LatentGNN: Learning Efficient Non-local Relations for Visual Recognition

Songyang Zhang, Shipeng Yan, Xuming He

ShanghaiTech University

Songyang Zhang
sy.zhangbuaa@gmail.com

June 13, 2019

上海科技大学 ShanghaiTech University

PLUS LAB SHANGHAITECH UNIVERSITY

ICML International Conference On Machine Learning

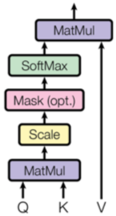
Goal & Motivation

Goal

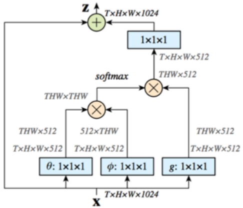
Learning efficient feature augmentation with Non-local relations for visual recognitions.

Motivation

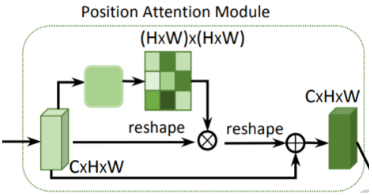
- To model the non-local feature context by a **Graph Neural Network (GNN)**.
  - **Self-attention Mechanism, Non-local network** as special examples of **Graph Neural Network** with truncated inference.
- To reduce the complexity of a fully-connected GNN by introducing a **latent representation**.



Attention is All You Need(Vaswani et al)



Non-local Network(Wang et al)

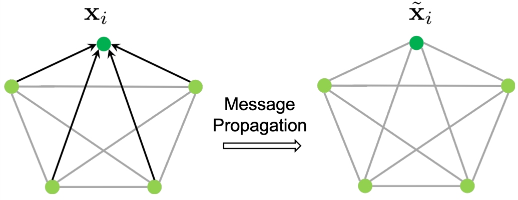


Dual Attention Network(Fu et al)

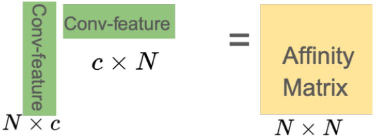
Non-local Features with GNN

Notation

- **Input**: Grid/Non-grid Conv-feature,
$$\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N]^{\mathsf{T}}, \mathbf{x}_i \in \mathbb{R}^c$$
- **Output**: Context-aware Conv-feature,
$$\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \cdots, \tilde{\mathbf{x}}_N]^{\mathsf{T}}, \tilde{\mathbf{x}}_i \in \mathbb{R}^c$$
- **Each Location**:
$$\tilde{\mathbf{x}}_i = h\left(\frac{1}{Z_i(\mathbf{X})}\sum_{j=1}^{N} g(\mathbf{x}_i, \mathbf{x}_j)\,\mathbf{W}^\top \mathbf{x}_j\right) \tag{1}$$
- **Matrix Form**:
$$\tilde{\mathbf{X}} = h(\mathbf{A}(\mathbf{X})\mathbf{X}\mathbf{W}), \quad \mathbf{X}_{\text{aug}} = \lambda \cdot \tilde{\mathbf{X}} + \mathbf{X} \tag{2}$$
  - $g(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^{\mathsf{T}}\mathbf{x}_j$: Pair-wise relations function
  - h: Element-wise activation function(ReLU)
  - $Z_i(\mathbf{X})$: Normalization factor
  - $\mathbf{W} \in \mathbb{R}^{c\times c}$: Weight matrix of the linear mapping
  - λ: Scaling parameter



Non-local features with GNN

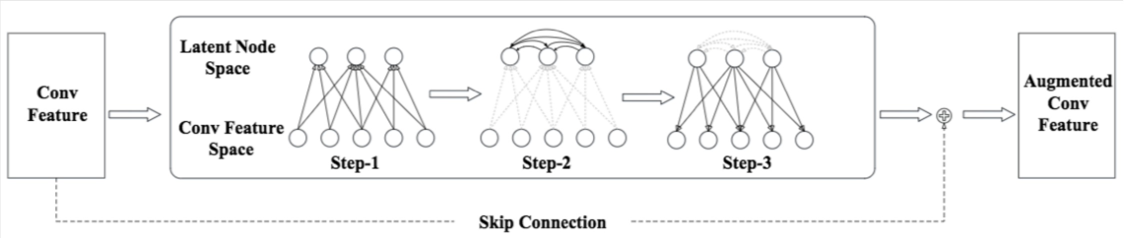


If $N = 500 \times 500$, A requires 500GB of storage!!!

