Visual Commonsense Representation Learning via Causal Inference

Tan Wang[1,3], Jianqiang Huang[2,3], Hanwang Zhang[3], Qianru Sun[4]
[1]University of Electronic Science and Technology of China [2]Damo Academy, Alibaba Group
[3]Nanyang Technological University [4]Singapore Management University
wangt97@hotmail.com, jianqiang.jqh@gmail.com, hanwangzhang@ntu.edu.sg, qianrusun@smu.edu.sg

Abstract

*We present a novel unsupervised feature representation learning method, Visual Commonsense Region-based Convolutional Neural Network (**VC R-CNN**[1]), to serve as an improved visual region encoder for high-level tasks such as captioning and VQA. Given a set of detected object regions in an image (e.g., using Faster R-CNN), like any other unsupervised feature learning methods (e.g., word2vec), the proxy training objective of VC R-CNN is to predict the contextual objects of a region. However, they are fundamentally different: the prediction of VC R-CNN is by using **causal intervention**: $P(Y|do(X))$, while others are by using the conventional **likelihood**: $P(Y|X)$. We extensively apply VC R-CNN features in prevailing models of two popular tasks: Image Captioning and VQA, and observe consistent performance boosts across all the methods, achieving many new state-of-the-arts[2].*

1. Introduction

Today's computer vision systems are good at telling us "what" (*e.g.*, classification [5], segmentation [4]) and "where" (*e.g.*, detection [9]), yet bad at knowing "why", by asking for high-level *commonsense*. That is still elusive, even for our human philosophers [3], not to mention for machines.

It is not hard to spot the "cognitive errors" committed by machines due to the lack of common sense. As shown in Figure 1, by using only the visual features, *e.g.*, the prevailing Faster R-CNN [9] based Up-Down [1], machine usually fails to describe the exact visual relationships (the captioning example), or, even if the prediction is correct, the underlying visual attention is not reasonable (the VQA example). Previous works blame this for dataset bias without further justification [6], *e.g.*, the large concept co-occurrence gap in Figure 1; but here we take a closer look at it by appreciating the difference between the "visual" and "commonsense" features. As the "visual" only tells "what"/"where"

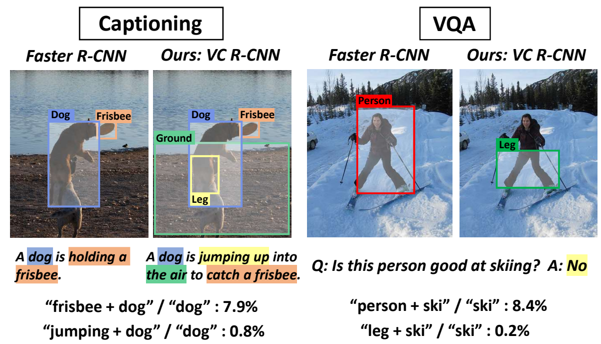

Figure 1. Examples of "cognitive errors" in image captioning and VQA due to the dataset bias. The ratio ./. denotes the co-occurrence% in captions and VQA questions. By comparing with the Faster R-CNN [9] based features [1], our VC R-CNN features can correct the errors, *e.g.*, more accurate visual relationships and visual attentions, by being more commonsense awareness.

about `person` or `leg` *per se*, it is just a more descriptive symbol than its correspondent English word; when there is bias, *e.g.*, there are more `person` than `leg` regions co-occur with the word "ski", the visual attention is thus more likely to focus on the `person` region.

We are certainly not the first to believe that visual features should include more commonsense knowledge. There is a trend in our community towards *weakly-supervised* learning features from large-scale vision-language corpus [8]. However, despite the challenge in trading off between annotation cost and noisy multimodal pairs, common sense is not always recorded in text due to the reporting bias [10], *e.g.*, most may say "people walking on road" but few will point out "people walking with legs". In fact, we humans naturally learn common sense in an *unsupervised fashion* by exploring the physical world, and we wish that machines can also imitate in this way.

