SUPPLEMENTARY MATERIAL
MOVIE: REVISITING MODULATED CONVOLUTIONS FOR VISUAL COUNTING AND BEYOND

Anonymous authors
Paper under double-blind review

1 RELATED WORK FOR COUNTING TASKS

Specialized counting. Counting for specialized objects has a number of practical applications Marsden et al. (2018), including but are not limited to cell counting Xie et al. (2018), crowd counting Sindagi & Patel (2017), vehicle counting Onoro-Rubio & López-Sastre (2016), wild-life counting Arteta et al. (2016), *etc*. While less general, they are important computer vision applications to respective domains, *e.g*. medical and surveillance. Standard convolution filters are used extensively in state-of-the-art models Cheng et al. (2019) to produce density maps that approximate the target count number in a local neighborhood. However, such models are designed to deal exclusively with a *single* category of interest, and usually require point-level supervision Sindagi & Patel (2017) in addition to the ground-truth overall count number for training.

Another line of work on specialized counting is psychology-inspired Cutini & Bonato (2012), which focuses on the phenomenon coined 'subitizing' Kaufman et al. (1949), that humans and animals can immediately tell the number of salient objects in a scene using holistic cues Zhang et al. (2015). It is specialized because the number of objects is usually limited to be small (*e.g*. up to 4 Zhang et al. (2015)).

General visual counting. Lifting the restrictions of counting one category or a few items at a time, more general task settings for counting have been introduced. Generalizing to multiple semantic classes and more instances, common object counting Chattopadhyay et al. (2017) as a task has been explored with a variety of strategies such as detection Ren et al. (2015), ensembling Galton (1907), or segmentation Laradji et al. (2018). The most recent development in this direction Cholakkal et al. (2019) also adopts a density map based approach, achieving state-of-the-art with weak, image-level supervision alone. Even more general is the setting of open-ended counting, where the counting target is expressed in natural language questions Acharya et al. (2019). This allows more advanced 'reasoning' task to be formulated involving objects, attributes, relationships, and more. Our module is designed for these general counting tasks, with the modulation coming from either a question or a class embedding.

2 IMPLEMENTATION DETAILS

In our experiments, query representations $\mathbf{q}\in\mathbb{R}^D$ have two types: questions and categories.

Question representation. We use LSTM and Self Attention (SA) layers Vaswani et al. (2017) for question encoding Yu et al. (2019). It was shown in Natural Language Processing (NLP) research that adding SA layers helps to produce informative and discriminative language representations Devlin et al. (2019); and we also empirically observe better results (0.7% improvement in accuracy and 0.1 reduction in RMSE according to our analysis on HowMany-QA Trott et al. (2018) *val* set). Specifically, a question (or sentence in NLP) consisting of N words is first converted into a sequence $\mathbf{Q}^0=\{w_1^0,\dots,w_N^0\}$ of N 300-dim GloVe word embeddings Pennington et al. (2014), which are then fed into a one-directional LSTM followed by a stacked of L=4 layers of self attention:

$$\mathbf{Q}^1 = \overrightarrow{\text{LSTM}}(\mathbf{Q}^0), \tag{1}$$

$$\mathbf{Q}^l = \text{SA}_l(\mathbf{Q}^{l-1}), \tag{2}$$

where $\mathbf{Q}^l=\{w_1^l, \ldots, w_N^l\}$, $l\in\{2, \ldots, L+1\}$ is the D=512 dimensional embedding for each word in the question after the $(l-1)$-th SA layer. We cap all the questions to the same length N as in common practice and pad all-zero vectors to shorter questions Nguyen & Okatani (2018).

Our design for the self-attention layer closely follows Vaswani et al. (2017) and uses multi-head attentions (h=8 heads) with each head having $D_k=D/h$ dimensions and attend with a separate set of keys, queries, and values. Layer normalization and feed-forward network are included without position embeddings.

