arXiv:2002.12204v1 [cs.CV] 27 Feb 2020

Visual Commonsense R-CNN

Tan Wang[1,3], Jianqiang Huang[2], 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**), 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**: $P(Y|X)$. This is also the core reason why VC R-CNN can learn "sense-making" knowledge like* `chair` *can be sat — while not just "common" co-occurrences such as* `chair` *is likely to exist if* `table` *is observed. We extensively apply VC R-CNN features in prevailing models of three popular tasks: Image Captioning, VQA, and VCR, and observe consistent performance boosts across all the methods and tasks, achieving many new state-of-the-arts*[1].

1. Introduction

> *"On the contrary, Watson, you can see everything. You fail, however, to reason from what you see."*
>
> *–Sherlock Holmes, The Adventure of the Blue Carbuncle*

Today's computer vision systems are good at telling us "what" (*e.g.*, classification [21, 29], segmentation [20, 37]) and "where" (*e.g.*, detection [53, 36], tracking [28, 32]), yet bad at knowing "why", *e.g.*, why is it `dog`? Note that the "why" here does not merely mean by asking for *visual* reasons — attributes like furry and four-legged — that are already well-addressed by machines; beyond, it means by asking for high-level *commonsense* reasons — such as `dog` barks [15] — that are still elusive, even for the Aristotelian and pre-Galilean philosophers [19, 56], not to mention for machines.

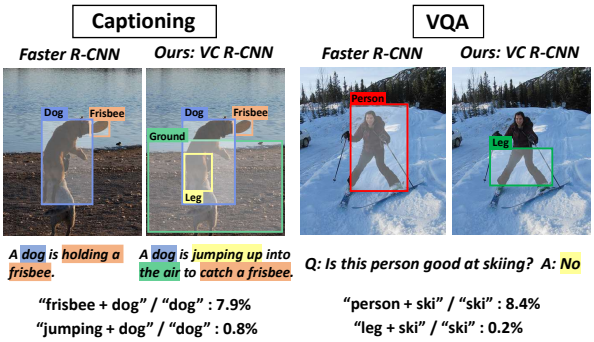


Figure 1. Examples of "cognitive errors" in image captioning and VQA due to the dataset bias. The ratio ./. denotes the co-occurrence% in ground-truth text (captioning: captions, VQA: questions). By comparing with the Faster R-CNN [53] based features [2], our VC R-CNN features can correct the errors, *e.g.*, more accurate visual relationships and visual attentions, by being more commonsense awareness.

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 [53] based Up-Down [2], 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 [22, 42, 52, 7], *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" 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. On the other hand, if we could use the "commonsense" features, the action of "ski" can focuses on the `leg` region

[1] https://github.com/Wangt-CN/VC-R-CNN

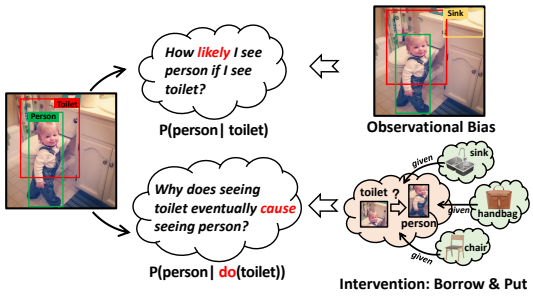


Figure 2. The illustration of why $P(Y|do(X))$ learns common sense while $P(Y|X)$ does not. Thanks to intervention, $P(Y|do(X))$ can "borrow" objects from other images and "put" them into the local image, to perform further justifications if X truly causes Y regardless of the unobserved confounders, and thus alleviate the observational bias.

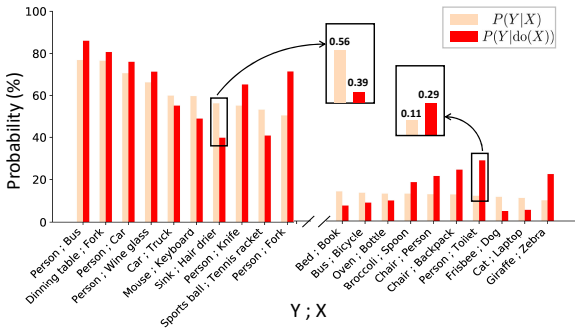


Figure 3. The sensible difference between the likelihood before (*i.e.*,$P(Y|X)$) and after intervention (*i.e.*, $P(Y|do(X))$) in MS-COCO. The object is represented by the 80 ground-truth class labels. Only 20 pairs are visualized to avoid clutter.

because of the common sense: we ski with legs.

We are certainly not the first to believe that visual features should include more commonsense knowledge, rather than just visual appearances. There is a trend in our community towards *weakly-supervised* learning features from large-scale vision-language corpus [39, 58, 59]. However, despite the major challenge in trading off between annotation cost and noisy multimodal pairs, common sense is not always recorded in text due to the reporting bias [64, 35], *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 [43, 11, 49]: 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 [12, 41], is far from effective in down-stream tasks. The reason is that the commonsense 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`.

Intrigued, we perform a toy MS-COCO [34] experiment with ground-truth object labels — by using a mental apparatus, *intervention*, that makes us human [48] — to screen out the existence of confounders and then eliminate their effect. We compare the difference between *association* $P(Y|X)$ and *causal intervention* $P(Y|do(X))$ [47]. As illustrated in Figure 2, you can intuitively understand the intervention as the following deliberate experiment: 1) "borrow" objects Z from other images, 2) "put" them around X and Y, then 3) test if X still causes the existence of Y given Z. The "borrow" and "put" is the spirit of intervention, which implies that the chance of Z is only dependent on us (probably subject to a prior), but independent on X or Y. By doing so, as shown in Figure 3, $P(\texttt{sink}|do(\texttt{dryer}))$ is lower because the most common restroom context such as `towel` is forced to be seen as fair as others. Therefore, by using $P(\texttt{sink}|do(\texttt{dryer}))$ as the learning objective, the bias from the context `towel` will be alleviated.

More intrigued, $P(\texttt{person}|do(\texttt{toilet}))$ is higher. Indeed, `person` and `toilet` co-occur rarely due to privacy. However, human's *seeing* is fundamentally different from machine's because our instinct is to seek the *causality* behind any association [48] — and here comes the common sense. As opposed to the passive observation $P(Y|X)$: "How likely I see person if I see toilet", we keep asking "Why does seeing toilet eventually cause seeing person?" by using $P(Y|do(X))$. Thanks to intervention, we can increase $P(Y|do(X))$ by "borrowing" non-local context that might not be even in this image, for the example in Figure 2, objects usable by `person` such as `chair` and `handbag` — though less common in the restroom context — will be still fairly "borrowed" and "put" in the image together with the common `sink`. Therefore, the `toilet` region can implicitly learn the common sense: *usable*. We will revisit this example formally in Section 3.1.

So far, we are ready to present our unsupervised region feature learning method: Visual Commonsense R-CNN (**VC R-CNN**), as illustrated in Figure 4, which uses Region-based Convolutional Neural Network (R-CNN) [53] as the visual backbone, and the causal intervention as the training objective. Besides its novel learning fashion, we also design a novel algorithm for the *do*-operation, which is an effective approximation for the imaginative intervention (cf. Section 3.2). The delivery of VC R-CNN is a region fea-

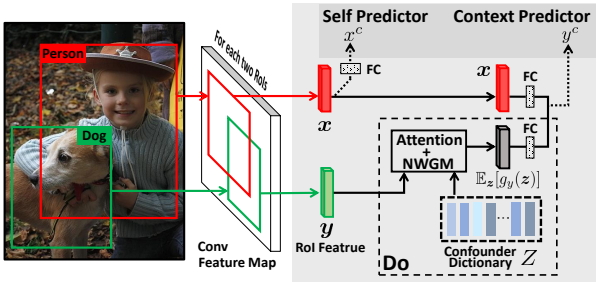


Figure 4. The overview of VC R-CNN. Any R-CNN backbone (*e.g.*, Faster R-CNN [53]) can be used to extract regions of interest (RoI) on the feature map. Each RoI is then fed into two sibling branches: a **Self Predictor** to predict its own class of $\boldsymbol{x}$, and a **Context Predictor** to predict its context labels, *e.g.*, y^c, with our **Do** intervention. The architecture is trained end-to-end with a multi-task loss.

ture extractor for any region proposal, and thus it is fundamental and ready-to-use for many high-level vision tasks such as Image Captioning [66], VQA [3], and VCR [73]. Through extensive experiments in Section 5, VC R-CNN shows significant and consistent improvements over strong baselines — the prevailing methods in each task. Unlike the recent "Bert-like" methods [39, 58] that require huge GPU computing resource for pre-training features and fine-tuning tasks, VC R-CNN is light and non-intrusive. By "light", we mean that it is just as fast and memory-efficient as Faster R-CNN [53]; by "non-intrusive", we mean that re-writing the task network is not needed, all you need is `numpy.concatenate` and then ready to roll.

