Position-Aware Recalibration Module:

Learning From Feature Semantics and Feature Position

Xu Ma, Song Fu
CAV Group, University of North Texas

Talk Pipeline

- Introduction
- Position-Aware Recalibration Module
  - Framework
  - Similarity Function
  - Semantic Normalization
  - Recalibration
  - Multi-head PRM
- Experiments
- Conclusion

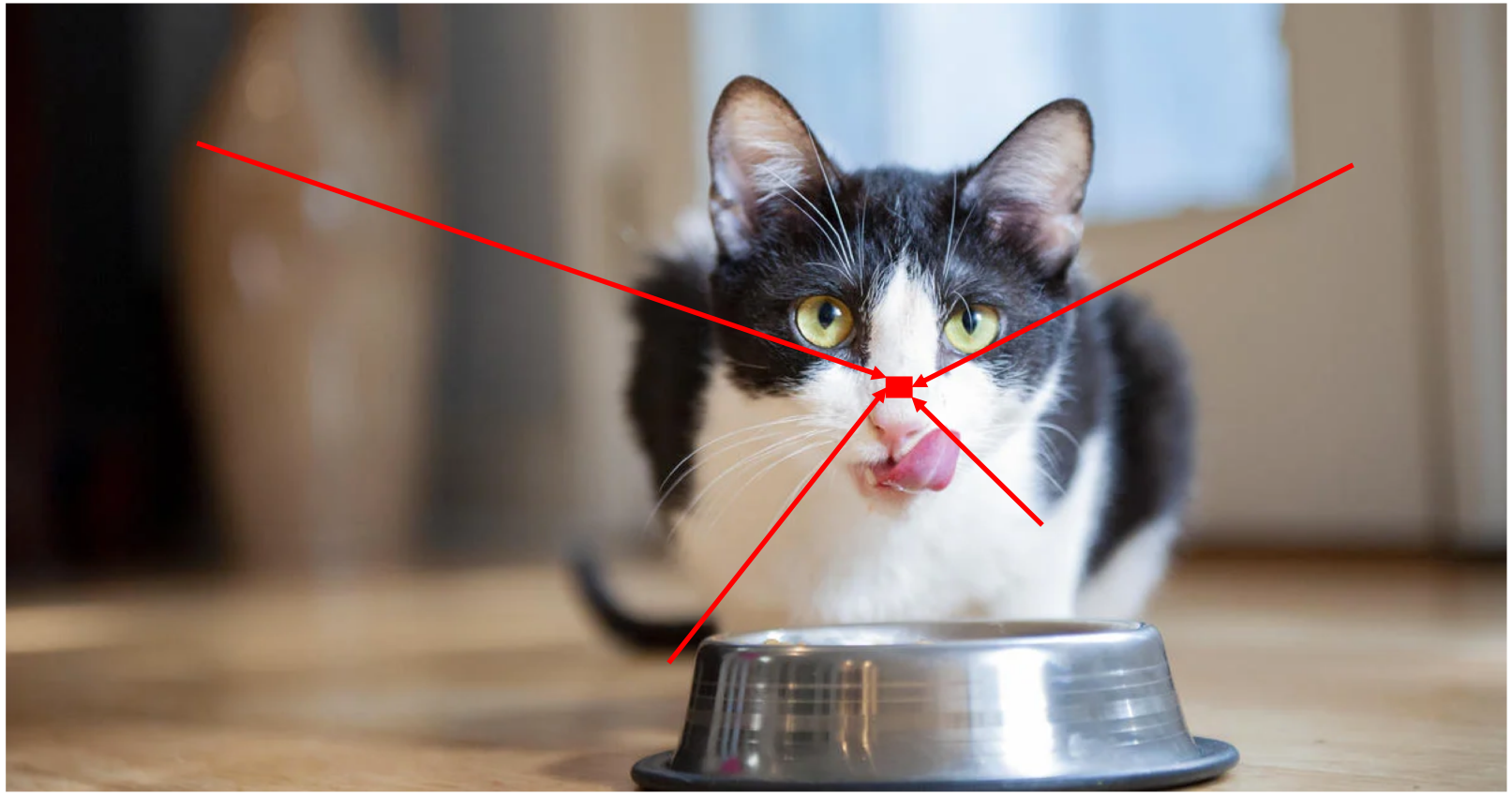
Introduction

How folks understand a given scene **effectively**?

By analyzing the target representation and considering the surrounding context.

