Improving VQA Performance with Mixture of Detectors features

Team HDU-UCAS-USYD with members

Zhou Yu[1], Jun Yu[1], Chenchao Xiang[1], Liang Wang[1], Dalu Guo[3], Qingming Huang[2], Jianping Fan[1] and Dacheng Tao[3]

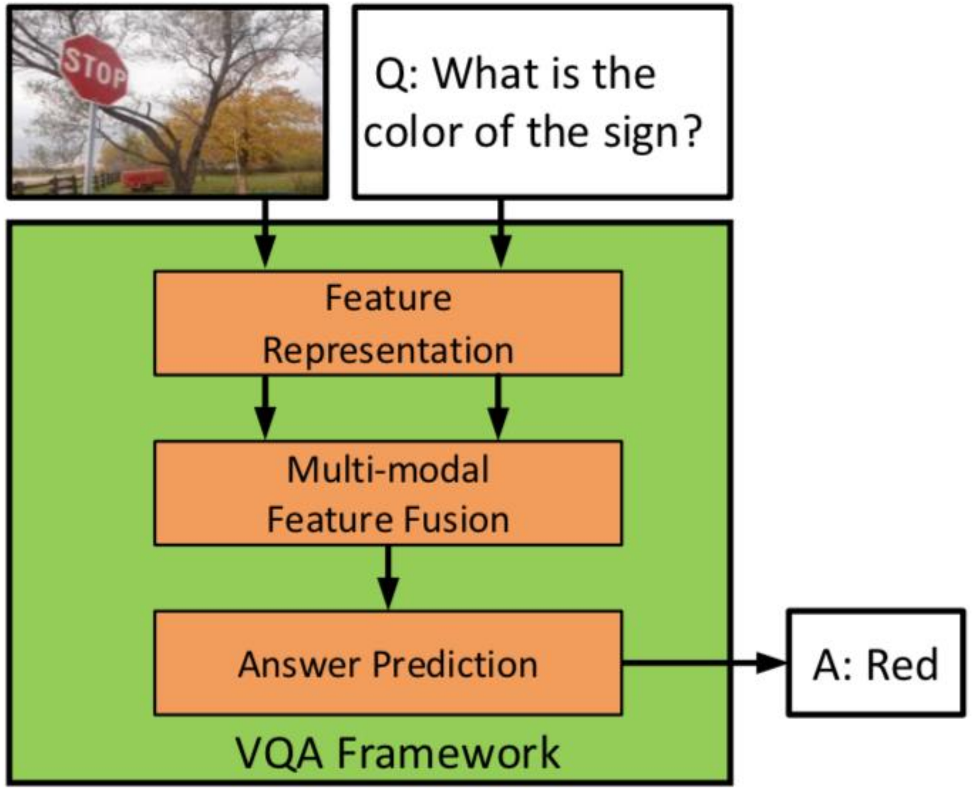
1. Hangzhou Dianzi University, China
2. University of Chinese Academy of Science, China
3. The University of Sydney, Australia

Outline

- Background
- Mixture of Detectors (MoD) features
- Implementation Details & Experimental Results
- Conclusions & Future works

Background

- Key components for VQA
    1. Feature representation (**feature**)
    2. Multi-modal feature fusion (**model**)
    3. Answer Prediction (**loss**)



Previous SOTA approaches

- Feature representation
  - LSTM (concat with 300D GloVe feature) for questions
  - bottom-up attention visual features extracted from Faster R-CNN

- Multi-modal feature fusion
  - Attention modeling: visual attention, question-attention, co-attention
  - Fusion: Concat, MCB, MLB, MUTAN, MFB, MFH

- Answering modeling
  - Answer sampling+softmax, Cross-entropy, multi-label KLD

Previous SOTA approaches

- Feature representation **0.8% improvement over LSTM w/o GloVe**
  - LSTM (**concat with 300D GloVe feature**) for questions
  - **bottom-up attention** visual features extracted from Faster R-CNN

2.5% improvement over ResNet-152 res5c features

- Multi-modal feature fusion **0.5% improvement over only visual attention**
  - Attention modeling: visual attention, question-attention, **co-attention**
  - Fusion: Concat, MCB, MLB, MUTAN, MFB, **MFH**

