GCN and NAS in Semantic Segmentation

Speaker：Xia Li

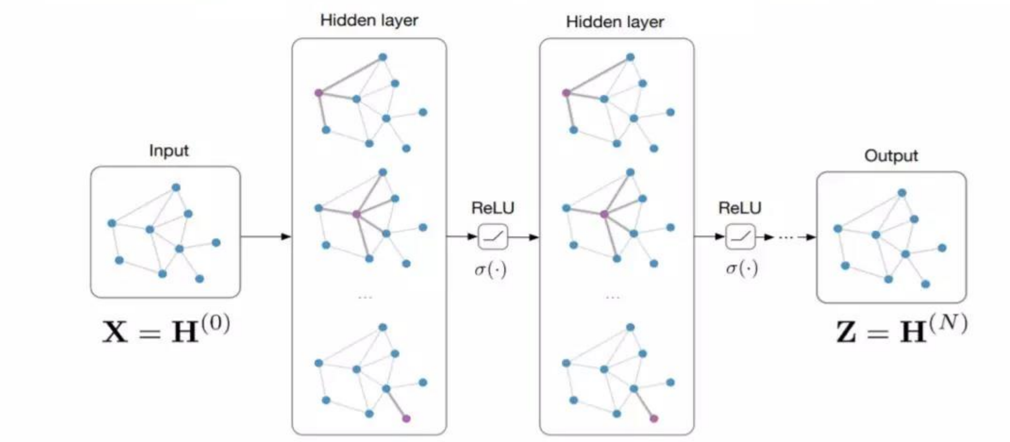
Date： 7th, Apr, 2019

Outline

1. GCN in Semantic Segmentation
    1. A^2Net
    2. GloRe
    3. SGR
    4. GCU
2. NAS in Semantic Segmentation
    1. DPC
    2. Auto-DeepLab

1. GCN in Semantic Segmentation

Input: Feature matrix $\mathbf{X} \in \mathbb{R}^{N \times E}$, preprocessed adjacency matrix $\hat{\mathbf{A}}$



$$\mathbf{H}^{(l+1)} = \sigma\left(\mathbf{A}^{(l)}\mathbf{H}^{(l)}\mathbf{W}^{(l)}\right)$$

Or $$\mathbf{H}^{(l+1)} = \sigma\left(\left(I - \hat{\mathbf{A}}^{(l)}\right)\mathbf{H}^{(l)}\mathbf{W}^{(l)}\right)$$

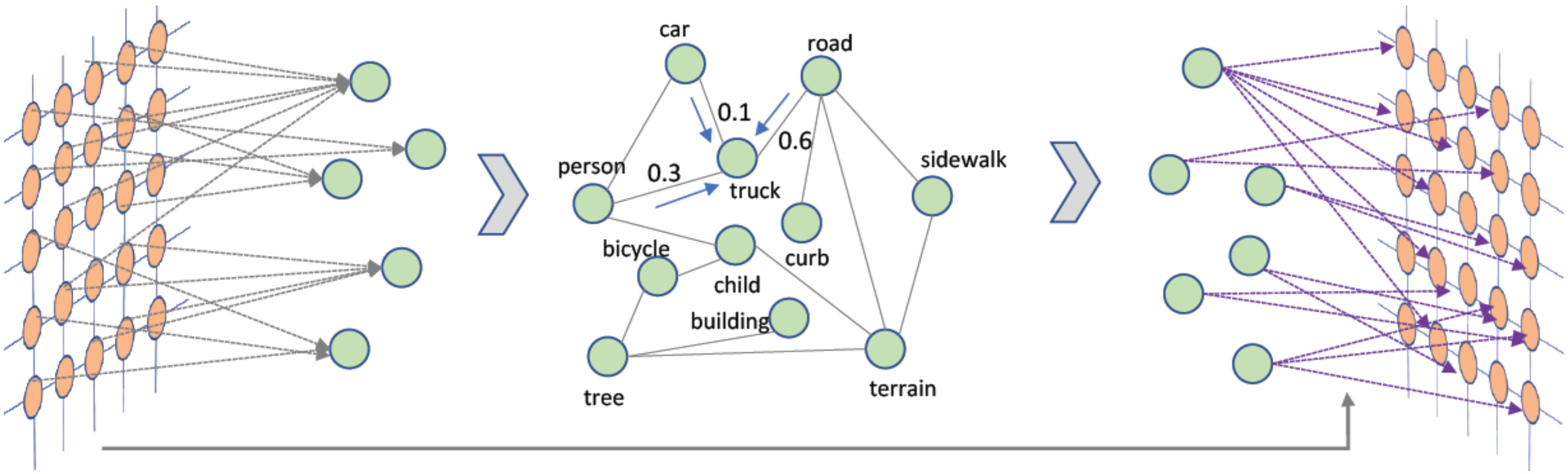
Where $$\hat{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}, D = diag\left(A \times \vec{1}\right)$$

Characteristics of GCN

1. Non-grid structure
2. Well-defined adjacency matrix A

How to apply on semantic segmentation task?

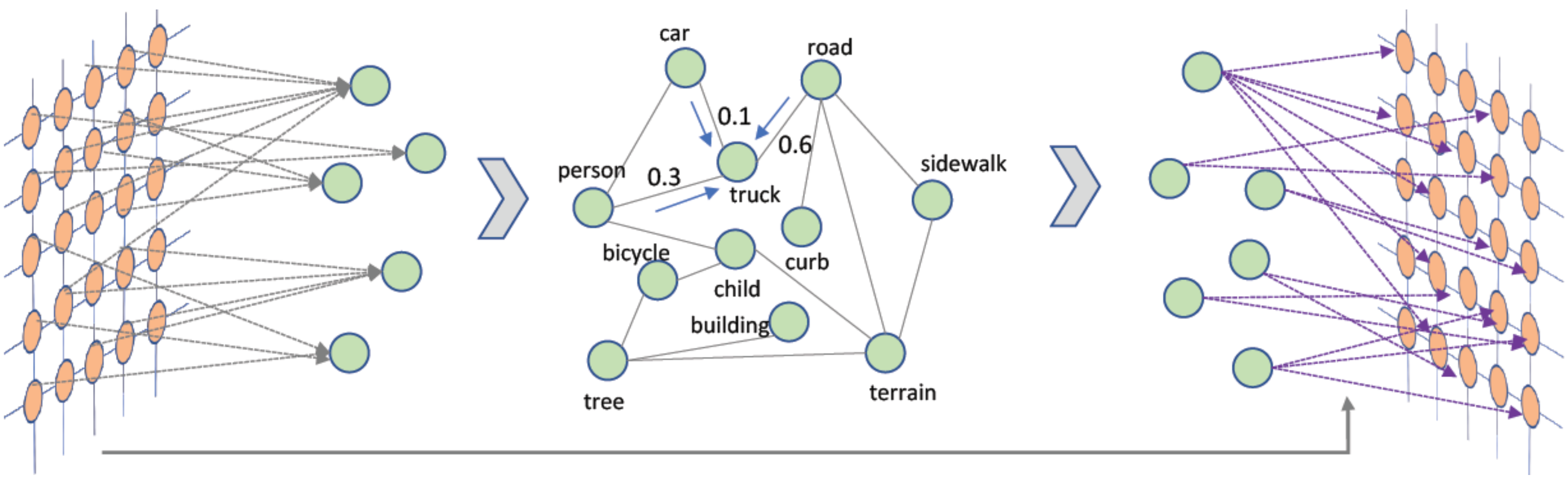
1.0 Prerequisites



In general, 3 steps are needed:

1. Pixels to entities
2. GCN upon entities
3. Entities to pixels

1.0 Prerequisites



In general, 3 steps are needed:

1. Pixels to entities
2. GCN upon entities
3. Entities to pixels

Difficulties:

1. How to proj and re-proj?
2. How to define A?

1.1 A^2Net

For every spatial input location *i*

$$\mathbf{z}_i = \underbrace{\mathbf{F}_{\mathrm{distr}}\left(\underbrace{\mathbf{G}_{\mathrm{gather}}(X)}_{\text{Proj}}, \mathbf{v}_i\right)}_{\text{Re-proj}}$$

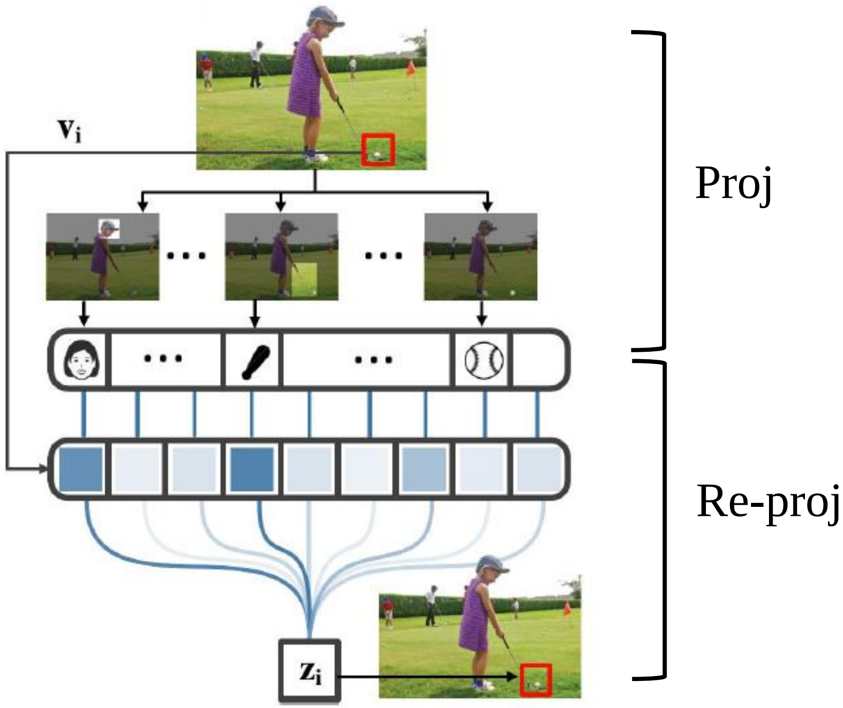
Chen, Yunpeng, et al. "A^ 2-Nets: Double Attention Networks." *Advances in Neural Information Processing Systems*. 2018.