Given the final $\mathbf{Q}^{L+1}$, to get the conditional vector $\mathbf{q}$, we resort to a summary attention mechanism Nguyen & Okatani (2018). A two-layer 512-dim MLP with ReLU non-linearity is applied to compute an attention score s_n for each word representation w_N^{L+1}. We normalize all scores by softmax to derive attention weights α_n and then compute an aggregated representation $\mathbf{q}$ via a weighted summation over $\mathbf{Q}^{L+1}$.

Object detection based counting. We train a Faster R-CNN Ren et al. (2015) with feature pyramid networks Lin et al. (2017) using the latest implementation on Detectron2 Wu et al. (2019). For fair comparison, we also use a ResNet-50 backbone He et al. (2016) pre-trained on ImageNet, the same for our counting module. The detector is trained on the *train2014* split of COCO images, which is referred as the *train* set for common object counting Chattopadhyay et al. (2017). We train the network for 90K iterations, reducing learning rate by 0.1× at 60K and 80K iterations – starting from a base learning rate of 0.02. The batch size is set to 16. Both left-right flipping and scale augmentation (randomly sampling shorter-side from {640, 672, 704, 736, 768, 800}) are used. The reference AP Ren et al. (2015) on COCO *val2017* split is 37.1. We directly convert the testing output of the detector to the per-category counting numbers.

3 Definition of RMSE Variants

Besides accuracy and RMSE,[1] object counting Chattopadhyay et al. (2017) additionally proposed several variants of RMSE to evaluate a system's counting ability. For convenience, we also include them here. The standard RMSE is defined as:

$$\text{RMSE} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(\hat{c}_i - c_i)^2}, \tag{3}$$

where $\hat{c}_i$ is ground-truth, c_i is prediction, and N is the number of examples. Focusing more on non-zero counts, RMSE-nz tries to evaluate a model's counting ability on harder examples where the answer is at least one:

$$\text{RMSE-nz} = \sqrt{\frac{1}{M_{nz}}\sum_{i\in\{i|\hat{c}_i>0\}}(\hat{c}_i - c_i)^2}, \tag{4}$$

where M_{nz} is the number of examples where ground-truth is non-zero. To penalize the mistakes when the count number is small (as making a mistake of 1 when the ground-truth is 2 is more serious than when the ground-truth is 100), rel-RMSE is proposed as:

$$\text{rel-RMSE} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\frac{(\hat{c}_i - c_i)^2}{\hat{c}_i + 1}}. \tag{5}$$

And finally, rel-RMSE-nz is used to calculate the relative RMSE for non-zero examples – both challenging and aligned with human perception.

4 Common Objects Counting on VOC

As mentioned in Sec. **??** of the main paper, we present the performance of MoVie on *test* split of Pascal VOC counting dataset in Tab. 1. Different from COCO, the VOC dataset is much smaller with

[1] https://en.wikipedia.org/wiki/Root-mean-square_deviation

Method	Instance supervision	RMSE ↓	RMSE-nz ↓	rel-RMSE ↓	rel-RMSE-nz ↓
LC-ResFCN (2018)	✓	0.31	1.20	**0.17**	0.61
LC-PSPNet (2018)	✓	0.35	1.32	0.20	0.70
glance-noft-2L (2017)	✗	0.50	1.83	0.27	0.73
CountSeg (2019)	✗	**0.29**	**1.14**	**0.17**	0.61
MoVie	✗	0.36	1.37	0.18	**0.56**

Table 1: **Common object counting** on VOC *test* set with various RMSE metrics.

	# Train params (M)	# Test params (M)	Train mem (G)	Train speed (s/iter)
MCAN-Large (2019)	218.2	218.2	10.5	0.84
MCAN-Large + MoVie	260.2	241.2	11.7	0.89

Table 2: Adding MoVie as a module to MCAN. Training speed is ∼5% slower, and the additional parameters during testing is minimal (∼10%) as it uses the joint branch only.