We apologize humbly to disclaim that VC R-CNN provides a philosophically correct definition of "visual common sense". We only attempt to step towards a **computational** definition in two intuitive folds: 1) common: unsupervised learning from the observed objects, and 2) sense-making: pursuing the causalities hidden in the observed objects. VC R-CNN not only re-thinks the conventional likelihood-based learning in our CV community, but also provides a promising direction — causal inference [48] — via practical experiments.

2. Related Work

Multimodal Feature Learning. With the recent success of pre-training language models (LM) [11, 51, 49] in NLP, several approaches [39, 58, 59, 9] seek weakly-supervised learning from large, unlabelled multi-modal data to encode visual-semantic knowledge. However, all these methods suffer from the reporting bias [64, 35] of language and the great memory cost for downstream fine-tuning. In contrast, our VC R-CNN is unsupervised learning only from images and the learned feature can be simply concatenated to the original representations.

Un-/Self-supervised Visual Feature Learning [13, 61, 41, 27, 74]. They aim to learn visual features through an elaborated proxy task such as denoising autoencoders [6, 65], context & rotation prediction [12, 16] and data augmentation [31]. The context prediction is learned from correlation while image rotation and augmentation can be regarded as applying the random controlled trial [48], which is active and non-observational (physical); by contrast, our VC R-CNN learns from the observational causal inference that is passive and observational (imaginative).

Visual Common Sense. Previous methods mainly fall into two folds: 1) learning from images with commonsense knowledge bases [64, 71, 55, 57, 67, 75] and 2) learning actions from videos [17]. However, the first one limits the common sense to the human-annotated knowledge, while the latter is essentially, again, learning from correlation.

Causality in Vision. There has been a growing amount of efforts in marrying complementary strengths of deep learning and causal reasoning [47, 46] and have been explored in several contexts, including image classification [8, 38], reinforcement learning [44, 10, 5] and adversarial learning [26, 24]. Lately, we are aware of some contemporary works on visual causality such as visual dialog [50] and scene graph generation [60]. Different from their task-specific causal inference, VC R-CNN offers a generic feature extractor.

3. Sense-making by Intervention

We detail the core technical contribution in VC R-CNN: causal intervention and its implementation.

3.1. Causal Intervention

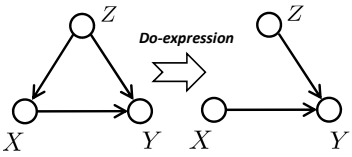


Figure 5. The causal intervention $P(Y|do(X))$. Nodes denote variables and arrows denote the direct causal effects.

As shown in Figure 5 (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:

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

where the confounder Z introduces the observational bias via $P(z|X)$. For example, as recorded in Figure 6, when $P(z{=}\texttt{sink}|X{=}\texttt{toilet})$ is large while $P(z{=}\texttt{chair}|X{=}\texttt{toilet})$ is small, most of the likelihood sum in Eq. (1) will be credited to $P(Y{=}\texttt{person}|X{=}\texttt{toilet},z{=}\texttt{sink})$, other than $P(Y{=}\texttt{person}|X{=}\texttt{toilet},z{=}\texttt{chair})$, so, the prediction from `toilet` to `person` will be eventually focused on `sink` rather than `toilet` itself, *e.g.*, the

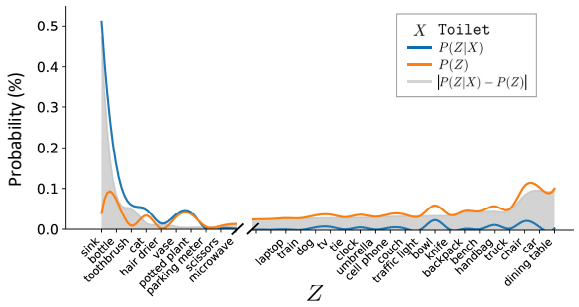


Figure 6. A case study of the differences between $P(z|\texttt{Toilet})$ and $P(z)$ from MS-COCO ground-truth object labels. Only 29 labels of Z are shown to avoid clutter.

learned features of a region `toilet` are merely its surrounding sink-like features.

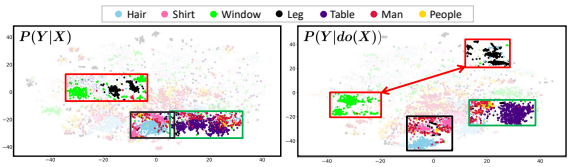
As illustrated in Figure 5 (right), if we intervene X, *e.g.*, $do(X$=`toilet`$)$, the causal link between Z and X is cut-off. By applying the Bayes rule on the new graph, we have:

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

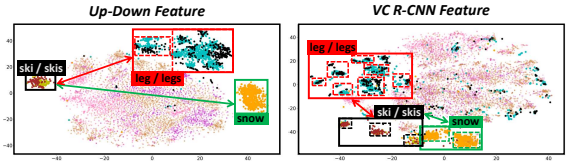
Compared to Eq. (1), z is no longer affected by X, and thus the intervention deliberately forces X to incorporate every z fairly, subject to its prior $P(z)$, into the prediction of Y. Figure 6 shows the gap between the prior $P(z)$ and $P(z|\texttt{toilet})$, $z \in Z$ is the set of MS-COCO labels. We can use this figure to clearly explain the two interesting key results by performing intervention. Please note that $P(Y|X,z)$ remains the same in both Eq. (1) and Eq. (2),

Please recall Figure 3 for the sensible difference between $P(Y|X)$ and $P(Y|do(X))$. First, $P(\texttt{person}|do(\texttt{toilet}))$>$P(\texttt{person}|\texttt{toilet})$ is probably because the number of classes z such that $P(z|\texttt{toilet})$>$P(z)$ is smaller than those such that of $P(z|\texttt{toilet}) < P(z)$, *i.e.*, the left grey area is smaller than the right grey area in Figure 6, making Eq. (1) smaller than Eq. (2). Second, we can see that z making $P(z) < P(z|X)$ is mainly from the common restroom context such as `sink`, `bottle`, and `toothbrush`. Therefore, by using intervention $P(Y|do(X))$ as the feature learning objective, we can adjust between "common" and "sense-making", and thus alleviate the observational bias.

Figure 7(a) visualizes the features extracted from MS-COCO images by using the proposed VC R-CNN. Promisingly, compared to $P(Y|X)$ (left), $P(Y|do(X))$ (right) successfully discovers some sensible common sense. For example, before intervention, `window` and `leg` features in red box are close due to the street view observational bias, *e.g.*, people walking on street with window buildings; after intervention, they are clearly separated. Interestingly, VC R-CNN `leg` features are closer to `head` while `window` features are closer to `wall`. Similarly, the



(a) Object features learned by correlation $P(Y|X)$ and intervention $P(Y|do(X))$ (our VC R-CNN).



(b) Object features of Up-Down features and our VC R-CNN.

Figure 7. The t-SNE visualization [40] of object features trained on MS-COCO with Up-Down [2] provided Faster R-CNN labels. Features out of the label legend are faded out to avoid clutter.

`Hair`/`Shirt` (black box) and `Table`/`Man` features (green box) are all separated after intervention. Furthermore, Figure 7(b) shows the features of `ski`, `snow` and `leg` on same MS-COCO images via Up-Down (left) and our VC R-CNN (right). We can see the `ski` feature of our VC R-CNN is reasonably closer to `leg` and `snow` than Up-Down. Interestingly, VC R-CNN merges into sub-clusters (dashed boxes), implying that the common sense is actually multi-facet and varies from context to context.

