \lambda \tilde{\mathbf{X}} + \mathbf{X} \quad (15)$$

where λ is a scaling parameter.

As our latent graph neural network only takes a set of conv-features as input and is fully differentiable, it can be easily integrated with the existing deep ConvNets and applied to a variety of visual cognition tasks. In practice, we usually stack several graph convolutional layers into one feature augmentation module. In this work, we evaluate our feature augmentation module on several visual recognition tasks, including object detection and segmentation, and point cloud semantic segmentation. For object detection and instance segmentation, we insert our LatentGNN into the backbone network, at the end of each ResNet layer. For point cloud semantic segmentation, the LatentGNN module is inserted after each point feature extraction block.

4. Experiments

In this section, we conduct a series of experiments to validate the effectiveness of our method. We evaluate our model on three visual recognition tasks, including object detection and instance segmentation on the MSCOCO 2017 dataset (Lin et al., 2014), and point cloud semantic segmentation on the ScanNet dataset (Dai et al., 2017a). In each task, we compare our approach to the state-of-the-art non-local methods in terms of prediction accuracy and computational complexity.

Below we present our experimental analysis on the MSCOCO dataset in Sec. 4.1, followed by our results on the ScanNet dataset in the Sec. 4.2. In both tasks, we first introduce implementation details of our method, and then report the ablative study and comparisons of quantitative results on each dataset.

4.1. Object Detection & Segmentation on MSCOCO

4.1.1. NETWORK ARCHITECTURE

To utilize our LatentGNN in the task of object detection and instance segmentation, we adopt the standard setup of MaskRCNN (He et al., 2017), in which ResNet/ResNeXt with FPN are used as the backbone architecture. We integrate our network module with the backbone network at different stages in the following experiments, and denote them as **+LatentGNN**. For ResNet backbones, we insert a LatentGNN into different residual blocks (Res3, Res4 or Res5) and refer them to as stage-3,4,5 respectively.

To build a basic LatentGNN module with a single kernel, we first reduce the channel dimension of feature map with a 1×1 convolutional bottleneck layer. We then use a learnable graph adjacency matrix for the message propagation from visible to latent nodes and in the reverse direction, which are implemented by a linear layer followed by a non-linear activation function (e.g. ReLU). The number of latent nodes d is different for each stage of the backbone network due to different resolution of the feature maps and we set $d = \{150, 100, 50\}$ for the stage-3, stage-4 and stage-5, respectively. Moreover, we extend the basic LatentGNN module by introducing a learnable low-rank adjacency matrix with multiple kernels. In the multi-kernel LatentGNN, we aggregate the multiple visible features generated by the *latent-to-visible propagation*, followed by a skip-connection to fuse them with the original conv feature.

For fair comparison, in addition to reporting the results in (Wang et al., 2018), we implement a Non-local Neural Network using the same backbone network and graph affinity matrix A_{sim} . Following (Wang et al., 2018), we insert the non-local block on the top of the last stage in the backbone network, which is denoted as '+NL Block'.

4.1.2. TRAINING SETTING

We train our models for object detection and instance segmentation on the MSCOCO dataset and utilize the open-source implementation (Massa & Girshick, 2018). MSCOCO dataset is a challenging dataset that contains 115K images over 80 categories for training, 5K for validation and 20K for testing. Following the training procedure in (Massa & Girshick, 2018), we resize the images so that the shorter side is 800 pixels. All experiments are conducted on 8 GPUs with 2 images per GPU (effective minibatch size 16) for 90K iterations, with a learning rate of 0.02 which is decreased by 10 at the 60K and 80K iteration. We use a weight decay of 0.0001 and momentum of 0.9. For ResNeXt, we train the model with 1 image per GPU and the same number of iterations, with a starting learning rate of 0.01. The backbone of all models are trained on ImageNet classification (Deng et al., 2009).

