北京理工大學
BEIJING INSTITUTE OF TECHNOLOGY

Few Shot Segmentation

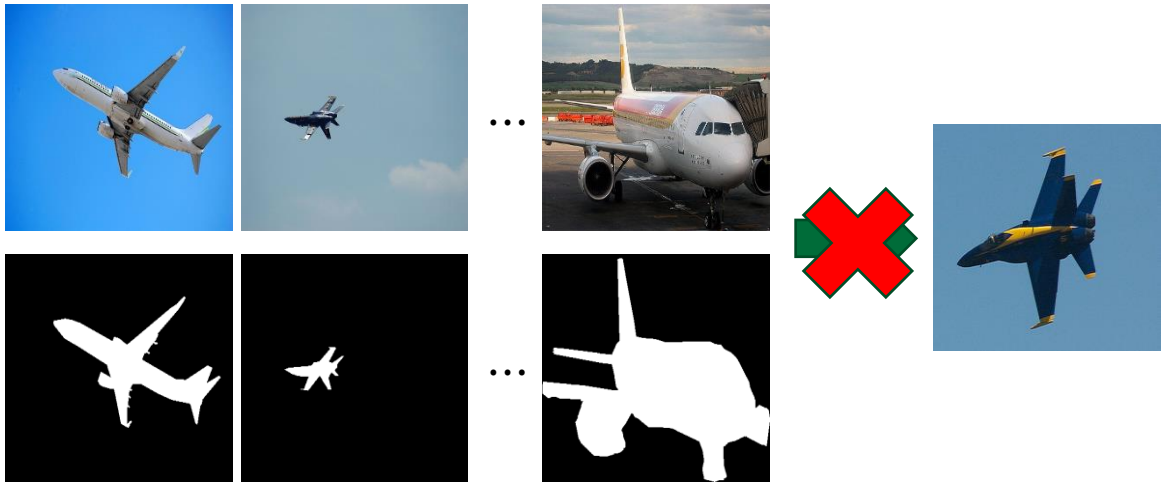
方致远 张泽康

2021/5/14

Motivation

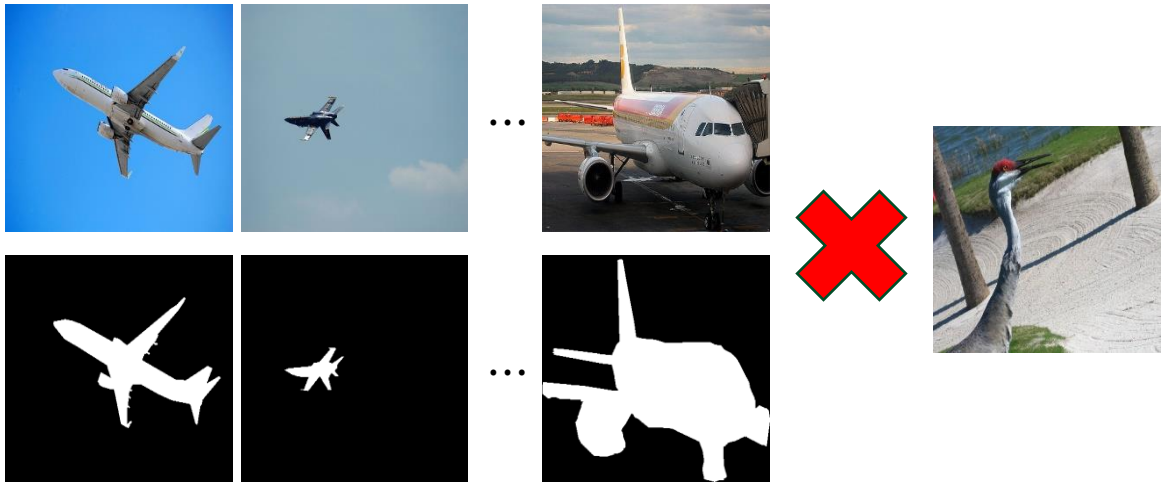
Defects of Deep-learning-based Segmentation

- Eager for large amount of samples of target
- Cannot segment unseen objects



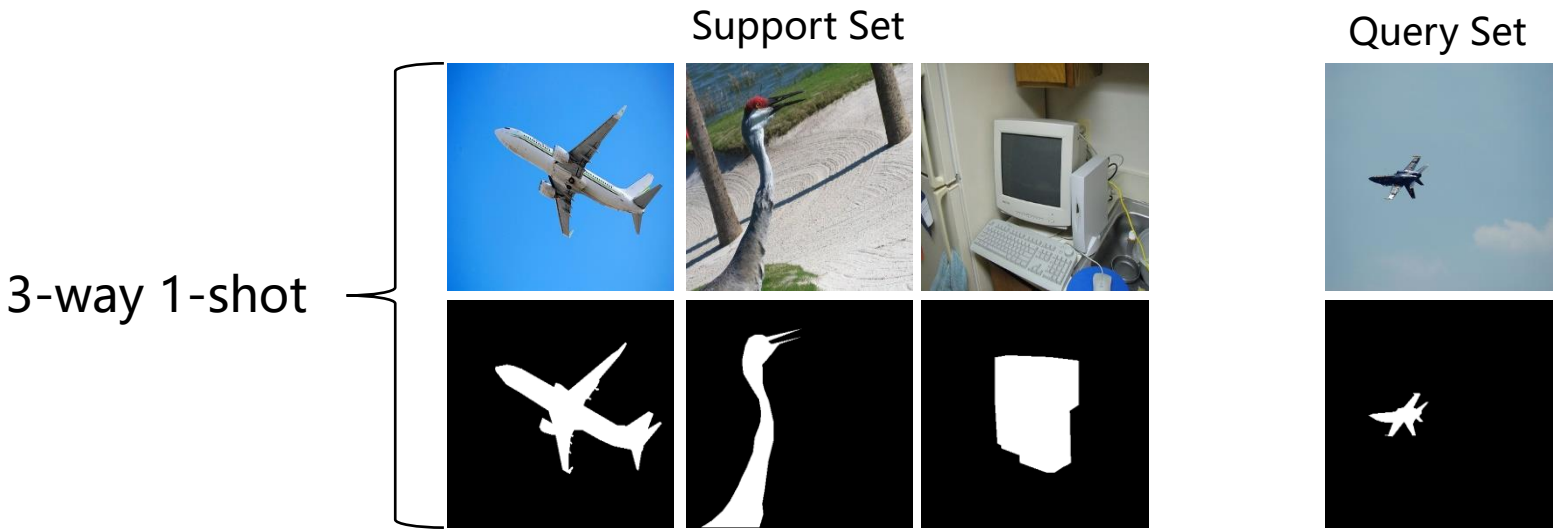
The aims of Few-shot Segmentation

- Segment unseen object with only a few references after training

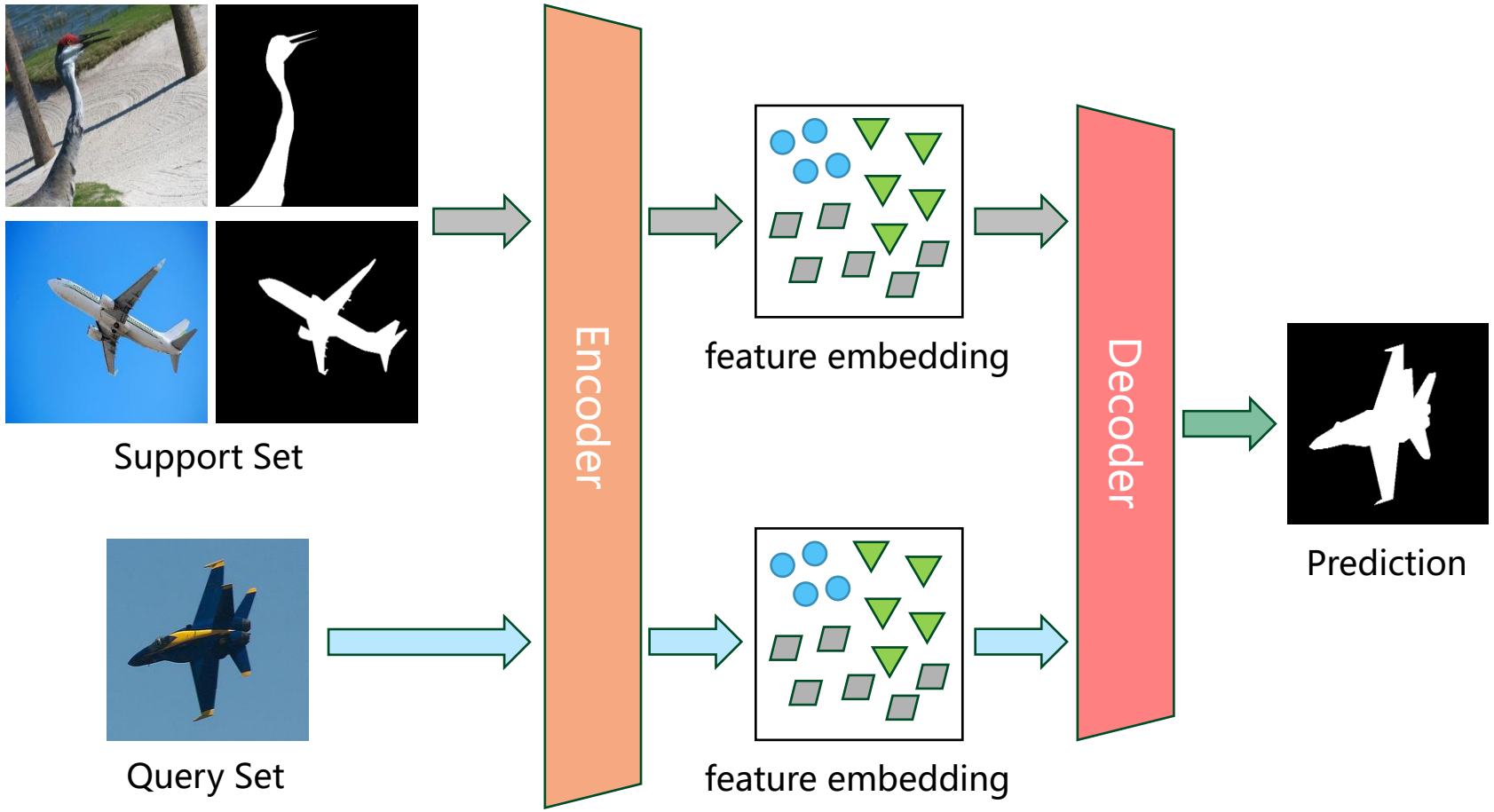


Implementation

- Divide classes C of dataset into seen C_{seen} and unseen C_{unseen}, $C_{seen} \cap C_{unseen} = \emptyset$
- $D_{train} = \{(S_i, Q_i)\}_{i=1}^{N_{train}}$, $D_{test} = \{(S_i, Q_i)\}_{i=1}^{N_{test}}$
- $S_i = \{(I_{c,k}, M_{c,k}) \mid c \in c_{seen}$ for training $else\ c \in c_{unseen}\}$, $Q_i = \{(I_{c,n}, M_{c,n})\}$
- (S_i, Q_i) called episode is the input to the model during training/testing stage



