Supplementary Material:
In Defense of Grid Features for Visual Question Answering

Huaizu Jiang[1,2], Ishan Misra[2], Marcus Rohrbach[2], Erik Learned-Miller[1], and Xinlei Chen[2]
[1]UMass Amherst, [2]Facebook AI Research (FAIR)
{hzjiang,elm}@cs.umass.edu, {imisra,mrf,xinleic}@fb.com

A. Overview

In this supplementary material, we provide:

- Details of hyperparameters of all models in Section B;
- Results of using region features from a Feature Pyramid Network (FPN) in Section C, which is used in [3];
- Details of Pyramid Pooling Module (PPM) [8] in Section D, which is used in end-to-end training.

B. Details of Hyperparameters

model	dataset	optimizer	# iterations	batch size	initial lr	lr decay	lr schedule	gradient clip
Faster R-CNN	VG/COCO	SGD	90K	16	0.02	0.1	[60K, 80K]	-
[3]	VQA 2.0, `train`	Adamax [4]	12K	512	0.01	0.1	[5K, 7K, 9K, 11K]	0.25
[3]	VQA 2.0, `train` + `vqa-eval`	Adamax	22K	512	0.01	0.1	[15K, 18K, 20K, 21K]	0.25
MCAN [7]	VQA 2.0 `trainval` + VG	Adam	234K[1]	64	5e-5	0.02	[180K, 216K]	-
[3]	VizWiz	Adamax	24K	128	0.005	0.01	[14K]	0.25
[1][2]	COCO Karpathy split	Adamax	50K	256	0.002	0.1	[15K, 25K, 35K, 45K]	0.25
e2e [3]	VQA 2.0, `train` +`vqa-eval`	Adamax	22K	512	0.002	0.1	[15K, 18K, 20K, 21K]	1

Table 1: Summary of hyperparameters. We follow the default setting for most of the models. For the image captioning model [1, 3], the default initial learning rate is 0.01. We found 0.002 leads to slightly better results. For the end-to-end trained Pythia (e2e Pythia in the last row), we use initial learning rate of 0.002 and a larger value of 1 for the gradient clip when fine-tuning the ResNet model for feature extraction.

Hyperparameters of different models are summarized in Table 1. For the SGD optimizer, the momentum is 0.9 and weight decay is 0.0001. For the Adamax optimizer, β_1 and β_2 are 0.9 and 0.999, respectively. No weight decay is used. For the Adam optimizer used in MCAN [7], β_1 and β_2 are 0.9 and 0.98, respectively. No weight decay is used.

We follow the default setting of hyperparameters for most of models. For the image captioning model [1, 3], the default initial learning rate is 0.01. We found 0.002 leads to slightly better results. For the end-to-end trained Pythia (e2e Pythia in the last row), we use an initial learning rate of 0.002 and a larger value of 1 for the gradient clip when fine-tuning the ResNet model for feature extraction.

C. Region Features from FPN

In the Pythia implementation [3] of bottom-up attention [1], a Feature Pyramid Network (FPN) model [5] is used to compute region features. This is different from the original Faster R-CNN model [6] used, and it is commonly believed that FPN can offer *better* object detection quality. Therefore, to reach a more solid conclusion, in this appendix we show extended results from the main paper to compare our grid features with FPN region features. The FPN model uses an entire ResNet

[1]In the MCAN paper, the model is trained for 13 epochs, where each epoch contains 17,967 iterations.
[2]We use the implementation provided in [3].

	# features (N)	test-dev accuracy	inference time breakdown (ms)				
			shared conv.	region feat. comp.	region selection	VQA	total
R	100	66.13	9	326	548	6	889
	608	66.22	9	322	544	7	882
R w/ FPN	100	66.01	11	311	690	5	1017
	608	66.36	12	323	690	7	1032
G	608	66.27	11	-	-	7	18

Table 2: This table extends Table 2 in the main paper for **speed and accuracy comparisons** with added rows for region features with FPN. Results are reported on VQA 2.0 test-dev with accuracy and inference time breakdown measured in milliseconds per image. Despite the advantages which FPN features have that 1) pools features from higher-resolution feature maps; and 2) fine-tunes the fc7 layer [3] when training VQA; our grid features achieve comparable VQA accuracy to all region features and are much faster.

	VQA 2.0 accuracy				time (ms)
	Yes/No	Number	Other	Overall	
[3]	-	-	-	68.31	-
R	84.73	46.88	58.98	68.21	929
R, w/ FPN	83.88	45.13	58.12	67.26	1069
G	84.13	45.98	58.76	67.76	39

(a)

	VQA 2.0 accuracy				time (ms)
	Yes/No	Number	Other	Overall	
[7]	87.39	52.78	60.98	70.93	-
R	88.19	54.38	62.19	72.01	963
R, w/ FPN	87.77	54.72	62.16	71.87	1100
G	88.46	55.68	62.85	72.59	72

(b)

	VizWiz accuracy					time (ms)
	Yes/No	Number	Other	Un. Ans.	Overall	
[3]	-	-	-	-	54.22	-
R	73.17	28.89	83.63	35.62	54.28	874
R, w/ FPN	73.00	27.11	82.02	33.59	52.50	1051
G	75.17	24.89	83.68	35.35	54.17	38

(c)