1.1 A^2Net

For every spatial input location i

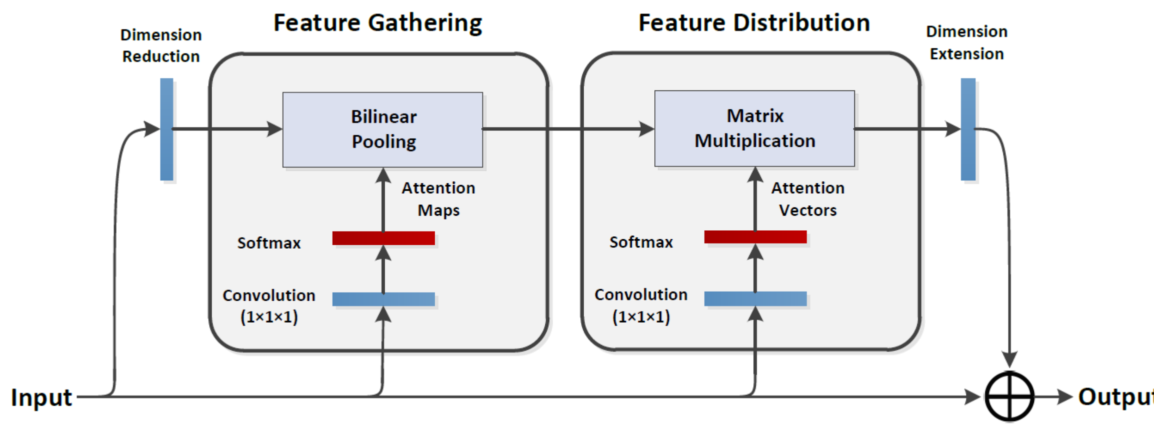
$$\mathbf{z}_i = \mathbf{F}_{\text{distr}}\big(\underbrace{\mathbf{G}_{\text{gather}}(X)}_{\text{Proj}}, \mathbf{v}_i\big)$$

Re-proj



Chen, Yunpeng, et al. "A^ 2-Nets: Double Attention Networks." *Advances in Neural Information Processing Systems*. 2018.

1.1 A^2Net



C × HW

$$\mathbf{z}_i = \mathbf{F}_{\text{distr}}\left(\mathbf{G}_{\text{gather}}(X), \mathbf{v}_i\right)$$

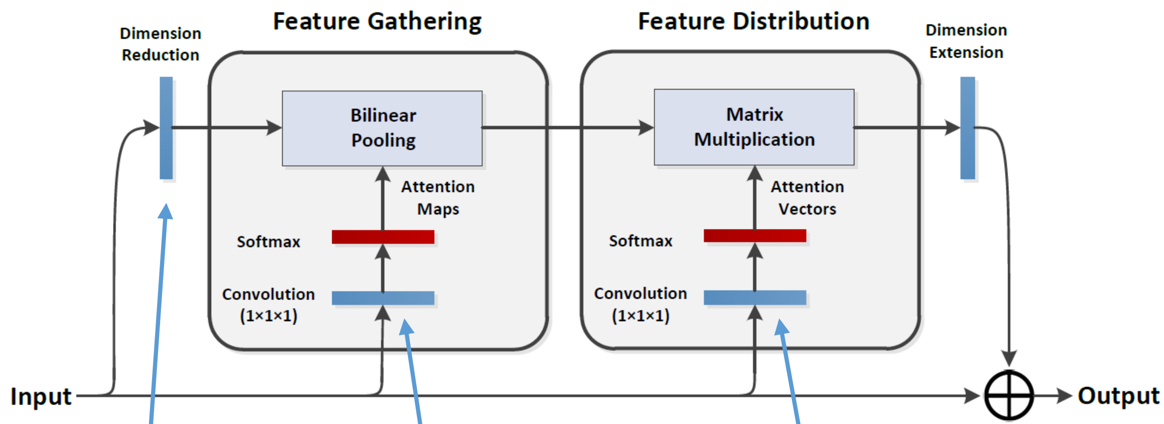
1 × 1 Conv　　1 × 1 Conv　　1 × 1 Conv

$$= \underbrace{\left[\underbrace{\phi(X; W_\phi)}_{\text{C} \times \text{HW}}\text{softmax}\left(\underbrace{\theta(X; W_\theta)}_{\text{HW} \times \text{K}}\right)^\top\right]}_{\text{C} \times \text{K}} \text{softmax}\left(\underbrace{\rho(X; W_\rho)}_{\text{K} \times \text{HW}}\right)$$

C × HW

H: Height
W: Width
C: Number of the channels
K: Number of global descriptors

1.1 A^2Net



Using four 1×1 convolutions

1. Two construct bottleneck
2. Another two used for attention,
   Which can be merged as one.

A^2Net

$$\left[\phi(X;W_\phi)\mathrm{softmax}\left(\theta(X;W_\theta)\right)^\top\right]\mathrm{softmax}\left(\rho(X;W_\rho)\right)$$

$$O(HWCK)$$

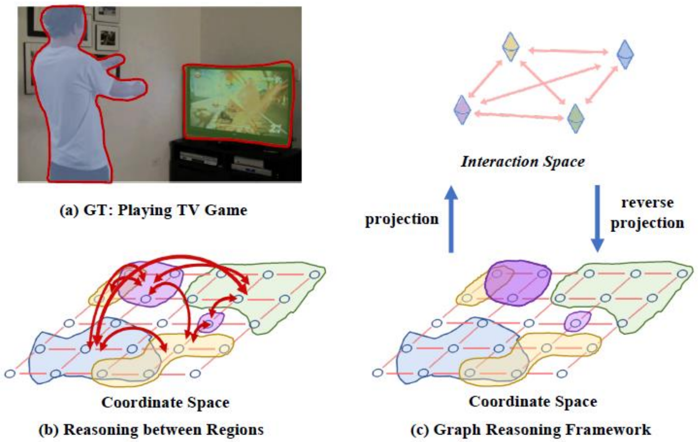
Comparison with Nonlocal:
Reduce the complexity by using the multiplication law.

Nonlocal

$$\phi\left(X;W_\phi\right) softmax\left(\theta\left(X;W_\theta\right)^\top \rho\left(X;W_\rho\right)\right)$$

$$O\left((HW)^2(C+K)\right)$$

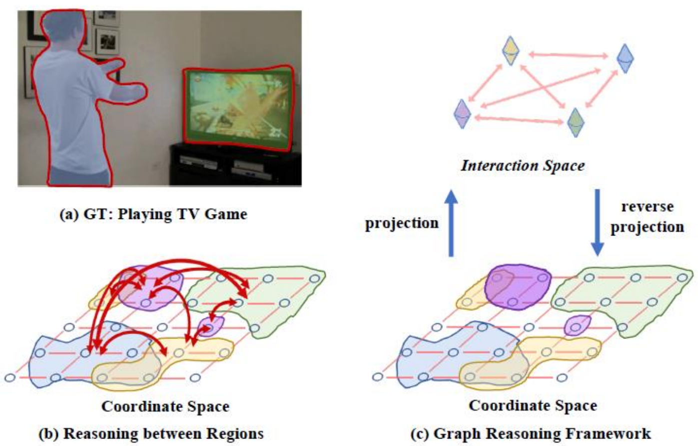
1.2 GloRe



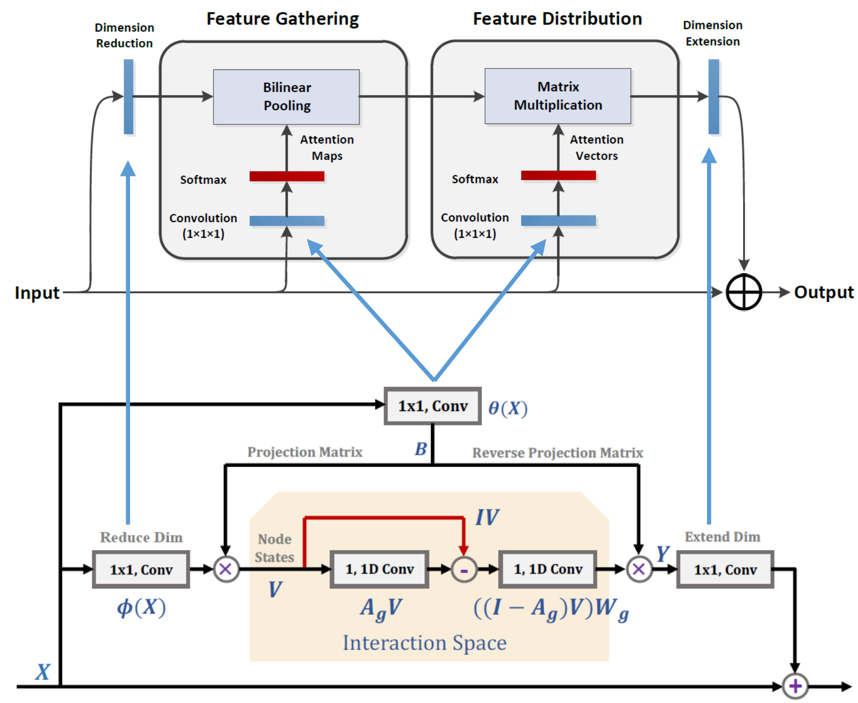
First paper about GCN in Seg published on Arxiv.org.