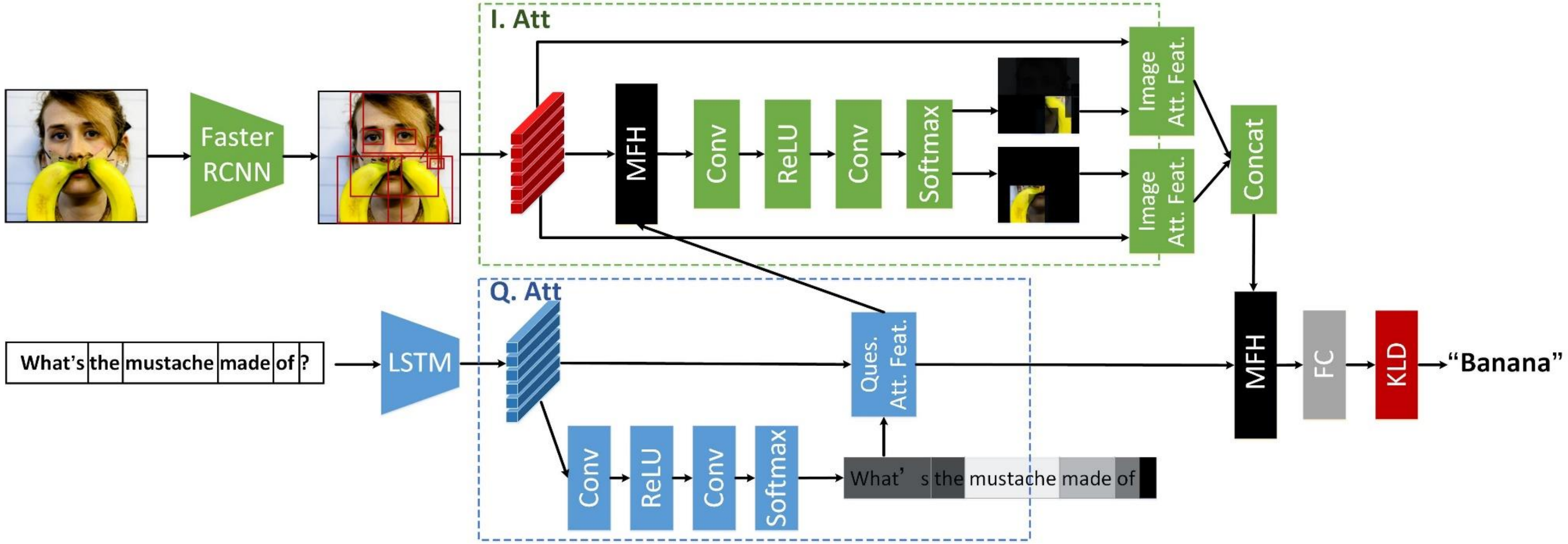
1.6% improvement over MCB w/o attention

- Answering modeling
  - Answer sampling+softmax, Cross-entropy, **multi-label KLD**