A successful example is the unsupervised learning of word vectors in our sister NLP community: a word representation X is learned by predicting its contextual word Y, *i.e.*, $P(Y|X)$ in a neighborhood window. However, its counterpart in our own community, such as learning by predicting surrounding objects or parts [2], is far from effec-

[1]Please refer to the full version of this paper in [11] for better clarity.
[2]https://github.com/Wangt-CN/VC-R-CNN

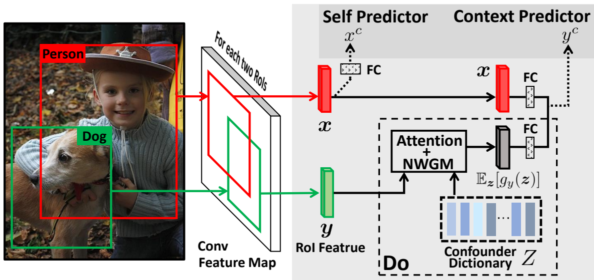

Figure 2. The overview of VC R-CNN. Any R-CNN backbone (*e.g.*, Faster R-CNN [9]) can be used to extract regions of interest (RoI) on the feature map. Each RoI is then fed into two sibling branches: a **Self Predictor** to predict its own class, *e.g.*, x^c, and a **Context Predictor** to predict its context labels, *e.g.*, y^c, with our **Do** calculus. The architecture is trained with a multi task loss.

tive in down-stream tasks. The reason is that the common-sense knowledge, in the form of language sentences, has already been recorded in discourse; in contrast, once an image has been taken, the explicit knowledge why objects are contextualized will never be observed, so the true common sense that **causes** the existence of objects X and Y might be **confounded** by the spurious *observational bias*, *e.g.*, if `keyboard` and `mouse` are more often observed with `table` than any other objects, the underlying common sense that `keyboard` and `mouse` are parts of `computer` will be wrongly attributed to `table`.

In this paper, we proposed an unsupervised region feature learning method: Visual Commonsense R-CNN (**VC R-CNN**), as illustrated in Figure 2, which uses Region-based Convolutional Neural Network (R-CNN) [9] as the visual backbone, and the **causal intervention** as the training objective. Besides its novel learning fashion, we also design a novel algorithm as an effective approximation for the imaginative intervention (cf. Section 2.2). The delivery of VC R-CNN is a region feature extractor for any region proposal, and thus it is fundamental and ready-to-use for many high-level vision tasks such as Image Captioning and VQA. Through extensive experiments in Section 3, VC R-CNN shows significant and consistent improvements over strong baselines — the prevailing methods in each task.

2. Sense-making by Intervention

2.1. Causal Intervention

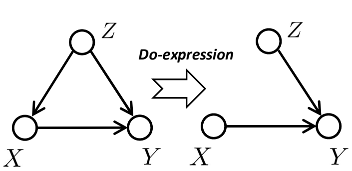

Figure 3. The causal intervention $P(Y|do(X))$. Nodes denote variables and arrows denote the direct causal effects.

As shown in Figure 3 (left), our visual world exists many confounders $z \in Z$ that affects (or causes) either X or Y, leading to spurious correlations by only learning from the likelihood $P(Y|X)$. To see this, by using Bayes rule:

$$P(Y|X) = \sum\nolimits_z P(Y|X,z)\,\underline{P(z|X)}, \tag{1}$$

where the confounder Z introduces the observational bias via $P(z|X)$. As illustrated in Figure 3 (right), if we intervene X, *e.g.*, $do(X)$, the causal link between Z and X is cut-off. By applying Bayes rule on the new graph, we have:

$$P(Y|do(X)) = \sum\nolimits_z P(Y|X,z)\,\underline{P(z)}. \tag{2}$$

Compared to Eq. (1), z is no longer affected by X, and thus the intervention deliberately forces X to incorporate every z fairly, subject to its prior $P(z)$, into the prediction of Y. Therefore, by using intervention $P(Y|do(X))$ as the feature learning objective, we can adjust between "common" and "sense-making", thus alleviate the observational bias.

2.2. The Proposed Implementation

To implement the theoretical and imaginative intervention in Eq. (2), we propose the proxy task of predicting the local context labels of Y's RoI. For the confounder set Z, since we can hardly collect all confounders in real world, we approximate it to a fixed confounder dictionary $\boldsymbol{Z} = [\boldsymbol{z}_1, ..., \boldsymbol{z}_N]$ in the shape of $N \times d$ matrix for practical use, where N is the category size in dataset (*e.g.*, 80 in MS-COCO) and d is the feature dimension of RoI. Each entry $\boldsymbol{z}_i$ is the averaged RoI feature of the i-th category samples in dataset. The feature is pre-trained by Faster R-CNN.