Introduction

In Computer Vision, we have the **Attention Mechanism**

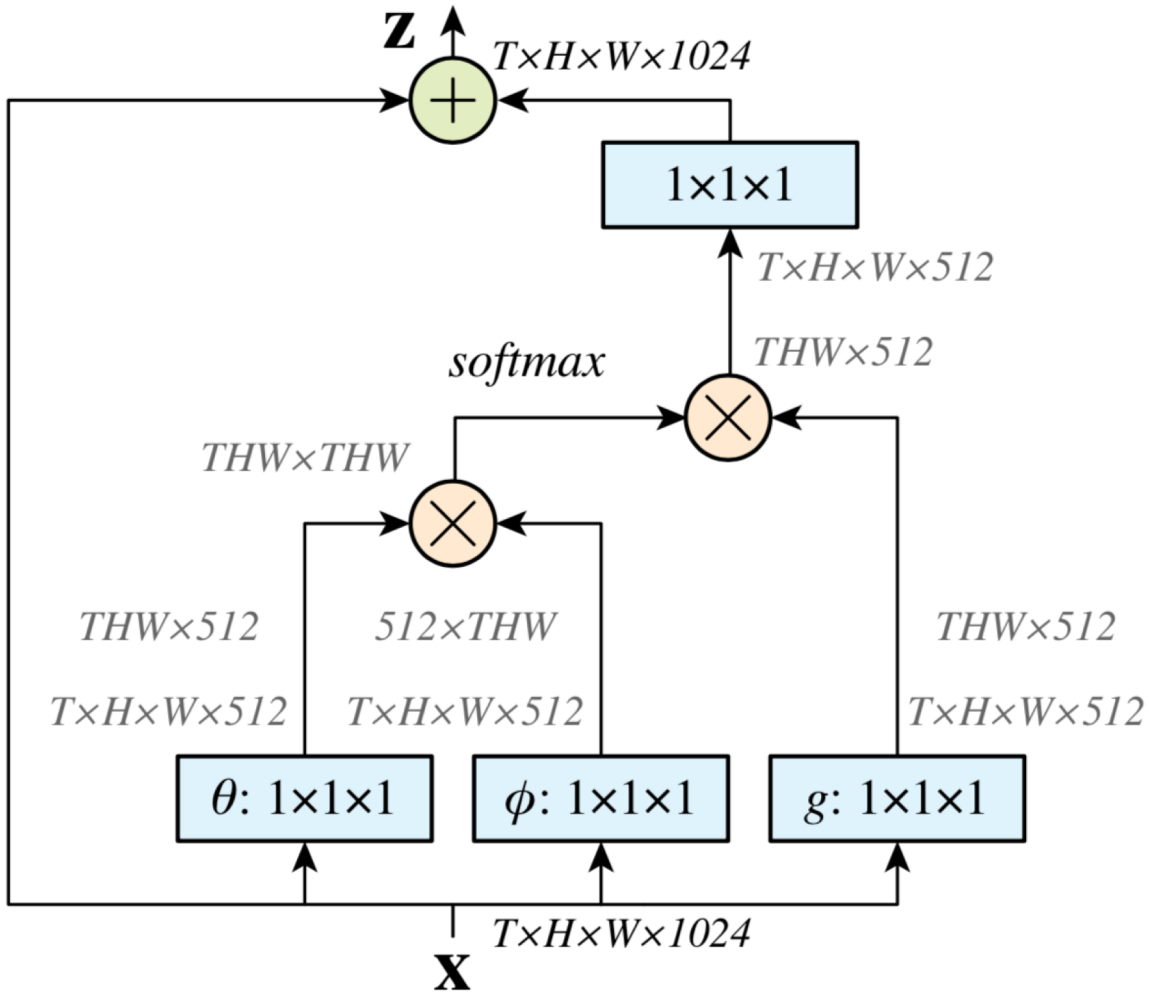


[1] Wang, Xiaolong, et al. "Non-local neural networks." *CVPR*. 2018.

Introduction

A classical Attention Mechanism: **Non-Local operation**

$$\mathbf{y}_i = \frac{1}{\mathcal{C}(\mathbf{x})} \sum_{\forall j} f(\mathbf{x}_i, \mathbf{x}_j) g(\mathbf{x}_j)$$



[1] Wang, Xiaolong, et al. "Non-local neural networks." *CVPR*. 2018.