20 object categories Everingham et al. (2015). We can see that MoVie achieves comparable results to the state-of-the-art method, CountSeg Cholakkal et al. (2019) in two relative metrics (rel-RMSE and rel-RMSE-nz) and falls behind in RMSE and RMSE-nz. In contrast, as shown in Tab. **??**, MoVie outperforms CountSeg on COCO with a significant margin on three RMSE metrics. The performance difference in two datasets suggests that MoVie scales better than CountSeg in terms of dataset size and number of categories. Moreover, the maintained advantage on relative metrics indicates the output of MoVie is better aligned with human perception Chattopadhyay et al. (2017).

5 Counting Module for VQA Architectures

As mentioned in **??**, we integrate MoVie into VQA models as a counting module. Naturally, such an integration leads to changes in model size, training speed *etc*. We report the added computational costs in Tab. 2 for our three-branch fusion scheme into MCAN. We see the training speed is only ∼5% slower, and the additional parameters used during testing is kept minimal (∼10%). Note that since the integration of MoVie mainly benefits 'number' questions, it is different from general model size increase.

Similar to MCAN, we also conducted experiments incorporating MoVie to Pythia Jiang et al. (2018), the 2018 VQA challenge winner, where we trained using the VQA 2.0 *train+val*, and evaluated on *test-dev* using the server. We observe even more significant improvements on Pythia for 'number' related questions (Tab. 3). MoVie also improves the performance of the network in all other categories, verifying that our MoVie generalizes to multiple VQA architectures.

6 Visualization of Where MoVie Helps

When MoVie is used as a counting module for generic VQA tasks, we fuse features pooled from MoVie and the features used by state-of-the-art VQA models (*e.g*. MCAN Yu et al. (2019)) to jointly predict the answer. Then a natural question arise: where does MoVie help? To answer this question, we want visualize how important MoVie and the original VQA features contribute to the final answer produced by the joint model. We conduct this study for each of the 55 question types listed in the VQA 2.0 dataset Antol et al. (2015) for better insights.

Specifically, suppose a fused representation in the joint branch is denoted as $\mathbf{o}=f(\mathbf{i}+\mathbf{v})$, where $\mathbf{i}$ is from the VQA model, $\mathbf{v}$ is from MoVie, and $f(\cdot)$ is the function consisting of layers applied after the features are summed up. We can compute two variants of this representation $\mathbf{o}$: one *without* $\mathbf{v}$: $\mathbf{o}_{-\mathbf{v}}=f(\mathbf{i})$, and one *without* $\mathbf{i}$: $\mathbf{o}_{-\mathbf{i}}=f(\mathbf{v})$. The similarity score is then computed between two pairs via dot-product: $p_{\mathbf{i}}=\mathbf{o}^T\mathbf{o}_{-\mathbf{v}}$ and $p_{\mathbf{v}}=\mathbf{o}^T\mathbf{o}_{-\mathbf{i}}$. Given one question, we assign a score of 1 to MoVie if $p_{\mathbf{i}}>p_{\mathbf{v}}$, and otherwise 0. The scores within each question type are then averaged, and produces the probability of how MoVie is chosen over the base VQA model for that particular question type.

Method	Test set	Yes/No ↑	Number ↑	Other ↑	Overall ↑
MCAN-Large + X-101 (2020)	*test-dev*	88.46	55.68	62.85	72.59
MCAN-Large + X-101 + MoVie		88.39	57.05	63.28	72.91
Pythia + X-101 (2018)	*test-dev*	84.13	45.98	58.76	67.76
Pythia + X-101 + MoVie		85.15	53.25	59.31	69.26

Table 3: **VQA accuracy** of Pythia with and without MoVie on VQA 2.0 *test-dev* set.