Implementation



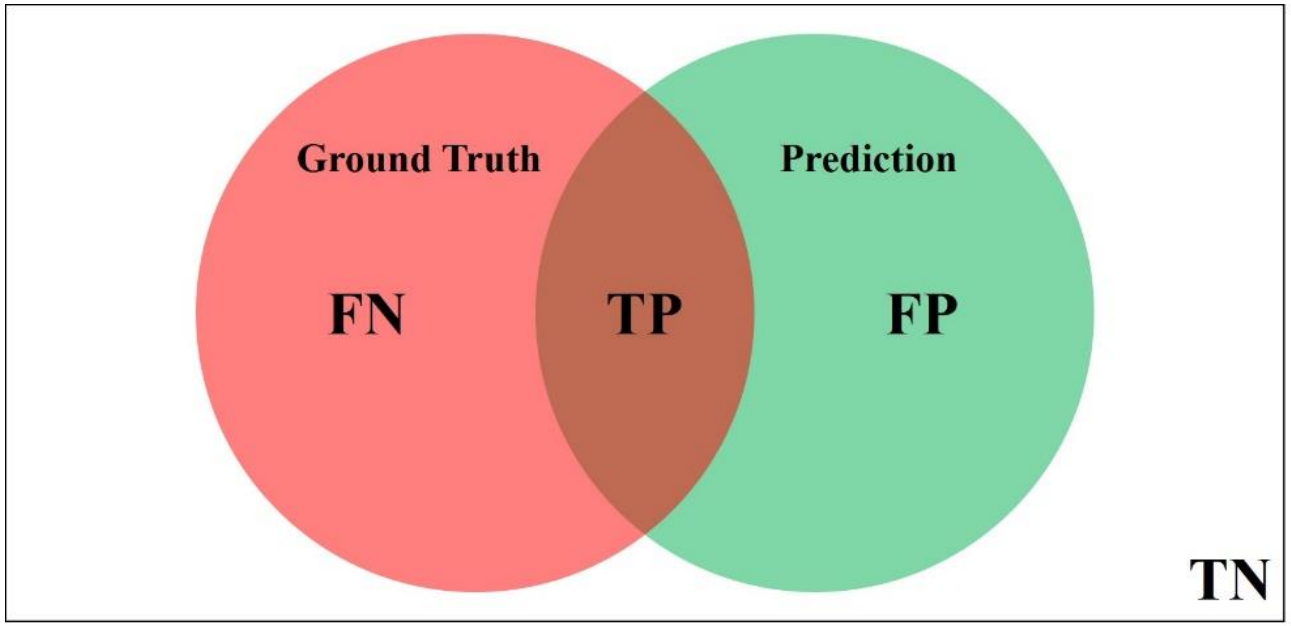
Dataset & Metric

Dataset:

- Pascal VOC 2012
- MS COCO

Metric:

- Mean Intersection over Union (mIoU)
- $mIoU = \frac{TP}{FP+FN+TP}$



Paper list

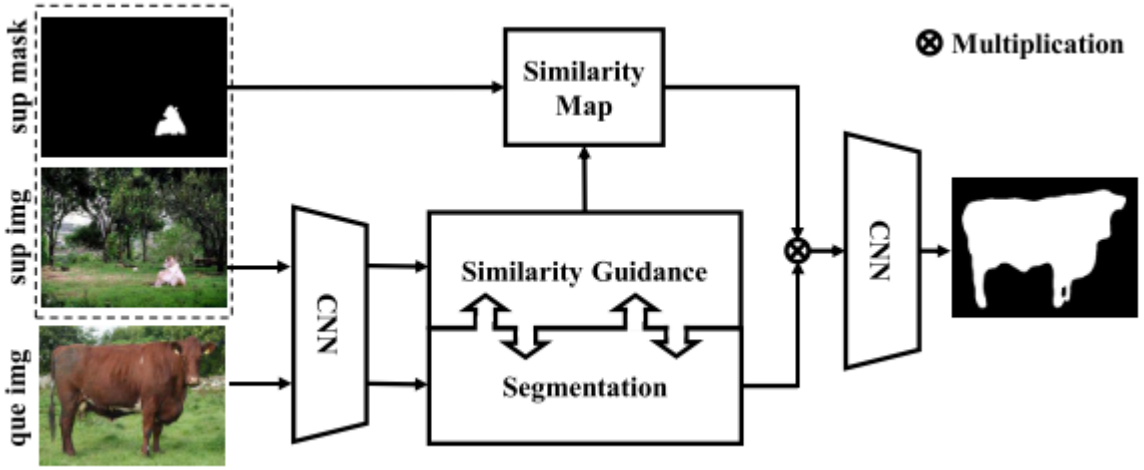
- SG-One: Similarity Guidance Network for One-Shot Semantic Segmentation (2018)
- PANet: Few-Shot Image Semantic Segmentation with Prototype Alignment (2019)
- Prior Guided Feature Enrichment Network for Few-Shot Segmentation (2020)
- Adaptive Prototype Learning and Allocation for Few-Shot Segmentation (2021)
- Edge-Labeling Graph Neural Network for Few-shot Learning (2019)

SG-One: Similarity Guidance Network for One-Shot Semantic Segmentation

SG-One

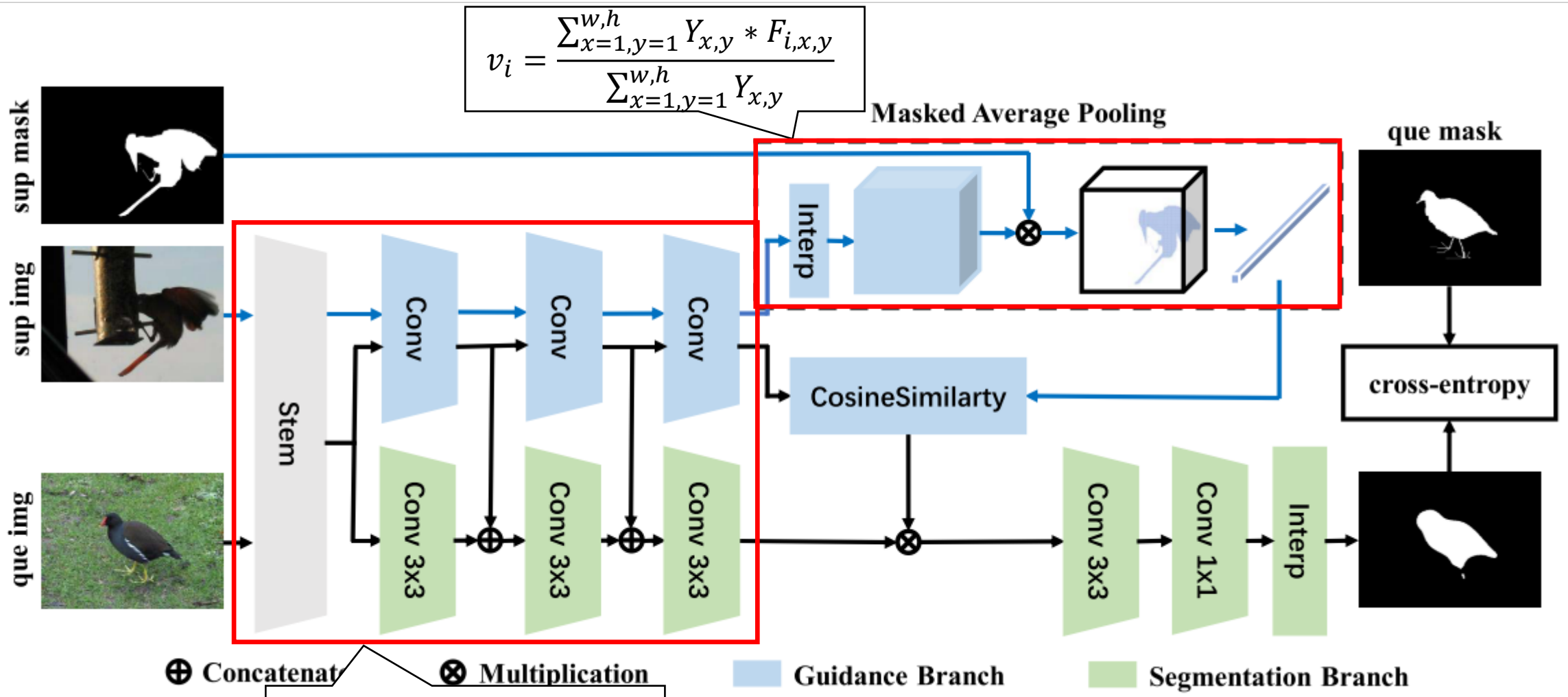
Key points:

- Masked Average Pooling strategy
- Build relationship between support and query with cosine similarity
- Unified network structure



SG-One

$$v_i = \frac{\sum_{x=1,y=1}^{w,h} Y_{x,y} * F_{i,x,y}}{\sum_{x=1,y=1}^{w,h} Y_{x,y}}$$



SG-One

K-shot Testing:

- Ensemble the K segmentation mask with equation:

$$\hat{Y}_{x,y} = max(\hat{Y}^1_{x,y}, \hat{Y}^2_{x,y}, \cdots, \hat{Y}^K_{x,y})$$

- Average the K representative vectors

MEAN IOU RESULTS OF ONE-SHOT SEGMENTATION ON THE PASCAL-5^i DATASET. THE BEST RESULTS ARE IN BOLD

Methods[1]	PASCAL-5^0	PASCAL-5^1	PASCAL-5^2	PASCAL-5^3	Mean
1-NN	25.3	44.9	**41.7**	18.4	32.6
LogReg	26.9	42.9	37.1	18.4	31.4
Siamese	28.1	39.9	31.8	25.8	31.4
OSVOS [37]	24.9	38.8	36.5	30.1	32.6
OSLSM [15]	33.6	55.3	40.9	33.5	40.8
co-FCN [16]	36.7	50.6	44.9	32.4	41.1
SG-One(Ours)	**40.2**	**58.4**	**48.4**	**38.4**	**46.3**

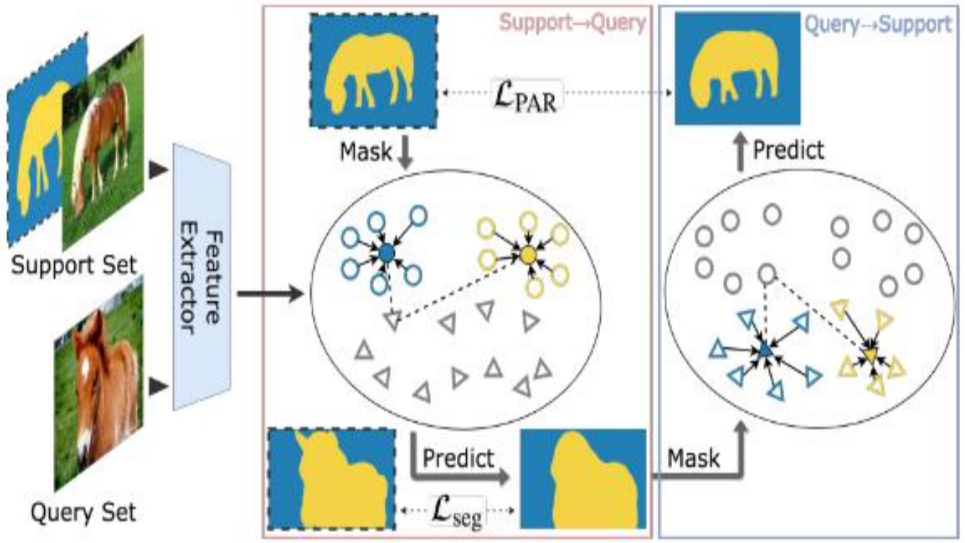
PANet: Few-Shot Image Semantic Segmentation with Prototype Alignment

PANet

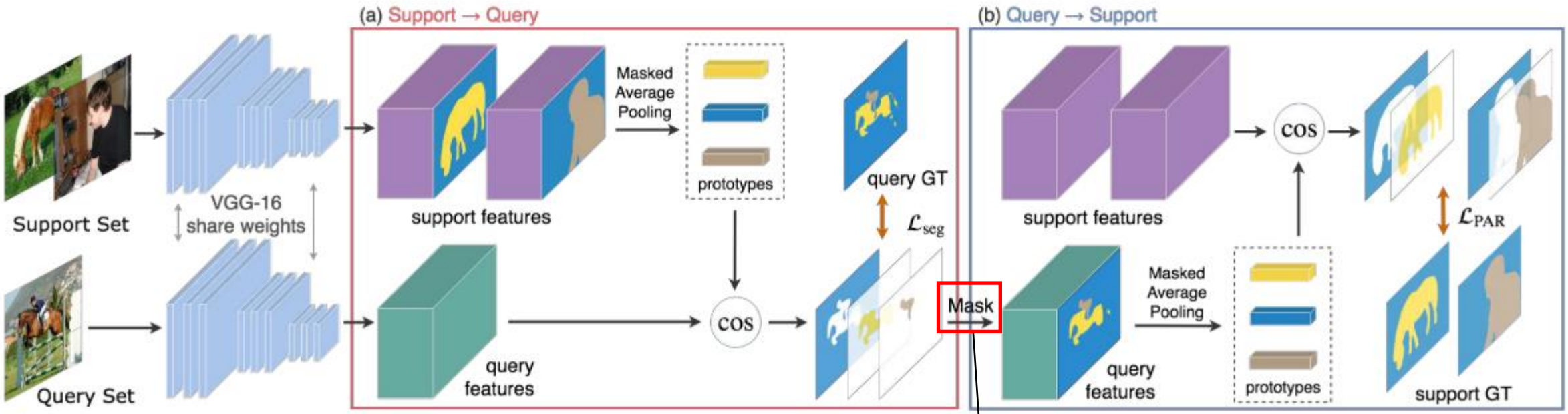
北京理工大学 BEIJING INSTITUTE OF TECHNOLOGY

Key points:

- Propose a non-parametric metric learning to extract knowledge and perform segmentation
- Design a Prototype Alignment structure to fully exploit knowledge from the support



PANet


(a) Support → Query
(b) Query → Support
Support Set
Query Set
VGG-16
share weights
support features
Masked
Average
Pooling
prototypes
query GT
$\mathcal{L}_{seg}$
query
features
COS
Mask
support features
COS
$\mathcal{L}_{PAR}$
Masked
Average
Pooling
query
features
prototypes
support GT
$\widehat{M}_q^{(x,y)} = \underset{j}{\operatorname{argmin}} \widetilde{M}_{q;j}^{(x,y)}$

PANet

北京理工大学 BEIJING INSTITUTE OF TECHNOLOGY

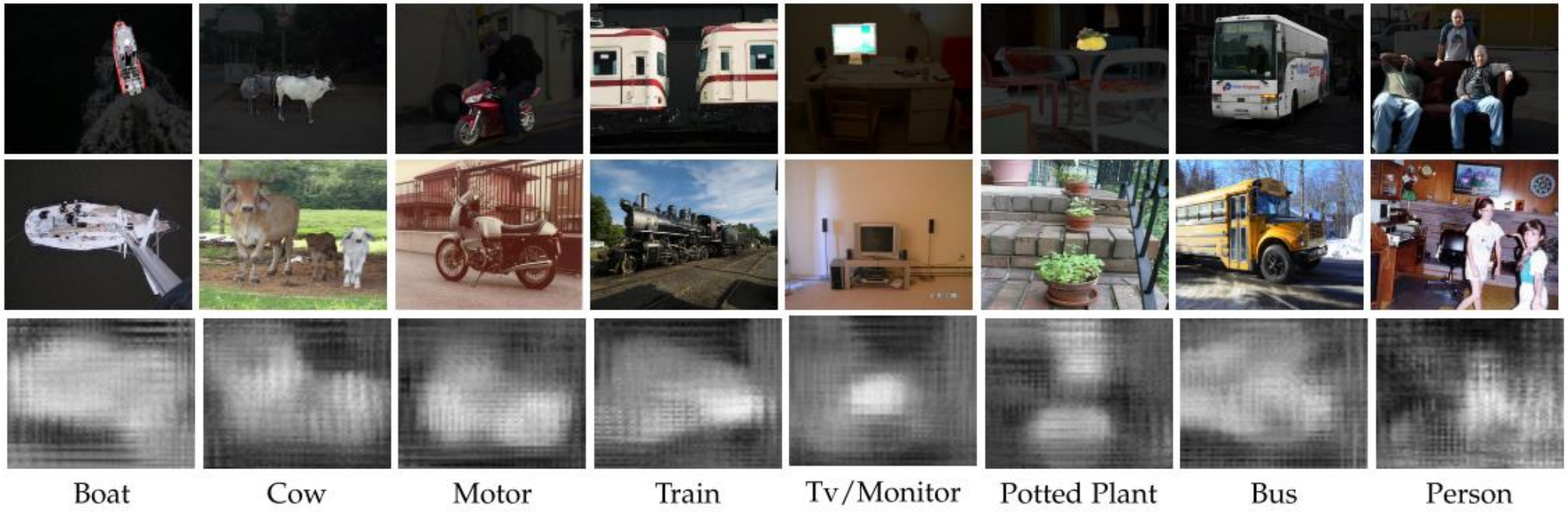
Method	1-shot					5-shot					Δ	#Params
	split-1	split-2	split-3	split-4	Mean	split-1	split-2	split-3	split-4	Mean	Mean	
OSLSM [21]	33.6	55.3	40.9	33.5	40.8	35.9	58.1	42.7	39.1	43.9	3.1	272.6M
co-FCN [16]†	36.7	50.6	44.9	32.4	41.1	37.5	50.0	44.1	33.9	41.4	0.3	34.2M
SG-One [28]	40.2	**58.4**	48.4	38.4	46.3	41.9	58.6	48.6	39.4	47.1	0.8	19.0M
PANet-init	30.8	40.7	38.3	31.4	35.3	41.6	52.7	51.6	40.8	46.7	**11.4**	14.7M
PANet	**42.3**	58.0	**51.1**	**41.2**	**48.1**	**51.8**	**64.6**	**59.8**	**46.5**	**55.7**	7.6	**14.7M**

Prior Guided Feature Enrichment Network for Few-Shot Segmentation

PFENet

PFENet mainly wants to solve following two problems:

- High-level features using in a few-shot model may cause performance drop
  - Using fixed high-level features to yield the prior mask
- Spatial Inconsistency
  - Proposing a FPN-like Feature Enrichment Module with residual blocks