0.3% improvement over AS+softmax

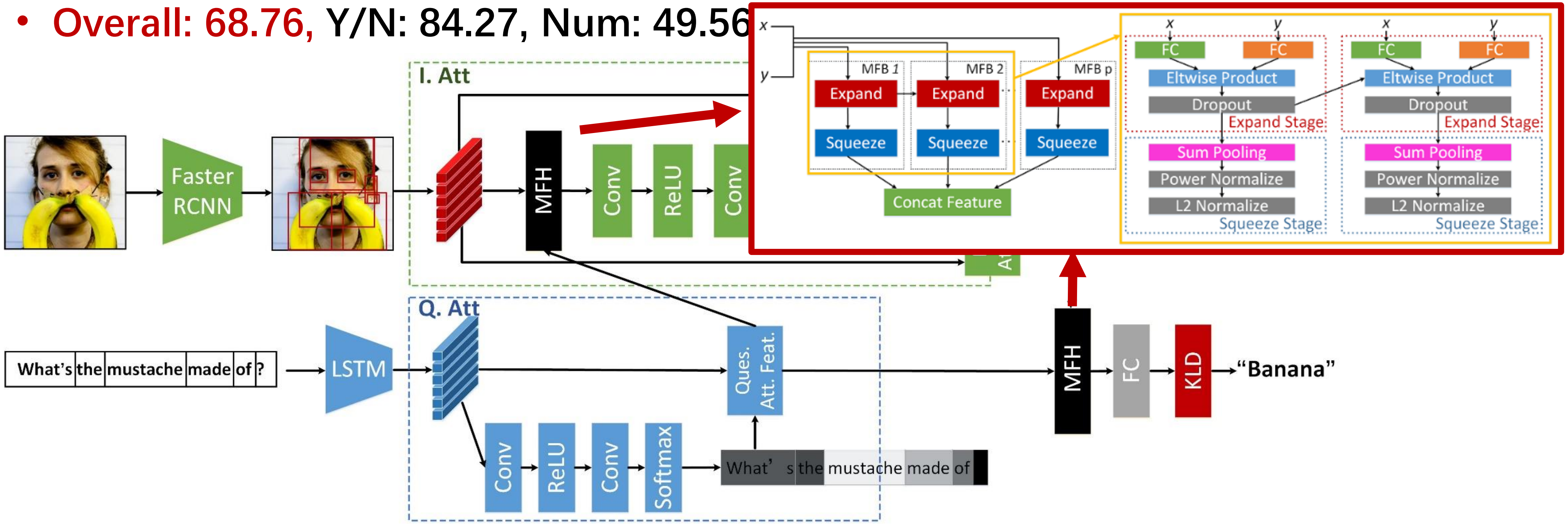
Our reference model

- 1 layer LSTM(w/ GloVE) + Bottom-up attention feature (K=[10,100]) + MFH-CoAtt (# Q. glimpses=2, # I. glimpses=2) + KLD
- VQA-2.0, train on <train+val> , test on <test-dev>
- **Overall: 68.76, Y/N: 84.27, Num: 49.56, Other: 59.89**



Our reference model

- 1 layer LSTM(w/ GloVE) + Bottom-up attention feature (K=[10,100]) + MFH-CoAtt (# Q. glimpses=2, # I. glimpses=2) + KLD
- VQA-2.0, train on <train+val> , test on <test-dev>
- **Overall: 68.76, Y/N: 84.27, Num: 49.56**



Yu *et al.*, Beyond Bilinear: Generalized Multi-modal Factorized High-order Pooling for Visual Question Answering, IEEE TNNLS 10.1109/TNNLS.2018.2817340 (https://github.com/yuzcccc/vqa-mfb)

Improvement

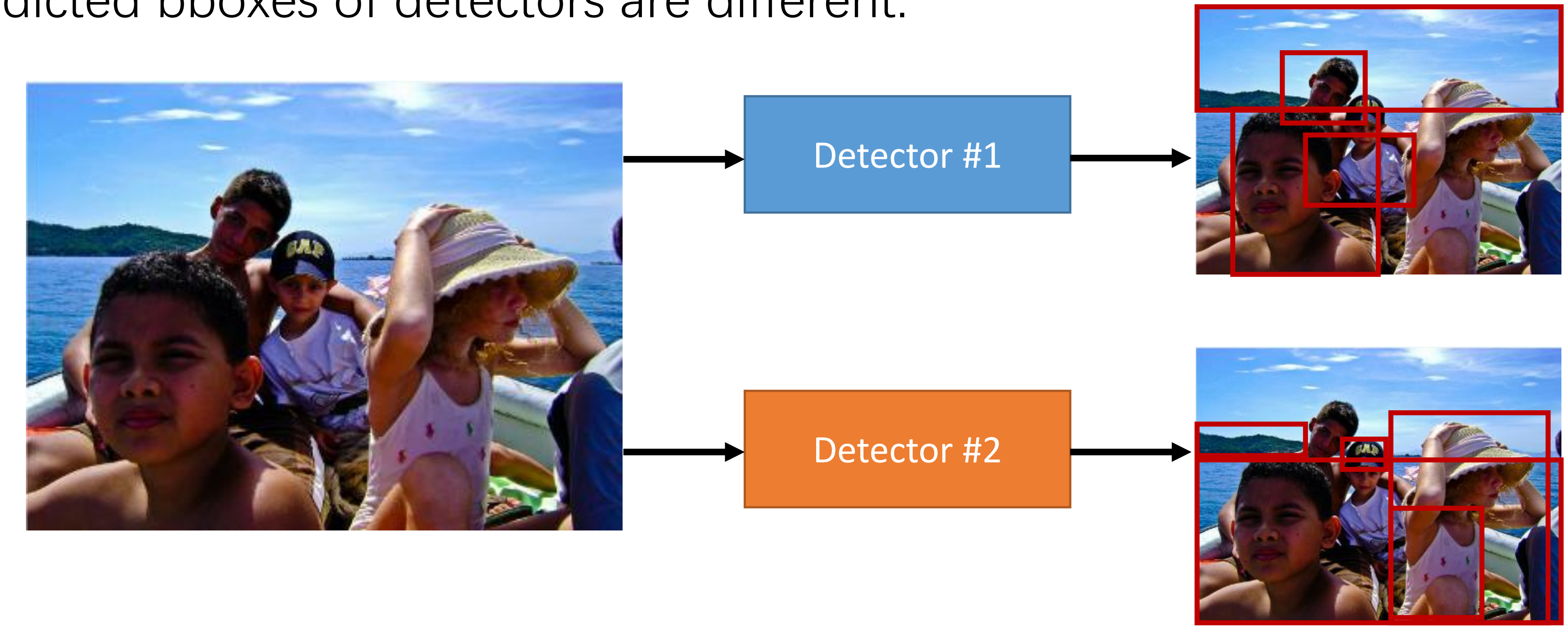
- Inspiration:
  - The **representation capacity** of visual features is the bottleneck for VQA
  - Current Bottom-up attention features (Faster R-CNN with ResNet-101) is good, but can be better