Chen, Yunpeng, et al. "Graph-Based Global Reasoning Networks."
IEEE Conference on Computer Vision and Pattern Recognition. 2019.

1.2 GloRe



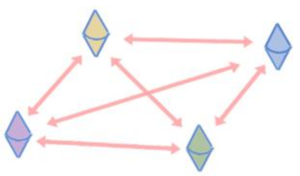
First paper about GCN in Seg published on Arxiv.org.



Using five 1 × 1 convolutions
To execute global reasoning.

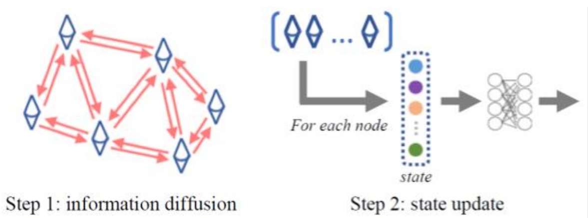
Chen, Yunpeng, et al. "Graph-Based Global Reasoning Networks."
IEEE Conference on Computer Vision and Pattern Recognition. 2019.

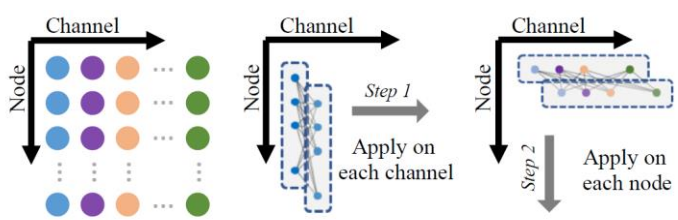
1.2 GloRe

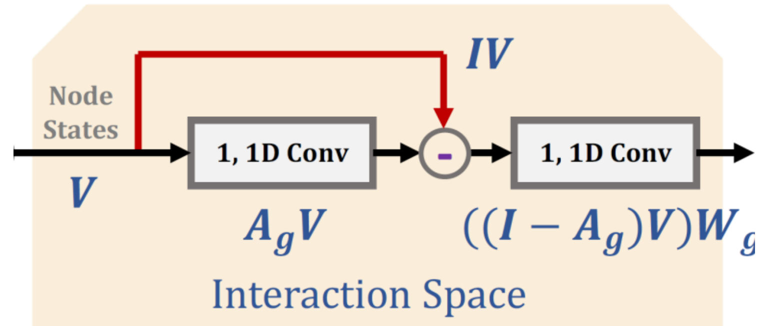


Interaction Space

How?







$$\mathbf{Z} = GVW_g = ((I - A_g)V)W_g$$

K × K　　K × C　　C × C

A_g : learned parameter of a Conv1d

W_g: learned parameter of a Conv1d

$$O(HWCK + K^2C + C^2K)$$

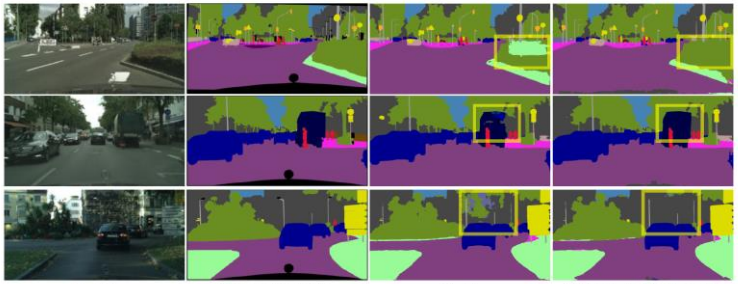
1.2 GloRe

Table 3: Semantic segmentation results on Cityscapes validation set. ImageNet pre-trained ResNet-50 is used as the backbone CNN.

FCN	multi-grid	+1 GloRe unit	+2 GloRe unit	mIoU	Δ mIoU
✓				75.79%	
✓	✓			76.45%	0.66%
✓	✓	✓		**78.25%**	**2.46%**
✓	✓		✓	77.84%	2.05%

Table 4: Semantic segmentation results on Cityscapes test set. All networks are evaluated by the testing server. Our method is trained without using extra "coarse" training set.

Method	Backbone	IoU cla.	iIoU cla.	IoU cat.	iIoU cat.
DeepLab-v2 [4]	ResNet101	70.4%	42.6%	86.4%	67.7%
PSPNet [36]	ResNet101	78.4%	56.7%	90.6%	78.6%
PSANet [37]	ResNet101	80.1%			
DenseASPP [35]	ResNet101	80.6%			
FCN + 1 GloRe unit	ResNet50	79.5%	60.3%	91.3%	81.5%
FCN + 1 GloRe unit	ResNet101	**80.9%**	**62.2%**	**91.5%**	**82.1%**



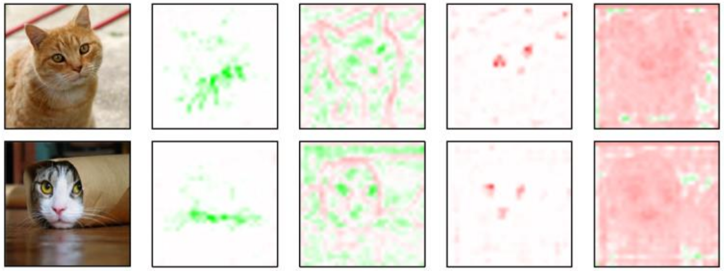
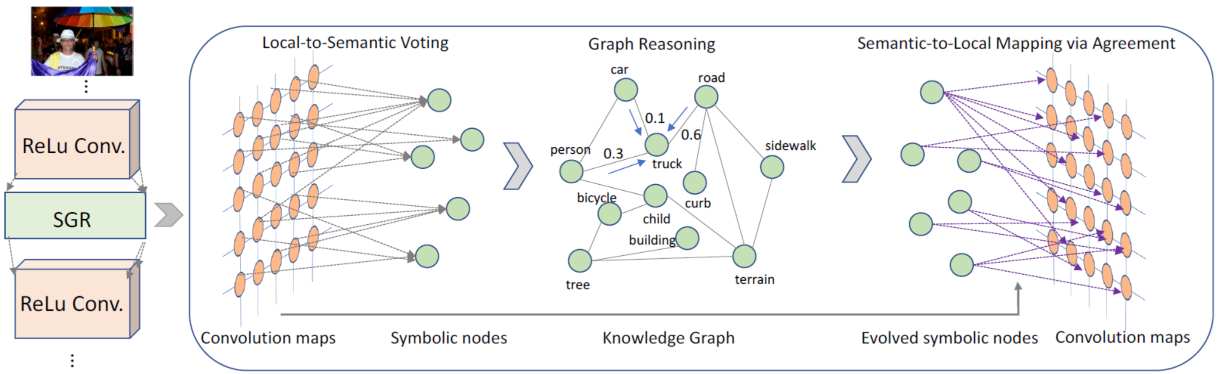
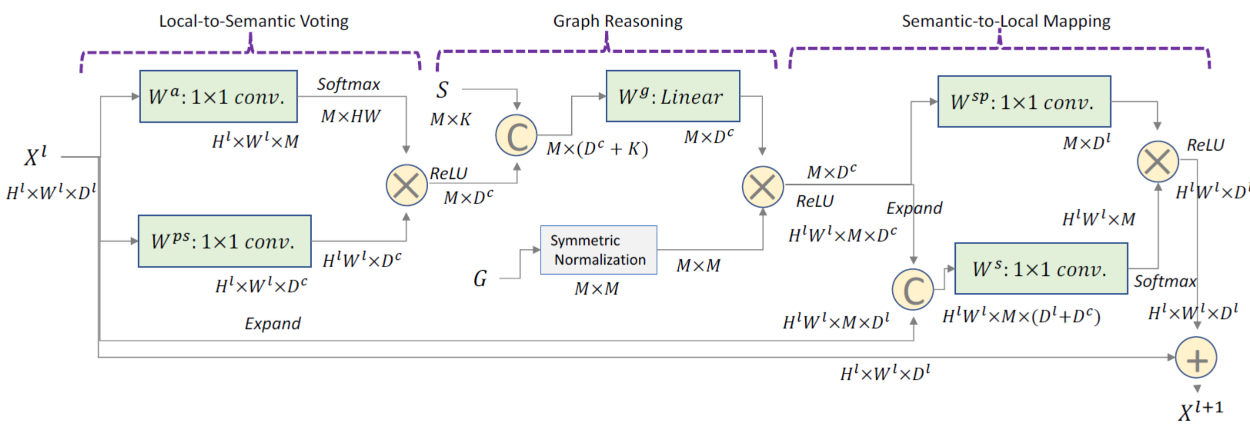


Figure 7: Visualization of the learned projection weights (best viewed in color). Red color denotes positive and green negative values, color brightness denotes magnitude.