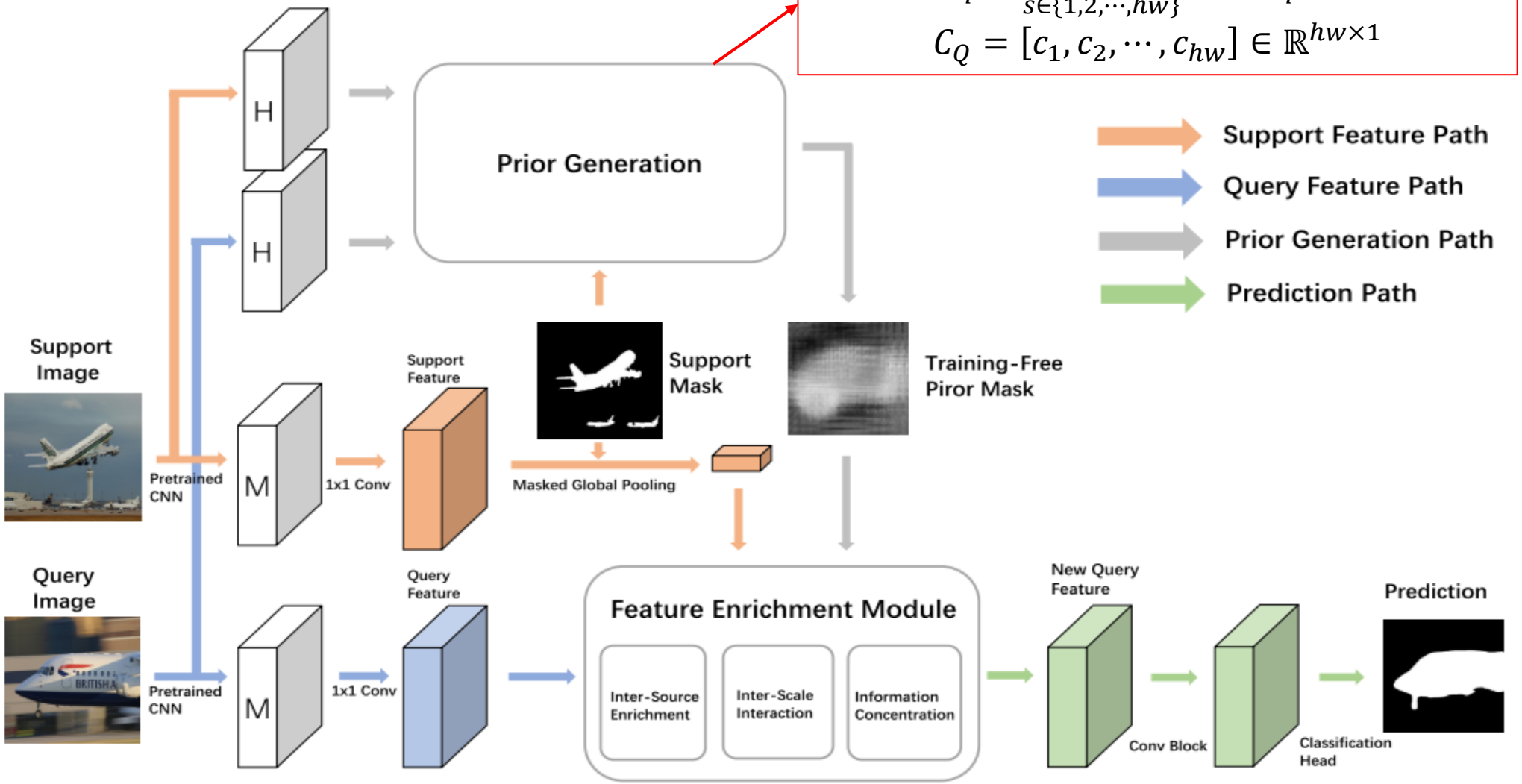


PFENet

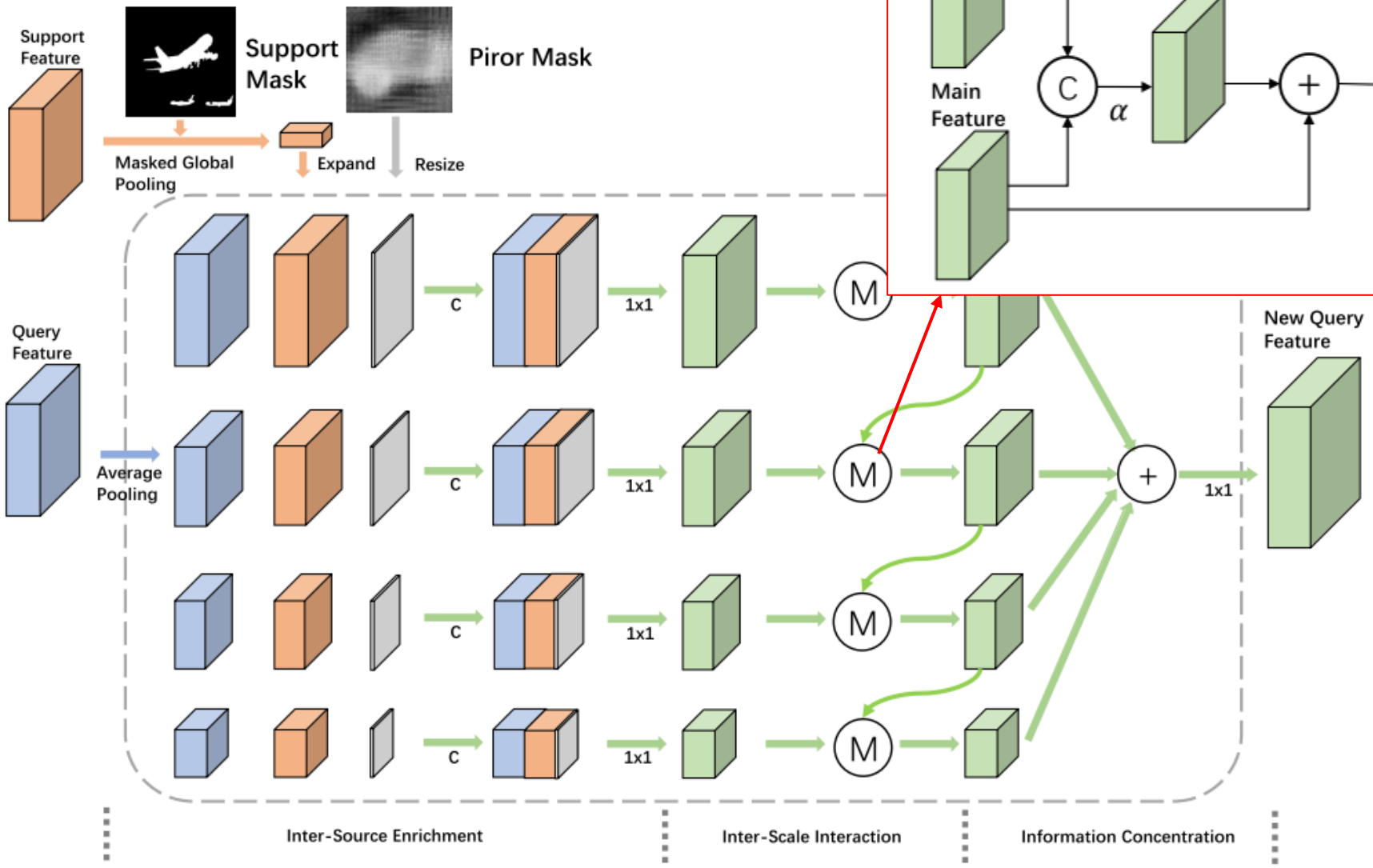
$$\cos(x_q, x_s) = \frac{x_q^T x_s}{\|x_q\|\|x_s\|} \quad q, s \in \{1,2,\cdots,hw\}$$

$$c_q = \max_{s\in\{1,2,\cdots,hw\}}(\cos(x_q, x_s))$$

$$C_Q = [c_1, c_2, \cdots, c_{hw}] \in \mathbb{R}^{hw\times 1}$$



PFENet


Support Feature
Support Mask
Piror Mask
Masked Global Pooling
Expand
Resize
Query Feature
Average Pooling
C
1x1
M
New Query Feature
Inter-Source Enrichment
Inter-Scale Interaction
Information Concentration
Auxiliary Feature
Main Feature
α
β
Refined Feature

PFENet

北京理工大学 BEIJING INSTITUTE OF TECHNOLOGY

Methods	1-Shot					5-Shot					Params
	Fold-0	Fold-1	Fold-2	Fold-3	Mean	Fold-0	Fold-1	Fold-2	Fold-3	Mean	
VGG-16 Backbone											
$OSLSM_{2017}$ [33]	33.6	55.3	40.9	33.5	40.8	35.9	58.1	42.7	39.1	44.0	276.7M
$co\text{-}FCN_{2018}$ [29]	36.7	50.6	44.9	32.4	41.1	37.5	50.0	44.1	33.9	41.4	34.2M
$SG\text{-}One_{2018}$ [58]	40.2	58.4	48.4	38.4	46.3	41.9	58.6	48.6	39.4	47.1	19.0M
AMP_{2019} [35]	41.9	50.2	46.7	34.7	43.4	41.8	55.5	50.3	39.9	46.9	34.7M
$PANet_{2019}$ [45]	42.3	58.0	51.1	41.2	48.1	51.8	64.6	**59.8**	46.5	55.7	14.7M
$FWBF_{2019}$ [28]	47.0	59.6	52.6	48.3	51.9	50.9	62.9	56.5	50.1	55.1	-
Ours	**56.9**	**68.2**	**54.4**	**52.4**	**58.0**	**59.0**	**69.1**	54.8	**52.9**	**59.0**	**10.4M**
ResNet-50 Backbone											
$CANet_{2019}$ [54]	52.5	65.9	51.3	51.9	55.4	55.5	67.8	51.9	53.2	57.1	19.0M
$PGNet_{2019}$ [53]	56.0	66.9	50.6	50.4	56.0	54.9	67.4	51.8	53.0	56.8	17.2M
Ours	**61.7**	**69.5**	**55.4**	**56.3**	**60.8**	**63.1**	**70.7**	**55.8**	**57.9**	**61.9**	**10.8M**
ResNet-101 Backbone											
$FWBF_{2019}$ [28]	51.3	64.5	**56.7**	52.2	56.2	54.8	67.4	**62.2**	55.3	59.9	-
Ours	**60.5**	**69.4**	54.4	**55.9**	**60.1**	**62.8**	**70.4**	54.9	**57.6**	**61.4**	**10.8M**