Improvement

- Inspiration:
  - The **representation capacity** of visual features is the bottleneck for VQA
  - Current Bottom-up attention features (Faster R-CNN with ResNet-101) is good, but can be better
- Our initial solution
  - Replace Faster R-CNN (w/ ResNet-101) with a better model, e.g., FPN (w/ ResNet-152)
  - Migrate the project from *Caffe* to ***Detectron***, and train the FPN model (ResNet-152 model) on Visual Genome.
  - We obtain the new 1024-D bottom-up features. However, the performance is not as competitive as the original 2048-D features ☹

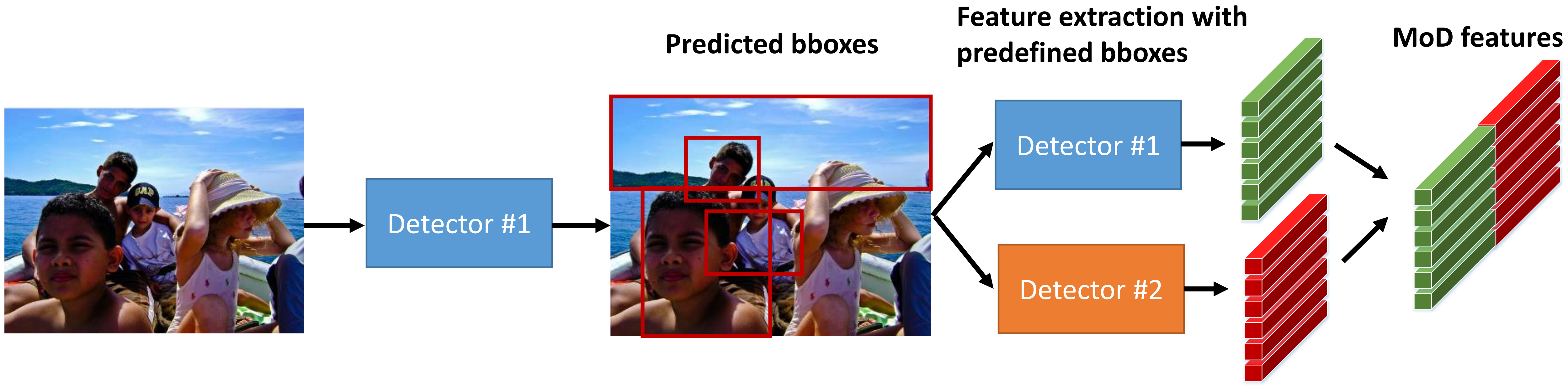
Mixture of Detectors (MoD) features

- Combine the bottom-up attention features from multiple object detectors
  - Can not directly combine the two features **w/o alignment**, as the predicted bboxes of detectors are different.