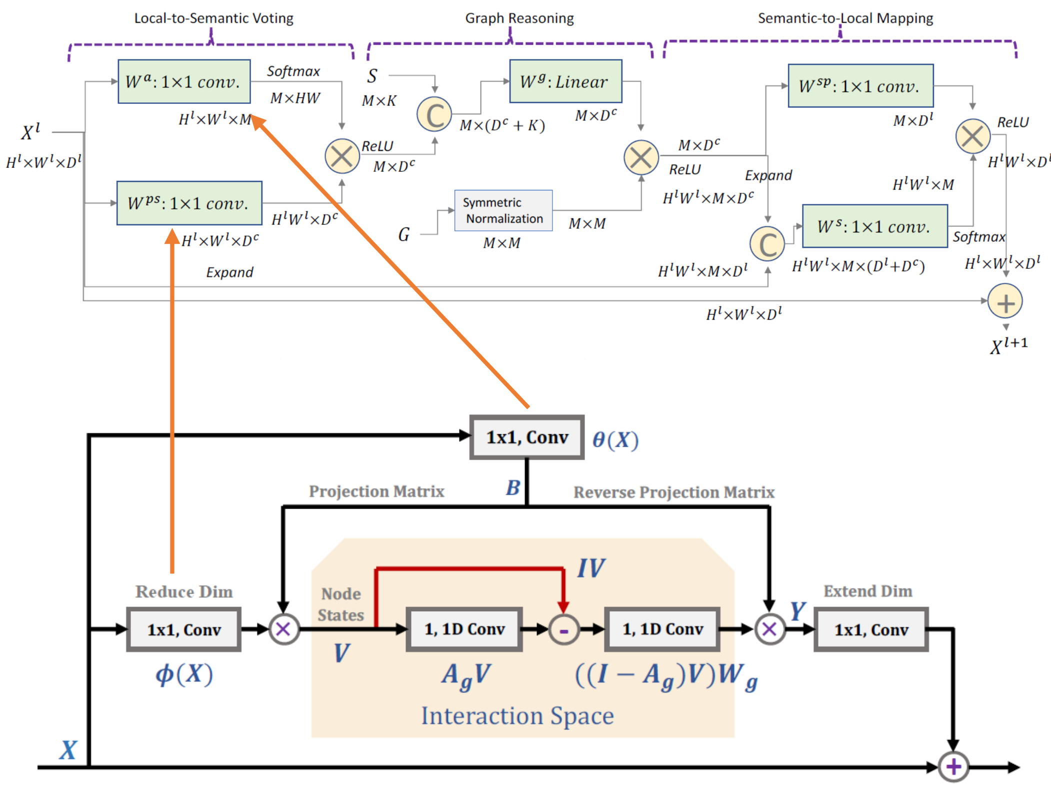
1.3 SGR





Liang, Xiaodan, et al. "Symbolic graph reasoning meets convolutions." Advances in Neural Information Processing Systems. 2018.

1.3 SGR

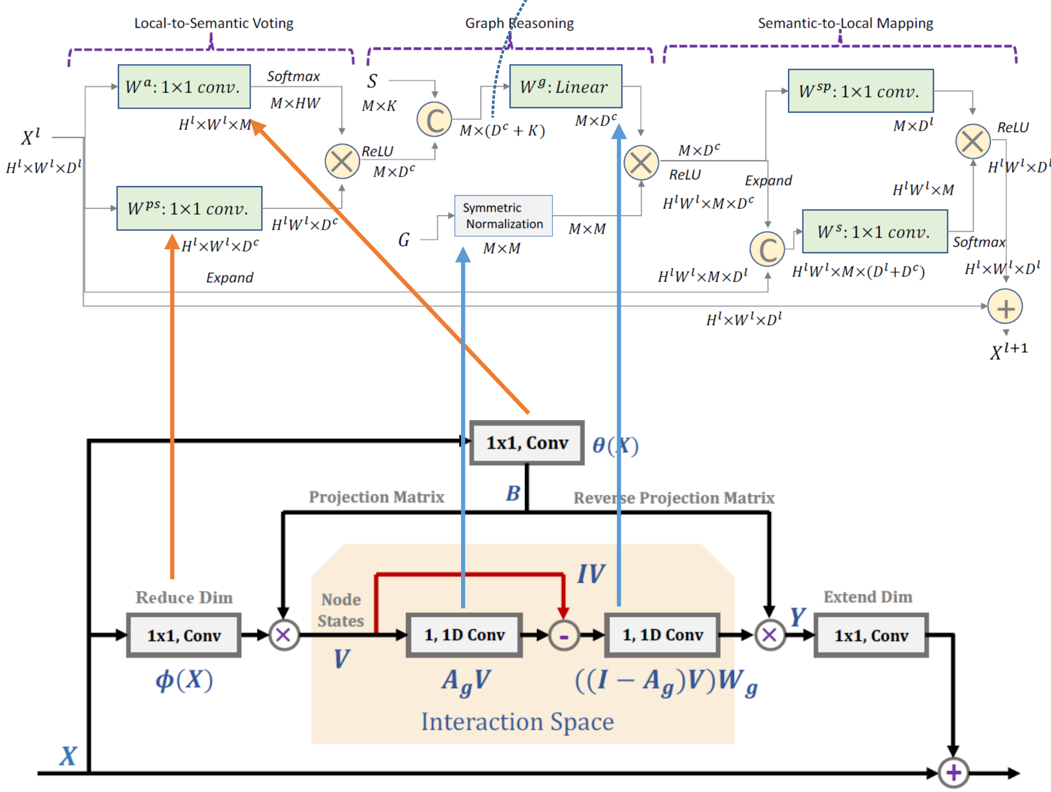


$$H_n^{ps} = \sum_{x_i} a_{x_i \to n} x_i W^{ps}$$

$$a_{x_i \to n} = \frac{\exp(W_n^{aT} x_i)}{\sum_{n \in \mathcal{N}} \exp(W_n^{aT} x_i)}$$

$$\left[\phi(X; W_\phi)\mathrm{softmax}\left(\theta(X; W_\theta)\right)^\top\right]\mathrm{softmax}\left(\rho(X; W_\rho)\right)$$

1.3 SGR



$$H^g = \sigma(A^g B W^g)$$

$$B = [\sigma(H^{ps}), \mathcal{S}] \in \mathbf{R}^{M\times(D^c+K)}$$

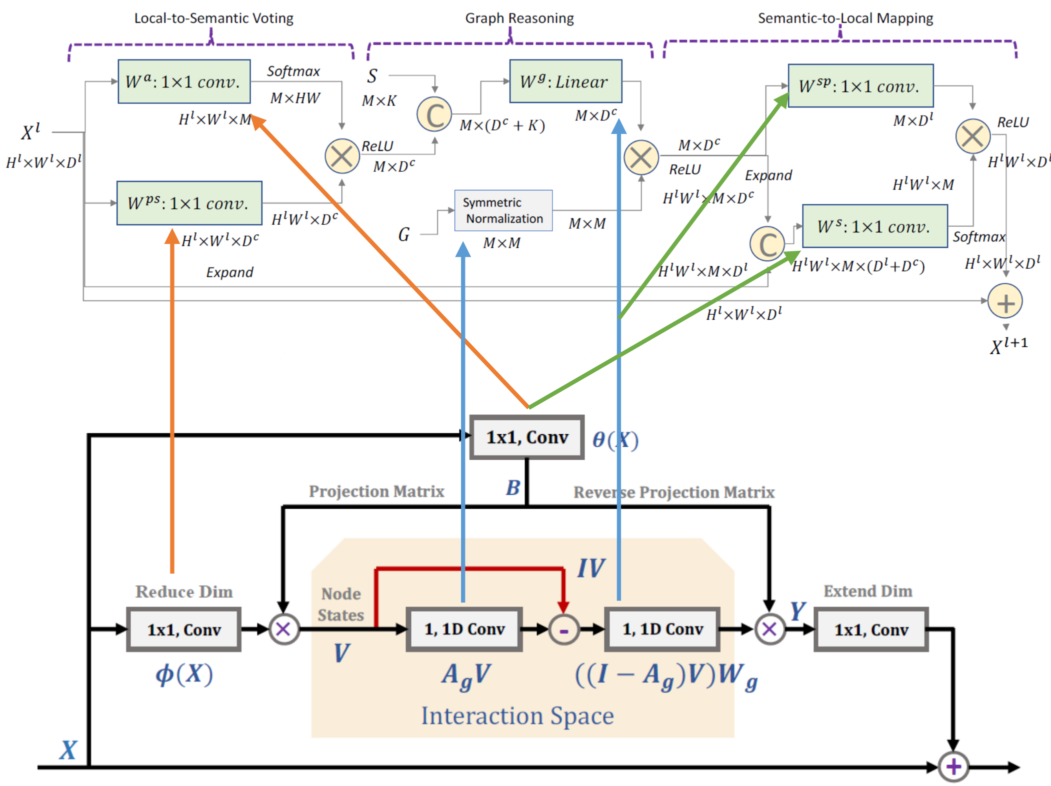
linguistic embedding

Different definitions of A:

- SGR - pre-defined according to priors
- GloRe - learnable parameters

$$\mathbf{Z} = GVW_g = ((I - A_g)V)W_g$$

1.3 SGR



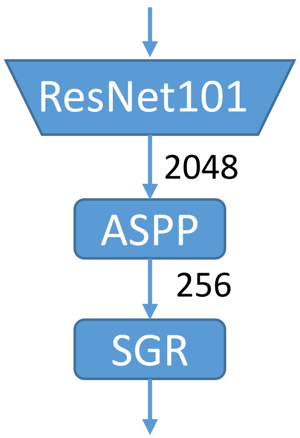
$$a_{h^g \to x_i} = \frac{\exp(W^{sT}[h^g, x_i])}{\sum_{x_i} \exp(W^{sT}[h^g, x_i])}$$

$$X^{l+1} = \sigma(A^{sp} H^g W^{sp}) + X^l$$

$$\mathbf{Z} = GVW_g = ((I - A_g)V)W_g$$

$$\left[\phi(X; W_\phi)\operatorname{softmax}\left(\theta(X; W_\theta)\right)^\top\right] \operatorname{softmax}\left(\rho(X; W_\rho)\right)$$

1.3 SGR



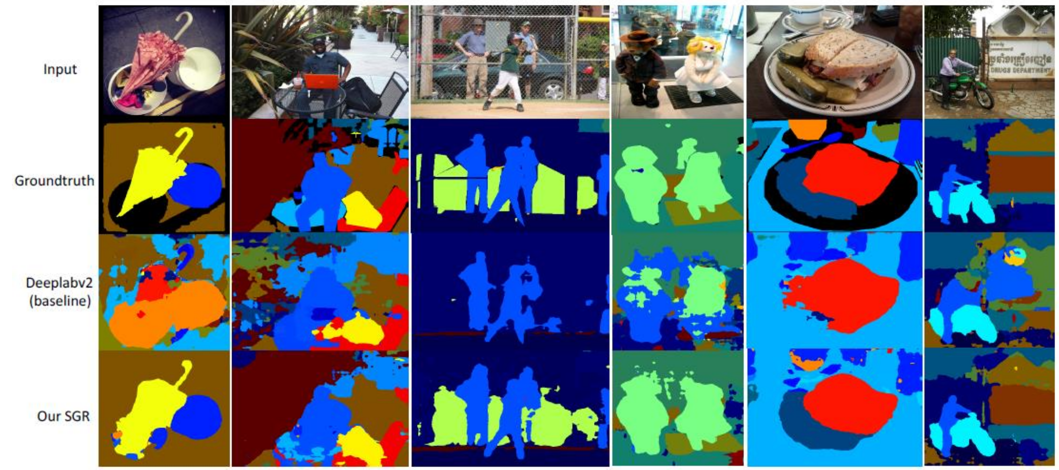


Figure 3: Qualitative comparison results on Coco-stuff dataset.

Method	Class acc.	acc.	mean IoU
FCN [31]	38.5	60.4	27.2
DeepLabv2 (ResNet-101) [6]	45.5	65.1	34.4
DAG RNN + CRF [42]	42.8	63.0	31.2
OHE + DC + FCN [15]	45.8	66.6	34.3
DSSPN (ResNet-101) [27]	47.0	68.5	36.2
SGR (w/o residual)	47.9	68.4	38.1
SGR (scene graph)	49.1	69.6	38.3
SGR (concurrence graph)	48.6	69.5	38.4
SGR (w/o mapping)	47.3	67.9	37.2
SGR (ConvBlock4)	47.6	68.3	37.5
Our SGR (ResNet-101)	49.3	69.9	38.7
Our SGR (ResNet-101 2-layer)	49.4	69.7	38.8
Our SGR (ResNet-101 Hybrid)	**49.8**	**70.5**	**39.1**

Table 1: Comparison on Coco-Stuff test set (%). All our models are based on ResNet-101.

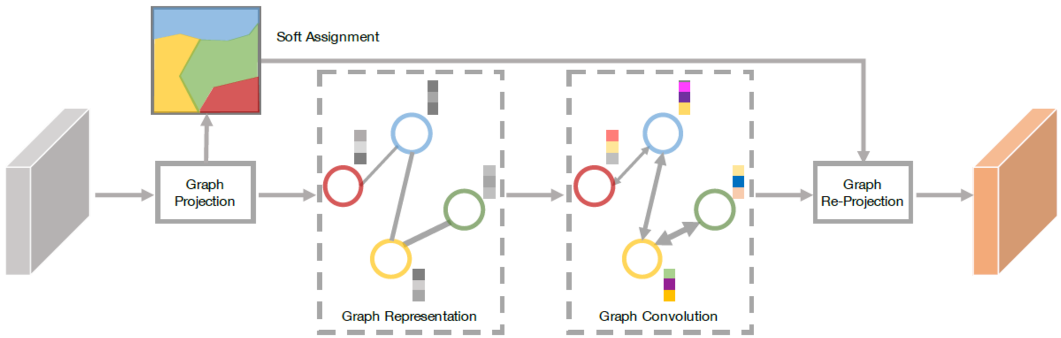
Method	mean IoU (%)
FCN [31]	37.8
CRF-RNN [51]	39.3
ParseNet [30]	40.4
BoxSup [8]	40.5
HO CRF [1]	41.3
Piecewise [29]	43.3
VeryDeep [44]	44.5
DeepLab-v2 (ResNet-101) [6]	45.7
RefineNet (Res152) [28]	47.3
Our SGR (ResNet-101)	50.8
Our SGR (Transfer convs)	51.3
Our SGR (Transfer SGR)	**52.5**

Table 2: Comparison on PASCAL-Context test set(%).

Method	mean IoU	pixel acc.
FCN [31]	29.39	71.32
SegNet [2]	21.64	71.00
DilatedNet [47]	32.31	73.55
CascadeNet [52]	34.90	74.52
ResNet-101, 2 conv [45]	39.40	79.07
PSPNet (ResNet-101)DA_AL [50]	41.96	80.64
Conditional Softmax [38]	31.27	72.23
Word2Vec [10]	29.18	71.31
Joint-Cosine [49]	31.52	73.15
DeepLabv2 (ResNet-101) [6]	38.97	79.01
DSSPN (ResNet-101) [27]	42.03	81.21
Our SGR (ResNet-101)	**44.32**	**81.43**

Table 3: Comparison on the ADE20K val set [52] (%). "Conditional Softmax [38]", "Word2Vec [10]" and "Joint-Cosine [49]" use VGG as backbone. We use "DeepLabv2 (ResNet-101) [6]" as baseline.

1.4 GCU



$$\tilde{X} = \mathrm{GCU}(x) = G_{reproj}(G_{conv}(G_{proj}(X)))$$

Graph Projection

GCU

$$q_{ij}^{k} = \frac{\exp\left(-\|(x_{ij} - w_k)/\sigma_k\|_2^2/2\right)}{\sum_k \exp\left(-\|(x_{ij} - w_k)/\sigma_k\|_2^2/2\right)}$$

$$z_k = \frac{z_k'}{\|z_k'\|_2}, \quad z_k' = \frac{1}{\sum_{ij} q_{ij}^k} \sum_{ij} q_{ij}^k \left(x_{ij} - w_k\right)/\sigma_k$$

A^2Net

$$q_{ij}^{k} = \frac{\exp\left(x_{ij}^{\top} w_k\right)}{\sum_k \exp\left(x_{ij}^{\top} w_k\right)}$$

$$z_k = \frac{1}{\sum_{ij} q_{ij}^k} \sum_{ij} q_{ij}^k x_{ij}$$

$$\boxed{\left[\phi(X; W_\phi)\mathrm{softmax}\left(\theta(X; W_\theta)\right)^{\top}\right]}\mathrm{softmax}\left(\rho(X; W_\rho)\right)$$

Li, Yin, and Abhinav Gupta. "Beyond Grids: Learning Graph Representations for Visual Recognition." Advances in Neural Information Processing Systems. 2018.

