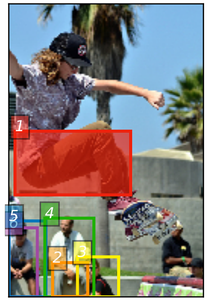
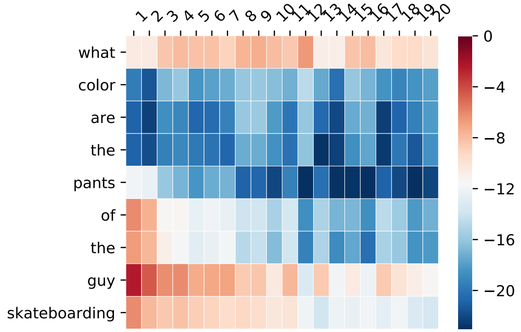
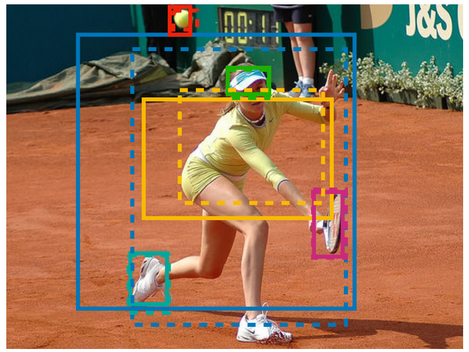






Bilinear Attention Networks

2018 VQA Challenge runner-up
(1st single model)

Jin-Hwa Kim, *Jaehyun Jun*, *Byoung-Tak Zhang*

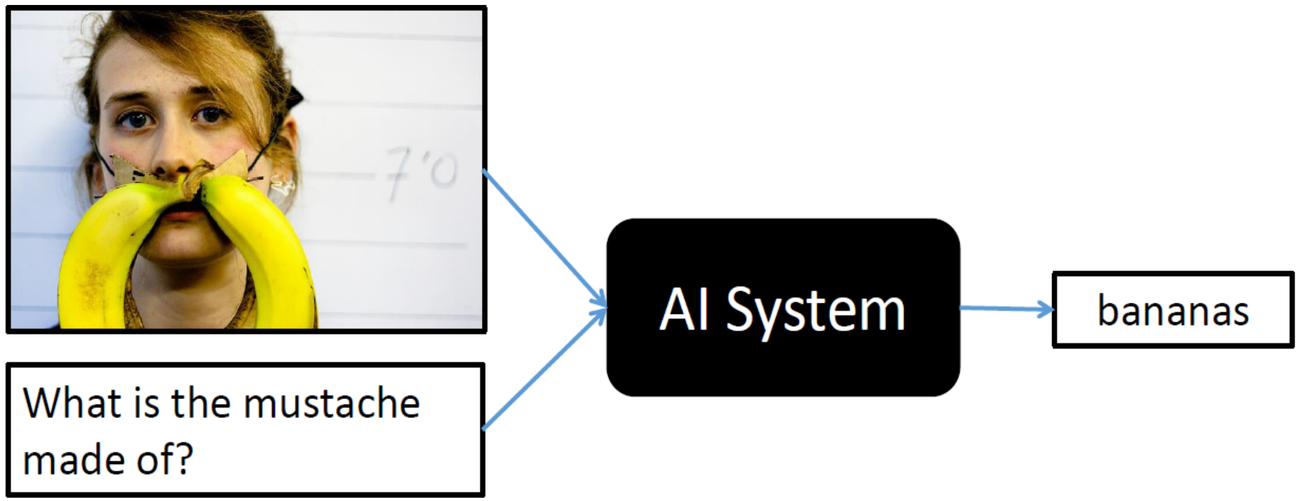
Biointelligence Lab.
Seoul National University

SEOUL NATIONAL UNIVERSITY

CVPR 2018
SALT LAKE CITY • JUNE 18-22

Visual Question Answering

- Learning joint representation of multiple modalities
- Linguistic and visual information



Credit: visualqa.org

VQA 2.0 Dataset

	Images	Questions	Answers
Train	80K	443K	4.4M
Val	40K	214K	2.1M
Test	80K	447K	unknown

- 204K MS COCO images
- 760K questions (10 answers for each question from unique AMT workers)
- train, val, test-dev (remote validation), test-standard (publications)
- and test-challenge (competitions), test-reserve (check overfitting)

$$accuracy = \min(\frac{\#\text{ humans that provided that answer}}{3}, 1)$$

- VQA Score is the avg. of 10 choose 9 accuracies →

# annotations	VQA Score
0	0.0
1	0.3
2	0.6
3	0.9
>3	1.0

https://github.com/GT-Vision-Lab/VQA/issues/1
https://github.com/hengyuan-hu/bottom-up-attention-vqa/pull/18

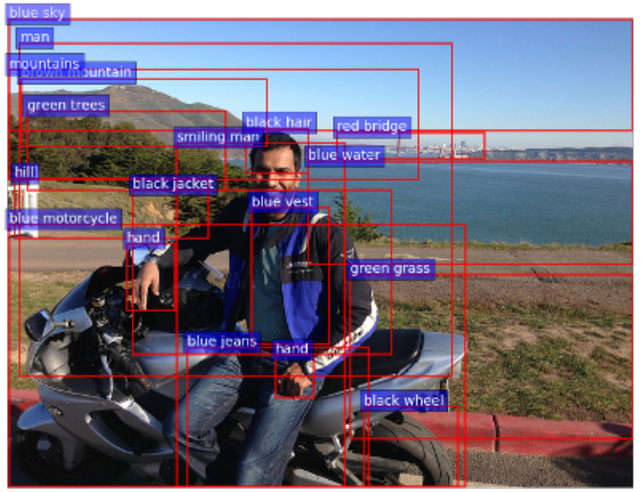
Goyal et al., 2017

Objective

- Introducing bilinear attention
  - *Interactions* between **words** and **visual concepts** are meaningful
  - Proposing an efficient method (with the same time complexity) on top of low-rank bilinear pooling
- Residual learning of attention
  - Residual learning with attention mechanism for incremental inference
- Integration of counting module (Zhang et al., 2018)