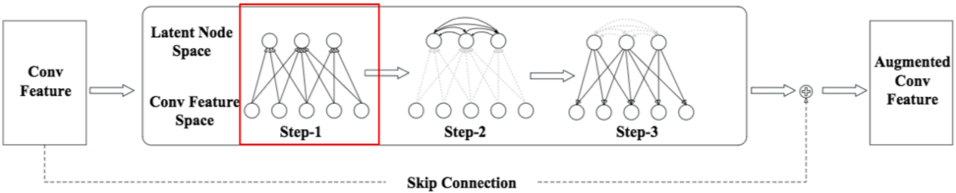
Latent Graph Neural Network

LatentGNN

- **Key Idea**: Introduce a latent space for efficient global context encoding
- Conv-feature Space: $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N]^\mathsf{T}, \mathbf{x}_i \in \mathbb{R}^c$
- Latent Space: $\mathbf{Z} = [\mathbf{z}_1, \cdots, \mathbf{z}_d]^\mathsf{T}, \mathbf{z}_i \in \mathbb{R}^c, d \ll N$



Latent Graph Neural Network



Step-1: **Visible-to-Latent** Propagation(Bipartite Graph)

- **Each Latent Node:**

$$\mathbf{z}_k = \sum_{j=1}^{N} \frac{1}{m_k(\mathbf{X})} \psi(\mathbf{x}_j, \theta_k) \mathbf{W}^{\mathsf{T}} \mathbf{x}_j, \quad 1 \leq k \leq d \tag{3}$$

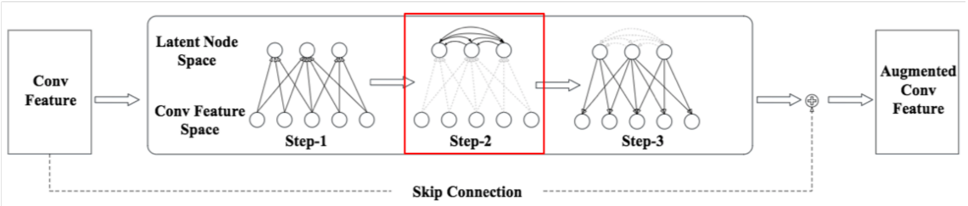
- **Matrix Form**:

$$\mathbf{Z} = \mathbf{\Psi}(\mathbf{X})^{\mathsf{T}} \mathbf{X} \mathbf{W} \tag{4}$$

$$\mathbf{\Psi}(\mathbf{X}) = [\psi(\mathbf{x}_1), \cdots, \psi(\mathbf{x}_N)]^{\mathsf{T}} \in \mathbb{R}^{N \times d}, \quad \psi(\mathbf{x}_i) = [\frac{\psi(\mathbf{x}_i, \theta_1)}{m_1(\mathbf{X})}, \cdots, \frac{\psi(\mathbf{x}_i, \theta_d)}{m_d(\mathbf{X})}]^{\mathsf{T}} \tag{5}$$

  - $\psi(\mathbf{x}_j, \theta_k)$: : encode the affinity between node $\mathbf{x}_j$ and node $\mathbf{z}_k$
  - $m_k(\mathbf{X})$: the normalization factor

Latent Graph Neural Network



Step-2: **Latent-to-Latent** Propagation(Fully-connected Graph)

- **Each Latent Node:**

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

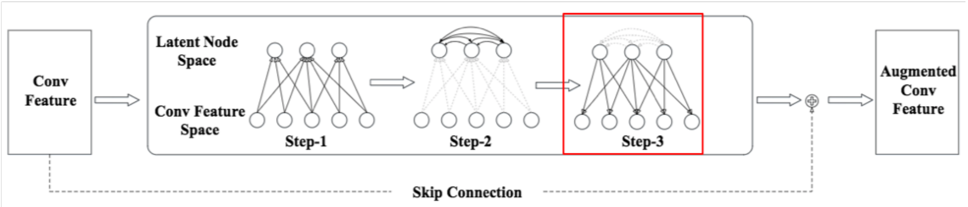
- **Matrix Form**:

$$\mathbf{F}_{\mathbf{X}} = [f(\phi_i, \phi_j, \mathbf{X})]_{d\times d} \tag{7}$$

$$\tilde{\mathbf{Z}} = \mathbf{F}_{\mathbf{X}}\mathbf{Z} \tag{8}$$

  - $f(\phi_k, \phi_j, \mathbf{X})$: data-dependent pair-wise relations between two latent nodes