Graph Convolution

GCU

$$\tilde{Z} = f(\mathcal{A}Z^T W_g)$$
$$\mathcal{A} = Z^T Z$$

A^2Net

$$\mathbf{Z} = GVW_g = ((I - A_g)V)W_g$$

Different definitions of A:

- SGR - pre-defined according to priors
- GloRe - learnable parameters
- GCU – sample-independent

Graph Reprojection

GCU

$$\tilde{X} = \mathcal{Q}\tilde{Z}^T$$

A^2Net

$$\left[\phi(X; W_\phi)\mathrm{softmax}\left(\theta(X; W_\theta)\right)^\top\right]\mathrm{softmax}\left(\rho(X; W_\rho)\right)$$

Graph Convolution

GCU

$$\tilde{Z} = f(\mathcal{A}Z^T W_g)$$
$$\mathcal{A} = Z^T Z$$

A^2Net

$$\mathbf{Z} = GVW_g = ((I - A_g)V)W_g$$

Different definitions of A:
- SGR - pre-defined according to priors
- GloRe - learnable parameters
- GCU – sample-independent

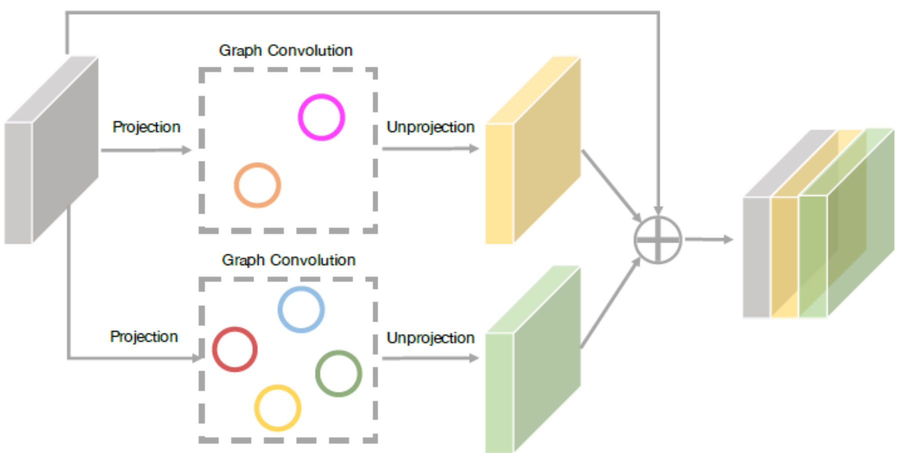
Graph Reprojection

GCU

$$\tilde{X} = \mathcal{Q}\tilde{Z}^T$$

A^2Net

$$\left[\phi(X; W_\phi)\mathrm{softmax}\left(\theta(X; W_\theta)\right)^\top\right] \mathrm{softmax}\left(\rho(X; W_\rho)\right)$$



$$\hat{X} = X \oplus \mathrm{GCU}_{k_1}(X) \oplus \ldots \oplus \mathrm{GCU}_{k_n}(X)$$

Graph Convolution

GCU

$$\tilde{Z} = f(\mathcal{A}Z^T W_g)$$
$$\mathcal{A} = Z^T Z$$

A^2Net

$$\mathbf{Z} = GVW_g = ((I - A_g)V)W_g$$

Different definitions of A:
- SGR - pre-defined according to priors
- GloRe - learnable parameters
- GCU – sample-independent

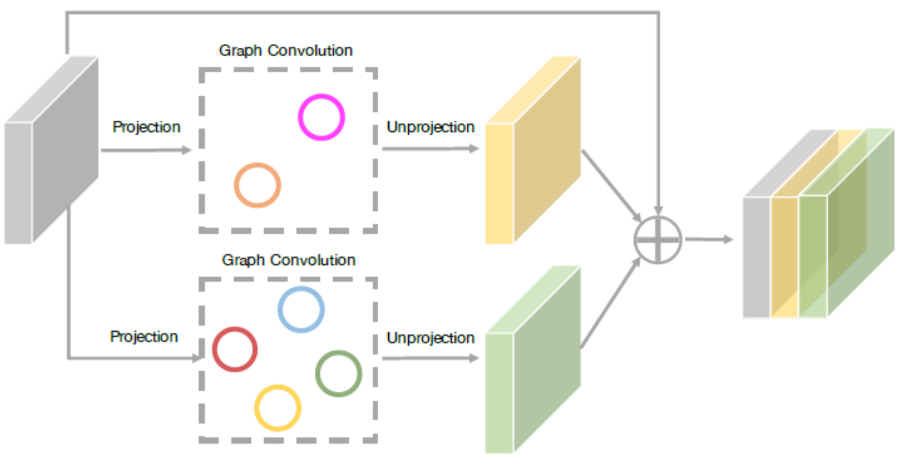
Graph Reprojection

GCU

$$\tilde{X} = \mathcal{Q}\tilde{Z}^T$$

A^2Net

$$\left[\phi(X; W_\phi)\mathrm{softmax}\left(\theta(X; W_\theta)\right)^\top\right] \mathrm{softmax}\left(\rho(X; W_\rho)\right)$$



$$\hat{X} = X \oplus \mathrm{GCU}_{k_1}(X) \oplus \ldots \oplus \mathrm{GCU}_{k_n}(X)$$

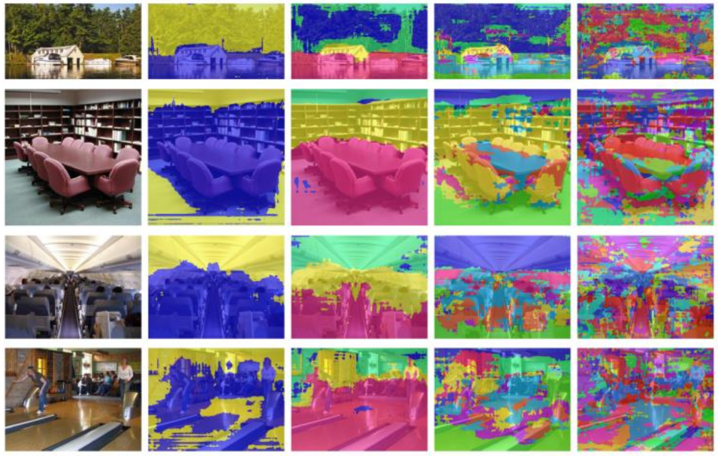


Figure 4: Visualization of the assignment matrix in GCU for semantic segmentation (with ResNet 50). From left to right: input image, pixel-to-vertex assignments with 2, 4, 8 and 32 vertices. Pixels with the same color are assigned to the same vertex. Vertices are colored consistently across images.

1.4 GCU

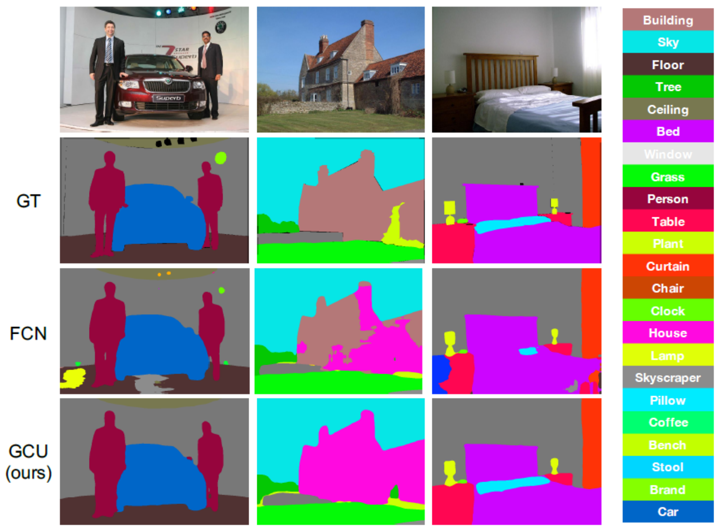


Figure 3: Visualization of segmentation results on ADE20K (with ResNet 50). Our method produces "smoother" maps–regions that are similar are likely to be labeled as the same category.

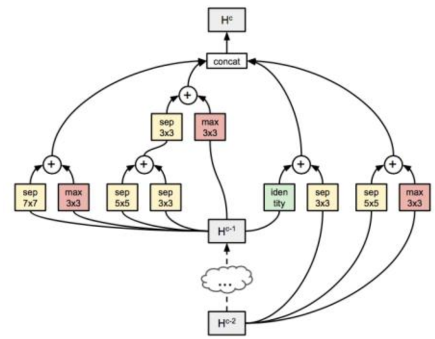
Backbone	Method	PixAcc%	mIoU%
VGG16 [42]	FCN-8s [12]	71.32	29.39
	SegNet [41]	71.00	21.64
	DilatedNet [17]	73.55	32.31
	CascadeNet [37]	74.52	34.90
Res50 [38]	Dilated FCN	76.51	35.60
	PSPNet [13]	80.76	**42.78**
	EncNet [14]	79.73	41.11
	GCU (ours)	79.51	**42.60**
Res101 [38]	RefineNet [19]	-	40.20
	PSPNet [13]	81.39	43.29
	EncNet [14]	81.69	**44.65**
	GCU (ours)	81.19	**44.81**

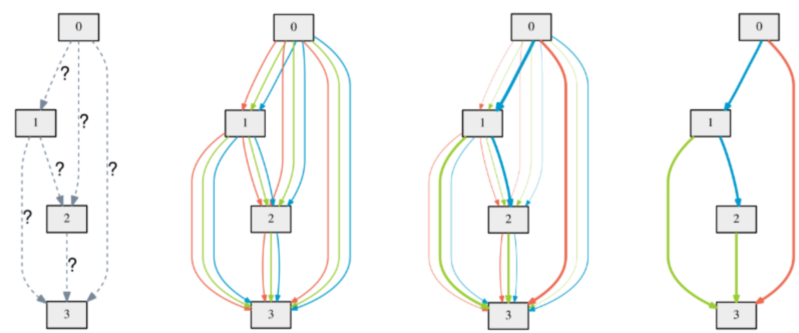
Table 1: Results of semantic segmentation on ADE20K. mIoU scores within 0.5% of the best result are marked. With ResNet 50, our method improves Dilated FCN by 7%. With ResNet 101, our method outperforms PSPNet by 1.5%.