Specifically, given X's RoI feature $\boldsymbol{x}$ and its contextual Y's RoI whose class label is y^c, Eq. (2) can be implemented as $\sum_{\boldsymbol{z}} P(y^c|\boldsymbol{x},\boldsymbol{z})\,P(\boldsymbol{z})$. The last layer of the network for label prediction is the Softmax layer: $P(y^c|\boldsymbol{x},\boldsymbol{z}) = \mathit{Softmax}(f_y(\boldsymbol{x},\boldsymbol{z}))$, where $f_y(\cdot)$ calculates the logits for N categories, and the subscript y denotes that $f(\cdot)$ is parameterized by Y's RoI feature $\boldsymbol{y}$, motivated by the intuition that the prediction for y^c should be characterized by Y. In summary, the implementation is defined as:

$$P(Y|do(X)) := \mathbb{E}_{\boldsymbol{z}}[\mathit{Softmax}(f_y(\boldsymbol{x},\boldsymbol{z}))]. \tag{3}$$

Note that $\mathbb{E}_{\boldsymbol{z}}$ requires expensive sampling.

Normalized Weighted Geometric Mean (NWGM). We apply NWGM [12] to approximate the above expectation. In a nutshell, NWGM[3] effeciently moves the outer expectation into the Softmax as:

$$\mathbb{E}_{\boldsymbol{z}}[\mathit{Softmax}(f_y(\boldsymbol{x},\boldsymbol{z}))] \overset{\text{NWGM}}{\approx} \mathit{Softmax}(\mathbb{E}_{\boldsymbol{z}}[f_y(\boldsymbol{x},\boldsymbol{z})]). \tag{4}$$

In this paper, we use the linear model $f_y(\boldsymbol{x},\boldsymbol{z}) = \boldsymbol{W}_1\boldsymbol{x} + \boldsymbol{W}_2 \cdot g_y(\boldsymbol{z})$, where $\boldsymbol{W}_1, \boldsymbol{W}_2 \in \mathbb{R}^{N \times d}$ denote the fully connected layer. Then the Eq. (4) can be derived as:

$$\mathbb{E}_{\boldsymbol{z}}[f_y(\boldsymbol{x},\boldsymbol{z})] = \boldsymbol{W}_1\boldsymbol{x} + \boldsymbol{W}_2 \cdot \mathbb{E}_{\boldsymbol{z}}[g_y(\boldsymbol{z})]. \tag{5}$$

Note that the above approximation is reasonable, because the effect on Y comes from both X and confounder Z (cf. the right Figure 3). Next, the key is to compute $\mathbb{E}_{\boldsymbol{z}}[g_y(\boldsymbol{z})]$.

[3] The detailed derivation about NWGM can be found in the Supp..

Feature	Model	B4	M	R	C	Model	B4	M	R	C
Origin [1, 7]	Up-Down	36.3	27.7	56.9	120.1	AoANet†	38.9	29.2	58.2	129.8
Obj		36.7	27.8	57.5	122.3		38.1	28.4	58.2	126.0
Only VC		34.5	27.1	56.5	115.2		35.8	27.6	56.8	118.1
+Det		37.5	28.0	58.3	125.9		38.8	28.8	58.7	128.0
+Cor		38.1	28.3	58.5	127.5		38.8	28.9	58.7	128.6
+VC		**39.5**	29.0	59.0	130.5		**39.5**	**29.3**	**59.3**	**131.6**

Table 1. The image captioning performances of representative two models with ablative features on Karpathy split. The metrics: B4, M, R and C denote BLEU@4, METEOR, ROUGE-L and CIDEr-D respectively. The grey row highlight our features and the underline denotes the current SOTA results.

Model	BLEU-4		METEOR		ROUGE-L		CIDEr-D	
Metric	c5	c40	c5	c40	c5	c40	c5	c40
Up-Down [1]	36.9	68.5	27.6	36.7	57.1	72.4	117.9	120.5
AoANet [7]	37.3	68.1	28.3	37.2	57.9	72.8	124.0	126.2
Up-Down+VC	37.8	69.1	28.5	37.6	58.2	73.3	124.1	126.2
AoANet†+VC	**38.4**	**69.9**	**28.8**	**38.0**	**58.6**	**73.8**	**125.5**	**128.1**

Table 2. The performances of various single models on the online MS-COCO test server. Up-Down+VC and AoANet†+VC are the short for concatenated on [1] in Up-Down and AoANet†.

Computing $\mathbb{E}_{\boldsymbol{z}}[g_y(\boldsymbol{z})]$. We encode $g_y(\cdot)$ as the Scaled Dot-Product Attention to assign weights for different confounders in dictionary $\boldsymbol{Z}$ with specific $\boldsymbol{y}$. Specifically, given the $\boldsymbol{y}$ and confounder dictionary $\boldsymbol{Z}$, we can have $\mathbb{E}_{\boldsymbol{z}}[g_y(\boldsymbol{z})] = \sum_z [Softmax(\boldsymbol{q}^T \boldsymbol{K}/\sqrt{\sigma}) \odot \boldsymbol{Z}]P(\boldsymbol{z})$, where $\boldsymbol{q} = \boldsymbol{W}_3\boldsymbol{y}$, $\boldsymbol{K} = \boldsymbol{W}_4\boldsymbol{Z}^T$, $P(\boldsymbol{z})$ denotes the prior statistic probability and $\odot$ is the element-wise product, $\boldsymbol{W}_3$ and $\boldsymbol{W}_4$ are the embedding matrices that map each vector to the common subspace for similarity measure, σ denotes the first dimension of $\boldsymbol{W}_3, \boldsymbol{W}_4$ as a constant scaling factor.