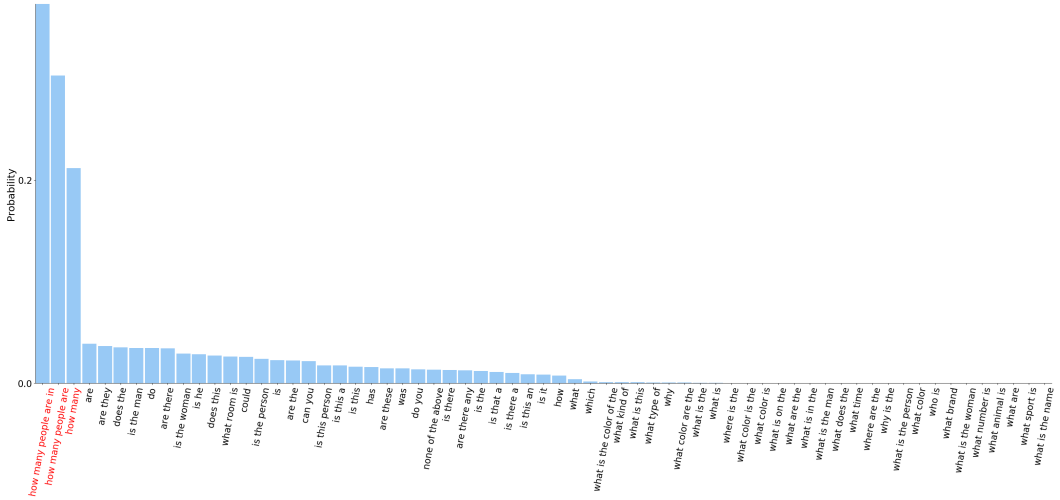

Figure 1: Visualization of where MoVie helps MCAN for different question types on VQA 2.0 *val* set. We compute the probability by assigning each question to MoVie based on similarity scores (see Sec. 6 for detailed explanations). The top contributed question types are counting related, confirming that state-of-the-art VQA models that perform global fusion are not ideally designed for counting, and the value of MoVie with local fusion. Best viewed on a computer screen with zoom.

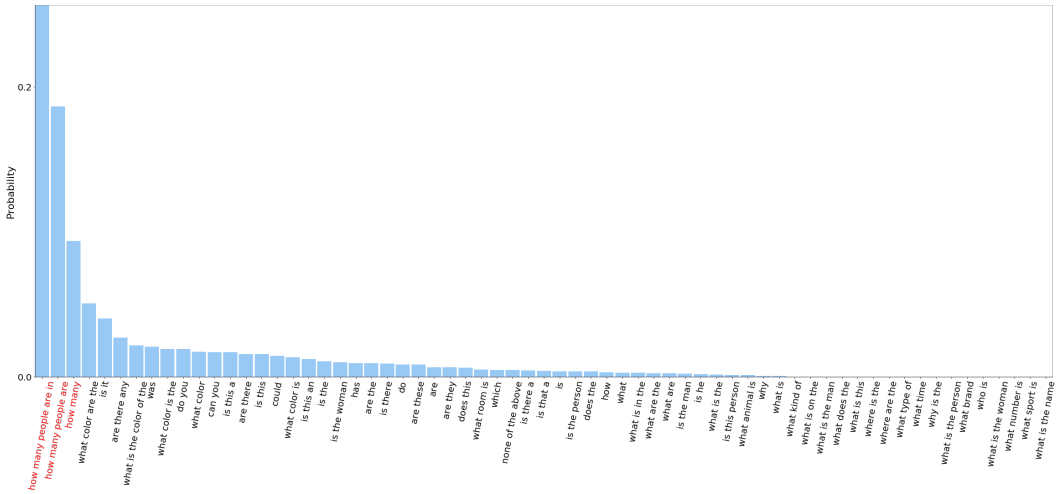

Figure 2: Similar to Fig. 1 but with Pythia. Best viewed on a computer screen with zoom.