2. NAS in Semantic Segmentation

Key components of Network Architecture Search (NAS)

1. Search space
    1. Block level
    2. Cell level
2. Proxy task
    1. Low-resolution image
3. Search strategy
    1. Reinforcement learning
    2. Evolutionary algorithm
    3. Bayesian optimization
    4. Differentiable methods

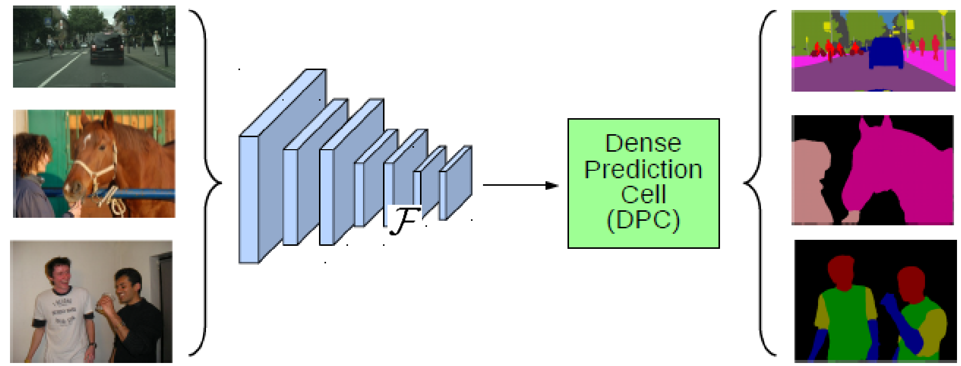




(a) (b) (c) (d)

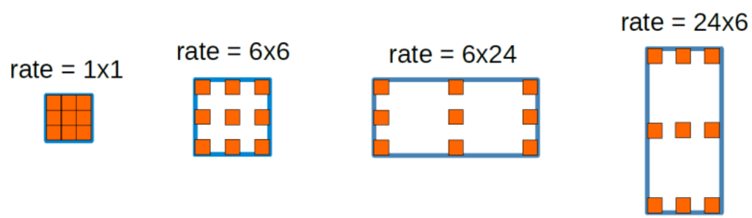
2.1 DPC

1. Search space
   - Head
   - Cell level
2. Search strategy
   - Random search



Cell definition: (X_i, OP_i, Y_i)

$$\mathcal{X}_i = \{\mathcal{F}, Y_1, \ldots, Y_{i-1}\}$$
$$Y = concat(Y_1, Y_2, \ldots, Y_{\mathcal{B}})$$



OP_i
- Convolution with a 1×1 kernel.
- $3{\times}3$ atrous separable convolution with rate $r_h{\times}r_w$, where r_h and $r_w \in \{1, 3, 6, 9, \ldots, 21\}$.
- Average spatial pyramid pooling with grid size $g_h \times g_w$, where g_h and $g_w \in \{1, 2, 4, 8\}$.

In total 81 operators

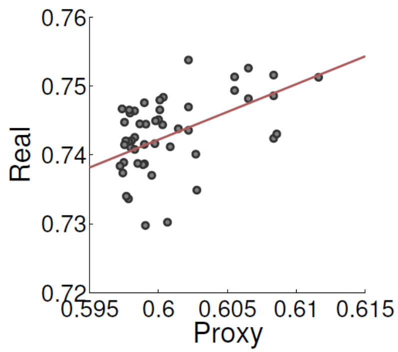
$$\mathcal{B}! \times 81^{\mathcal{B}} \approx 4.2 \times 10^{11}$$

Chen, Liang-Chieh, et al. "Searching for efficient multi-scale architectures for dense image prediction." Advances in Neural Information Processing Systems. 2018.

2.1 DPC

3. Proxy task
   - Small backbone
   - Fix backbone
   - Early stopping

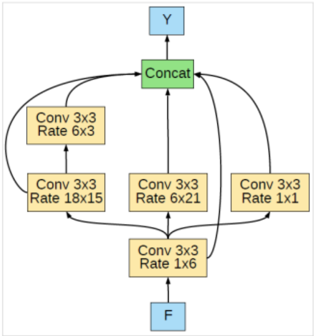
From 1 week to 90 minutes

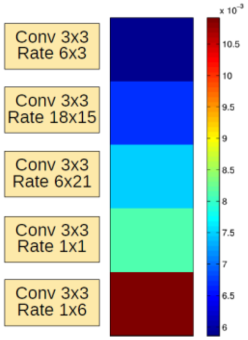


$\rho = 0.46$

Spearman's rank correlation coefficient

Using 370 GPUs over one week
Explore 28k DPC architectures





Method	road	sidewalk	building	wall	fence	pole	light	sign	vege.	terrain	sky	person	rider	car	truck	bus	train	mbike	bicycle	mIOU
PSPNet [97]	**98.7**	86.9	93.5	58.4	63.7	67.7	76.1	80.5	93.6	72.2	95.3	86.8	71.9	96.2	77.7	91.5	83.6	70.8	77.5	81.2
Mapillary Research [6]	98.4	85.0	93.7	**61.8**	**63.9**	67.7	77.4	80.8	93.7	71.9	95.6	86.7	72.8	95.7	79.9	93.1	**89.7**	72.6	78.2	82.0
DeepLabv3+ [14]	**98.7**	87.0	**93.9**	59.5	63.7	**71.4**	**78.2**	**82.2**	**94.0**	73.0	**95.9**	88.0	73.3	96.4	78.0	90.9	83.9	73.8	78.9	82.1
DPC	**98.7**	**87.1**	93.8	57.7	63.5	71.0	78.0	82.1	**94.0**	**73.3**	95.4	**88.2**	**74.5**	**96.5**	**81.2**	**93.3**	89.0	**74.1**	**79.0**	**82.7**

Table 2: Cityscapes *test* set performance across leading competitive models.

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mIOU
EncNet [95]	95.3	76.9	94.2	80.2	85.3	96.5	90.8	96.3	47.9	93.9	**80.0**	92.4	96.6	90.5	91.5	70.9	93.6	66.5	87.7	80.8	85.9
DFN [93]	96.4	78.6	95.5	79.1	86.4	97.1	91.4	95.0	47.7	92.9	77.2	91.0	96.7	92.2	91.7	76.5	93.1	64.4	88.3	81.2	86.2
DeepLabv3+ [14]	97.0	77.1	**97.1**	79.3	**89.3**	97.4	93.2	96.6	**56.9**	95.0	79.2	93.1	97.0	**94.0**	**92.8**	71.3	92.9	72.4	91.0	84.9	87.8
ExFuse [96]	96.8	**80.3**	97.0	**82.5**	87.8	96.3	92.6	96.4	53.3	94.3	78.4	94.1	94.9	91.6	92.3	**81.7**	**94.8**	70.3	90.1	83.8	87.9
MSCI [48]	96.8	76.8	97.0	80.6	**89.3**	97.4	**93.8**	**97.1**	56.7	94.3	78.3	93.5	97.1	**94.0**	**92.8**	72.3	92.6	**73.6**	90.8	**85.4**	**88.0**
DPC	**97.4**	77.5	96.6	79.4	87.2	**97.6**	90.1	96.6	56.8	**97.0**	77.0	**94.3**	**97.5**	93.2	92.5	78.9	94.3	70.1	**91.4**	84.0	87.9

Table 4: PASCAL VOC 2012 *test* set performance.

2.3 Auto-DeepLab

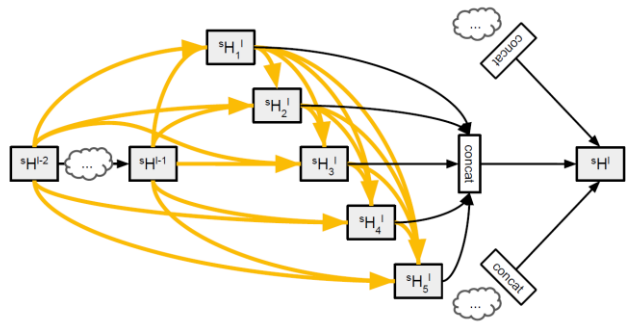
1. Search space
   1. Cell level (I_1, I_2, O_1, O_2, C)

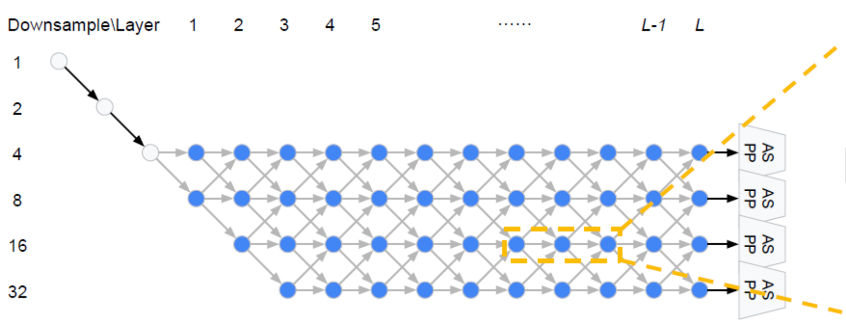
      For the l-th cell in the i-th block

      $$I_i^l \in \{H^{l-2}, H^{l-1}, \{H_1^l, \cdots, H_i^{l-1}\}\}$$

      O
      - 3×3 depthwise-separable conv
      - 5×5 depthwise-separable conv
      - 3×3 atrous conv with rate 2
      - 5×5 atrous conv with rate 2
      - 3×3 average pooling
      - 3×3 max pooling
      - skip connection
      - no connection (zero)