Mixture of Detectors (MoD) features

- We use the predicted bboxes of one model and extract bottom-up features from each detector using a **Fast-R-CNN** like strategy
- The extracted features are **aligned** and we simply concat them to obtain the MoD features



Implementation Details

- Two object detectors trained on Visual Genome with different backbone models
  - Detector #1: the original bottom-up model, i.e., Faster-RCNN (ResNet-101), **2048-D** output feature for each bbox
  - Detector #2: FPN (ResNet-152 pre-trained on ImageNet-5k) , **1024-D** output feature for each bbox
  - MOD feature: 2048+1024=**3072-D**
- Two strategies in bboxes generation
  - Dynamic K range from 10~100, K is the number of predicted bboxes*
  - Fix K=100

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

Experimental Results (single model)

- Trained on <train+val>, tested on <test-dev>

Models	Overall (%)	Y/N (%)	Num (%)	Others (%)
Reference model (K=[10,100])	68.76	84.27	49.56	59.89
MoD (K=[10,100])	69.47 (+0.71)	85.35 (+1.08)	49.85 (+0.29)	60.39 (+0.50)
MoD (K=100)	**69.82 (+1.06)**	85.86 (+1.59)	49.37 (-0.19)	60.79 (+0.90)

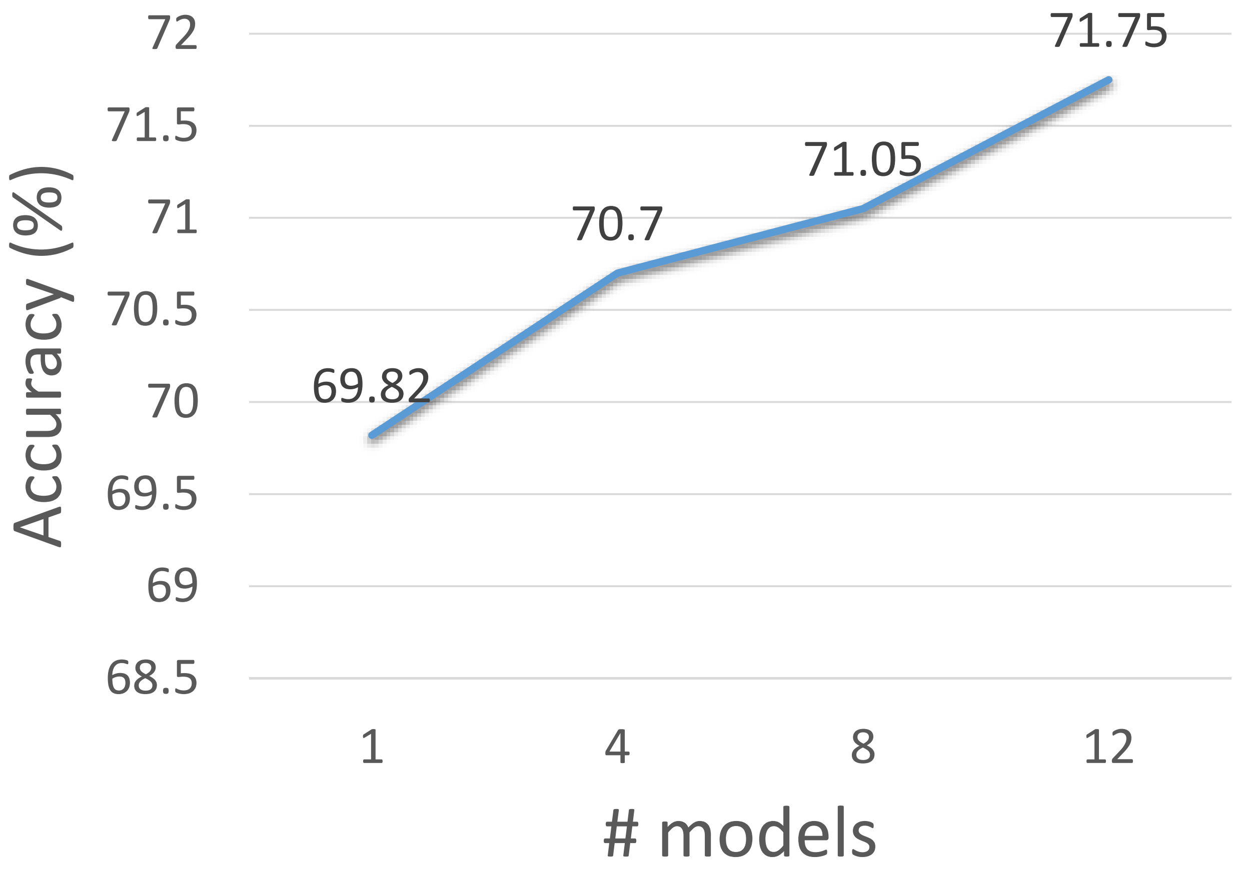
Experimental Results (single model)