Methods	*Backbone*	1-Shot					5-Shot				
		Fold-0	Fold-1	Fold-2	Fold-3	Mean	Fold-0	Fold-1	Fold-2	Fold-3	Mean
Class mIoU Evaluation											
$FWBF_{2019}$ [28]	VGG-16	18.4	16.7	19.6	25.4	20.0	20.9	19.2	21.9	28.4	22.6
Ours	VGG-16	**33.4**	**36.0**	**34.1**	**32.8**	**34.1**	**35.9**	**40.7**	**38.1**	**36.1**	**37.7**
$PANet_{2019}$† [45]	VGG-16	-	-	-	-	20.9	-	-	-	-	29.7
Ours†	VGG-16	**35.4**	**38.1**	**36.8**	**34.7**	**36.3**	**38.2**	**42.5**	**41.8**	**38.9**	**40.4**
$FWBF_{2019}$ [28]	ResNet-101	19.9	18.0	21.0	28.9	21.2	19.1	21.5	23.9	30.1	23.7
Ours	ResNet-101	**34.3**	**33.0**	**32.3**	**30.1**	**32.4**	**38.5**	**38.6**	**38.2**	**34.3**	**37.4**
Ours†	ResNet-101	36.8	41.8	38.7	36.7	38.5	40.4	46.8	43.2	40.5	42.7

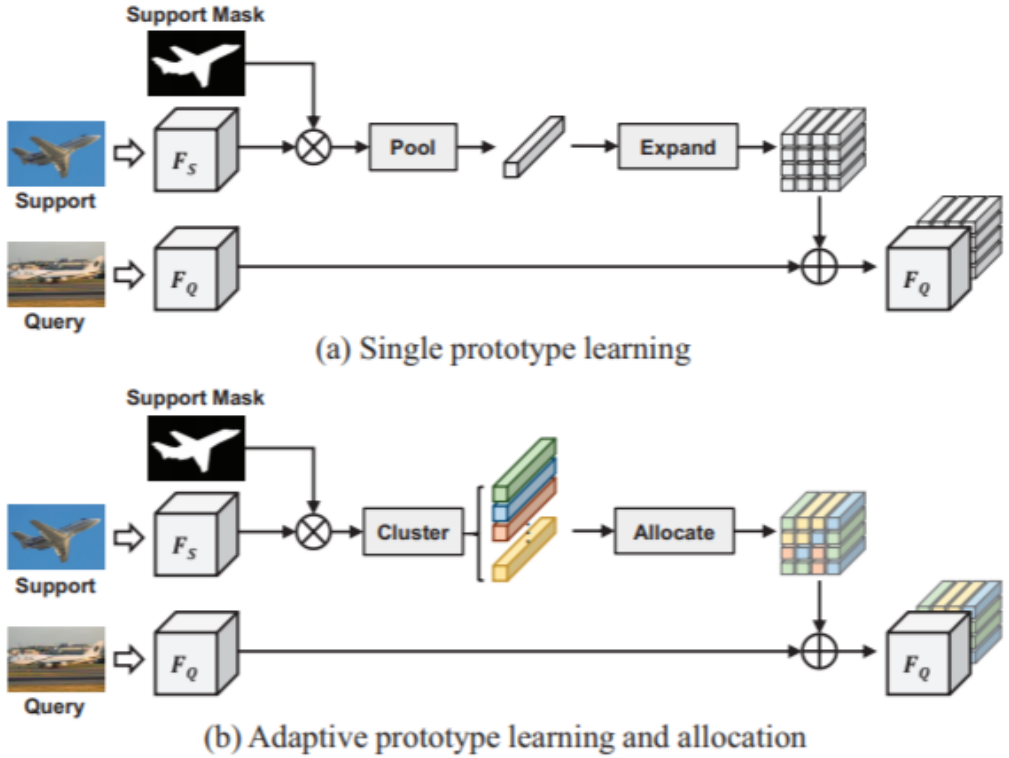
Adaptive Prototype Learning and Allocation for Few-Shot Segmentation

Adaptive Prototype Learning and Allocation for Few-Shot Segmentation(CVPR2021)

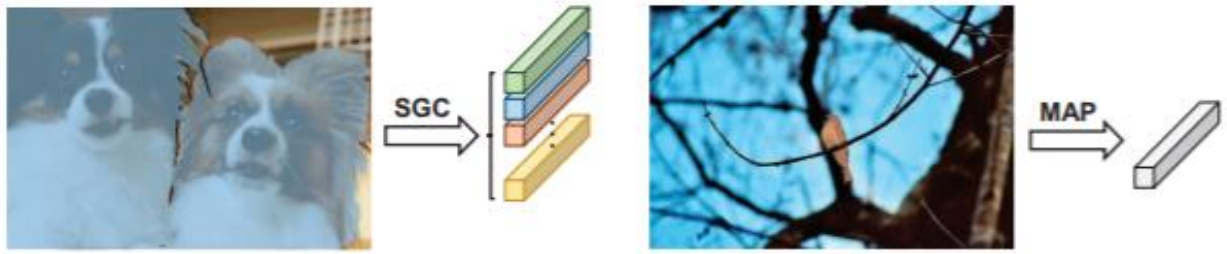
北京理工大学 BEIJING INSTITUTE OF TECHNOLOGY

Motivation

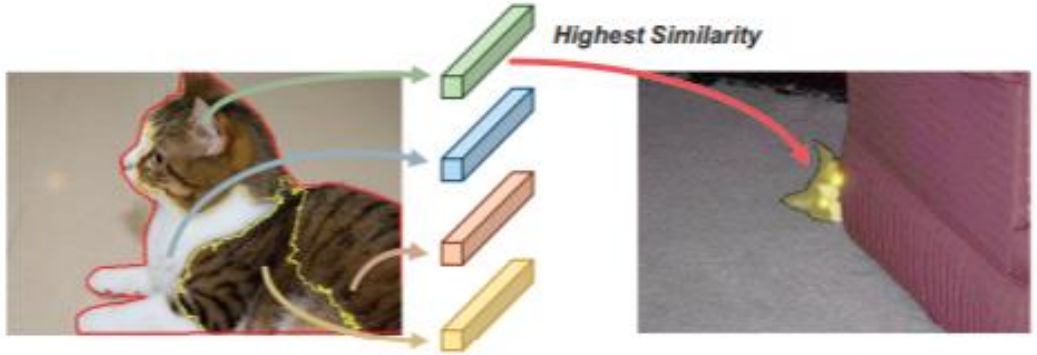
Using one prototype to represent all the information may lead to ambiguities



Adaptive Prototype Learning and Allocation for Few-Shot Segmentation



(a) SGC is adaptive to object scale variation



(b) GPA is adaptive to object shape variation

Adaptive Prototype Learning and Allocation for Few-Shot Segmentation

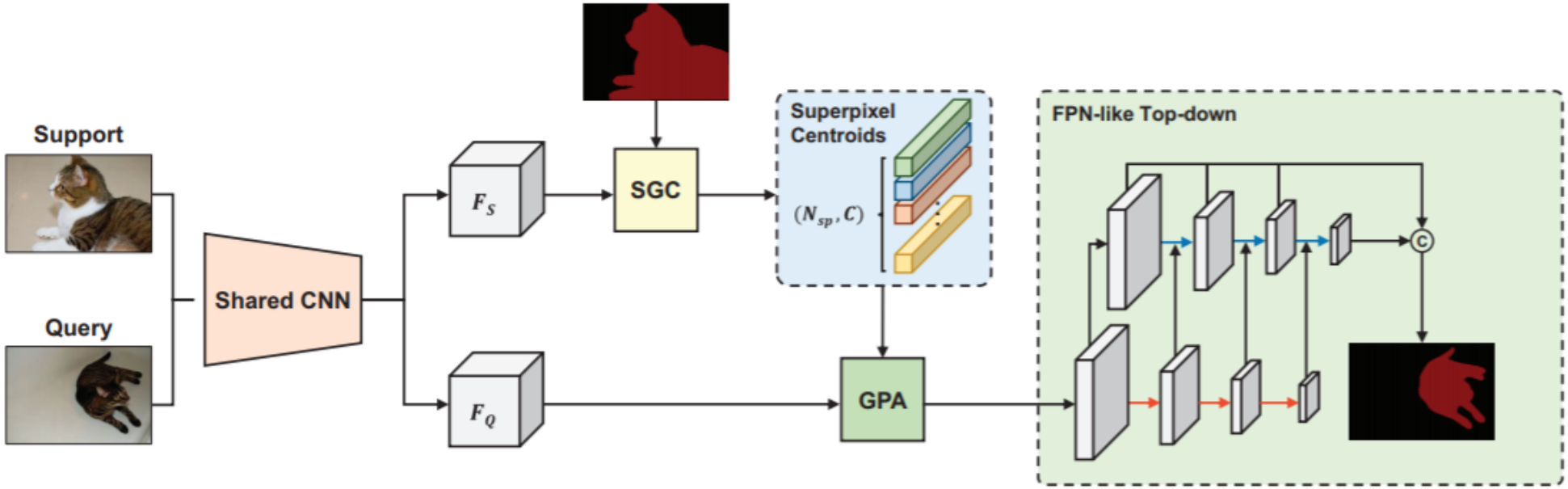


Figure 2. Overall architecture of the proposed Adaptive Superpixel-guided Network.