https://github.com/peteanderson80/bottom-up-attention

Preliminary



- Question embedding (fine-tuning)
  - Use the **all outputs of GRU** (every time steps)
  - $\mathbf{X} \in R^{N \times \rho}$ where N is hidden dim., and $\rho = |\{x_i\}|$ is # of tokens
- Image embedding (fixed **bottom-up-attention**)
  - Select 10-100 detected objects (rectangles) using pre-trained Faster RCNN, to extract *rich* features for each object (1600 classes, **400 attributes**)
  - $\mathbf{Y} \in R^{M \times \Phi}$ where M is feature dim., and $\phi = |\{y_j\}|$ is # of objects

Low-rank Bilinear Pooling

- Bilinear model and its approximation (Wolf et al., 2007, Pirsiavash et al., 2009)

$$f_i = \mathbf{x}^T \mathbf{W}_i \mathbf{y} \approx \mathbf{x}^T \mathbf{U}_i \mathbf{V}_i^T \mathbf{y} = \mathbb{1}^T (\mathbf{U}_i^T \mathbf{x} \circ \mathbf{V}_i^T \mathbf{y})$$

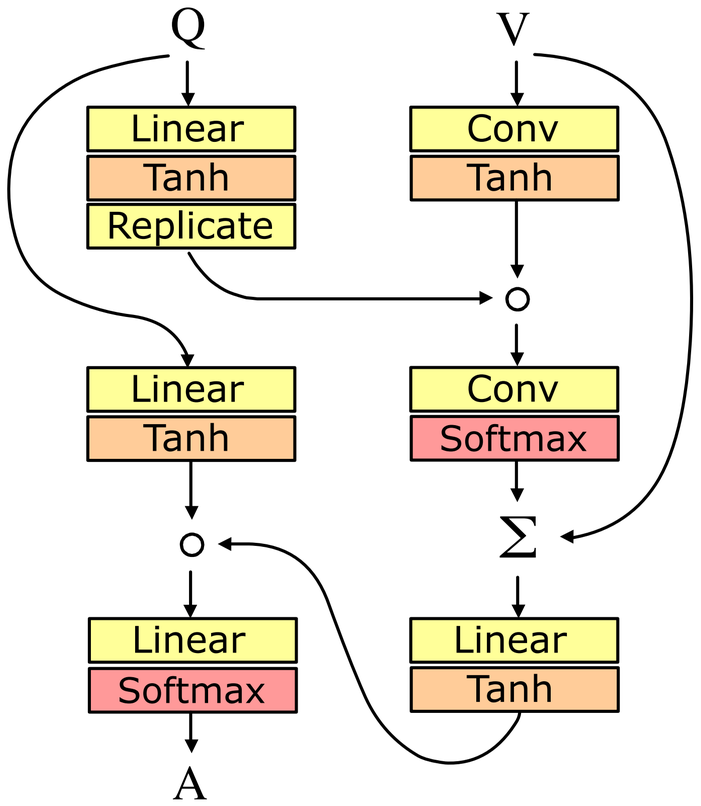
- Low-rank bilinear pooling (Kim et al., 2017)

$$\mathbf{f} = \mathbf{P}^T (\mathbf{U}^T \mathbf{x} \circ \mathbf{V}^T \mathbf{y})$$

  For vector output, instead of using three-dimensional tensors **U** and **V**, replace the vector of ones with a pooling matrix **P** (use three two-dimensional tensors).

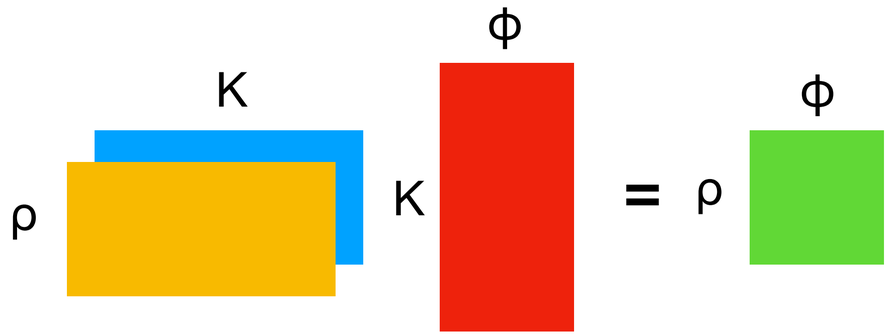
Unitary Attention

- This pooling is used to get attention weights with a question embedding (single-channel) vector and visual feature vectors (multi-channel) as the two inputs.
- We call it *unitary attention* since a question embedding vector queried the feature vectors, *unidirectionally*.