3.2. The Proposed Implementation

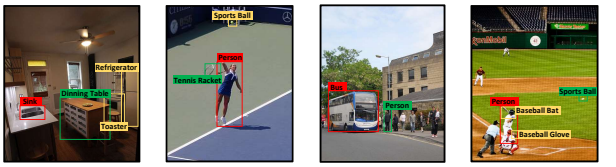
To implement the theoretical and imaginative intervention in Eq. (2), we propose the proxy task of predicting the local context labels of Y's RoI. For the confounder set Z, since we can hardly collect all confounders in real world, we approximate it to a fixed confounder dictionary $\boldsymbol{Z} = [\boldsymbol{z}_1, ..., \boldsymbol{z}_N]$ 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 $\sum_z P\left(y^c|\boldsymbol{x},\boldsymbol{z}\right)P\left(\boldsymbol{z}\right)$. The last layer of the network for label prediction is the Softmax layer: $P\left(y^c|\boldsymbol{x},\boldsymbol{z}\right) = \mathit{Softmax}(f_y(\boldsymbol{x},\boldsymbol{z}))$, 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:

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

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

Normalized Weighted Geometric Mean (NWGM). We apply NWGM [68] to approximate the above expectation.



X: Sink Y: Dinning Table
Z: Toaster 0.75
Refrigerator 0.11
Microwave 0.01

X: Person Y: Tennis Racket
Z: Baseball Bat 0.78
Baseball Glove 0.05
Sports Ball 0.02

X: Bus Y: Person
Z: Stop Sign 0.97
Skateboard 0.002
Fire Hydrant 0.0005

X: Person Y: Sports Ball
Z: Baseball Glove 0.55
Sports Ball 0.32
Baseball Bat 0.09

Figure 8. The visualizations of the top 3 confounders given RoI feature $\boldsymbol{x}$ (red box) and $\boldsymbol{y}$ (green box), while numbers denote the attention weight. We can see that our model can recognize reasonable confounders $\boldsymbol{z}$, *e.g.*, the common context (yellow boxes).

In a nutshell, NWGM[2] effeciently moves the outer expectation into the Softmax as:

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

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:

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

Note that the above approximation is reasonable, because the effect on Y comes from both X and confounder Z (cf. the right Figure 5).

Computing $\mathbb{E}_{\boldsymbol{z}}[g_y(\boldsymbol{z})]$. We encode $g_y(\cdot)$ as the Scaled Dot-Product Attention [62] 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. Figure 8 visualizes the top 3 confounders randed by the soft attention weights. Note that they are the cancer in learning "sense-making" features from $P(Y|X)$.

Neural Causation Coefficient (NCC). Due to the fact that the causality from the confounders as the category averaged features are not yet verified, that is, $\boldsymbol{Z}$ may contain colliders (or v-structure) [47] causing spurious correlations when intervention. To this end, we apply NCC [38] to remove possible colliders from $\boldsymbol{Z}$. Given $\boldsymbol{x}$ and $\boldsymbol{z}$, $NCC(\boldsymbol{x} \to \boldsymbol{z})$ outputs the relative causality intensity from $\boldsymbol{x}$ to $\boldsymbol{z}$. Then we discard the training samples with strong collider causal intensities above a threshold.

4. VC R-CNN

Architecture. Figure 4 illustrates the VC R-CNN architecture. Similar to Faster/Mask R-CNN [53, 20], VC R-CNN takes an entire image as input and then generates region bounding box proposals from a CNN backbone (*e.g.*, ResNet101 [21]). Then, for each ground-truth bounding box, the RoIAlign Layer is utilized to extract the object level representation. 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 (note that we do not have the "background" class). 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:

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

Feature Extractor. We consider VC R-CNN as a visual commonsense feature extractor for any region proposal. Then the extracted features are directly concatenated to the original visual feature utilized in any downstream tasks. It is worth noting that we do NOT recommend early concatenations for some models that contain a self-attention architecture such as AoANet [23]. The reasons are two-fold. First, as the computation of these models are expensive, early concatenation significantly slows down the training. Second, which is more crucial, the self-attention essentially and implicitly applies $P(Y|X)$, which contradicts to causal intervention. We will detail this finding in Section 5.4.

5. Experiments

5.1. Datasets

We used the two following datasets for unsupervised learning VC R-CNN.

MS-COCO Detection [34]. It is a popular benchmark dataset for classification, detection and segmentation in our community. It contains 82,783, 40,504 and 40,775 images for training, validation and testing respectively with 80 annotated classes. Since there are 5K images from downstream image captioning task which can be also found in MS-COCO validation split, we removed those in training. Moreover, recall that our VC R-CNN relies on the context prediction task, thus, we discarded images with only one annotated bounding box.

Open Images [30]. We also used a much larger dataset called Open Images, a huge collection containing 16M

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

Model	Feature	MS-COCO				Open Images			
		B4	M	R	C	B4	M	R	C
Up-Down	Origin [2]	36.3	27.7	56.9	120.1	36.3	27.7	56.9	120.1
	Obj	36.7	27.8	57.5	122.3	36.7	27.8	57.5	122.3
	Only VC	34.5	27.1	56.5	115.2	35.1	27.2	56.6	115.7
	+Det	37.5	28.0	58.3	125.9	37.4	27.9	58.2	125.7
	+Cor	38.1	28.3	58.5	127.5	38.3	28.4	58.8	127.4
	+VC	39.5	29.0	59.0	130.5	39.1	28.8	59.0	130.0
AoANet†	Origin[3] [23]	38.9	28.9	58.8	128.4	38.9	28.9	58.8	128.4
	Obj	38.1	28.4	58.2	126.0	38.1	28.4	58.2	125.9
	Only VC	35.8	27.6	56.8	118.1	35.8	27.9	56.7	118.5
	+Det	38.8	28.8	58.7	128.0	38.7	28.6	58.7	127.7
	+Cor	38.8	28.9	58.7	128.6	38.9	28.8	58.7	128.2
	+VC	**39.5**	**29.3**	**59.3**	**131.6**	39.3	29.1	59.0	131.5
SOTA	AoANet [23]	38.9	29.2	58.2	129.8	38.9	29.2	58.2	129.8

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 in each model. AoANet† indicates the AoANet without the refine encoder. Note that the Origin and Obj share the same results in MS-COCO and Open Images since they does not contain our new trained features.

bounding boxes across 1.9M images, making it the largest object detection dataset. For Open Images, we chose images with more than three annotations from the official training set, results in about 1.07 million images consisting of 500 classes.

5.2. Implementation Details

We trained our VC R-CNN on 4 Nvidia 1080Ti GPUs with a total batch size of 8 images for 180K iterations (each mini-batch has 2 images per GPU). The learning rate was set to 0.0002 which was decreased by 10 at every 120K iteration and ResNet-101 was set to the image feature extraction backbone. We used SGD as the optimizer with weight decay of 0.0001 and momentum of 0.9 following [53]. To construct the confounder dictionary $\boldsymbol{Z}$, we first employed the pre-trained official ResNet-101 model on Faster R-CNN with ground-truth boxes as the input to extract the RoI features for each object. For training on Open Images, we first trained a vanilla Faster R-CNN model. Then $\boldsymbol{Z}$ is built by making average on RoIs of the same class and is fixed during the whole training stage.

5.3. Comparative Designs

To evaluate the effectiveness of our VC R-CNN feature (VC), we present three representative vision-and-language downstream tasks 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 three ablative feature settings: 1) **Obj**: the features based on Faster R-CNN, we adopted the popular used bottom-up feature [2]; 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, *e.g.*, bottom-up 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. Moreover, since some downstream models may have different settings in the original papers, we also quoted their results for clear comparison. For each downstream task, we detail the problem settings, dataset and evaluation metrics as below.

[3]Since we cannot achieve performances reported in original paper using the official code even with the help of author, here we show ours and original results can be referred at the bottom line SOTA of Table 1.

Model	BLEU-4		METEOR		ROUGE-L		CIDEr-D	
Metric	c5	c40	c5	c40	c5	c40	c5	c40
Up-Down [2]	36.9	68.5	27.6	36.7	57.1	72.4	117.9	120.5
SGAE [69]	37.8	68.7	28.1	37	58.2	73.1	122.7	125.5
CNM [70]	37.9	68.4	28.1	36.9	58.3	72.9	123.0	125.3
AoANet [23]	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 [2] in Up-Down and AoANet†.

Model	Feature	CHs	Chi	Model	Feature	CHs	Chi
Up-Down	Obj	12.8	8.1	AoANet†	Obj	12.6	8.0
	+Det	12.0	7.5		+Det	9.5	6.2
	+Cor	11.2	7.1		+Cor	10.4	6.5
	+VC	**10.3**	**6.5**		+VC	**8.8**	**5.5**

Table 3. Hallucination analysis [54] of various models on MS-COCO Karpathy test split to measure object hallucination for image captioning. The lower, the better.