- Trained on <train+val>, tested on <test-dev>

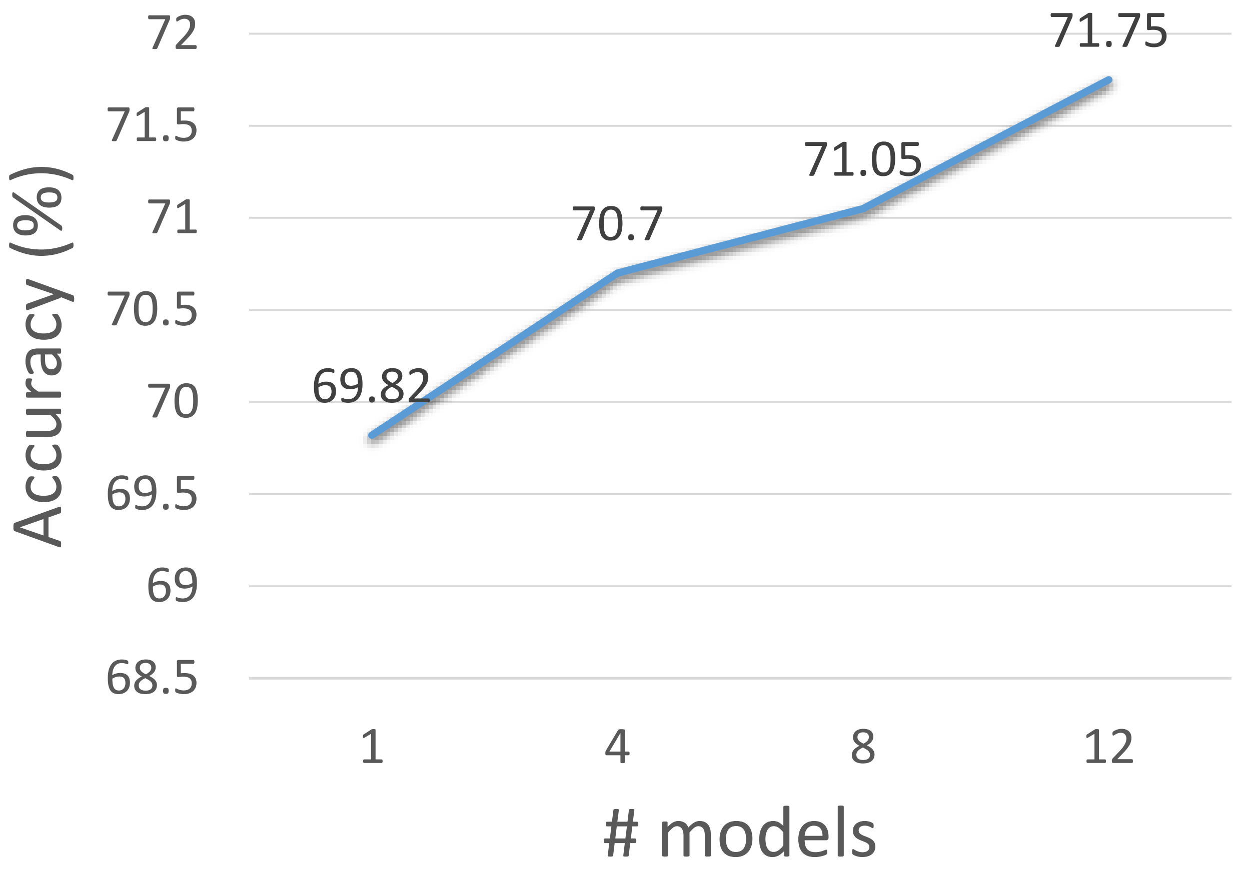
Models	Overall (%)	Y/N (%)	Num (%)	Others (%)
Reference model (K=[10,100])	68.76	84.27	49.56	59.89
MoD (K=[10,100])	69.47 (+0.71)	85.35 (+1.08)	49.85 (+0.29)	60.39 (+0.50)
MoD (K=100)	**69.82 (+1.06)**	85.86 (+1.59)	49.37 (-0.19)	60.79 (+0.90)