Introduction

Such an operation is **problematic**：

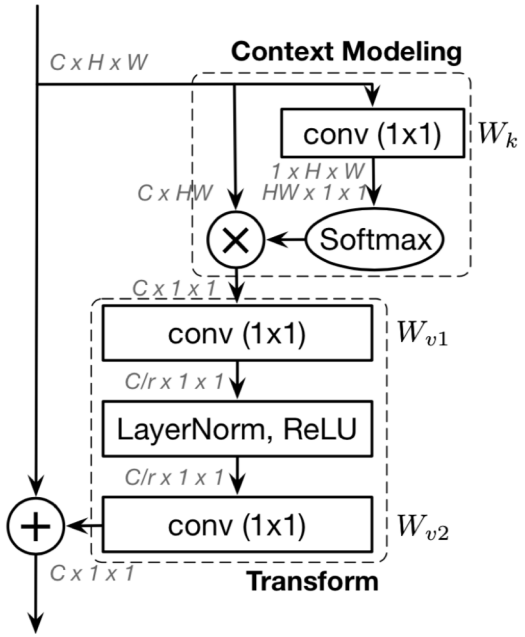
- High computational complexity
- Missing position information

Introduction

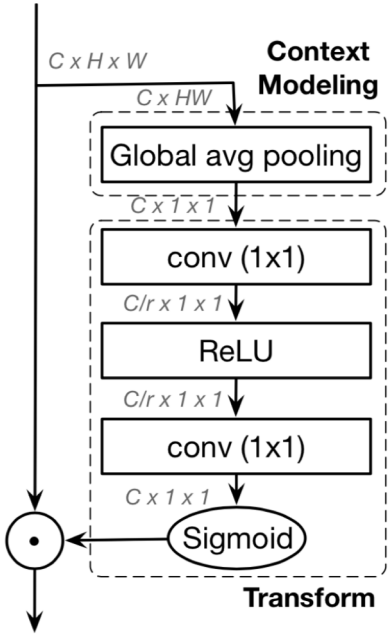
Problem 1: High computational complexity:

Complexity:
4 Conv(1024, 512), 2 matrix multiplication.

Improvement :
Change from query-specific to query-independent.



(GC Block)



(SE Block)

[2] Cao, Yue, et al. "Gcnet: Non-local networks meet squeeze-excitation networks and beyond." ICCV. 2019.
[3] Hu, Jie, et al. "Squeeze-and-excitation networks." CVPR. 2018.

Introduction

Problem 2: Missing position information :

Problem:
Even if we disrupt the spatial position of the features, there will be no change in the results for query-specific or query-independent operations.

Related Work :

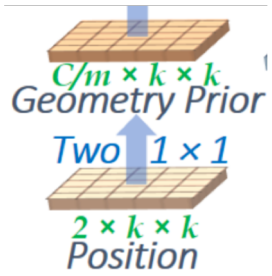
1) Transformer[4] (NLP)

Positional Embedding

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

2) LRNet[5]

Geometry Prior



3) AANet[6]

Relative positional encodings

$$O_h = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k^h}} + S_H^{rel} + S_W^{rel}\right)V$$

4)Explore Self-Attention[7]