Model	Stage	Kernels	AP _{box}	AP ⁵⁰ _{box}	AP ⁷⁵ _{box}	AP _{sem}	AP ⁵⁰ _{sem}	AP ⁷⁵ _{sem}	FLOPS	#Params
ResNet-50 ¹	-	-	38.0	59.6	41.5	34.6	56.4	36.5	-	-
+NL Block ¹	Stage4	1	39.0	61.1	41.9	35.5	58.0	37.4	+10.67G	+ 2.09M
ResNet-50(1x) ²	-	-	37.8	59.1	41.2	34.2	55.8	36.3	-	-
+ NL Block ²	Stage4	1	38.7	60.2	42.2	35.0	57.0	37.1	+10.67G	+ 2.09M
+ LatentGNN	Stage3	1	38.2	59.7	41.7	34.7	56.3	36.8	+1.48G	+ 0.06M
+ LatentGNN	Stage4	1	39.0	60.7	42.6	35.2	57.6	37.4	+1.11G	+ 0.20M
+ LatentGNN	Stage5	1	38.8	61.0	42.0	35.0	57.6	37.0	+0.97G	+ 0.81M
+ LatentGNN	Stage345	1	39.5	61.6	43.2	35.6	58.3	37.7	+3.59G	+1.07M
ResNet-101(1x)	-	-	39.9	61.3	43.8	35.9	58.2	38.1	-	-
+ LatentGNN	Stage4	1	41.0	63.2	45.0	36.9	59.6	39.4	+1.11G	+ 0.20M
+ LatentGNN	Stage345	1	41.4	63.7	45.2	37.2	60.1	39.5	+3.59G	+1.07M
ResNeXt-101(1x)	-	-	42.1	64.1	45.9	37.8	60.3	39.5	-	-
+ LatentGNN	Stage4	1	43.0	65.3	46.9	38.5	61.9	40.9	+1.11G	+ 0.20M
+ LatentGNN	Stage345	1	43.2	65.6	47.2	38.8	62.1	41.0	+3.59G	+1.07M

Table 1. Performance of Non-local NNs and LatentGNNs with different backbone and different stages on MSCOCO 2017 validation set. All experiments are conducted with 1x learning rate schedule as in (He et al., 2017).

4.1.3. QUANTITATIVE RESULTS AND ANALYSIS

The ablation study is conducted on the MSCOCO validation set. All standard MSCOCO metrics including AP, AP⁵⁰, AP⁷⁵ for both bounding boxes and segmentation masks are reported below.

Different Stage We first investigate at which stage we should add the LatentGNN layer to augment the feature with contextual information. We insert one LatentGNN layer for different residual stages, based on the ResNet-50 backbone, to build the network. The quantitative results shown in Table.1 demonstrate the effects of our LatentGNN layer for each of and all three stages.

We note that incorporating LatentGNN layer into the stage3 has a small gain compared with the baseline method as the lower-level feature maps lack semantic information. Meanwhile, the stage4 equipped with a single LatentGNN layer could achieve 1.2 points improvement for box AP and 1.0 improvements for segmentation AP. Moreover, it is straightforward to incorporate our LatentGNN layer into multiple residual stages in order to augment the feature maps with multiscale contextual information. While easy to implement in our framework due to its efficient computation, it is not applicable to the Non-local Neural Networks. Our results show that adding LatentGNN module to the last three stages(3,4 and 5) together can achieve 39.5 for box mAP and 35.6 for segmentation mAP, which has 1.7 and 1.4 performance gain, respectively. The quantitative results indicate that our LatentGNN can achieve a large margin improvement with less FLOPS and fewer learnable parameters compared to the standard non-local block.

¹Results reported by (Wang et al., 2018).

²Our implementation based on (Massa & Girshick, 2018).

Different Backbones We validate our method on different backbone networks, ranging from ResNet-50, ResNet-101 to ResNeXt-101 (Xie et al., 2017). It is worth noting that even when adopting stronger backbones, the gain of LatentGNN compared to the baseline is still significant, which demonstrates that our LatentGNN is complementary to the capacity of current models. Equipped with different backbone, we also follow the same strategy to insert the LatentGNN layer into each residual stage. With feature augmentation on the last three stages, we can achieve 1.5 AP_{box} and 1.3 AP_{box} improvement with ResNet-101, and 1.1 AP_{box} and 1.0 AP_{box} improvement with ResNeXt-101.

Mixture of Low-rank Adjacency Matrices Using single kernel usually has limited representation power as aforementioned, and hence we also investigate the LatentGNN with a mixture of low-rank matrices for capturing the complex non-local relations. We conduct an ablation study to explore the influence of multiple low-rank adjacency matrices and report the performance in Table.2. The quantitative performance indicates the mixture model is more powerful to represent the semantic context than the single kernel, and using three low-rank adjacency matrices can achieve 1.4 AP_{box} and 1.2 AP_{box} gain compared to the baseline. However, we note that inserting such mixture LatentGNN module into multiple stages achieves little gain, possibly due to the global context encoded by the multi-stage model.