Image Captioning. Image captioning aims to generate textual description of an image. We trained and evaluated on the most popular "Karpathy" split built on MS-COCO dataset, where 5K images for validation, 5K for testing, and the rest for training. The sentences were tokenized and changed to lowercase. Words appearing less than 5 times were removed and each caption was trimmed to a maximum of 16 words. Five standard metrics were applied for evaluating the performances of the testing models: CIDEr-D [63], BLEU [45], METROT [4], ROUGE [33] and SPICE [1].

Visual Question Answering (VQA). The VQA task requires answering natural language questions according to the images. We evaluated the VQA model on VQA2.0 [18]. Compared with VQA1.0 [3], VQA2.0 has more question-image pairs for training (443,757) and validation (214,354), and all the question-answer pairs are balanced. Before training, we performed standard text pre-processing. Questions were trimed to a maximum of 14 words and candidate answer set was restricted to answers appearing more than 8 times. The evaluation metrics consist of three pre-type accuracies (*i.e.*, "Yes/No", "Number" and "Other").

Visual Commonsense Reasoning (VCR). In VCR, given

Model	Feature	MS-COCO				Open Images			
		Y/N	Num	Other	All	Y/N	Num	Other	All
Up-Down	Obj [2]	80.3	42.8	55.8	63.2	80.3	42.8	55.8	63.2
	Only VC	77.8	37.9	51.6	59.8	77.9	38.1	51.1	59.9
	+Det	81.8	44.5	56.8	64.5	81.9	44.7	56.5	64.6
	+Cor	81.5	44.6	57.1	64.7	81.3	44.7	57.0	64.6
	+VC	82.5	46.0	57.6	65.4	82.8	45.7	57.4	65.4
MCAN	Obj [72]	84.8	49.4	58.4	67.1	84.8	**49.4**	58.4	67.1
	Only VC	80.8	40.7	48.9	60.1	81.0	40.8	49.1	60.3
	+Det	84.8	49.2	58.8	67.2	84.9	49.3	58.4	67.2
	+Cor	85.0	49.2	58.9	67.4	85.1	49.1	58.6	67.3
	+VC	**85.2**	**49.4**	**59.1**	**67.7**	85.1	49.1	58.9	67.5
SOTA	MCAN	84.8	**49.4**	58.4	67.1	84.8	**49.4**	58.4	67.1

Table 4. Accuracy (%) of various ablative features on VQA2.0 validation set. For Up-Down and MCAN, since the Obj is same with the original paper and achieve almost equal results , here we just merge the two rows.

Model	test-dev				test-std
	Y/N	Num	Other	All	All
Up-Down [2]	81.82	44.21	56.05	65.32	65.67
BAN [25]	85.46	50.66	60.50	69.66	-
DFAF [14]	86.09	53.32	60.49	70.22	70.34
MCAN [72]	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 5. Single model accuracies (%) on VQA2.0 test-dev and test set, where Up-Down+VC and MCAN+VC are the short for Object-VC R-CNN feature in Up-Down and MCAN.

a challenging question about an image, machines need to present two sub-tasks: answer correctly (Q→A) and provide a rationale justifying its answer (QA→R). The VCR dataset [73] contains over 212K (training), 26K (validation) and 25K (testing) derived from 110K movie scenes. The model was evaluated in terms of 4-choice accuracy and the random guess accuracy on each sub-task is 25%.

5.4. Results and Analysis

Results on Image Captioning. We compared our VC representation with ablative features on two representative approaches: Up-Down [2] and AoANet [23]. For Up-Down model shown in Table 1, 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. However, only utilizing the pure VC feature (*i.e.*, Only VC) would hurt the model performance. The reason can be obvious. Even for human it is insufficient to merely know the common sense that "apple is edible" for specific tasks, we also need visual features containing objects and attributes (*e.g.*, "what color is the apple") which are encoded by previous representations. 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 [23] proposed an "Attention on Attention" module on feature encoder and caption decoder for refining 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 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.

Model	Feature	MS-COCO		Open Images	
		Q → A	QA → R	Q → A	QA → R
R2C	Origin [73]	63.8	67.2	63.8	67.2
	Obj	65.9	68.2	65.9	68.2
	Only VC	64.1	66.7	64.3	66.8
	+Det	66.1	68.5	66.1	68.3
	+Cor	66.5	68.9	66.6	69.1
	+VC	67.4	69.5	67.2	69.9
ViLBERT	Origin [39]	70.3	70.4	70.3	70.4
	Obj	71.1	72.2	71.1	72.2
	Only VC	69.6	70.1	69.7	70.2
	+Det	71.2	72.3	71.1	72.0
	+Cor	71.3	72.4	71.2	72.5
	+VC	**71.5**	72.8	**71.5**	**72.9**
SOTA	ViLBERT [39]	70.3	70.4	70.3	70.4

Table 6. Experimental results on VCR with various visual features. For the ViLBERT we present the original version since the latest code is not available.

Moreover, since the existing metrics fall short to the dataset bias, we also applied a new metric CHAIR [54] to measure the object hallucination (*e.g.*, "hallucinate" objects not in image). The lower is better. As shown in Table 3, we can see that our VC feature performs the best on both standard and CHAIR metrics, thanks to our proposed intervention that can encode the visual commonsense knowledge.

Results on VQA. In Table 4, we applied our VC feature on classical Up-Down [2] and recent state-of-the-art method MCAN [72]. 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 5 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, which only achieves a slight improvement.

Results on VCR. We present two representative methods R2C [73] and ViLBERT [39] in this emerging task on the validation set. Note that as the R2C applies the ResNet backbone for residual feature extraction, here for fair comparison we followed the uniform experimental settings and switched it to the bottom-up features. The re-implementation results are shown as the Obj model in Table 6. From the comparison with ablative visual representations, our +VC feature still shows the best performances similar to the above two tasks.

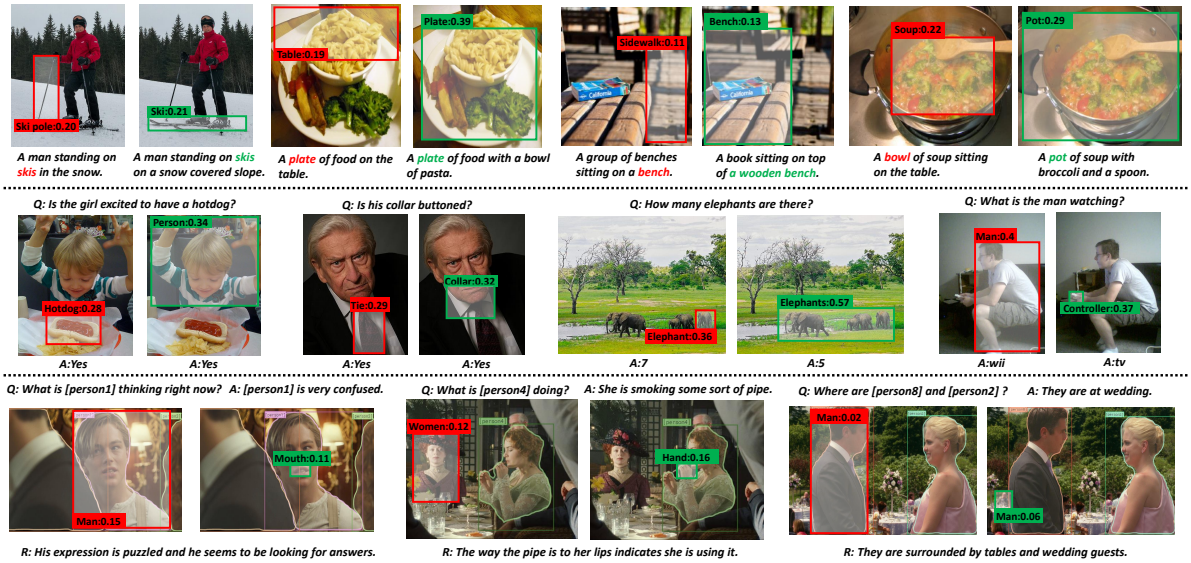


Figure 9. Qualitative examples of utilizing our VC feature (right) compared with using Obj feature (left). Boxes in images denote the attention region labeled with name and attention weight. Three rows represent Image Captioning, VQA and VCR task respectively.