Position encoding

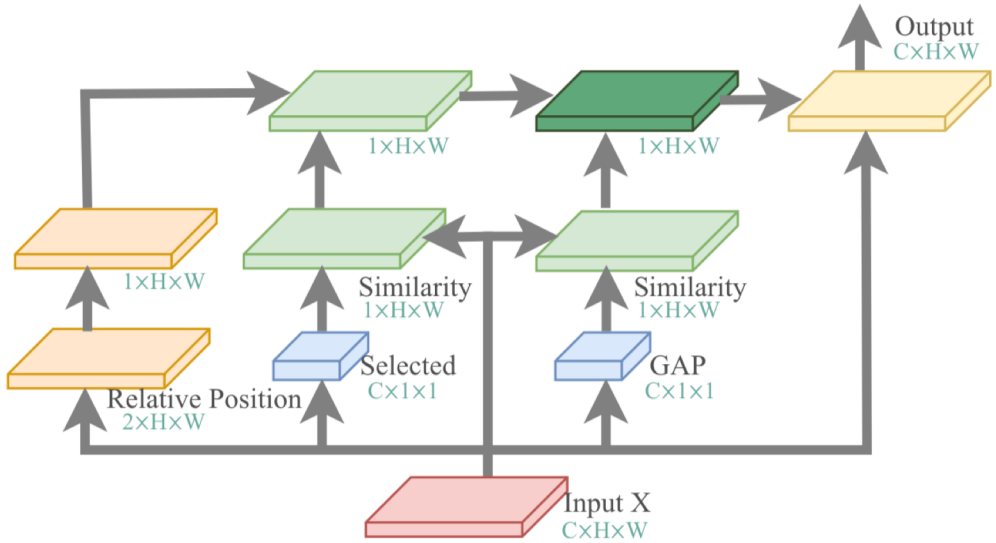
$$cat[p_i - p_j, \delta(x_i, x_j)]$$

[4]Vaswani, et al. "Attention is all you need." NeurIPS. 2017.
[5]Hu, Han, et al. "Local relation networks for image recognition." ICCV. 2019.
[6]Bello, et al. "Attention augmented convolutional networks." ICCV. 2019.
[7]Jiaya Jia, et al. "Exploring Self-attention for Image Recognition." CVPR. 2020.

PRM framework

Position-Aware Recalibration Module

We mainly improve CNNs by dealing with the following problems:
1) Efficiency; 2) Positional Information.



$$\mathbf{y} = sigmoid\left(\mathcal{N}\left(\mathbf{S}\right)\right) \otimes \mathbf{x}$$
$$s.t. \quad \mathbf{S} = \alpha\phi\left(\mathbf{x}, q\right) * \mathbf{D} + \beta\phi\left(\mathbf{x}, z\right),$$
$$\mathbf{D} = f_p\left(\left|p_{\mathbf{x}} - p_q\right|\right),$$

Fig. A basic Position-Aware Recalibration Module

Similarity Function

We explore the similarity from two aspects:

1) The most distinct feature; 2) the global context.

$$\phi(\mathbf{x}, q) \qquad \phi(\mathbf{x}, z)$$

- **Cosine similarity:** $$\phi(\mathbf{x}_i, q) = \frac{\mathbf{x}_i^{\mathsf{T}} q}{\max(\|\mathbf{x}_i\|_2 * \|q\|_2, \epsilon)},$$
- **L1-norm similarity:** $$\phi(\mathbf{x}_i, q) = \sum_{c=1}^{C'} - |\mathbf{x}_i^c - q^c|$$
- **Dot-product similarity:** $$\phi(\mathbf{x}_i, q) = \mathbf{x}_i^{\mathsf{T}} q$$ (default)

[8] https://www.zhihu.com/question/19734616/answer/349132554

PRM

Missing position information :

Our Solution: Gaussian Distribution.

Basic idea: the closer the distance, the greater the impact.

$$f_p\left(|p_{\mathbf{x}} - p_q|\right) = \frac{1}{d\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\theta|p_{\mathbf{x}} - p_q|}{d}\right)^2}$$

Step1: calculate the geometric relative position (in a range of [0,1]);
Step2: combine the width/height using learnable parameter **Theta**;
Step3: Encoding with a Gaussian Distribution [9].

[9] Luo, Wenjie, et al. "Understanding the effective receptive field in deep convolutional neural networks." NeurIPS. 2016.

Semantic Normalization

1) Calculate the mean/std of the similarity S over spatial dimension:

$$\mu = \frac{1}{HW}\sum_{i=1}^{HW}\mathbf{S}_i, \quad \sigma = \left(\frac{1}{HW}\sum_{i=1}^{HW}(\mathbf{S}_i - \mu)^2\right)^{\frac{1}{2}}$$

2) Normalize:

$$\mathbf{S} = f_s(\mathbf{S}) = \frac{\mathbf{S} - \mu}{\sigma + \epsilon},$$

3) Apply affine transformation:

$$\mathbf{S} = \lambda\mathbf{S} + \xi$$

PRM

Recalibration

1) Rescale using Sigmoid function;

2) Recalibrate using element—wise multiplication.

$$\mathbf{y} = sigmoid\left(\mathcal{N}\left(\mathbf{S}\right)\right) \otimes \mathbf{x}$$
$$s.t. \quad \mathbf{S} = \alpha\phi\left(\mathbf{x}, q\right) * \mathbf{D} + \beta\phi\left(\mathbf{x}, z\right),$$
$$\mathbf{D} = f_p\left(\left|p_{\mathbf{x}} - p_q\right|\right),$$

Input feature map

Normalization

Similarity of mean

Similarity of max

Position encoding

PRM

Is only one max-value point sufficient?