4.2. Point Cloud Segmentation on ScanNet

4.2.1. NETWORK ARCHITECTURE

We also validate our method on non-grid data, and conduct 3D point cloud semantic segmentation experiments on the ScanNet dataset, a 3D indoor scene benchmark with 1513 scanned and reconstructed indoor scenes in total. We fol-

LatentGNN: Learning Efficient Non-local Relations for Visual Recognition

Model	Stage	Kernels	AP _{box}	AP _{box} ⁵⁰	AP _{box} ⁷⁵	AP _{sem}	AP _{sem} ⁵⁰	AP _{sem} ⁷⁵	FLOPS	#Params
ResNet-50(1x)	-	-	37.8	59.1	41.2	34.2	55.8	36.3	-	-
+LatentGNN	Stage4	1	39.0	60.7	42.6	35.2	57.6	37.4	+1.11G	+ 0.20M
	Stage4	2	39.0	60.7	42.7	35.3	57.6	37.6	+1.30G	+ 0.29M
	Stage4	3	39.2	61.0	42.8	35.4	57.6	37.7	+1.48G	+0.38M
+LatentGNN	Stage345	1	39.5	61.6	43.2	35.6	58.3	37.7	+3.59G	+1.07M
	Stage345	3	39.5	61.7	43.3	35.7	58.4	37.8	+5.13G	+1.89M

Table 2. Performance of LatentGNN with a mixture of low-rank adjacency matrices on the MSCOCO validation set.

Model	Kernels	Scale	Pixel Accuracy	Voxel Accuracy	Class Pixel Accuracy	Class Voxel Accuracy	FLOPS	#Params
3DCNN(Dai et al., 2017a)	-	-	-	73.0	-	-	-	-
PointNet(Qi et al., 2017a)	-	-	-	73.9	-	-	-	-
PointCNN(Li et al., 2018)	-	-	85.1	-	-	-	-	-
PointNet++(Qi et al., 2017b)	-	Single Scale	81.5	83.2	51.7	53.1	-	-
PointNet++(Qi et al., 2017b)	-	Multi Scale	-	84.5	-	-	-	-
+NL Block	1	Single Scale	82.3	84.0	53.1	54.5	+31M	+0.70M
+LatentGNN	1	Single Scale	82.6	84.2	53.2	54.6	+15M	+0.31M
+LatentGNN	3	Single Scale	83.7	85.2	56.0	57.6	+30M	+0.54M

Table 3. Performance of LatentGNN on ScanNet based on PointNet++.

low the PointNet++ framework (Qi et al., 2017b), which introduces a hierarchical neural network to learn multi-scale feature descriptor for the point cloud. The PointNet++ uses a *set abstraction module* (SA) as encoder and a *feature propagation module* (FP) as decoder, which implement down-sampling and up-sampling of point cloud respectively. In particular, the set abstraction module (SA) learns local feature with increasing contextual scales and generates semantic feature of the point cloud. Hence, we insert the LatentGNN module at the end of each set abstraction layer to augment the feature with long-range dependencies, with latent dimension of {80,40,20,10} for four set abstraction layers, respectively.

4.2.2. TRAINING SETTING

We implement a multi-GPU version of the PointNet++ and conduct a series of experiments on ScanNet dataset for the task of semantic segmentation, which is to predict semantic object label for points in indoor scans. We use 1201 scenes for training and 312 scenes for test. We remove RGB information in all our experiments and convert point cloud label prediction into voxel labeling following (Dai et al., 2017a). For training, we take the preprocessed ScanNet data with 8192 points per data chunk. Specifically, all models are trained with a batch-size at 64 on 4 GPUs and the mini-batch size is set to 16 per GPU. We train our models for 200 epochs with Adam as optimizer, starting from the learning rate at 0.001 and reduce it using exponential schedule with the decay rate at 0.7. We report the average voxel classification accuracy as the evaluation metric.

4.2.3. RESULTS

We compare our model with the PointNet++ baseline and the Non-local Network module, and the performance is reported in Table. 3. Our proposed mixture LatentGNN model achieves the best performance in terms of per-voxel accuracy. The quantitative results show our LatentGNN achieves a large-margin performance gain at 2% on voxel accuracy compared with PointNet++. We note that we do not use the multi-scale grouping(MSG) in *set abstraction module*, which addresses the issue of non-uniform sampling density and may further improve the performance. Our single-kernel LatentGNN module also outperforms the Non-local Network module with much fewer parameters and merely half of computation cost. The results demonstrate the advantage of our efficient feature augmentation module.