Component	Setting	CIDEr-D	Accuracy
Expectation	$\mathbb{E}_{\boldsymbol{z}}[\boldsymbol{z}]$	128.9	67.2
NCC	w/o NCC	131.5	**67.7**
Dictionary	Random Dictionary	127.5	66.9
	Context Dictionary	*Unstable Training*	
	Fixed Dictionary	**131.6**	**67.7**

Table 7. Ablation studies of our proposed intervention trained on MS-COCO and evaluated with CIDEr-D (captioning) and Accuracy (VQA) on Karpathy testset and VQA2.0 validation set.

Results on Open Images. To evaluate the transfer ability and flexibility of the learned visual commonsense feature, we also performed our proposed VC R-CNN on a large image detection collection. The results can be referred to Table 1&4&6. We can see that the performances are extremely close to the VC feature trained on MS-COCO, indicating the stability of our learned semantically meaningful representation. Moreover, while performing VCR with the dataset of movie clip, which has quite diverse distributions compared to the captioning and VQA built on MS-COCO, our VC R-CNN trained on Open Images achieves the reasonable better results.

5.5. Qualitative Analysis

We visualize several examples with our VC feature and previous Up-Down feature [2] for each task in Figure 9. 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.

5.6. Ablation Study

To evaluate our proposed intervention implementation, we carry out different settings for each module in our VC R-CNN and report results on captioning and VQA in Table 7. $\mathbb{E}_{\boldsymbol{z}}[\boldsymbol{z}]$ denotes utilizing statistical $P(z)$ by counting from the dataset without attention. Random Dictionary denotes initializing the confounder dictionary by randomization rather than the average RoI feature, while the Context Dictionary encodes contexts in each image as a dynamic dictionary set. The default setting is the fixed confounder dictionary with our attention module and NCC, which gives the best results. We can observe that random dictionary and $\mathbb{E}_{\boldsymbol{z}}[\boldsymbol{z}]$ would hurt the performance, which demonstrates the effectiveness of our implementation. Moreover, we can find that NCC refining just brings a little difference to the downstream task performance. The potential reason is that NCC just provides a qualitative prediction and may have deviation when applying on real-world visual feature. We will continue exploring NCC in the future work.

6. 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.

Acknowledgments We would like to thank all reviewers for their constructive comments. This work was partially supported by the NTU-Alibaba JRI and the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 1 grant.

References

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *ECCV*, pages 382–398. Springer, 2016. 6

[2] 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*, pages 6077–6086, 2018. 1, 4, 6, 7, 8

[3] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015. 3, 6

[4] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACLW*, pages 65–72, 2005. 6

[5] Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Rosemary Ke, Sébastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher Pal. A meta-transfer objective for learning to disentangle causal mechanisms. *arXiv preprint arXiv:1901.10912*, 2019. 3

[6] Yoshua Bengio, Eric Laufer, Guillaume Alain, and Jason Yosinski. Deep generative stochastic networks trainable by backprop. In *ICML*, pages 226–234, 2014. 3

[7] Remi Cadene, Corentin Dancette, Hedi Ben-younes, Matthieu Cord, and Devi Parikh. Rubi: Reducing unimodal biases in visual question answering. *arXiv preprint arXiv:1906.10169*, 2019. 1

[8] Krzysztof Chalupka, Pietro Perona, and Frederick Eberhardt. Visual causal feature learning. *arXiv preprint arXiv:1412.2309*, 2014. 3

[9] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Learning universal image-text representations. *arXiv preprint arXiv:1909.11740*, 2019. 3

[10] Ishita Dasgupta, Jane Wang, Silvia Chiappa, Jovana Mitrovic, Pedro Ortega, David Raposo, Edward Hughes, Peter Battaglia, Matthew Botvinick, and Zeb Kurth-Nelson. Causal reasoning from meta-reinforcement learning. *arXiv preprint arXiv:1901.08162*, 2019. 3

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2, 3

[12] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, pages 1422–1430, 2015. 2, 3

[13] Justin Domke, Alap Karapurkar, and Yiannis Aloimonos. Who killed the directed model? In *CVPR*, pages 1–8. IEEE, 2008. 3

[14] Peng Gao, Zhengkai Jiang, Haoxuan You, Pan Lu, Steven CH Hoi, Xiaogang Wang, and Hongsheng Li. Dynamic fusion with intra-and inter-modality attention flow for visual question answering. In *CVPR*, pages 6639–6648, 2019. 7

[15] James J Gibson. The theory of affordances. *Hilldale, USA*, 1(2), 1977. 1

[16] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 3

[17] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *ICCV*, volume 1, page 3, 2017. 3

[18] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*, pages 6904–6913, 2017. 6

[19] Ibrahim Abou Halloun and David Hestenes. Common sense concepts about motion. *American journal of physics*, 53(11):1056–1065, 1985. 1

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 1, 5

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 5

[22] Lisa Anne Hendricks, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach. Women also snowboard: Overcoming bias in captioning models. In *ECCV*, pages 793–811. Springer, 2018. 1

[23] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *ICCV*, 2019. 5, 6, 7

[24] Diviyan Kalainathan, Olivier Goudet, Isabelle Guyon, David Lopez-Paz, and Michèle Sebag. Sam: Structural agnostic model, causal discovery and penalized adversarial learning. *arXiv preprint arXiv:1803.04929*, 2018. 3

[25] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear Attention Networks. In *NIPS*, pages 1571–1581, 2018. 7

[26] Murat Kocaoglu, Christopher Snyder, Alexandros G Dimakis, and Sriram Vishwanath. Causalgan: Learning causal implicit generative models with adversarial training. *arXiv preprint arXiv:1709.02023*, 2017. 3

[27] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In *CVPR*, pages 1920–1929, 2019. 3

[28] Matej Kristan, Jiri Matas, Ales Leonardis, Michael Felsberg, Luka Cehovin, Gustavo Fernandez, Tomas Vojir, Gustav Hager, Georg Nebehay, and Roman Pflugfelder. The visual object tracking vot2015 challenge results. In *ICCVW*, pages 1–23, 2015. 1

[29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 1

[30] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, et al. The open images dataset v4: Unified image classification, object detec-

tion, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982*, 2018. 5

[31] Hankook Lee, Sung Ju Hwang, and Jinwoo Shin. Rethinking data augmentation: Self-supervision and self-distillation. *arXiv preprint arXiv:1910.05872*, 2019. 3

[32] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, pages 4282–4291, 2019. 1

[33] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004. 6

[34] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 2, 5

[35] Xiao Lin and Devi Parikh. Don't just listen, use your imagination: Leveraging visual common sense for non-visual tasks. In *CVPR*, pages 2984–2993, 2015. 2, 3

[36] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37. Springer, 2016. 1

[37] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 1

[38] David Lopez-Paz, Robert Nishihara, Soumith Chintala, Bernhard Scholkopf, and Léon Bottou. Discovering causal signals in images. In *CVPR*, pages 6979–6987, 2017. 3, 5

[39] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265*, 2019. 2, 3, 7

[40] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 4

[41] Tomasz Malisiewicz and Alyosha Efros. Beyond categories: The visual memex model for reasoning about object relationships. In *NIPS*, pages 1222–1230, 2009. 2, 3

[42] Varun Manjunatha, Nirat Saini, and Larry S Davis. Explicit bias discovery in visual question answering models. In *CVPR*, pages 9562–9571, 2019. 1

[43] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013. 2

[44] Suraj Nair, Yuke Zhu, Silvio Savarese, and Li Fei-Fei. Causal induction from visual observations for goal directed tasks. *arXiv preprint arXiv:1910.01751*, 2019. 3

[45] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318. Association for Computational Linguistics, 2002. 6

[46] Judea Pearl. Interpretation and identification of causal mediation. *Psychological methods*, 19(4):459, 2014. 3

[47] Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016. 2, 3, 5

[48] Judea Pearl and Dana Mackenzie. *The book of why: the new science of cause and effect*. Basic Books, 2018. 2, 3

[49] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018. 2, 3

[50] Jiaxin Qi, Yulei Niu, Jianqiang Huang, and Hanwang Zhang. Two causal principles for improving visual dialog. 2020. 3

[51] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018. 3

[52] Sainandan Ramakrishnan, Aishwarya Agrawal, and Stefan Lee. Overcoming language priors in visual question answering with adversarial regularization. In *NIPS*, pages 1541–1551, 2018. 1