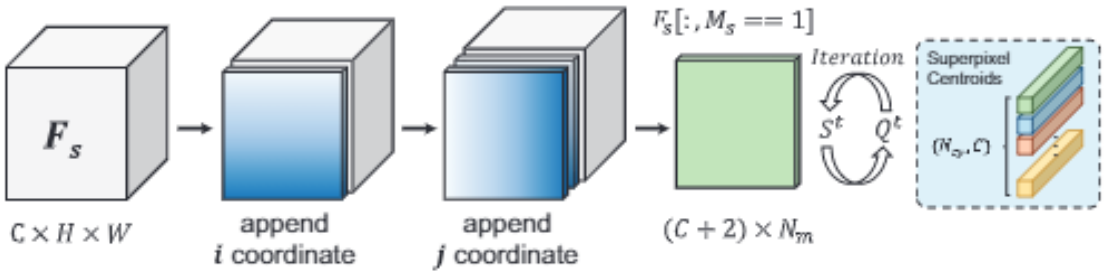
Super-pixel

Serve super-pixel centroids as prototypes

- concatenate the coordinates of each pixel with the support feature map
- Get foreground feature map with mask
- Iteration

$$D = \sqrt{(d_f)^2 + (d_s/r)^2}, \quad (1)$$

where d_f, d_s are the Euclidean distance for features and coordinate values, and r is a weighting factor. We filter



Adaptive Prototype Learning and Allocation for Few-Shot Segmentation

北京理工大学 BEIJING INSTITUTE OF TECHNOLOGY

Prototype Allocation

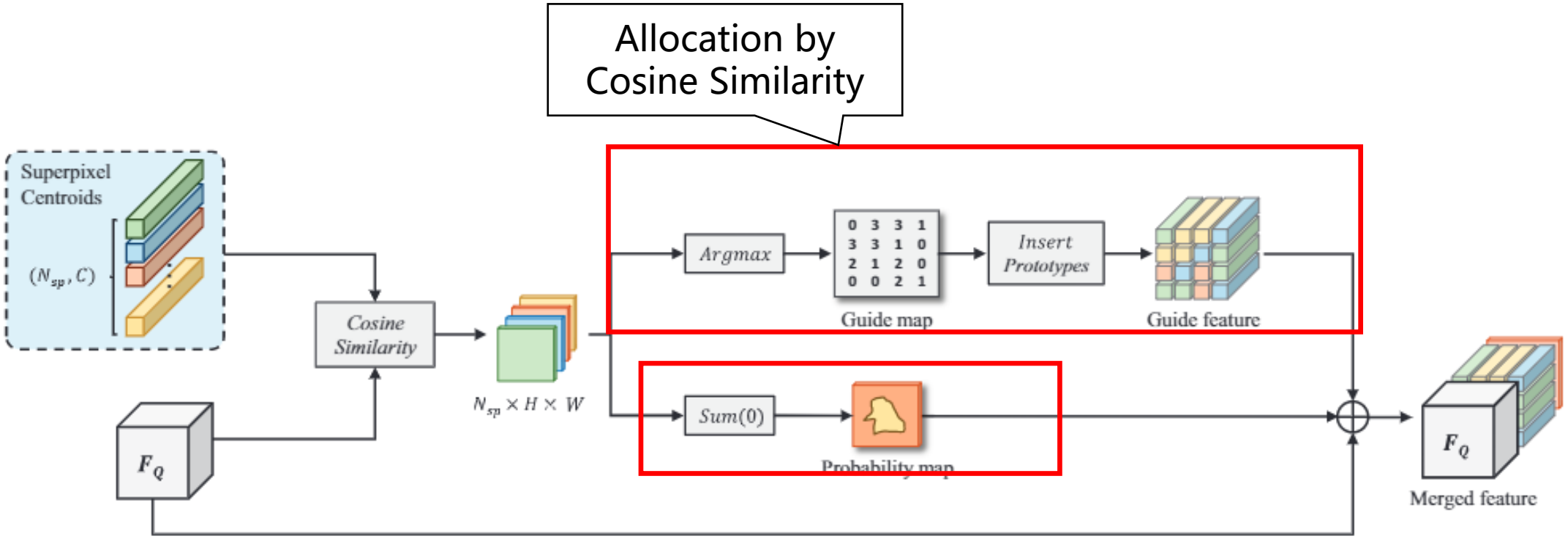


Figure 4. Illustration of proposed Guided Prototype Allocation.

Adaptability

Compute the number of prototypes with the number of pixels in the foreground

$$N_{sp} = \min(\left\lfloor \frac{N_m}{S_{sp}} \right\rfloor, N_{max}),$$

K-shot setting

Directly get all prototypes from K-shot

$$N_{sp} = max \sum_{i=1}^{k} N_{sp}^{i}$$

Adaptive Prototype Learning and Allocation for Few-Shot Segmentation

北京理工大学 BEIJING INSTITUTE OF TECHNOLOGY

Backbone	Methods	1-shot					5-shot					Δ
		s-0	s-1	s-2	s-3	mean	s-0	s-1	s-2	s-3	mean	
VGG-16	OSLSM [23]	33.60	55.30	40.90	33.50	40.80	35.90	58.10	42.70	39.10	43.95	3.15
	co-FCN [21]	36.70	50.60	44.90	32.40	41.10	37.50	50.00	44.10	33.90	41.40	0.30
	AMP [24]	41.90	50.20	46.70	34.40	43.40	40.30	55.30	49.90	40.10	46.40	3.00
	SG-One [42]	40.20	58.40	48.40	38.40	46.30	41.90	58.60	48.60	39.40	47.10	0.80
	PANet [32]	42.30	58.00	51.10	41.20	48.10	51.80	64.60	59.80	46.50	55.70	**7.60**
	FWB [20]	47.04	59.64	52.61	48.27	51.90	50.87	62.86	56.48	50.09	55.08	3.18
ResNet50	CANet† [40]	52.50	65.90	51.30	51.90	55.40	55.50	67.80	51.90	53.20	57.10	1.70
	PGNet† [39]	56.00	66.90	50.60	50.40	56.00	57.70	68.70	52.90	54.60	58.50	2.50
	RPMMs [34]	55.15	66.91	52.61	50.68	56.34	56.28	67.34	54.52	51.00	57.30	0.96
	SimPropNet [8]	54.82	67.33	54.52	52.02	57.19	57.20	68.50	58.40	56.05	60.04	2.85
	PPNet [18]	47.83	58.75	53.80	45.63	51.50	58.39	67.83	**64.88**	56.73	61.96	**10.46**
	PFENet [29]	**61.70**	**69.50**	55.40	**56.30**	**60.80**	63.10	**70.70**	55.80	**57.90**	61.90	1.10
	ASGNet (ours)	58.84	67.86	**56.79**	53.66	59.29	**63.66**	70.55	64.17	57.38	**63.94**	4.65
ResNet101	FWB [20]	51.30	64.49	56.71	52.24	56.19	54.84	67.38	62.16	55.30	59.92	3.73
	DAN† [31]	54.70	68.60	**57.80**	51.60	58.20	57.90	69.00	60.10	54.90	60.50	2.30
	PFENet [29]	**60.50**	**69.40**	54.40	**55.90**	**60.10**	62.80	70.40	54.90	**57.60**	61.40	1.30
	ASGNet (ours)	59.84	67.43	55.59	54.39	59.31	**64.55**	**71.32**	**64.24**	57.33	**64.36**	**5.05**

Edge-Labeling Graph Neural Network for Few-shot Learning

Edge-Labeling Graph Neural Network for Few-shot Learning (CVPR2019)