Kim et al., 2017

Bilinear Attention Maps



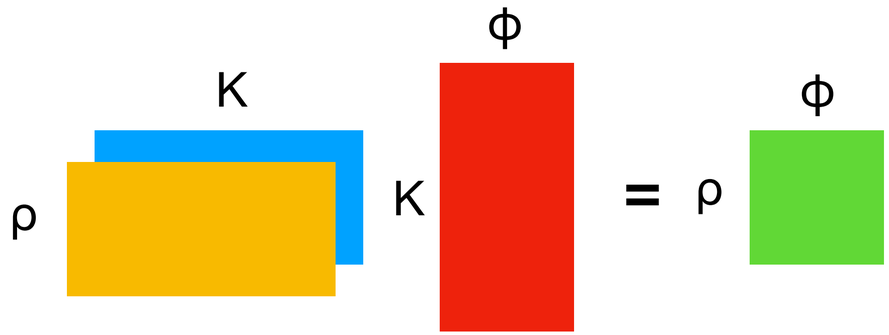
- **U** and **V** are linear embeddings
- **p** is a learnable projection vector

element-wise multiplication

$$\underset{\rho \times \phi}{\mathcal{A}} := \operatorname{softmax}\Big(\underbrace{\big((\mathbb{1} \cdot \underset{\rho\times K}{\mathbf{p}^T}) \circ \underset{\rho\times N}{\mathbf{X}^T} \underset{N\times K}{\mathbf{U}}\big)}_{\rho \times K} \underbrace{\mathbf{V}^T \mathbf{Y}}_{K \times \phi}\Big)$$

K×M M×ϕ

Bilinear Attention Maps



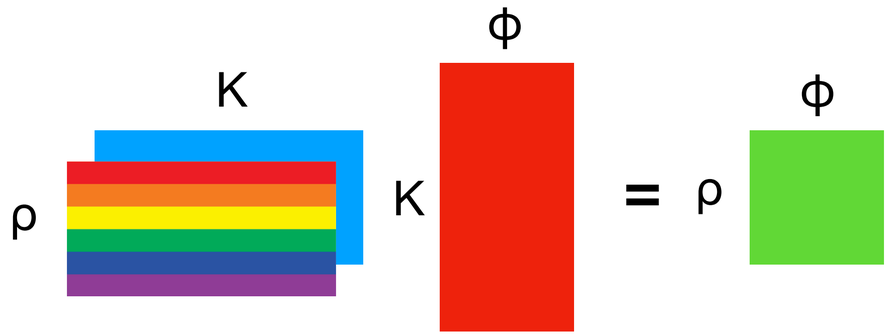
- Exactly the same approach with low-rank bilinear pooling

element-wise multiplication

$$\mathcal{A} := \text{softmax}\Big(\big((\mathbb{1} \cdot \mathbf{p}^T) \circ \mathbf{X}^T\mathbf{U}\big)\mathbf{V}^T\mathbf{Y}\Big)$$

$$A_{i,j} = \mathbf{p}^T\big((\mathbf{U}^T\mathbf{X}_i) \circ (\mathbf{V}^T\mathbf{Y}_j)\big).$$

Bilinear Attention Maps



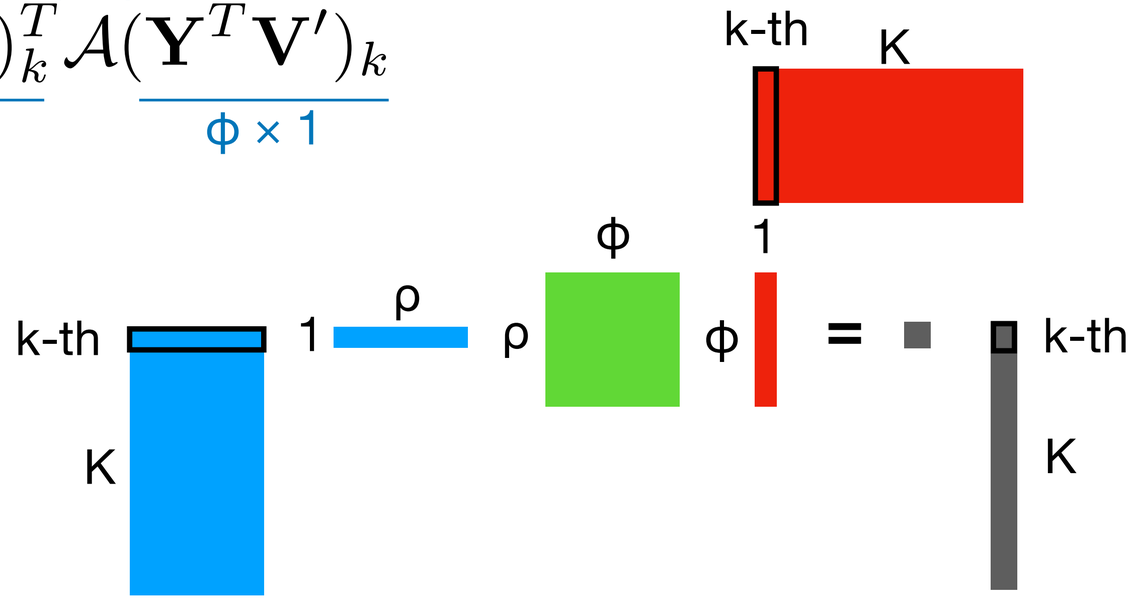
- Multiple bilinear attention maps are acquired by different projection vectors $\mathbf{p}_g$ as:

$$\mathcal{A}_g := \text{softmax}\Big(\big((\mathbb{1} \cdot \mathbf{p}_g^T) \circ \mathbf{X}^T\mathbf{U}\big)\mathbf{V}^T\mathbf{Y}\Big)$$

Bilinear Attention Networks

- Each multimodal joint feature is filled with following equation (*k* is the index of K; *broadcasting* in PyTorch let you avoid for-loop for this):

$$\mathbf{f}'_k = \underbrace{(\mathbf{X}^T\mathbf{U}')^T_k}_{1\times\rho}\overset{\rho\times\phi}{\mathcal{A}}\underbrace{(\mathbf{Y}^T\mathbf{V}')_k}_{\phi\times 1}$$



※ *broadcasting: automatically repeat tensor operations in api-level supported by Numpy, Tensorflow, Pytorch*