[53] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015. 1, 2, 3, 5, 6

[54] Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object hallucination in image captioning. *arXiv preprint arXiv:1809.02156*, 2018. 6, 7

[55] Fereshteh Sadeghi, Santosh K Kumar Divvala, and Ali Farhadi. Viske: Visual knowledge extraction and question answering by visual verification of relation phrases. In *CVPR*, pages 1456–1464, 2015. 3

[56] Barry Smith. The structures of the common-sense world. 1995. 1

[57] Zhou Su, Chen Zhu, Yinpeng Dong, Dongqi Cai, Yurong Chen, and Jianguo Li. Learning visual knowledge memory networks for visual question answering. In *CVPR*, pages 7736–7745, 2018. 3

[58] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. *arXiv preprint arXiv:1904.01766*, 2019. 2, 3

[59] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019. 2, 3

[60] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *CVPR*, 2020. 3

[61] Lucas Theis and Matthias Bethge. Generative image modeling using spatial lstms. In *NIPS*, pages 1927–1935, 2015. 3

[62] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017. 5

[63] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, pages 4566–4575, 2015. 6

[64] Ramakrishna Vedantam, Xiao Lin, Tanmay Batra, C Lawrence Zitnick, and Devi Parikh. Learning common sense through visual abstraction. In *ICCV*, pages 2542–2550, 2015. 2, 3

[65] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, pages 1096–1103. ACM, 2008. 3

[66] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, pages 3156–3164, 2015. 3

[67] Qi Wu, Peng Wang, Chunhua Shen, Anthony Dick, and Anton van den Hengel. Ask me anything: Free-form visual question answering based on knowledge from external sources. In *CVPR*, pages 4622–4630, 2016. 3

[68] 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*, pages 2048–2057, 2015. 4

[69] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *CVPR*, pages 10685–10694, 2019. 6

[70] Xu Yang, Hanwang Zhang, and Jianfei Cai. Learning to collocate neural modules for image captioning. *arXiv preprint arXiv:1904.08608*, 2019. 6

[71] Mark Yatskar, Vicente Ordonez, and Ali Farhadi. Stating the obvious: Extracting visual common sense knowledge. In *NAACL*, pages 193–198, 2016. 3

[72] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *CVPR*, pages 6281–6290, 2019. 7

[73] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning. In *CVPR*, pages 6720–6731, 2019. 3, 7

[74] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *ICCV*, pages 1476–1485, 2019. 3

[75] Yuke Zhu, Alireza Fathi, and Li Fei-Fei. Reasoning about object affordances in a knowledge base representation. In *ECCV*, pages 408–424. Springer, 2014. 3

Supplementary Material

In this Supplementary Material, we will further detail the following aspects omitted in the main paper. The detailed code guide and VC features can be referred to https://github.com/Wangt-CN/VC-R-CNN.

- Section A: the detailed derivation of the intervention in Section 3.1 Causal Intervention of the main paper .
- Section B: The details of our proposed implementation in Section 3.2 of the main paper.
- Section C: The details of the network architecture of our VC R-CNN in Section 4 in the main paper.
- Section D: more quantitative results of VC features concatenated on on different Faster R-CNN based representations.
- Section E: more qualitative visualizations compared our VC features with previous bottom-up representations [1].

A. The Do-Expression

In our main paper, we give the do-expression Eq. (2) comparing with the Bayes rule in an intuitive way for easier understanding. In this section, we further formally explain and prove the intervention (do calculus) in causal theory which is applied in our VC R-CNN.

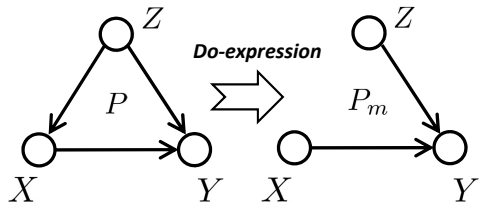


Figure 1. The do expression $P(Y|do(X))$ with a graph surgery. Nodes denote variables and arrows mean the direct causal effects.

As written in our main paper, in our visual world there may exists many "background factors" $z \in Z$, no matter known or unknown, that affect (or cause) either X or/and Y, leading to spurious correlations by only learning from the likelihood $P(Y|X)$. To avoid the confounder as shown in Figure 1, the causal intervention (do calculus) is achieved by cutting off the effect from Z to X in the form of a graph surgery. Here for clear clarification, we use P and P_m to distinguish the probabilities in the causal graph before and after surgery, respectively. Therefore, due to the definition of the Do-expression we can have:

$$P(Y|do(X)) = P_m(Y|X). \quad \textit{(Definition)} \tag{1}$$

Then the key to compute the causal effect lies in the observation P_m, the manipulated probability, shares two essential properties with P (*i.e.*, the original probability function that prevails in the preintervention model). First, the marginal probability $P(Z = z)$ is invariant under the intervention, because the process determining Z is not affected by removing the arrow from Z to X, *i.e.*, $P(z) = P_m(z)$. Second, the conditional probability $P(Y|X, z)$ is invariant, because the process by which Y responds to X and Z remains the same, regardless of whether X changes spontaneously or by deliberate manipulation:

$$P_m(Y|X, z) = P(Y|X, z). \quad \textit{(Invariance)} \tag{2}$$

Moreover, we can also use the fact that Z and X are independent under the intervention distribution. This tell us that $P_m(z|X) = P_m(z)$. Putting these considerations together, we have:

$$\begin{aligned} P(Y|do\,(X)) &= P_m\,(Y|X) \\ &= \sum\nolimits_z P_m\,(Y|X, z)\,P_m\,(z|X) \quad \textit{(Bayes Rule)} \\ &= \sum\nolimits_z P_m\,(Y|X, z)\,P_m\,(z) \quad \textit{(Independency)} \\ &= \sum\nolimits_z P\,(Y|X, z)\,P\,(z)\,, \end{aligned} \tag{3}$$

where $P\,(Y|X, z)$ denotes the conditional probability given X and confounder z and $P(z)$ is a prior probability of each object class.

The Eq. (3) is called the adjustment formula, which computed the association between X and Y for each value z of Z, then averages over all values. This procedure is referred to as "adjusting for Z" or "controlling for Z". Then with this final expression, we can measure the casual effects between X to Y directly from the data, since it consists only of conditional probabilities.

Moreover, in the main paper to show the difference between Bayes Rule and Intervention clearly, we propose an example about `person` and `toilet` by comparing $P(Z)$ and $P(Z|\texttt{toilet})$ on partial labels. Here in the Supplementary Material we present the integrated figure for whole 80 MS-COCO labels on both $P(Z)$, $P(Z|X)$ and $P(Y|X, Z)P(Z)$,$P(Y|X, Z)P(Z|X)$ in Figure 2 & 3. From Figure 2 we can see that the do intervention achieves "borrow" and "put" by applying $P(Z)$ to replace $P(Z|X)$, which can be also regarded as a kind of method to alleviate the previous long tail distribution (blue line).

B. Our Proposed Implementation

B.1. Normalized Weighted Geometric Mean.

In our main paper we just give the application of Normalized Weighted Geometric Mean (NWGM) due to the limited space, here we present the detailed derivation and reader can

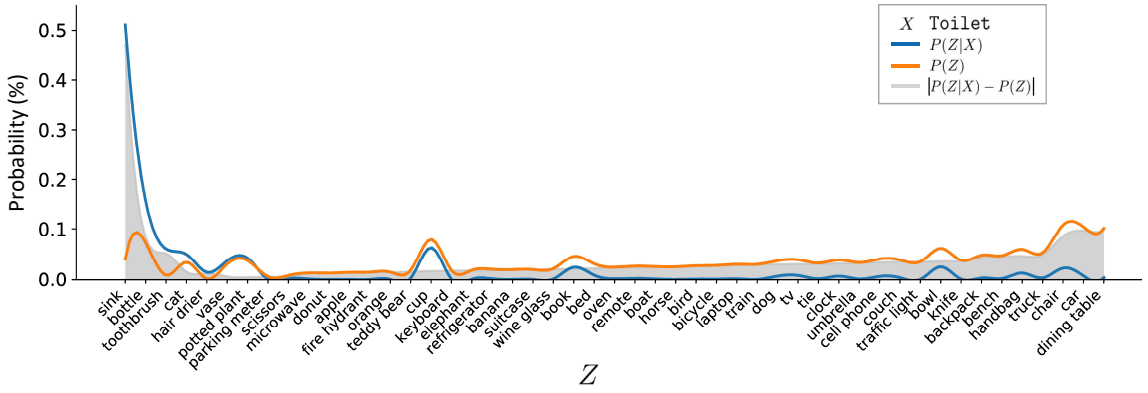


Figure 2. The case study of the differences between $P(z|\texttt{Toilet}$ and $P(z)$ from whole MS-COCO ground-truth object labels. Note that confounders that never appeared with X (*i.e.*, `Toilet`) is not contained.