- MoD brings **0.71%** improvement over the reference model with the same bboxes
- Using fix K=100 bring about **0.35%** improvement over K=[10,100] for MoD features, <u>but performance for <Num> type is even lower than the reference model</u>

Experimental Results (ensemble)



Experimental Results (ensemble)



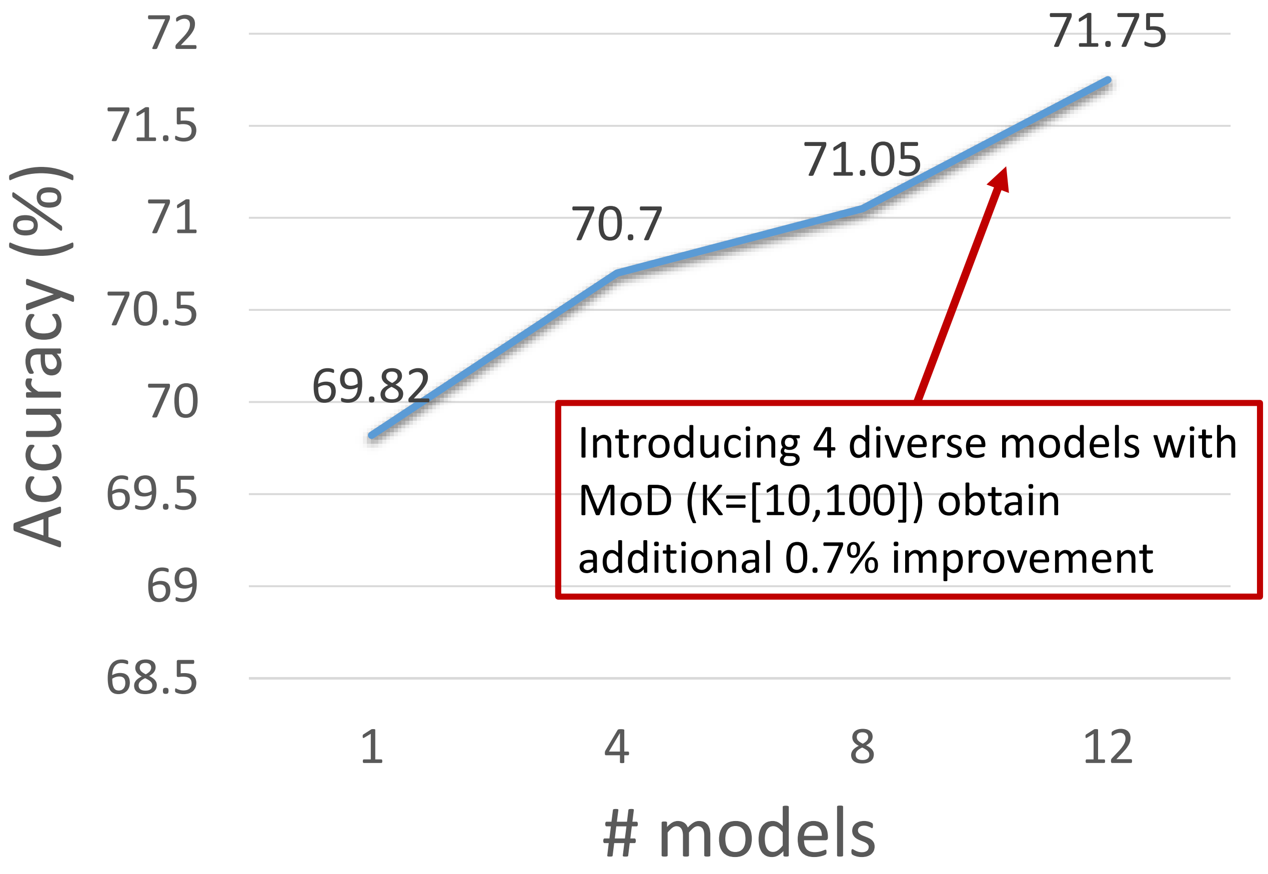
Submitted Final Results (12 models)