Bilinear Attention Networks

- One can show that this is equivalent to a bilinear attention model where each feature is pooled by low-rank bilinear approximation

$$\mathbf{f}'_k = (\mathbf{X}^T\mathbf{U}')^T_k \mathcal{A} (\mathbf{Y}^T\mathbf{V}')_k$$

$$\mathbf{f}'_k = \sum_{i=1}^{|\{\mathbf{x}_i\}|} \sum_{j=1}^{|\{\mathbf{y}_j\}|} \mathcal{A}_{i,j} (\mathbf{X}_i^T\mathbf{U}'_k)(\mathbf{V}'^T_k\mathbf{Y}_j)$$

$$= \sum_{i=1}^{|\{\mathbf{x}_i\}|} \sum_{j=1}^{|\{\mathbf{y}_j\}|} \mathcal{A}_{i,j} \underbrace{\mathbf{X}_i^T(\mathbf{U}'_k\mathbf{V}'^T_k)\mathbf{Y}_j}_{\textit{low-rank bilinear pooling}}$$

Bilinear Attention Networks

- One can show that this is equivalent to a bilinear attention model where each feature is pooled by low-rank bilinear approximation
- Low-rank bilinear feature learning ***inside*** bilinear attention

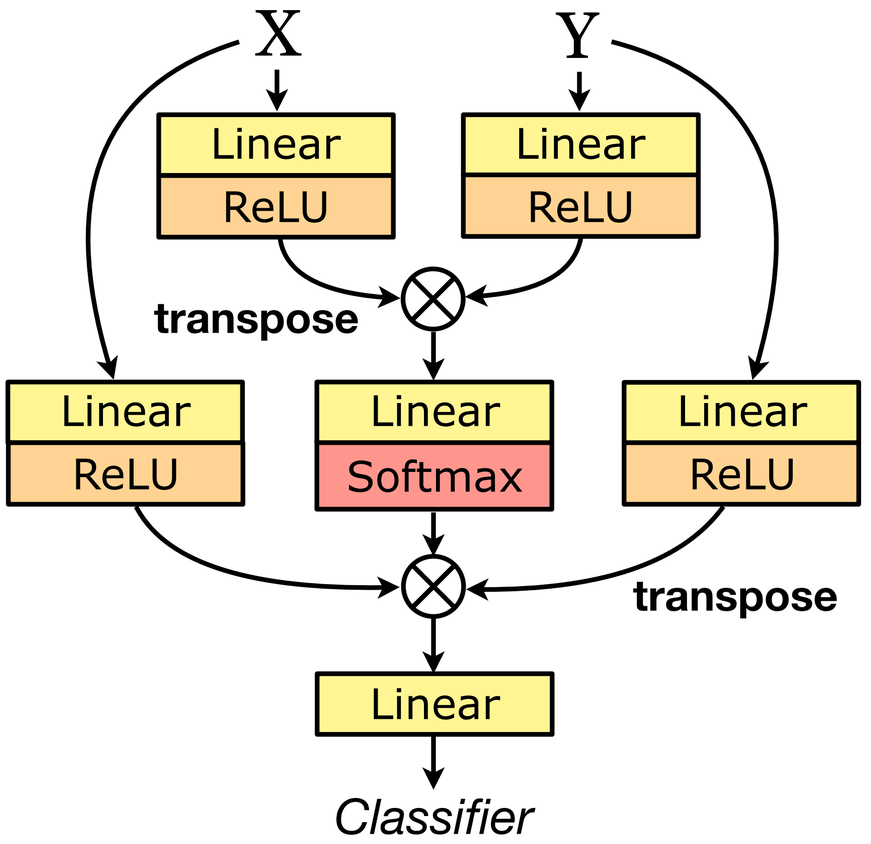
$$\mathbf{f}'_k = \sum_{i=1}^{|\{\mathbf{x}_i\}|} \sum_{j=1}^{|\{\mathbf{y}_j\}|} \mathcal{A}_{i,j} (\mathbf{X}_i^T \mathbf{U}'_k)(\mathbf{V}'^T_k \mathbf{Y}_j)$$

$$= \sum_{i=1}^{|\{\mathbf{x}_i\}|} \sum_{j=1}^{|\{\mathbf{y}_j\}|} \mathcal{A}_{i,j} \underbrace{\mathbf{X}_i^T (\mathbf{U}'_k \mathbf{V}'^T_k) \mathbf{Y}_j}_{\textit{low-rank bilinear pooling}}$$

Bilinear Attention Networks

- One can show that this is equivalent to a bilinear attention model where each feature is pooled by low-rank bilinear approximation
- Low-rank bilinear feature learning ***inside*** bilinear attention
- Similarly to MLB (Kim et al., ICLR 2017), activation functions can be applied