2.3. VC R-CNN

Architecture. Figure 2 illustrates the VC R-CNN architecture. VC R-CNN takes an image as input and generates feature map from a CNN backbone (*e.g.*, ResNet101 [5]). Then, unlike Faster R-CNN [9], we discard the Region Proposal Network (RPN). The ground-truth bounding boxes are directly utilized to extract the object level representation with the RoIAlign layer. Finally, each two RoI features $\boldsymbol{x}$ and $\boldsymbol{y}$ eventually branch into two sibling predictors: Self Predictor with a fully connected layer to estimate each object class, while Context Predictor with the approximated *do*-calculus in Eq. (3) to predict the context label.

Training Objectives. The Self-Predictor outputs a discrete probability distribution $p = (p[1], ..., p[N])$ over N categories. The loss can be defined as $L_{self}(p, x^c) = -log(p[x^c])$, where x^c is the ground-truth class of RoI X. The Context Predictor loss L_{cxt} is defined for each two RoI feature vectors. Considering X as the center object while Y_i is one of the K context objects with ground-truth label y_i^c, the loss is $L_{cxt}(p_i, y_i^c) = -log(p_i[y_i^c])$, where p_i is calculated by $p_i = P(Y_i|do(X))$ in Eq. (3) and $p_i = (p_i[1], ..., p_i[N])$ is the probability over N categories. Finally, the overall mulit-task loss for each RoI X is:

Feature	Model	Y/N	Num	Other	All	Model	Y/N	Num	Other	All
Obj [1, 13]	Up-Down	80.3	42.8	55.8	63.2	MCAN	84.8	**49.4**	58.4	67.1
Only VC		77.8	37.9	51.6	59.8		80.8	40.7	49.3	60.1
+Det		81.8	44.5	56.8	64.5		84.8	49.2	58.8	67.2
+Cor		81.5	44.6	57.1	64.7		85.0	49.2	58.9	67.4
+VC		82.5	46.0	57.6	65.4		**85.2**	**49.4**	**59.1**	**67.7**

Table 3. Accuracies of various ablative features on VQA2.0 validation set. For Up-Down and MCAN, since the Obj achieves almost equal results with original paper, we just merge the two rows.

$$L(X) = L_{self}(p, x^c) + \frac{1}{K}\sum_i L_{cxt}(p_i, y_i^c). \tag{6}$$

3. Experiments

3.1. Experimental Setup

Dataset: MS-COCO Detection. We apply our VC R-CNN on the MS-COCO dataset with 80 annotated classes.

Comparative Designs. To evaluate the effectiveness of our VC R-CNN feature (VC), we present two representative vision-and-language downstream tasks (*i.e.*, Image Captioning and VQA) in our experiment. For each task, a **classic** model and a **state-of-the-art** model were both performed for comprehensive comparisons. For each method, we used the following five ablative feature settings: 1) **Obj**: the features based on Faster R-CNN, we adopted the popular used bottom-up feature [1]; 2) **Only VC**: pure VC features; 3) **+Det**: the features from training R-CNN with single self detection branch without Context Predictor. "+" denotes the extracted features are concatenated with the original feature; 4) **+Cor**: the features from training R-CNN by predicting all context labels (*i.e.*, correlation) without the intervention; 5) **+VC**: our full feature with the proposed implemented intervention, concatenated to the original feature. For fair comparisons, we retained all the settings and random seeds in the downstream task models.