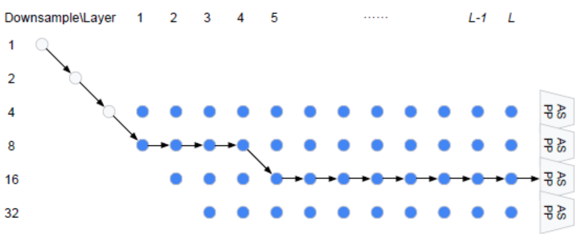
      C : element-wise addition
   2. Block level



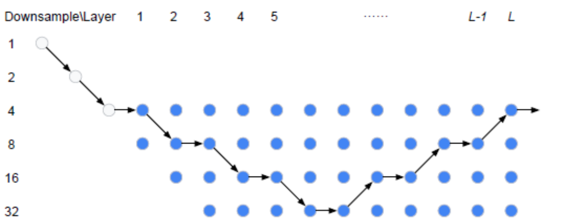


Liu, Chenxi, et al. "Auto-DeepLab: Hierarchical Neural Architecture Search for Semantic Image Segmentation." *arXiv preprint arXiv:1901.02985* (2019).

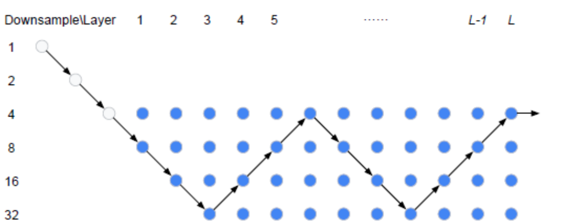
2.3 Auto-DeepLab



(a) Network level architecture used in DeepLabv3 [9].



(b) Network level architecture used in Conv-Deconv [56].



(c) Network level architecture used in Stacked Hourglass [55].

3. Continuous relaxation
   1. Cell level

$$H_i^l = \sum_{H_j^l \in \mathcal{I}_i^l} O_{j\to i}(H_j^l)$$

$$\bar{O}_{j\to i}(H_j^l) = \sum_{O^k \in \mathcal{O}} \alpha_{j\to i}^k O^k(H_j^l)$$

$$\sum_{k=1}^{|\mathcal{O}|} \alpha_{j\to i}^k = 1 \qquad \forall i, j$$

$$\alpha_{j\to i}^k \geq 0 \qquad \forall i, j, k$$

   2. Block level

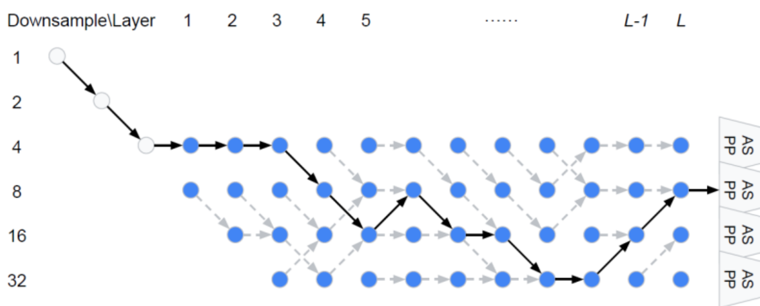
$$\begin{aligned} {}^sH^l = &\beta_{\frac{s}{2}\to s}^l \mathrm{Cell}({}^{\frac{s}{2}}H^{l-1}, {}^sH^{l-2}; \alpha) \\ &+ \beta_{s\to s}^l \mathrm{Cell}({}^sH^{l-1}, {}^sH^{l-2}; \alpha) \\ &+ \beta_{2s\to s}^l \mathrm{Cell}({}^{2s}H^{l-1}, {}^sH^{l-2}; \alpha) \end{aligned}$$

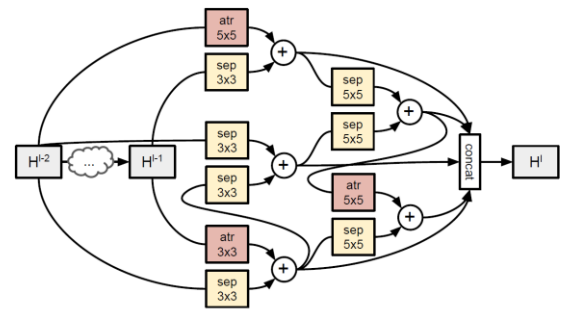
$$\beta_{s\to\frac{s}{2}}^l + \beta_{s\to s}^l + \beta_{s\to 2s}^l = 1 \qquad \forall s, l$$

$$\beta_{s\to\frac{s}{2}}^l \geq 0 \quad \beta_{s\to s}^l \geq 0 \quad \beta_{s\to 2s}^l \geq 0 \qquad \forall s, l$$

4. Search strategy
   1. Update network weights w by $\nabla_w \mathcal{L}_{trainA}(w, \alpha, \beta)$
   2. Update architecture α, β by $\nabla_{\alpha,\beta} \mathcal{L}_{trainB}(w, \alpha, \beta)$
5. Decoding discrete architecture
   1. Cell architecture
      - Argmax
   2. Block architecture
      - Viteerbi algorithm

2.3 Auto-DeepLab





Method	ImageNet	Coarse	mIOU (%)
FRRN-A [60]			63.0
GridNet [17]			69.5
FRRN-B [60]			71.8
Auto-DeepLab-S			79.9
Auto-DeepLab-L			80.4
Auto-DeepLab-S		✓	80.9
Auto-DeepLab-L		✓	82.1
ResNet-38 [81]	✓	✓	80.6
PSPNet [87]	✓	✓	81.2
Mapillary [4]	✓	✓	82.0
DeepLabv3+ [11]	✓	✓	82.1
DPC [6]	✓	✓	82.7
DRN_CRL_Coarse [90]	✓	✓	82.8

Table 4: Cityscapes test set results with *multi-scale* inputs during inference. **ImageNet:** Models pretrained on ImageNet. **Coarse:** Models exploit coarse annotations.

Method	ImageNet	COCO	mIOU (%)
Auto-DeepLab-S		✓	82.5
Auto-DeepLab-M		✓	84.1
Auto-DeepLab-L		✓	85.6
RefineNet [44]	✓	✓	84.2
ResNet-38 [81]	✓	✓	84.9
PSPNet [87]	✓	✓	85.4
DeepLabv3+ [11]	✓	✓	87.8
MSCI [43]	✓	✓	88.0

Table 6: PASCAL VOC 2012 test set results. Our Auto-DeepLab-L attains comparable performance with many state-of-the-art models which are pretrained on both **ImageNet** and **COCO** datasets. We refer readers to the official leader-board for other state-of-the-art models.

Method	ImageNet	mIOU (%)	Pixel-Acc (%)	Avg (%)
Auto-DeepLab-S		40.69	80.60	60.65
Auto-DeepLab-M		42.19	81.09	61.64
Auto-DeepLab-L		43.98	81.72	62.85
CascadeNet (VGG-16) [89]	✓	34.90	74.52	54.71
RefineNet (ResNet-152) [44]	✓	40.70	-	-
UPerNet (ResNet-101) [82] †	✓	42.66	81.01	61.84
PSPNet (ResNet-152) [87]	✓	43.51	81.38	62.45
PSPNet (ResNet-269) [87]	✓	44.94	81.69	63.32
DeepLabv3+ (Xception-65) [11] †	✓	45.65	82.52	64.09

Table 7: ADE20K validation set results. We employ *multi-scale* inputs during inference. †: Results are obtained from their up-to-date model zoo websites respectively. **ImageNet:** Models pretrained on ImageNet. Avg: Average of mIOU and Pixel-Accuracy.

6. References

[1] Chen, Yunpeng, et al. "A^ 2-Nets: Double Attention Networks." *Advances in Neural Information Processing Systems*. 2018.

[2] Chen, Yunpeng, et al. "Graph-Based Global Reasoning Networks." IEEE Conference on Computer Vision and Pattern Recognition. 2019.

[3] Liang, Xiaodan, et al. "Symbolic graph reasoning meets convolutions." Advances in Neural Information Processing Systems. 2018.

[4] Li, Yin, and Abhinav Gupta. "Beyond Grids: Learning Graph Representations for Visual Recognition." Advances in Neural Information Processing Systems. 2018.

[5] Chen, Liang-Chieh, et al. "Searching for efficient multi-scale architectures for dense image prediction." Advances in Neural Information Processing Systems. 2018.

[6] Liu, Chenxi, et al. "Auto-DeepLab: Hierarchical Neural Architecture Search for Semantic Image Segmentation." *arXiv preprint arXiv:1901.02985* (2019).

Thanks

Speaker : Xia Li