Bilinear Attention Networks

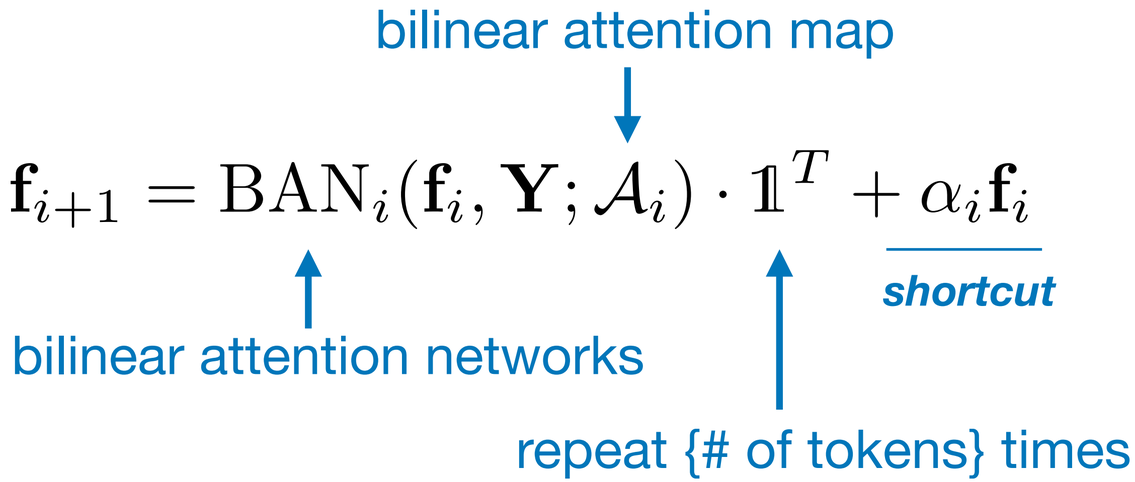


Time Complexity

- Assuming that $M \geq N > K > \phi \geq \rho$, the time complexity of bilinear attention networks is $O(KM\phi)$ where K denotes hidden size, since BAN consists of **matrix chain multiplication**
- Empirically, BAN takes 284s/epoch while unitary attention control takes 190s/epoch
- Largely due to the increment of input size for Softmax function, ϕ to $\phi \times \rho$

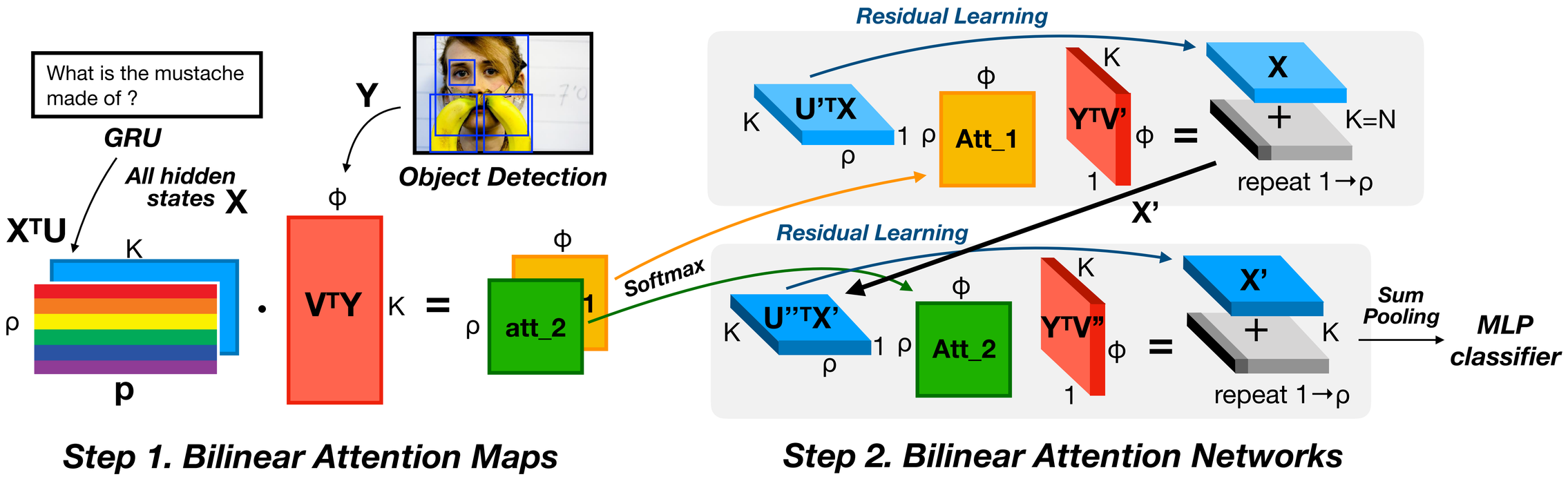
Residual Learning of Attention

- Residual learning exploits multiple attention maps ($\mathbf{f}_0$ is $\mathbf{X}$ and $\{\alpha_i\}$ is fixed to ones):



Overview

- After getting bilinear attention maps, we can stack multiple BANs.



Multiple Attention Maps

- Single model on validation score

	Validation VQA 2.0 Score	+%
Bottom-Up (Teney et al., 2017)	63.37	
BAN-1	65.36	+1.99
BAN-2	65.61	+0.25
BAN-4	65.81	+0.2
BAN-8	**66.00**	+0.19
BAN-12	**66.04**	+0.04

Residual Learning

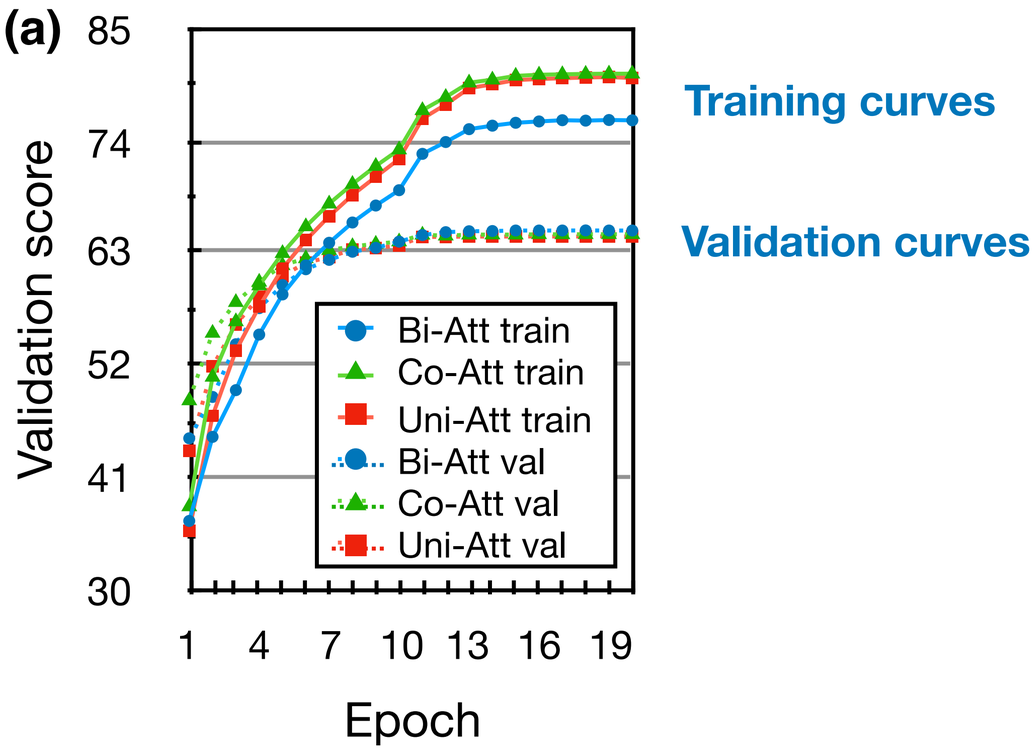
		Validation VQA 2.0 Score	+/-
	BAN-4 (Residual)	**65.81 ±0.09**	
$\sum_i \mathrm{BAN}_i(\mathbf{X}, \mathbf{Y}; A_i)$	**BAN-4 (Sum)**	64.78 ±0.08	-1.03
$\Big\Vert_i \mathrm{BAN}_i(\mathbf{X}, \mathbf{Y}; A_i)$	**BAN-4 (Concat)**	64.71 ±0.21	-0.07