Test-dev
All: 71.75;
Y/N: 87.32; Num: 52.15; Other:62.93

Test-std
All: 72.09
Y/N: 87.61; Num: 51.92; Other:63.19

Test-challenge
All: 71.91

Experimental Results (ensemble)



Submitted Final Results (12 models)

Test-dev

All: 71.75;

Y/N: 87.32; Num: 52.15; Other:62.93

Test-std

All: 72.09

Y/N: 87.61; Num: 51.92; Other:63.19

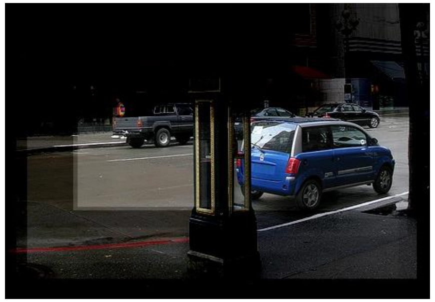
Test-challenge

All: 71.91

Visualizations

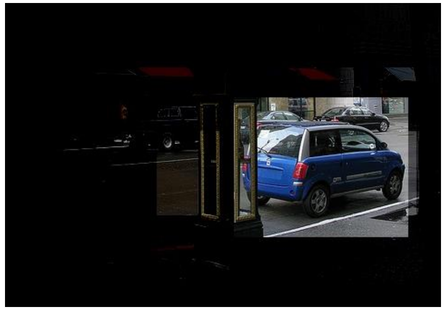
- MoD (K=100) vs. MoD (K=[10, 100]) on **<Others>**
  - Larger K tends to discover more details of the image, which makes its performances on <Y/N> and <Others> better

Q: What side of the street are cars parked on?



attention map for MoD (K=100)

→ A: both √



attention map for MoD (K=[10, 100])

→ A: right ×

Visualizations

- MoD (K=100) vs. MoD (K=[10, 100]) on **<Y/N>**

Q: Are the zebras in the wild?

There are fence and wood pile here

attention map for MoD (K=100)

A: Yes √

attention map for MoD (K=[10, 100])

A: No ×

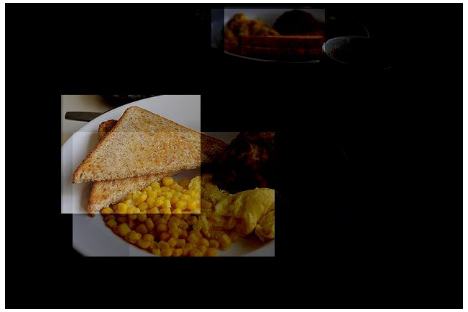
Visualizations

- MoD (K=100) vs. MoD (K=[10, 100]) on **<Num>**
  - Larger K leads to more redundant bboxes for one object, which makes it harder to learn correct visual attention

Q: How many sandwiches can you see?

attention map for MoD (K=100)

→ A: 6 ×



attention map for MoD (K=[10, 100])

→ A: 4 √

Take-away

- The **capability of visual features** are still the core for VQA (and other related tasks, e.g., visual grounding).
- Using **Mixture of Detectors (MoD)** features can still improve the VQA performance even with a strong reference model
- Fix K=100 is better than dynamic K=[10,100] on overall accuracy, but they both have advantages on some aspects over each other
- Ensemble of **diverse models** are important to further boost the performance

Q&A

- Special thanks to:
  - VQA Challenge organizers
  - Peter(@peteranderson80) to release the bottom-up-attention codes and models
  - FAIR for releasing the Detectron project

- Our papers and codes
  - Yu *et al.*, Multi-modal Factorized Bilinear Pooling with Co-attention Learning for Visual Question Answering, ICCV 2017
  - Yu *et al.*, Beyond Bilinear: Generalized Multi-modal Factorized High-order Pooling for Visual Question Answering, IEEE TNNLS 10.1109/TNNLS.2018.2817340
  - https://github.com/yuzcccc/vqa-mfb

Team HDU-UCAS-USYD with members

Zhou Yu[1], Jun Yu[1], Chenchao Xiang[1], Liang Wang[1], Dalu Guo[3], Qingming Huang[2], Jianping Fan[1] and Dacheng Tao[3]

1. Hangzhou Dianzi University, China
2. University of Chinese Academy of Science, China
3. The University of Sydney, Australia