5. Conclusion

In this paper, we have presented an efficient and flexible context-aware feature augmentation model for visual recognition tasks. We introduce a novel graph neural network with a set of latent nodes to capture long-range dependencies. The sparse structure of this graph enable us to develop a scalable and efficient message passing algorithm for computing the context features. In addition, by introducing a grouping structure in the latent nodes, we are able to build a rich model of pairwise relations. We have evaluated our method on grid-like conv-feature maps for object localization and irregular point clouds for semantic segmentation. Our results demonstrate that our approach outperforms the baseline and other non-local method, and in particular, with significant lower computation cost in model inference.

Acknowledgement

The authors would like to acknowledge the support of a NSFC Grant No. 61703195 and a NSF Shanghai Grant No. 18ZR1425100.

References

- Bruna, J., Zaremba, W., Szlam, A., and LeCun, Y. Spectral networks and locally connected networks on graphs. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.
- Byeon, W., Breuel, T. M., Raue, F., and Liwicki, M. Scene labeling with lstm recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint*, 2017.
- Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.
- Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., and Nießner, M. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017a.
- Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., and Wei, Y. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017b.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.
- Drineas, P. and Mahoney, M. W. On the nyström method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 2005.
- Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. *arXiv preprint*, 2017.
- Girshick, R. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.
- Girshick, R., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- Gori, M., Monfardini, G., and Scarselli, F. A new model for learning in graph domains. In *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN)*, 2005.
- Gould, S., Gao, T., and Koller, D. Region-based segmentation and object detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2009.
- He, K., Zhang, X., Ren, S., and Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European Conference on Computer Vision (ECCV)*, 2014.
- He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.
- He, X., Zemel, R. S., and Carreira-Perpiñán, M. Á. Multiscale conditional random fields for image labeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.
- Henaff, M., Bruna, J., and LeCun, Y. Deep convolutional networks on graph-structured data. *arXiv preprint*, 2015.
- Kearnes, S., McCloskey, K., Berndl, M., Pande, V., and Riley, P. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30(8):595–608, 2016.
- Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.
- Koller, D., Friedman, N., and Bach, F. *Probabilistic graphical models: principles and techniques*. MIT Press, 2009.
- Landrieu, L. and Simonovsky, M. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- Li, Y. and Gupta, A. Beyond grids: Learning graph representations for visual recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.
- Li, Y., Bu, R., Sun, M., Wu, W., Di, X., and Chen, B. Pointcnn: Convolution on x-transformed points. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *European conference on computer vision*, 2014.
- Luo, W., Li, Y., Urtasun, R., and Zemel, R. Understanding the effective receptive field in deep convolutional neural networks. In *Advances in Neural Information Processing Systems(NeurIPS)*, 2016.
- Massa, F. and Girshick, R. maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. <https://github.com/facebookresearch/maskrcnn-benchmark>, 2018.
- Nair, V. and Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, 2010.
- Osindero, S. and Hinton, G. E. Modeling image patches with a directed hierarchy of markov random fields. In *Advances in neural information processing systems*, pp. 1121–1128, 2008.
- Qi, C. R., Su, H., Mo, K., and Guibas, L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2017a.
- Qi, C. R., Yi, L., Su, H., and Guibas, L. J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems(NeurIPS)*, 2017b.
- Rahimi, A. and Recht, B. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems(NeurIPS)*, 2008.
- Rahimi, A. and Recht, B. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in Neural Information Processing Systems(NeurIPS)*, 2009.
- Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems(NeurIPS)*, 2015.
- Russell, C., Kohli, P., Torr, P. H., et al. Associative hierarchical crfs for object class image segmentation. In *Proceedings of the IEEE International Conference on Computer Vision(ICCV)*, 2009.
- Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE Transactions on Neural Networks*, 2009.
- Schölkopf, B., Smola, A. J., Bach, F., et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT Press, 2002.
- Shuai, B., Zuo, Z., Wang, B., and Wang, G. Scene segmentation with dag-recurrent neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.
- Strat, T. M. and Fischler, M. A. Context-based vision: recognizing objects using information from both 2 d and 3 d imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1991.
- Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. Graph attention networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.
- Wang, X., Girshick, R., Gupta, A., and He, K. Non-local neural networks. In *Proceedings of the IEEE International Conference on Computer Vision(CVPR)*, 2018.
- Williams, C. K. and Seeger, M. Using the nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems(NeurIPS)*, 2001.
- Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2017.
- Yu, F. and Koltun, V. Multi-scale context aggregation by dilated convolutions. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.
- Yue, K., Sun, M., Yuan, Y., Zhou, F., Ding, E., and Xu, F. Compact generalized non-local network. In *Advances in Neural Information Processing Systems(NeurIPS)*, 2018.
- Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. Pyramid scene parsing network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., and Torr, P. H. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international Conference on Computer Vision(ICCV)*, 2015.