Comparison with Co-attention

		Validation VQA 2.0 Score	+/-
BAN-1	**Bilinear Attention**	**65.36 ±0.14**	
	Co-Attention	64.79 ±0.06	-0.57
	Attention	64.59 ±0.04	-0.20

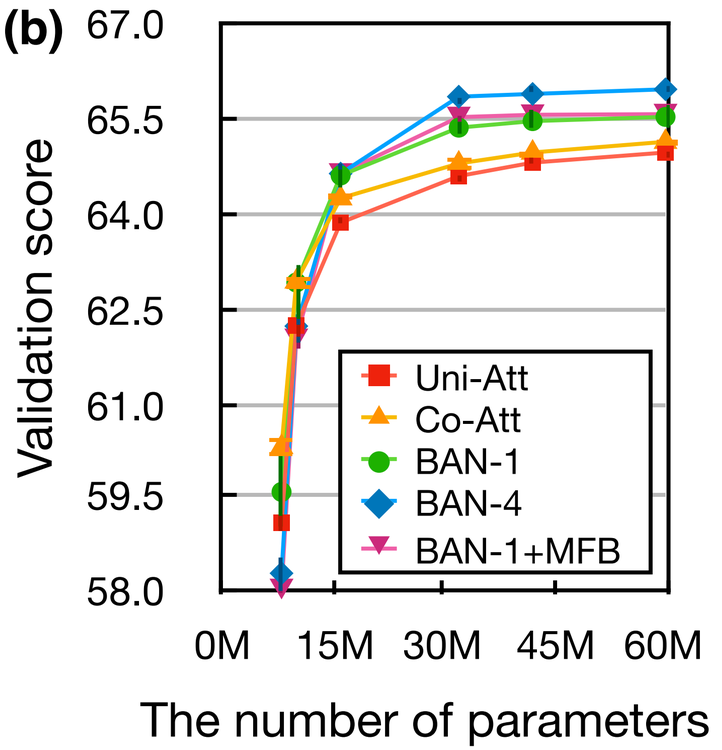
※ *The number of parameters is controlled (all comparison models have 32M parameters).*

Comparison with Co-attention



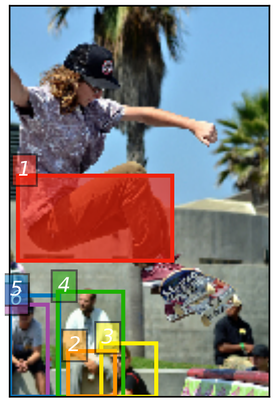
Training curves

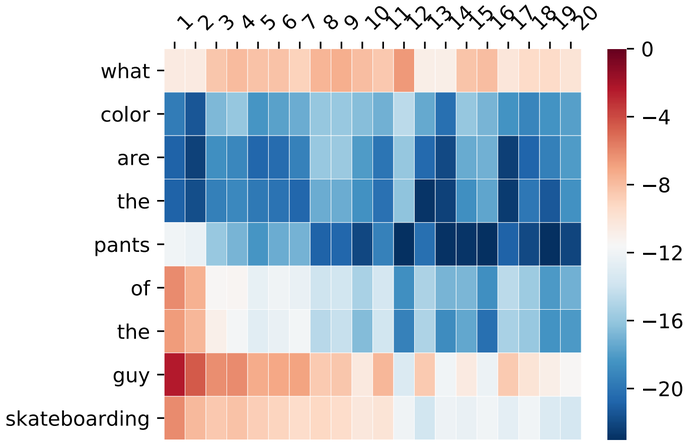
Validation curves

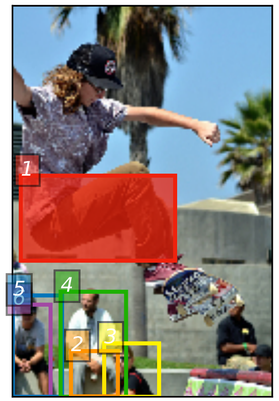


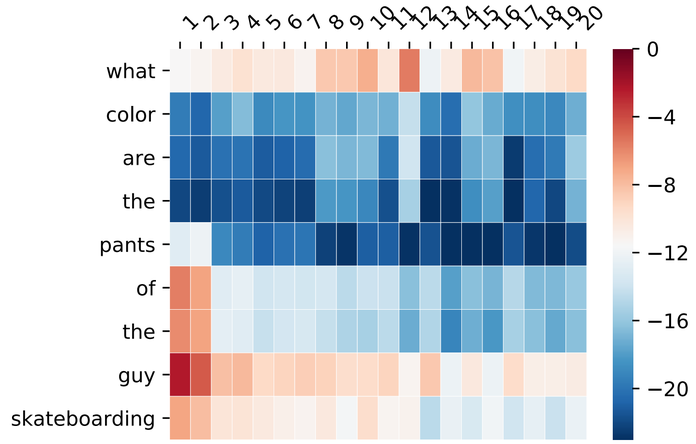
※ *The number of parameters is controlled (all comparison models have 32M parameters).*