Question:

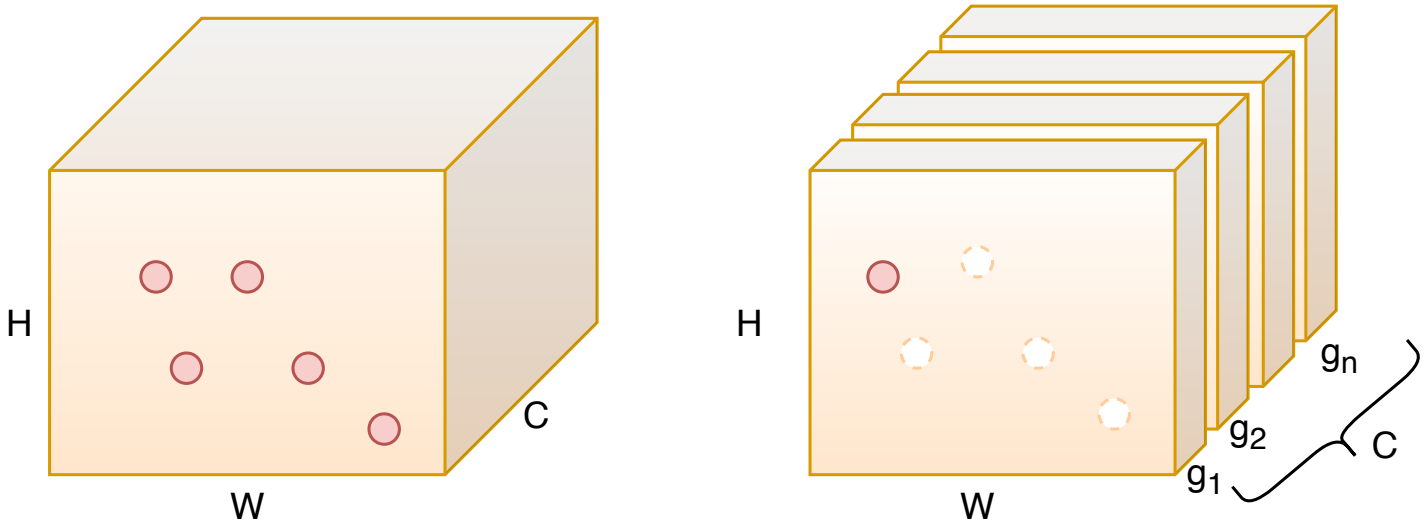
How do we decide the multiple key-points?

We do need multiple key points.

PRM

How do we decide the multiple key-points?

Two options:



(a) Select several points simultaneously

(b) Select for each group

Fig. Multiple Key-points scheme

(Reviewer's suggestion)

(Ours implementation)

Reason:

1) Complexity;
2) Decision of multiple points;
3) Performance;