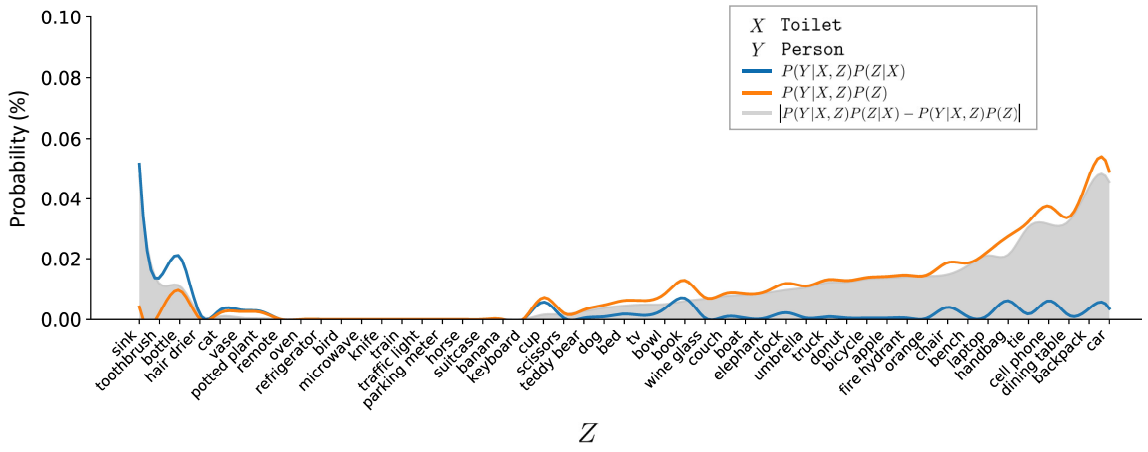


Figure 3. The case study of the differences between $P(\texttt{Person}|\texttt{Toilet}, z)P(z|\texttt{Toilet})$ and $P(\texttt{Person}|\texttt{Toilet}, z)P(z)$ from whole MS-COCO ground-truth object labels. Note that confounders that never appeared with X (*i.e.*, `Toilet`) is not contained.

also refer to the [8]. Recall that in the main paper we have defined the RoI feature $\boldsymbol{x}$ as the X, one of its context class label y^c as Y. For the confounder set Z, we denote it as a global 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 $\boldsymbol{x}$.

Here we first introduce the normalized weighted geometric mean in our softmax class label prediction:

$$\begin{aligned} \text{NWGM}[f_y(\boldsymbol{x}, \boldsymbol{z})] &= \frac{\prod_{\boldsymbol{z}} \exp(f_y(\boldsymbol{x}, \boldsymbol{z}))^{p(\boldsymbol{z})}}{\sum_j \prod_{\boldsymbol{z}} \exp(f_y(\boldsymbol{x}, \boldsymbol{z}))^{p(\boldsymbol{z})}} \\ &= \frac{\exp(\mathbb{E}_{\boldsymbol{z}}[f_y(\boldsymbol{x}, \boldsymbol{z})])}{\sum_j \exp(\mathbb{E}_{\boldsymbol{z}}[f_y(\boldsymbol{x}, \boldsymbol{z})])} \\ &= \mathit{Softmax}(\mathbb{E}_{\boldsymbol{z}}[f_y(\boldsymbol{x}, \boldsymbol{z})]), \end{aligned} \tag{4}$$

where $f_y(\cdot)$ calculates the logits for N categories. Note that the subscript y denotes that $f(\cdot)$ is parameterized by feature $\boldsymbol{y}$, motivated by the heuristics that the context prediction task for RoI Y is characterized by its visual feature. We can see that the most ingenious operation in Eq. (4) is to change the production $\prod$ to the sum $\sum$ by putting it into the exp. Moreover, from the results in [8, 2, 7], we know $\text{NWGM}[f_y(\boldsymbol{x}, \boldsymbol{z})] \approx \mathbb{E}_{\boldsymbol{z}}[\mathit{Softmax}(f_y(\boldsymbol{x}, \boldsymbol{z}))]$ under the softmax activation. Therefore Eq. (3) in the main paper can be further derived as:

$$P\left(Y|do(X)\right) \approx \mathit{Softmax}(\mathbb{E}_{\boldsymbol{z}}[f_y(\boldsymbol{x}, \boldsymbol{z})]). \tag{5}$$

Furthermore, 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 linear projection of the expectation of one variable equals to the linear projection of that and we can put $\mathbb{E}$ into the linear projection as $\mathit{Softmax}(\boldsymbol{W}_1\mathbb{E}_{\boldsymbol{z}}[\boldsymbol{x}] + \boldsymbol{W}_2 \cdot \mathbb{E}_{\boldsymbol{z}}[g_y(\boldsymbol{z})]))$. Since the RoI representation $\boldsymbol{x}$ remains the same, we can discard the $\mathbb{E}$ over $\boldsymbol{x}$, *i.e.*, $\mathit{Softmax}(\boldsymbol{W}_1\boldsymbol{x} + \boldsymbol{W}_2 \cdot \mathbb{E}_{\boldsymbol{z}}[g_y(\boldsymbol{z})]))$. That means the expectation of the outputs over all possible confounder $\boldsymbol{z}$ can be simply computed by feedforward propagation with

Index	Input	Operation	Output	Trainable Parameters
(1)	-	RoI feature	$\boldsymbol{x}$ (1024×1)	-
(2)	-	RoI feature	$\boldsymbol{y}$ (1024×1)	-
(3)	(2), $\boldsymbol{Z}$	Scale Dot-Product Attention	$\mathbb{E}_{\boldsymbol{z}}[g_y(\boldsymbol{z})]$ (1024×1)	$\boldsymbol{W}_3$ (512×1024) $\boldsymbol{W}_4$ (512×1024)
(4)	(1),(3)	Linear Addition Model	$\mathbb{E}_{\boldsymbol{z}}[f_y(\boldsymbol{x}, \boldsymbol{z})]$ (80×1)	$\boldsymbol{W}_1$ (80×1024) $\boldsymbol{W}_2$ (80×1024)
(5)	(1)	Feature Embedding	$\boldsymbol{W}\boldsymbol{x}$ (80×1)	$\boldsymbol{W}(80 \times 1024)$
(6)	(5)	Self Predictor	*Softmax*	-
(7)	(4)	Context Predictor	*Softmax*	-

Table 1. The detailed network architecture of our VC R-CNN.

the expectation vector $\mathbb{E}_{\boldsymbol{z}}[g_y(\boldsymbol{z})]$ as the input.

B.2. Neural Causation Coefficient (NCC)

Here we give a more detailed information about the usage of NCC and collider of our proposed implementations in the main paper. In our visual world sometimes there are no confounders in the structure like $X \rightarrow Z \leftarrow Y$ what we call "collider", as shown in Figure 4. Felix Elwert and Chris Winship [4] have illustrated this junction using three features of Hollywood actors: Talent (X), Celebrity (Z), and Beauty (Y). Here we are asserting that both talent and beauty contribute to an actor's success, but beauty and talent are completely unrelated to one another in the general population.

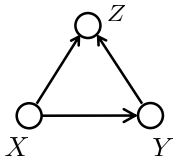


Figure 4. The causal graph structure of the "collider". Nodes denote variables, arrows denote the direct causal effects.

In this structure making the intervention on variable Z (*i.e.*, condition on Z) would create a spurious dependence between X and Y. The reason is that if X and Y are independent to begin with, conditioning on Z will make them dependent. For example, if we look only at famous actors (in other words, we observe the variable Celebrity = 1), we will see a negative correlation between talent and beauty: finding out that a celebrity is unattractive increases our belief that he or she is talented. This negative correlation is sometimes called collider bias or the "explain-away" effect. Therefore we cannot make the intervention as what we do before in the collider structure. For simplicity we would make a preliminary examination before training to eliminate the effect of collider in the whole dataset. We apply the neural causation inference model (NCC) [6] to detect the strong causal effect from $X \rightarrow Z$ and $Y \rightarrow Z$ with the RoI feature directly.