We take two models as examples. One is MCAN-Small Yu et al. (2019) + MoVie (three-branch), and the other one replaces MCAN-Small with Pythia Jiang et al. (2018). The visualizations are shown in Fig. 1 and Fig. 2, respectively. We sort the question types based on how much MoVie has contributed, *i.e.* the 'probability'. Some observations:

- MoVie shows significant contribution in the counting questions for both MCAN and Pythia: the top three question types are consistently 'how many people are in', 'how many people are', and 'how many', this evidence strongly suggests that existing models that fuse features *globally* between vision and language are not well suited for counting questions, and confirms the value of incorporating MoVie (that performs fusion *locally*) as a counting module for generic VQA models;
- The 'Yes/No' questions are likely benefited from MoVie as well, since the contribution of MoVie spreads in several question types belong to that category (*e.g.* 'are', 'are they', 'do', *etc.*) – this maybe because counting also includes 'verification-of-existing' questions such as 'are there people wearing hats in the image';
- For Pythia, we also find it likely helps 'color' related questions (*e.g.* 'what color are the', 'what color', *etc.*) and some other types – this strengthens our exploration that our model contributes beyond counting capability.

REFERENCES

Manoj Acharya, Kushal Kafle, and Christopher Kanan. Tallyqa: Answering complex counting questions. In *AAAI*, 2019.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *ICCV*, 2015.

Carlos Arteta, Victor Lempitsky, and Andrew Zisserman. Counting in the wild. In *ECCV*, 2016.

Prithvijit Chattopadhyay, Ramakrishna Vedantam, Ramprasaath R. Selvaraju, Dhruv Batra, and Devi Parikh. Counting everyday objects in everyday scenes. In *CVPR*, 2017.

Zhi-Qi Cheng, Jun-Xiu Li, Qi Dai, Xiao Wu, and Alexander Hauptmann. Learning spatial awareness to improve crowd counting. In *ICCV*, 2019.

Hisham Cholakkal, Guolei Sun, Fahad Shahbaz Khan, and Ling Shao. Object counting and instance segmentation with image-level supervision. In *CVPR*, 2019.

Simone Cutini and Mario Bonato. Subitizing and visual short-term memory in human and non-human species: a common shared system? *Frontiers in Psychology*, 2012.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *ACL*, 2019.

M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 2015.

Francis Galton. One vote, one value. *Nature*, 1907.

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

Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik Learned-Miller, and Xinlei Chen. In defense of grid features for visual question answering. *arXiv preprint arXiv:2001.03615*, 2020.

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.

Edna L Kaufman, Miles W Lord, Thomas Whelan Reese, and John Volkmann. The discrimination of visual number. *The American journal of psychology*, 1949.

Issam H Laradji, Negar Rostamzadeh, Pedro O Pinheiro, David Vazquez, and Mark Schmidt. Where are the blobs: Counting by localization with point supervision. In *ECCV*, 2018.

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.

Mark Marsden, Kevin McGuinness, Suzanne Little, Ciara E Keogh, and Noel E O'Connor. People, penguins and petri dishes: Adapting object counting models to new visual domains and object types without forgetting. In *CVPR*, 2018.

Duy-Kien Nguyen and Takayuki Okatani. Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering. In *CVPR*, 2018.

Daniel Onoro-Rubio and Roberto J López-Sastre. Towards perspective-free object counting with deep learning. In *ECCV*, 2016.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

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

Vishwanath A. Sindagi and Vishal M. Patel. A survey of recent advances in cnn-based single image crowd counting and density estimation. *PR Letters*, 2017.

Alexander Trott, Caiming Xiong, and Richard Socher. Interpretable counting for visual question answering. In *ICLR*, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2, 2019. URL https://github.com/facebookresearch/detectron2.

Weidi Xie, J Alison Noble, and Andrew Zisserman. Microscopy cell counting and detection with fully convolutional regression networks. *Computer methods in biomechanics and biomedical engineering: Imaging & Visualization*, 2018.

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

Jianming Zhang, Shugao Ma, Mehrnoosh Sameki, Stan Sclaroff, Margrit Betke, Zhe Lin, Xiaohui Shen, Brian Price, and Radomir Mech. Salient object subitizing. In *CVPR*, 2015.