Visualization



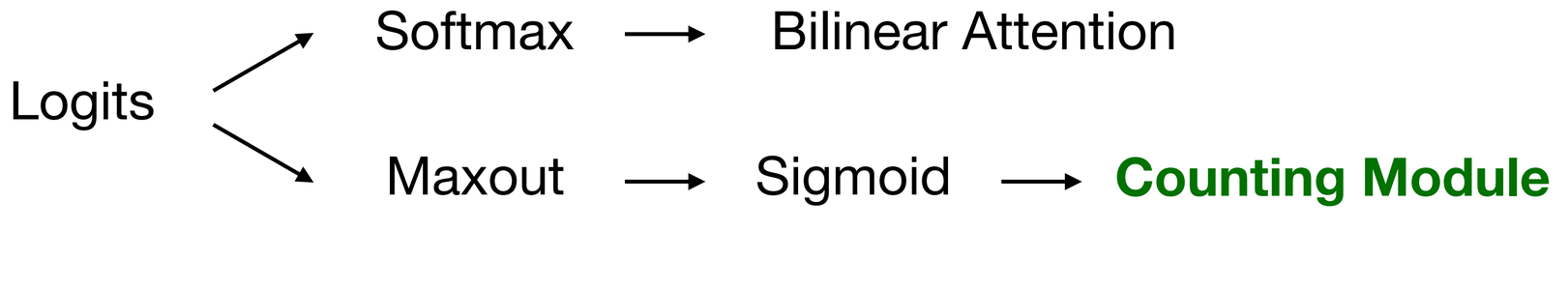


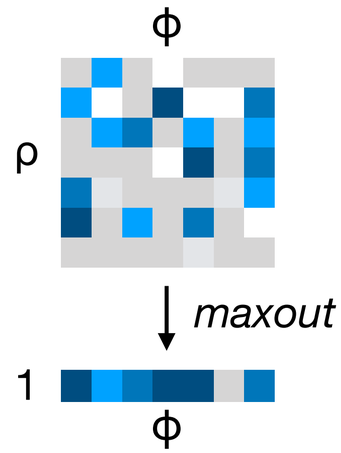




Integration of Counting module with BAN

- Counter (Zhang et al., 2018) gets a multinoulli distribution and detected box info
- **Our method** — *maxout* from bilinear attention distribution to unitary attention distribution, easy to adapt multitask learning





$$\mathbf{f}_{i+1} = \left(\mathrm{BAN}_i(\mathbf{f}_i, \mathbf{Y}; \mathcal{A}_i) + g_i(\mathbf{c}_i)\right) \cdot \mathbb{1}^T + \mathbf{f}_i$$