北京理工大学 BEIJING INSTITUTE OF TECHNOLOGY

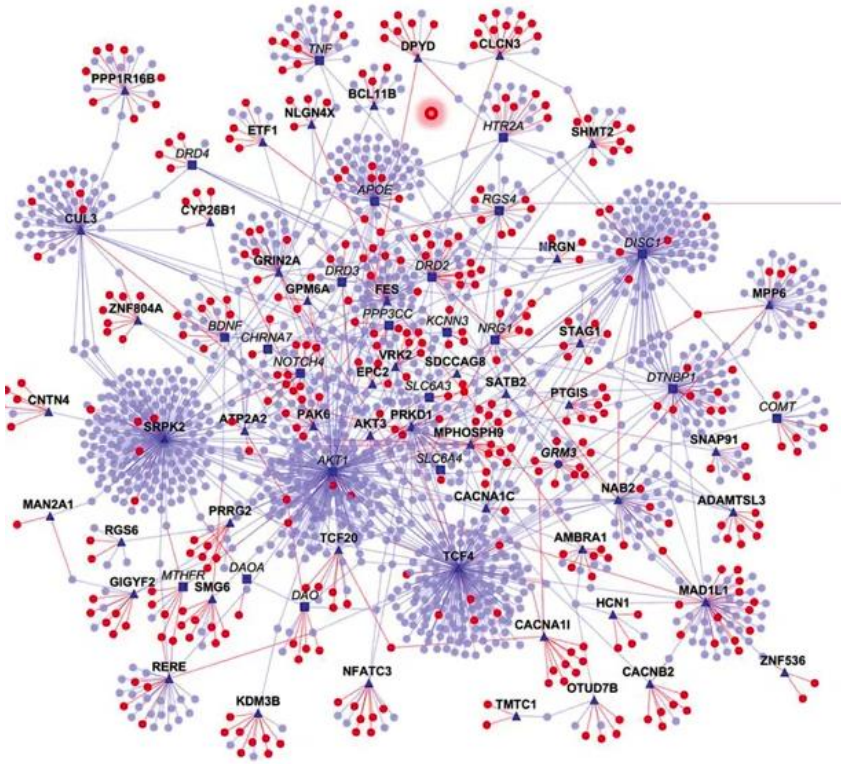
Graph Neural Network （GNN）

$$\mathbf{H} = F(\mathbf{H}, \mathbf{X})$$

$$\mathbf{O} = G(\mathbf{H}, \mathbf{X}_N)$$

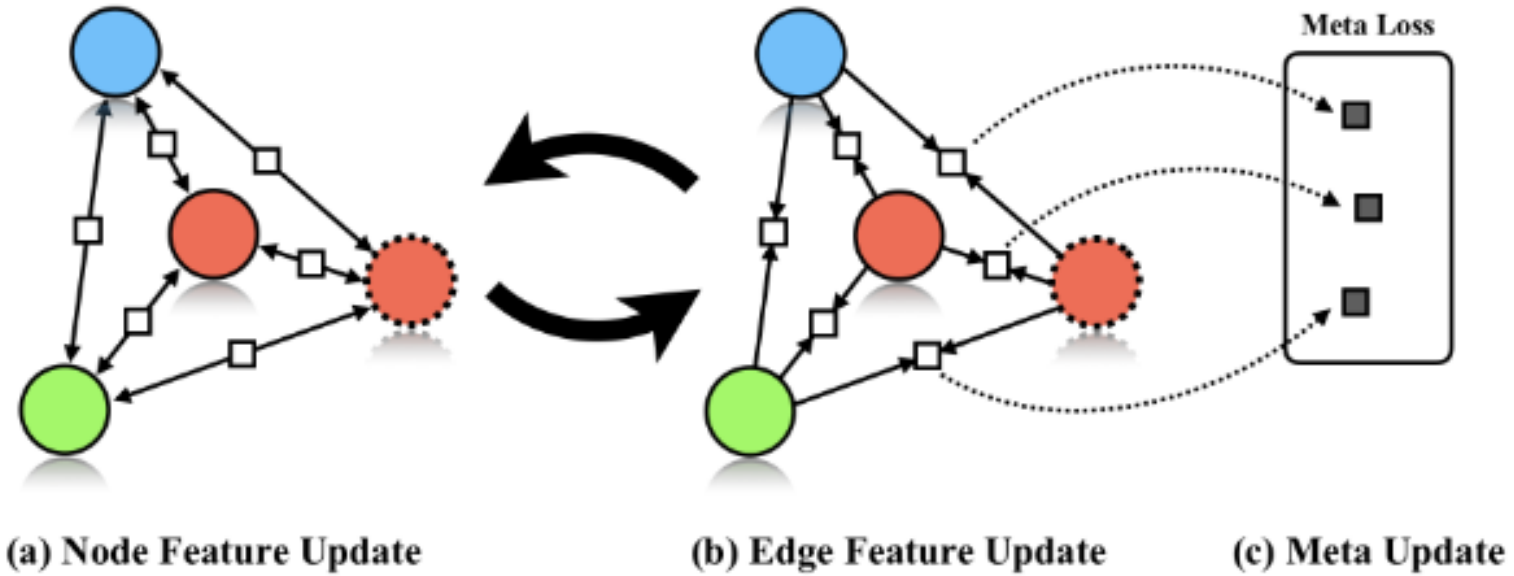
GCN

$$\mathbf{X}_s^{k+1} = \sigma\left(\left(\mathbf{I}_{|\mathcal{V}^s|} + \mathbf{D}_s^{-\frac{1}{2}}\mathbf{A}_s\mathbf{D}_s^{-\frac{1}{2}}\right)\mathbf{X}_s^k\mathbf{W}_s^k\right)$$



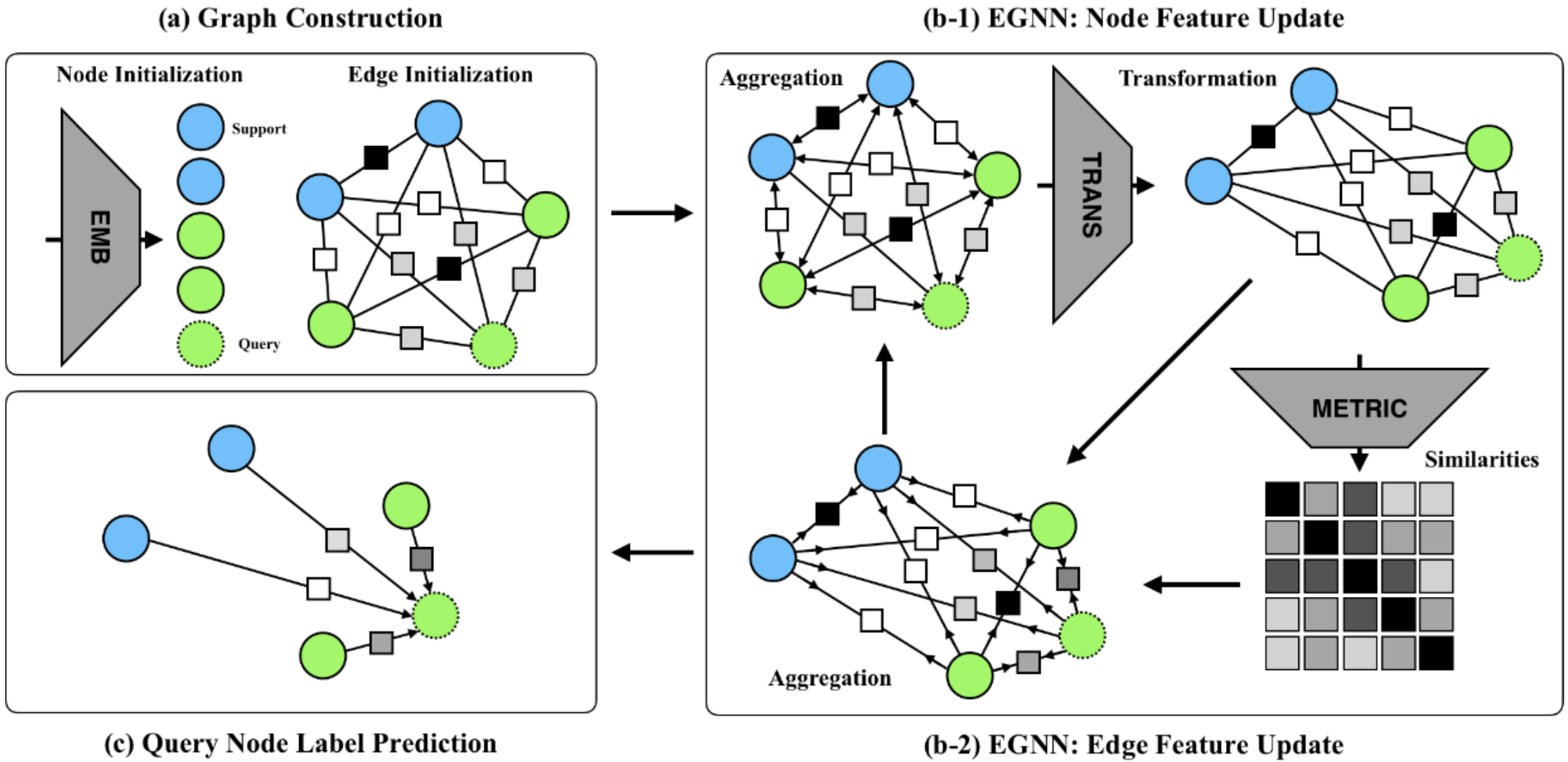
[1]Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. ICLR, 2017.

Edge-Labeling Graph Neural Network for Few-shot Learning (CVPR2019)



Not suitable for few-shot learning

Edge-Labeling Graph Neural Network for Few-shot Learning (CVPR2019)



Edge-Labeling Graph Neural Network for Few-shot Learning (CVPR2019)

北京理工大学 BEIJING INSTITUTE OF TECHNOLOGY

Initial edge label

$$\mathbf{e}_{ij}^0 = \begin{cases} [1||0], & \text{if } y_{ij} = 1 \text{ and } i,j \leq N \times K, \\ [0||1], & \text{if } y_{ij} = 0 \text{ and } i,j \leq N \times K, \\ [0.5||0.5], & \text{otherwise,} \end{cases}$$

Node update

$$\mathbf{v}_i^\ell = f_v^\ell([\sum_j \tilde{e}_{ij1}^{\ell-1}\mathbf{v}_j^{\ell-1} || \sum_j \tilde{e}_{ij2}^{\ell-1}\mathbf{v}_j^{\ell-1}]; \theta_v^\ell),$$

Edge update

$$\bar{e}_{ij1}^\ell = \frac{f_e^\ell(\mathbf{v}_i^\ell, \mathbf{v}_j^\ell; \theta_e^\ell) e_{ij1}^{\ell-1}}{\sum_k f_e^\ell(\mathbf{v}_i^\ell, \mathbf{v}_k^\ell; \theta_e^\ell) e_{ik1}^{\ell-1} / (\sum_k e_{ik1}^{\ell-1})}, \quad (4)$$

$$\bar{e}_{ij2}^\ell = \frac{(1 - f_e^\ell(\mathbf{v}_i^\ell, \mathbf{v}_j^\ell; \theta_e^\ell)) e_{ij2}^{\ell-1}}{\sum_k (1 - f_e^\ell(\mathbf{v}_i^\ell, \mathbf{v}_k^\ell; \theta_e^\ell)) e_{ik2}^{\ell-1} / (\sum_k e_{ik2}^{\ell-1})}, \quad (5)$$

$$\mathbf{e}_{ij}^\ell = \bar{\mathbf{e}}_{ij}^\ell / \|\bar{\mathbf{e}}_{ij}^\ell\|_1, \quad (6)$$

Edge-Labeling Graph Neural Network for Few-shot Learning (CVPR2019)