PRM

Multi-head PRM

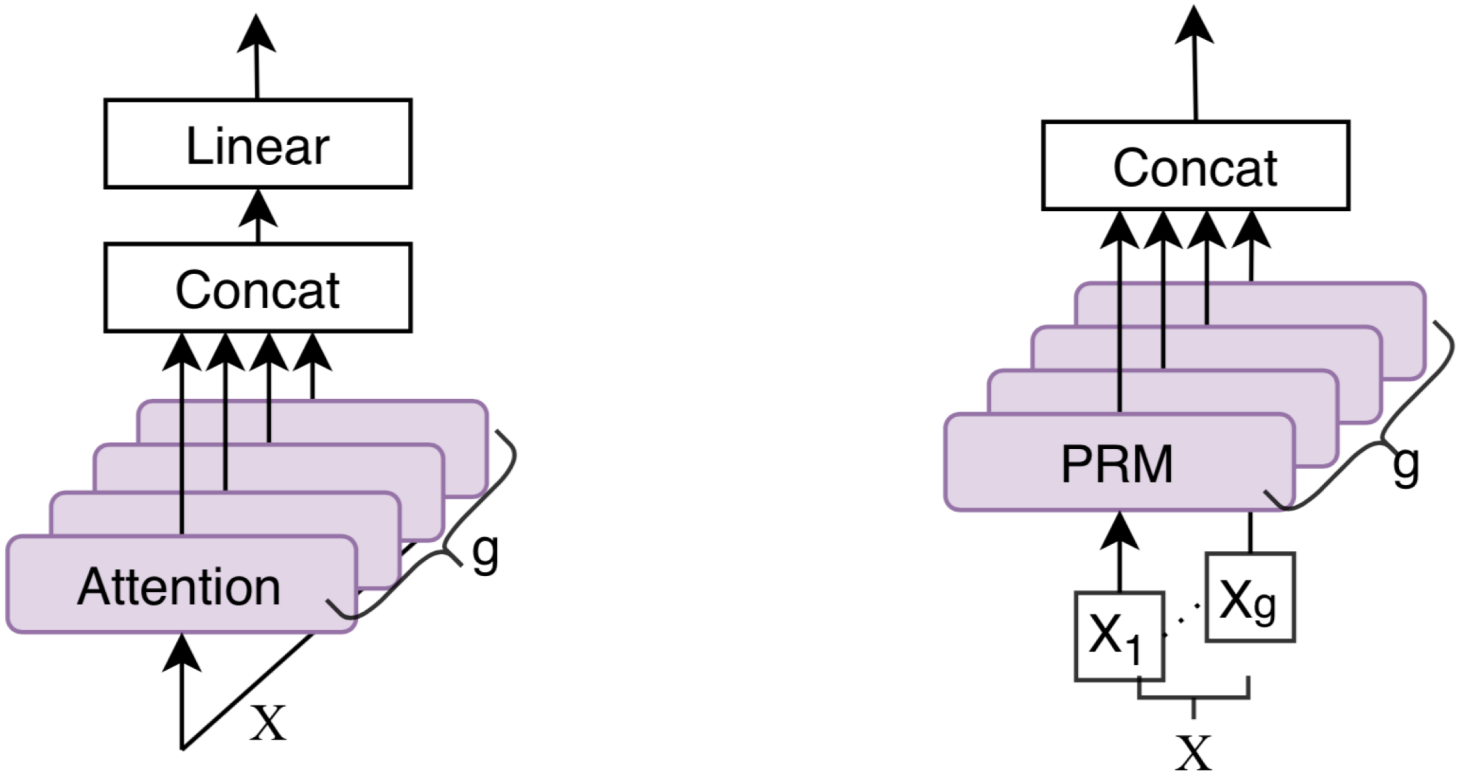


Fig. Left: multi-head attention; Right: multi-head PRM. Fig

Experiments

Comparison on ImageNet

Model	top-1 acc.	top-5 acc.	FLOPs (G)	Parameters (M)
ResNet50 [He *et al.*, 2016]	75.8974	92.7224	4.122	25.557
SE-ResNet50 [Hu *et al.*, 2018b]	77.2877	93.6478	4.130	28.088
GE-ResNet50 [Hu *et al.*, 2018a]	76.2357	92.9847	4.127	25.557
CBAM-ResNet50 [Woo *et al.*, 2018]	77.2840	93.6005	4.139	28.090
SK-ResNet50 [Li *et al.*, 2019b]	77.3657	93.5256	4.187	26.154
GC-ResNet50 [Cao *et al.*, 2019]	74.8966	92.2812	4.130	28.105
SGE-ResNet50 [Li *et al.*, 2019a]	77.5072	**93.6783**	4.127	25.560
PRM-ResNet50 (ours)	**77.6474**	93.6418	4.128	25.560

Table: Comparison results of classification accuracy (%) and complexity on ImageNet.

Experiments

Apply to other CNN models

Models	PRM	top-1	FLOPs	Parameters
ResNet50	w/o	75.8974	4.122G	25.56M
	w/	77.6474	4.128G	25.56M
MobileNetV2	w/o	71.0320	0.320G	3.51M
	w/	72.5466	0.321G	3.51M
MnasNet	w/o	71.7195	0.330G	4.38M
	w/	73.0147	0.331G	4.38M

Table: The performance of PRM on different CNN architectures.

Experiments

Ablation: Similarity function

network	similarity function	top-1	top-5
Resnet50	Cosine	77.6517	93.6711
	L1-norm	77.6012	93.5944
	Dotproduct	77.6474	93.6418
MobileNetV2	Cosine	72.5374	90.8714
	L1-norm	72.5570	90.7993
	Dotproduct	72.5466	90.8960

Table: The performance of different similarity functions.

Experiments

Ablation: Influence of each component

Feature dependency	Position encoding	Normal -ization	top-1	top-5
			68.9632	88.5902
✓			70.1863	89.4491
✓	✓		70.2388	89.4970
✓		✓	70.4467	89.5653
✓	✓	✓	70.5616	89.6635

Table: Ablation studies on each component of PRM. We conduct these ablation experiments based on ResNet-18.