	B4	B3	B2	B1	RL	M	C	S	time (ms)
[1]	36.2	-	-	77.2	56.4	27.0	113.5	20.3	-
R	36.2	46.8	60.4	76.4	56.5	27.7	113.9	20.8	1101
R, w/ FPN	35.7	46.5	60.3	76.6	56.4	27.5	113.1	20.6	1099
G	36.4	47.3	61.1	76.7	56.6	27.4	113.8	20.7	240

(d)

Table 3: This table extends Table 6 in the main paper for **generalization experiments**. From left to right: (a) Different *backbone*. We use a ResNeXt-101-32x8d instead of a ResNet-50 as the backbone. (b) Different *VQA model*. We use MCAN [7] implementation which is the state-of-the-art VQA model. (c) Accuracy on *VizWiz* using the same VQA models [3]. (d) *Image captioning* on COCO Karpathy test split. Abbreviations: BLEU4 (B4), BLEU3 (B3), BLEU2 (B2), BLEU1 (B1), ROUGE_L (RL), METEOR (M), CIDEr (C), and SPICE (S). Our grid features generalize well by achieving results at-par with bottom-up region features while being significantly faster.

model as the backbone, where the multi-scale feature maps of different blocks of the ResNet model are fused in a feature pyramid. Two randomly initialized fully-connect layers (denoted as fc6 and fc7 for simplicity) are added to predict object category, bounding box regression offsets, and attribute labels for each bounding box proposal. We follow the strategy used in [3] to compute region features. Specifically, we use the output of the fc6 layer as input to a VQA or image captioning model, where the fc7 layer is also used and *fine-tuned* during VQA training.

Accuracy on the VQA 2.0 test-dev set and breakdown inference time of the FPN model, using a ResNet50 as the backbone, are summarized in Table 2. Different from the trend observed in object detection [5], we find the FPN model, when used to provide region features for VQA, does not show clear advantage over the original C4 model [1], which in turn gives on-par results to our grid features. Speed-wise, despite the lighter pre-region computation, we find the region-related steps with FPN are still very expensive, and the efficiency advantage of our grid features is even more significant.

We also test the top 100 (N=100) regions using different backbones, VQA models, VQA tasks, and image captioning task, as we have done in Section 6 in the paper. Results are reported in Table 3a, 3b, 3c, and 3d. For the accuracy on the VQA 2.0 test-dev set and VizWiz, the FPN model's accuracy is lower than the results reported in [3], because grid features (from an ImageNet pre-trained ResNet-152 [2] model) are used in *addition* to the region features [3]. Using the MCAN model [7], the FPN model achieves better results than reported in [7] but still performs worse than C4 and our grid features.

D. Details of PPM

In Section 7 of the main paper, we introduce end-to-end training of the Pythia VQA model [3] with PPM (Pyramid Pooling Module) [8]. A detailed illustration of this module is provided in Fig. 1. Given a grid convolution feature map from a ResNet

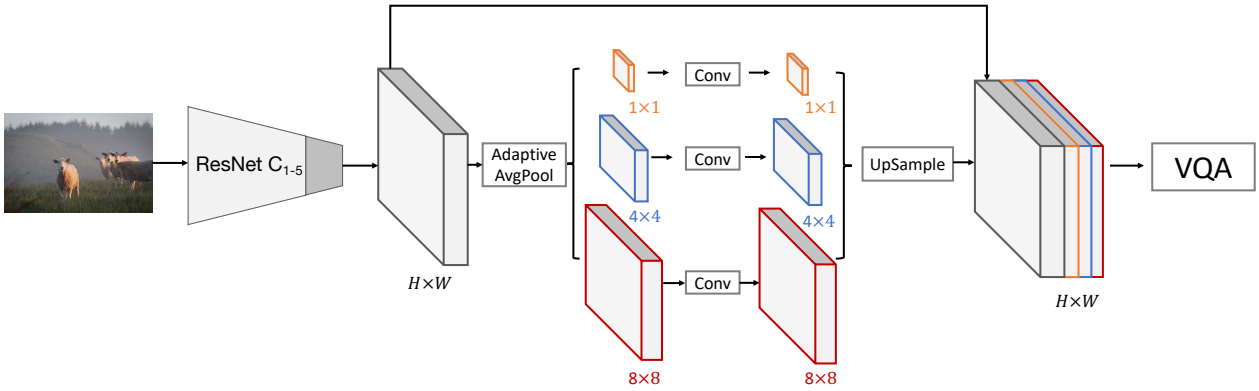

Figure 1: Illustration of **PPM (Pyramid Pooling Module)** [8] experimented in the end-to-end model for VQA. See Section D for details.

model, adaptive average pooling operations are performed at three different spatial resolutions: 1×1, 4×4, and 8×8. Three separate convolution layers (followed by batch normalization and ReLU) are added, where the kernel sizes are all set to 1 and output dimensions are all 512. Finally, the original grid feature map is concatenated together with the three ones obtained from PPM as the input for VQA.

References

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018. 1, 2

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[3] Yu Jiang, Vivek Natarajan, Xinlei Chen, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. Pythia v0.1: the winning entry to the vqa challenge 2018. *arXiv preprint arXiv:1807.09956*, 2018. 1, 2

[4] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 1

[5] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 1, 2

[6] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 1

[7] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *CVPR*, 2019. 1, 2

[8] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 1, 2, 3