inference steps for multi-glimpse attention

linear embedding of the i-th output of counter module

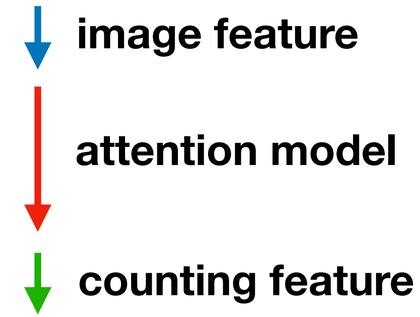
residual learning of attention

Comparison with State-of-the-arts

test-dev	Numbers
Zhang et al. (2018)	51.62
Ours	**54.04**

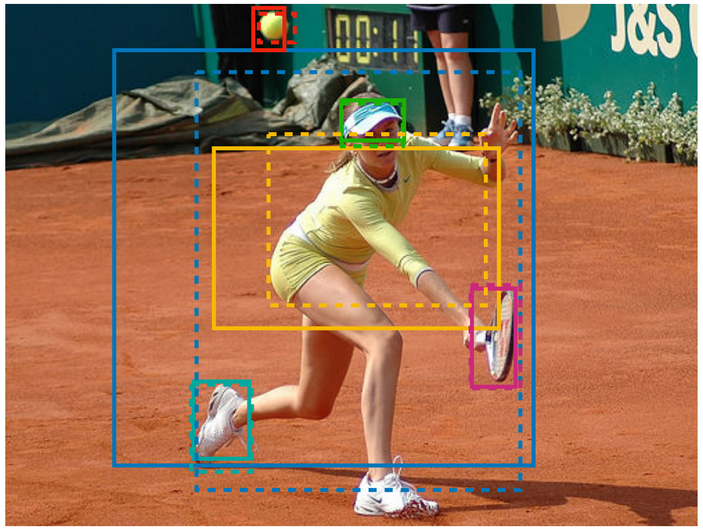
- Single model on test-dev score

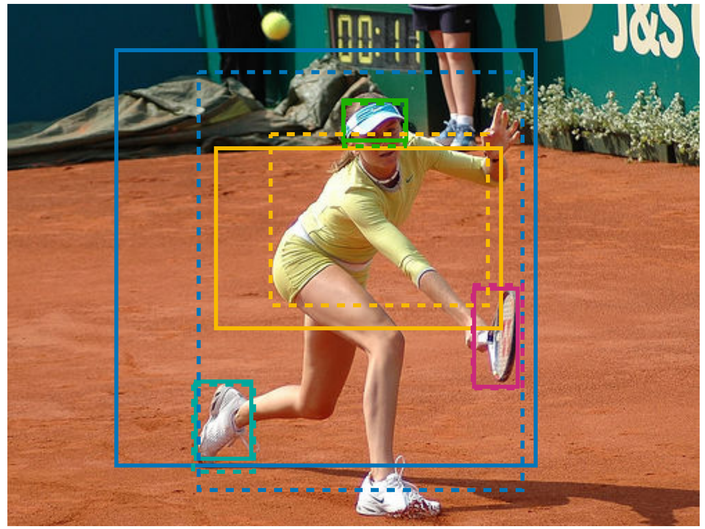
		Test-dev VQA 2.0 Score	+%
	Prior	25.70	
	Language-Only	44.22	+18.52%
2016 winner	MCB (ResNet)	61.96	+17.74%
2017 winner	Bottom-Up (FRCNN)	65.32	+3.36%
2017 runner-up	MFH (ResNet)	65.80	+0.48%
	MFH (FRCNN)	68.76	**+2.96%**
	BAN (Ours; FRCNN)	69.52	**+0.76%**
	BAN-Glove (Ours; FRCNN)	69.66	**+0.14%**
	BAN-Glove-Counter (Ours; FRCNN)	**70.04**	**+0.38%**



Flickr30k Entities

- Visual grounding task — mapping entity phrases to regions in an image





[/EN#40120/people A girl] in [/EN#40122/clothing a yellow tennis suit] , [/EN#40125/other green visor] and [/EN#40128/clothing white tennis shoes] holding [/EN#40124/other a tennis racket] in a position where she is going to hit [/EN#40121/other the tennis ball] .

Flickr30k Entities

- Visual grounding task — mapping entity phrases to regions in an image



[/EN#38656/people A male conductor] wearing [/EN#38657/clothing all black] leading [/EN#38653/people a orchestra] and [/EN#38658/people choir] on [/EN#38659/scene a brown stage] playing and singing [/EN#38664/other a musical number] .

Flickr30k Entities Recall@1,5,10

	R@1	R@5	R@10
Zhang et al. (2016)	27.0	49.9	57.7
SCRC (2016)	27.8	-	62.9
DSPE (2016)	43.89	64.46	68.66
GroundeR (2016)	47.81	-	-
MCB (2016)	48.69	-	-
CCA (2017)	50.89	71.09	75.73
Yeh et al. (2017)	53.97	-	-
Hinami & Satoh (2017)	65.21		
BAN (ours)	**66.69**	**84.22**	**86.35**

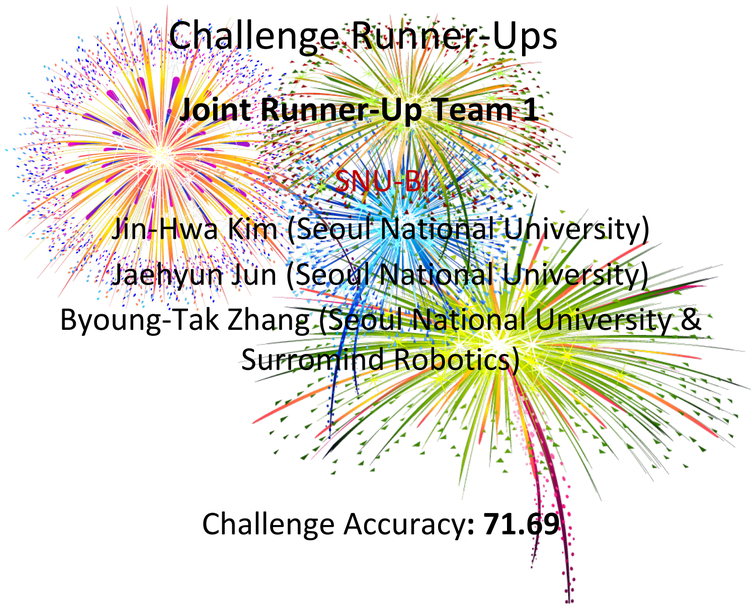
Flickr30k Entities Recall@1,5,10

	people	clothing	bodyparts	animals	vehicles	instruments	scene	other
#Instances	5,656	2,306	523	518	400	162	1,619	3,374
GroundeR (2016)	61.00	38.12	10.33	62.55	68.75	36.42	58.18	29.08
CCA (2017)	64.73	46.88	17.21	65.83	68.75	37.65	51.39	31.77
Yeh et al. (2017)	68.71	46.83	19.50	70.07	73.75	39.50	60.38	32.45
Hinami & Satoh (2017)	78.17	61.99	35.25	74.41	76.16	**56.69**	68.07	47.42
BAN (ours)	**79.90**	**74.95**	**47.23**	**81.85**	**76.92**	43.00	**68.69**	**51.33**

Conclusions

- Bilinear attention networks gracefully extends unitary attention networks,
- as low-rank bilinear pooling inside bilinear attention.
- Furthermore, residual learning of attention efficiently uses multiple attention maps.

2018 VQA Challenge runners-up (shared 2nd place)
1st single model (70.35 on test-standard)



arXiv:1805.07932

Code & pretrained model in PyTorch:

github.com/jnhwkim/ban-vqa

Thank you!

Any question?

We would like to thank Kyoung-Woon On, Bohyung Han, and Hyeonwoo Noh for helpful comments and discussion.
This work was supported by 2017 Google Ph.D. Fellowship in Machine Learning and Ph.D. Completion Scholarship from College of Humanities, Seoul National University, and the Korea government (IITP-2017-0-01772-VTT, IITP-R0126-16-1072-SW.StarLab, 2018-0-00622-RMI, KEIT-10044009-HRI.MESSI, KEIT-10060086-RISF).
The part of computing resources used in this study was generously shared by Standigm Inc.