Latent Graph Neural Network



Step-3: **Latent-to-Visible** Propagation(Bipartite Graph)

- **Each Visible Node:**

$$\tilde{\mathbf{x}}_i = h\left(\sum_{k=1}^{d} \psi(\mathbf{x}_i, \theta_k)\tilde{\mathbf{z}}_k\right), \quad 1 \le i \le N \tag{9}$$

- **Matrix Form**:

$$\tilde{\mathbf{X}} = h\left(\mathbf{\Psi}(\mathbf{X})\tilde{\mathbf{Z}}\right) \tag{10}$$

LatentGNN vs. GNN

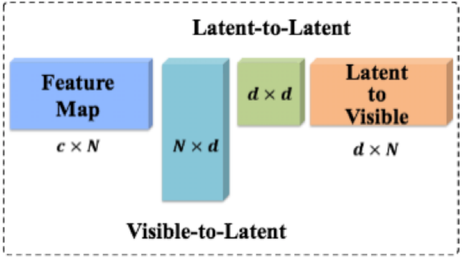
Overall Process

LatentGNN

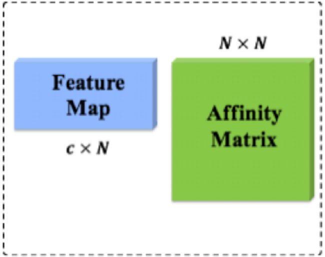
- $\tilde{\mathbf{X}} = h\left(\mathbf{\Psi}(\mathbf{X})\mathbf{F}_{\mathbf{X}}\mathbf{\Psi}(\mathbf{X})^{\mathsf{T}}\mathbf{X}\mathbf{W}\right)$
- $\mathbf{X}_{\text{aug}} = \lambda \cdot \tilde{\mathbf{X}} + \mathbf{X}$
- $\mathbf{A}(\mathbf{X}) = \mathbf{\Psi}(\mathbf{X})\mathbf{F}_{\mathbf{X}}\mathbf{\Psi}(\mathbf{X})^{\mathsf{T}}$

GNN

- $\tilde{\mathbf{X}} = h\left(\mathbf{A}(\mathbf{X})\mathbf{X}\mathbf{W}\right)$
- $\mathbf{X}_{\text{aug}} = \lambda \cdot \tilde{\mathbf{X}} + \mathbf{X}$
- $\mathbf{A}_{i,j} = \frac{1}{Z_i(\mathbf{X})} g(\mathbf{x}_i, \mathbf{x}_j), \mathbf{A}(\mathbf{X}) \in \mathbb{R}^{N \times N}$



$\approx$



$O(N \cdot d)$ $O(N \cdot N)$

Experimental Results

Grid Data: Object Detection/Instance Segmentation on MSCOCO

- **+NLBlock**: insert the non-local block in the last stage of the backbone.
- **+LatentGNN**: Integrate LatentGNN with the backbone at different stages.

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

Experimental Results

Grid Data: Ablation Study on MSCOCO

- Effects of different backbone networks.
- A mixture of low-rank matrices.

Model	Stage	Kernels	AP_{box}	AP^{50}_{box}	AP^{75}_{box}	AP_{sem}	AP^{50}_{sem}	AP^{75}_{sem}	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

Non-grid Data: Point Cloud Semantic Segmentation on ScanNet

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

Take Home Message

LatentGNN

- A novel graph neural network for efficient non-local relations learning.
  - Introduce a latent space for efficient message propagation
- Our model has a modularized design, which can be easily incorporated into any layer in deep ConvNet

Paper

Code(available soon)

Poster:
Thu, Jun 13, 2019
Pacific Ballroom #28

Thanks!