3.2. Results and Analysis

Results on Image Captioning. We compared our VC representation with ablative features on two representative approaches: Up-Down [1] and AoANet [7] in Table 1. For Up-Down model, we can observe that with our +VC trained on MS-COCO, the model can even outperform current SOTA method AoANet over most of the metrics. When comparing +VC with the +Det and +Cor without intervention, results also show absolute gains over all metrics, which demonstrates the effectiveness of our proposed causal intervention in representation learning. AoANet [7] proposed an "Attention on Attention" module on feature encoder and caption decoder with the self-attention mechanism. In our experiment, we discarded the AoA refining encoder (*i.e.*, AoANet†) rather than using full AoANet since the self-attentive operation on feature can be viewed as an indiscriminate correlation against our do-expression. From Table 1 we can observe that our +VC with AoANet† achieves a new SOTA performance. We also evaluated our feature

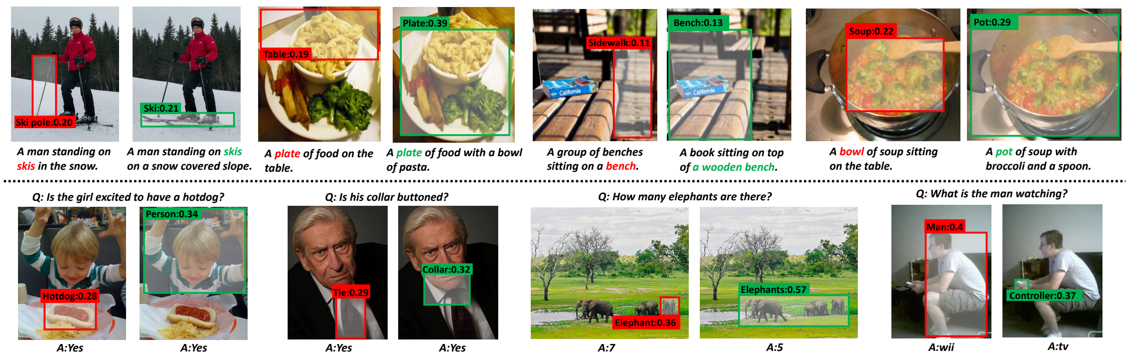

Figure 4. Qualitative examples of utilizing our VC feature (right) compared with using Obj feature (left). Boxes in images denote the attention region labeled with name and attention weight. Three rows represent Image Captioning, VQA and VCR task respectively.

Model	test-dev				test-std
	Y/N	Num	Other	All	All
Up-Down [1]	81.82	44.21	56.05	65.32	65.67
MCAN [13]	86.82	54.04	60.52	70.63	70.90
UP-Down+VC	84.26	48.50	58.86	68.15	68.45
MCAN+VC	**87.41**	**53.28**	**61.44**	**71.21**	**71.49**

Table 4. Single model accuracies on VQA2.0 test-dev and test set.

on the online COCO test server in Table 2. We can find our model also achieves the best single-model scores across all metrics outperforming previous methods significantly.

Results on VQA. In Table 3, we applied our VC feature on classical Up-Down [1] and recent state-of-the-art method MCAN [13]. From the results, our proposed +VC outperforms all the other ablative representations on three answer types, achieving the state-of-the-art performance. However, compared to the image captioning, the gains on VQA with our VC feature are less significant. The potential reason lies in the limited ability of the current question understanding, which cannot be resolved by "visual" common sense. Table 4 reports the single model performance of various models on both test-dev and test-standard sets. Although our VC feature is limited by the question understanding, we still receive the absolute gains by just feature concatenation compared to previous methods with complicated module stack.

Qualitative Analysis. We visualize several examples with our VC feature and previous Up-Down feature in Figure 4. Any other settings except for feature kept the same. We can observe that with our VC, models can choose more precise, reasonable attention area and explicable better performance.

4. Conclusions

We presented a novel unsupervised feature representation learning method called VC R-CNN that can be based on any R-CNN framework, supporting a variety of high-level tasks by using only feature concatenation. The key novelty of VC R-CNN is that the learning objective is based on causal intervention, which is fundamentally different from the conventional likelihood. Extensive experiments on benchmarks showed impressive performance boosts on almost all the strong baselines and metrics. In future, we intend to study the potential of our VC R-CNN applied in other modalities such as video and 3D point cloud.

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, 3, 4

[2] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015. 1

[3] Ibrahim Abou Halloun and David Hestenes. Common sense concepts about motion. *AM J PHYS*, 53(11), 1985. 1

[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 1

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 3

[6] Lisa Anne Hendricks, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach. Women also snowboard: Overcoming bias in captioning models. In *ECCV*, 2018. 1

[7] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *ICCV*, 2019. 3

[8] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NIPS*, 2019. 1

[9] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 1, 2, 3

[10] Ramakrishna Vedantam, Xiao Lin, Tanmay Batra, C Lawrence Zitnick, and Devi Parikh. Learning common sense through visual abstraction. In *ICCV*, 2015. 1

[11] Tan Wang, Jianqiang Huang, Hanwang Zhang, and Qianru Sun. Visual commonsense r-cnn. In *CVPR*, 2020. 1

[12] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 2

[13] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *CVPR*, 2019. 3, 4