北京理工大學 BEIJING INSTITUTE OF TECHNOLOGY

Algorithm 1: The process of EGNN for inference

Input: $\mathcal{G} = (\mathcal{V}, \mathcal{E}; \mathcal{T})$, where $\mathcal{T} = \mathcal{S} \bigcup \mathcal{Q}$,
$\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N \times K}$, $\mathcal{Q} = \{\mathbf{x}_i\}_{i=N \times K+1}^{N \times K+T}$
Parameters: $\theta_{emb} \cup \{\theta_v^\ell, \theta_e^\ell\}_{\ell=1}^{L}$
Output: $\{\hat{y}_i\}_{i=N \times K+1}^{N \times K+T}$
Initialize: $\mathbf{v}_i^0 = f_{emb}(\mathbf{x}_i; \theta_{emb})$, $\mathbf{e}_{ij}^0$, $\forall i, j$
for $\ell = 1, \cdots, L$ **do**
  /* Node feature update */
  for $i = 1, \cdots, |V|$ **do**
    $\mathbf{v}_i^\ell \leftarrow \texttt{NodeUpdate}(\{\mathbf{v}_i^{\ell-1}\}, \{\mathbf{e}_{ij}^{\ell-1}\}; \theta_v^\ell)$
  end
  /* Edge feature update */
  for $(i, j) = 1, \cdots, |E|$ **do**
    $\mathbf{e}_{ij}^\ell \leftarrow \texttt{EdgeUpdate}(\{\mathbf{v}_i^\ell\}, \{\mathbf{e}_{ij}^{\ell-1}\}; \theta_e^\ell)$
  end
end
/* Query node label prediction */
$\{\hat{y}_i\}_{i=N \times K+1}^{N \times K+T} \leftarrow \texttt{Edge2NodePred}(\{y_i\}_{i=1}^{N \times K}, \{\mathbf{e}_{ij}^L\})$

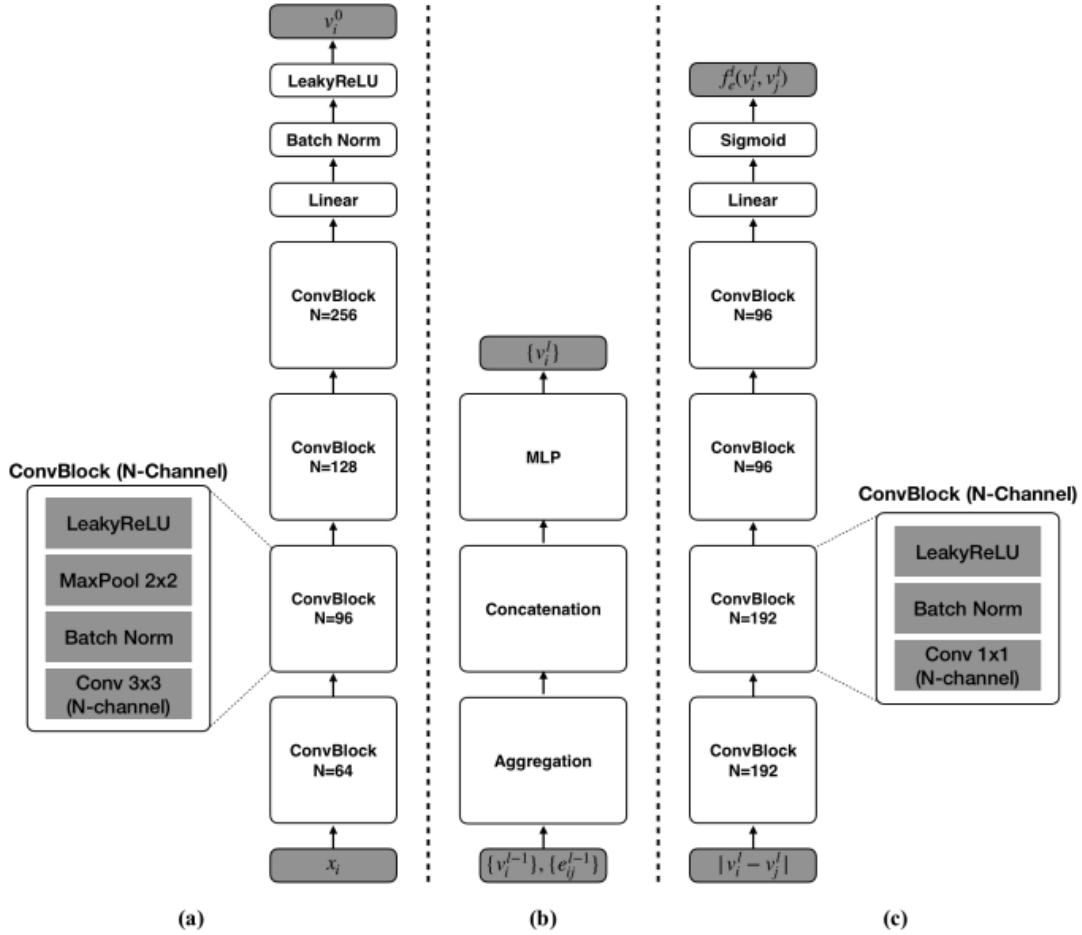


Figure 3: Detailed network architectures used in EGNN. (a) Embedding network f_{emb}. (b) Feature (node) transformation network f_v^ℓ. (c) Metric network f_e^ℓ.

德以明理 学以精工

Edge-Labeling Graph Neural Network for Few-shot Learning (CVPR2019)

北京理工大學 BEIJING INSTITUTE OF TECHNOLOGY

(a) *mini*ImageNet

Model	Trans.	5-Way 5-Shot
Matching Networks [2]	No	55.30
Reptile [46]	No	62.74
Prototypical Net [3]	No	65.77
GNN [6]	No	66.41
EGNN	No	**66.85**
MAML [4]	BN	63.11
Reptile + BN [46]	BN	65.99
Relation Net [5]	BN	67.07
MAML+Transduction [4]	Yes	66.19
TPN [12]	Yes	69.43
TPN (Higher K) [12]	Yes	69.86
EGNN+Transduction	Yes	**76.37**

(b) *tiered*ImageNet

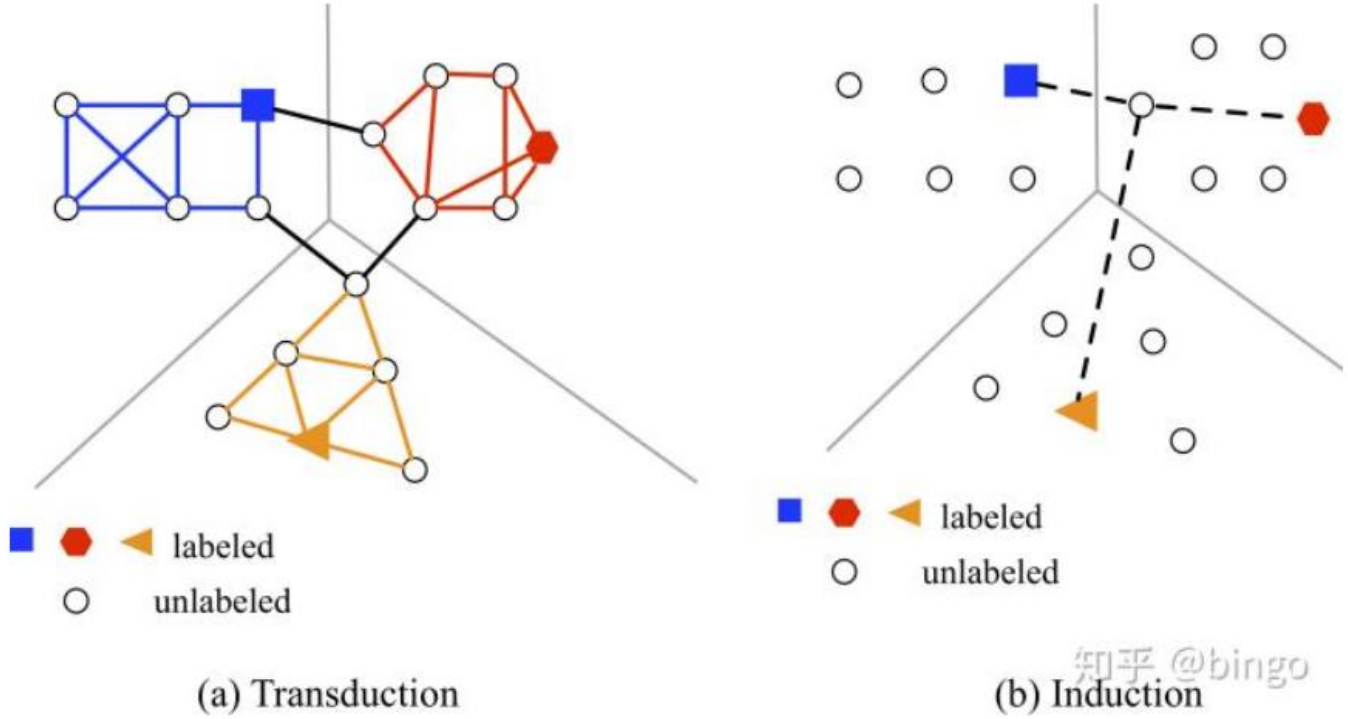
Model	Trans.	5-Way 5-Shot
Reptile [46]	No	66.47
Prototypical Net [3]	No	69.57
EGNN	No	**70.98**
MAML [4]	BN	70.30
Reptile + BN [46]	BN	71.03
Relation Net [5]	BN	71.31
MAML+Transduction [4]	Yes	70.83
TPN [12]	Yes	72.58
EGNN+Transduction	Yes	**80.15**

Edge-Labeling Graph Neural Network for Few-shot Learning (CVPR2019)

北京理工大学 BEIJING INSTITUTE OF TECHNOLOGY

Transductive inference

Classify the entire test set at once to alleviate the low-data problem



[2]Liu Y, Lee J, Park M, et al. Learning to propagate labels: Transductive propagation network for few-shot learning[C]//7th International Conference on Learning Representations, ICLR 2019. 2019.

Edge-Labeling Graph Neural Network for Few-