Experiments

Ablation: Influence of head number

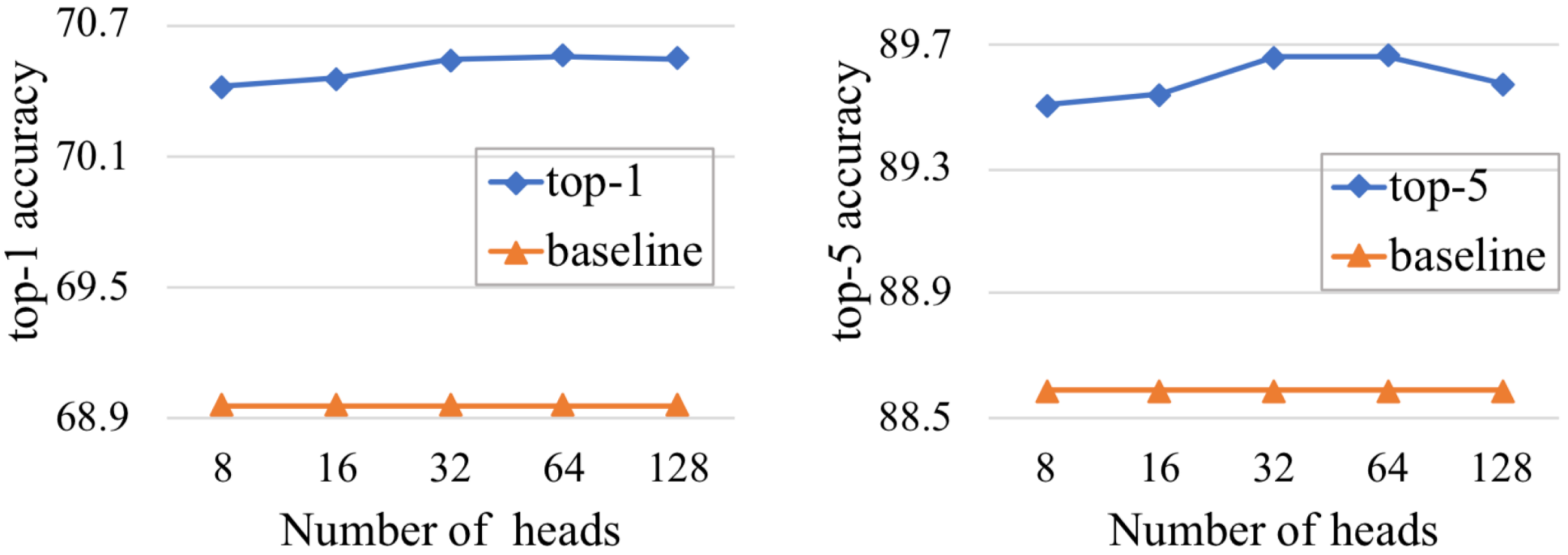


Fig: Influence of the head number in PRM .

Experiments

Visualization of PRM

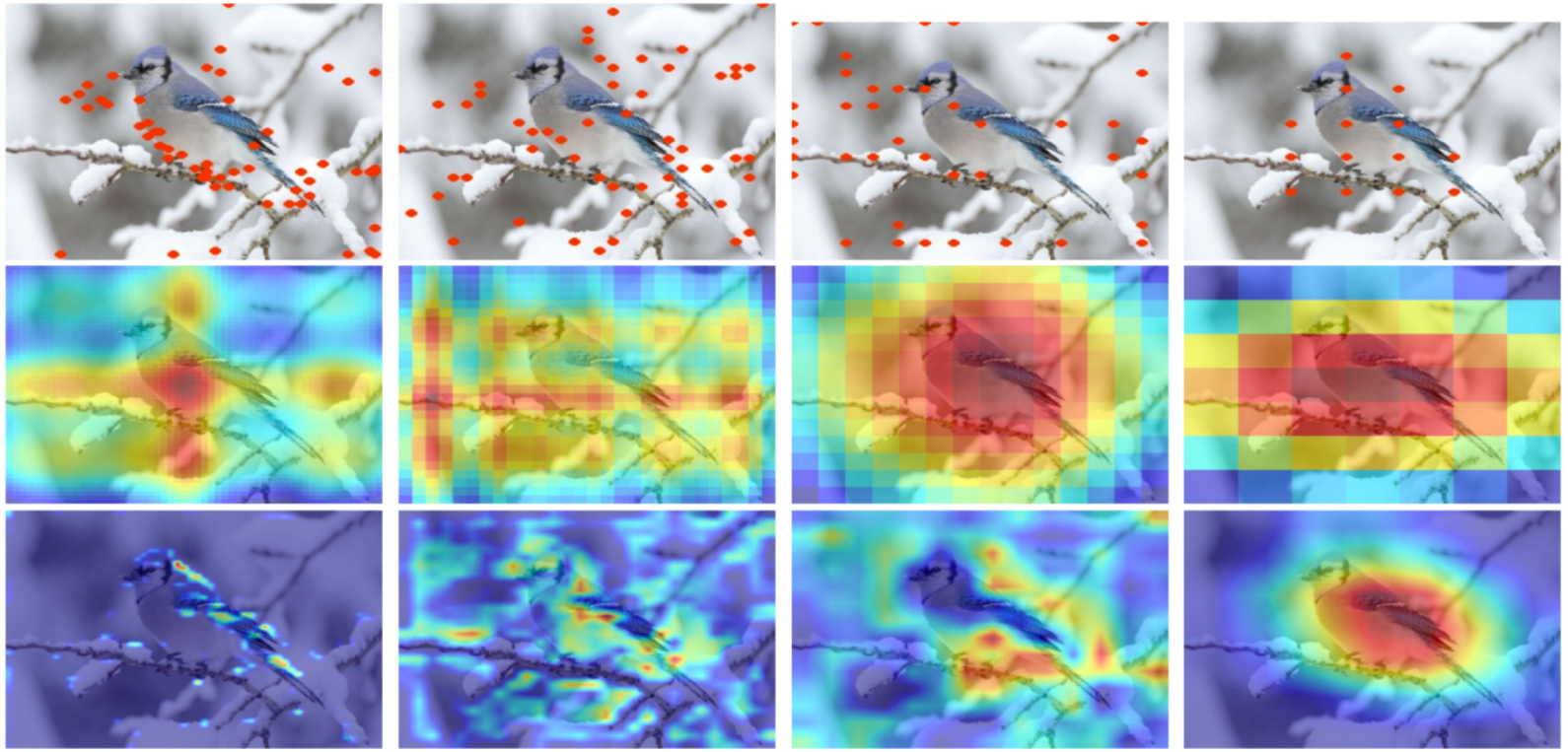


Fig: Query points visualization. **From top to bottom**: selected query points, the positional mask from all groups, and the attention map generated by Grad-CAM. **From left to right**: the corresponding results of each stage in ResNet50. Best viewed in color.

Experiments

Application on high-level visual tasks

Detector	Backbone	$AP_{50:95}$	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	GMac	Parameters(M)
RetinaNet	ResNet50 [He *et al.*, 2016]	36.2	55.9	38.5	19.4	39.8	48.3	239.32	37.74
RetinaNet	SE-ResNet50 [Hu *et al.*, 2018b]	37.4	57.8	**39.8**	20.6	**40.8**	50.3	239.43	40.25
RetinaNet	PRM-ResNet50 (ours)	**37.7**	**58.4**	39.7	**21.4**	40.6	**50.7**	239.32	37.74
Cascade R-CNN	ResNet50 [He *et al.*, 2016]	40.6	58.9	44.2	22.4	43.7	54.7	234.71	69.17
Cascade R-CNN	GC-ResNet50 [Cao *et al.*, 2019]	41.1	59.7	44.6	23.6	44.1	54.3	234.82	71.69
Cascade R-CNN	PRM-ResNet50 (ours)	**42.5**	**61.2**	**46.2**	**24.2**	**45.8**	**56.4**	234.71	69.17

Table 4: Detection performance (%) using different backbones on the MS-COCO validation dataset.

Conclusion

- We present a new module to recalibrate the convolutional neural network.
- We mainly deal with two issues: efficiency + positional information.
- We achieve a SOTA performance, with minimal parameters/FLOPs increase.
- The new method generalize well on other visual tasks.

Contact Me

Personal Website:	https://13952522076.github.io/
CAV Group:	https://www.cse.unt.edu/~qingyang/research.html
Github:	https://github.com/13952522076
Zhihu:	https://www.zhihu.com/people/ma-xu-41
Email:	xuma@my.unt.edu

Thank you.

Questions & Suggestions?