NCC has partly proven [6] to be efficient for transferring to real-world, visual cause-effect observational samples with just training on artificially constructed synthetic observational samples. Specifically, the n synthetic observational samples $S_i = \{(x_{ij}, y_{ij})\}_{j=1}^{m_i}$ are drawn from an heteroscedastic additive noise model $y_{ij} = f_i(x_{ij}) + v_{ij}e_{ij}$ for all $j = 1, ..., m_i$, The cause terms x_{ij} are drawn from a mixture of k_i Gaussians distributions. We construct each Gaussian by sampling its mean from Gaussian(0, r_i) , its standard deviation from Gaussian(0, s_i) followed by an absolute value, and its unnormalized mixture weight from Gaussian (0, 1) followed by an absolute value. NCC samples k_i from RandomInteger[1,5] and r_i ,s_i from Uniform[0,5]. NCC normalizes the mixture weights to sum to one and ${x_{ij}}_{j=1}^{m_i}$ to zero mean and unit variance. The noise term v_{ij} and e_{ij} are also sampled from Gaussian distribution and mechanism f_i is a cubic hermite spline which can be referred to [6]. Finally NCC is trained with two embedding layers and two classification layers followed by the softmax in a ternary classification task (causal, anticausal and no causation). Then while testing the model can be used to evaluate on the RoI feature vectors directly. The output $NCC(\boldsymbol{x} \rightarrow \boldsymbol{y})$ ranges from $(0, 1)$ denotes the relative causality intensity from $\boldsymbol{x}$ inferring $\boldsymbol{y}$.

However since the NCC model just can provide a qualitative prediction and may have huge deviation when applying on real-world feature which may affects the training procedure of our VC R-CNN, in our experiment we just discard few training samples with very strong collider causal structure (*i.e.*, $X \rightarrow Z \leftarrow Y$) by setting a threshold (we set 0.001 in our experiment). Moreover, we use the object-level RoI features extracted by the pretrained Faster R-CNN to pre-calculate the NCC score, which may also lead to a deviation since the pretrained RoI representations may not fully present the objects. From the Table 7 in the main paper we can also observe that NCC refining just brings a little difference to the downstream task performance. The potential reason is that our VC R-CNN can automatically learn the reasonable confounder attention during the large dataset training. We will continue exploring the usage of NCC and other causal discovery method in our future work.

Model	Feature	Cross-Entropy Loss						CIDEr Optimization					
		B@1	B@4	M	R	S	C	B@1	B@4	M	R	S	C
Up-Down	Obj	74.5	33.2	25.9	54.7	18.9	104.7	77.1	32.6	25.2	55.2	18.3	110.6
	Obj+Det	75.4	34.4	26	55.8	19.9	108.9	77.9	33.9	25.4	56.1	19.8	114.7
	Obj+Cor	75.6	34.5	26.1	55.2	19.6	108.7	78.0	34.1	25.6	56.0	19.9	115.2
	Obj+VC	76.3	35.3	26.3	56.3	20.2	111.6	79.1	35.7	25.9	57.0	20.5	119.7
AoANet	Obj	74.6	34.1	25.9	55.4	19.7	108.1	78.1	35.4	25.6	56.7	20.7	118.4
	Obj+Det	75.1	33.9	26.1	55.7	19.8	109.7	78.3	36.2	27.1	56.9	20.9	120.2
	Obj+Cor	75.5	34.3	26.2	55.9	20.1	110.8	78.7	36.8	27.5	57.2	21.1	121.1
	Obj+VC	76.0	35.0	26.4	56.1	20.5	112.2	79.1	37.2	29.0	57.6	21.5	123.5

Table 2. The image captioning performances of two models with ablative features (based on vanilla Faster R-CNN feature) on Karpathy split.

C. Network Architecture

Here we introduce the detailed network architectures of all the components of our VC R-CNN in Table 1. Given an image and the feature extraction backbone, any two RoI feature vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ were extracted as in Table 1 (1)(2). Then as the Section 3.2 The Proposed Implementation, we adopted the Scale Dot-Product Attention to refine confounders from the confounder dictionary Z as in Table 1 (3). A linear addition model $f_y(\boldsymbol{x}, \boldsymbol{z})$ was proposed to combine the effect on Y from both X and confounder Z. Finally we made the do calculus by Self Predictor and Context Predictor in Table 1 (6)(7).

D. More Quantitative Results

In the experiment of our main paper, we adopted the bottom-up feature [1] as our base feature. The bottom-up feature pretrained Faster R-CNN on ImageNet [3] and Visual Genome [5] to propose salient object level features with attribute rather than the uniform gird of equally-sized image regions, enable attention to be calculated at the level of semantically meaningful regions and bring a huge improvement in image-and-language tasks.

Here we also concatenated our VC feature onto the vanilla image region representations based on pretrained Faster R-CNN model with ResNet-101 on MS-COCO dataset. Note that for better comparison we utilized the bounding box coordinates of the bottom-up feature to control the number and location of the boxes and then applied new feature in the Image Captioning task. Results are shown in Table 2. We can also observe that concatenating with our VC feature can lead to a huge performance improvement, which demonstrates the stability of our VC feature and effectiveness of the proposed intervention.

E. More Qualitative Results

E.1. Failure Case

Failure in VC R-CNN. As shown in Figure 5, we can see that sometimes our VC R-CNN cannot make quite reasonable refinement for confounder dictionary via the Scaled Dot-Product Attention while predicting Y given X and Z, especially when there is no obvious relation between X and Y. For example while making the intervention between `dog` and `vase`, `chair` and `fork`, the model attends to the `giraffe` and `skateboard` respectively. To tackle this limitation, the better schedule of confounder exploring, for example choosing approapiate context objects as the confounder dictionary, will be tried in our future work.

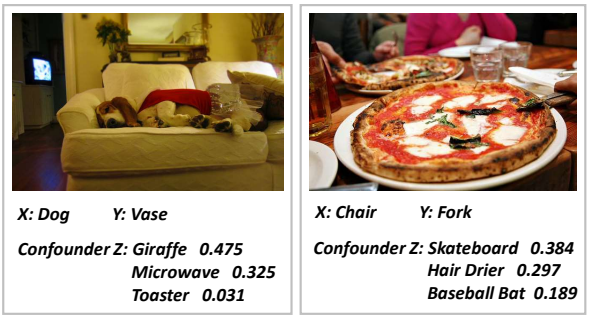


Figure 5. The examples of the failure case about confounder finding in VC R-CNN.

Failure in Downstream Tasks. Though we designed the intervention (do-expression) in unsupervised representation learning to prevent the cognition error and help machine learn the common sense, some attention errors still exist in downstream tasks. Here we present two examples in Figure 6. We can observe that in the VQA example (left), the model provides a reasonable but incorrect answer, while in image captioning the generated description does not cover every instance. The possible reason lies in two folds. First, the current detection technique is still limited, for example the Faster R-CNN cannot recognize the `kangaroo` on the stop sign. Second, we know that our VC R-CNN can find the probable and reasonable confounders from the confounder dictionary according to the given image. However, it may still fail to exploit the exact confounder (*e.g.*, `motorcycle` in VQA and `lamp`, `chair` in Image Captioning) to fully eliminate the correlation bias.

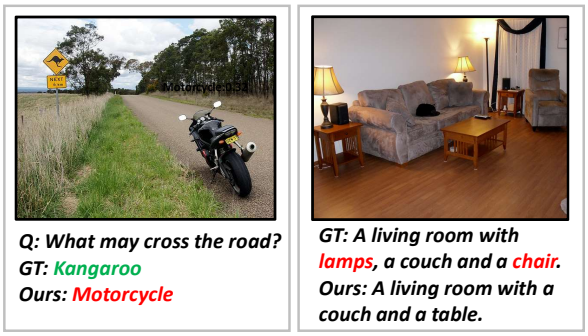


Figure 6. The examples of the failure case in downstream tasks.

E.2. Image Captioning

Figure 7 & 8 exhibit visualizations of utilizing our VC feature (right) compared with using Faster R-CNN feature (*i.e.*, bottom-up feature, left) with the classical Up-Down model in image captioning task. The boxes represent the attended regions when generating words with the same color. From the illustration we can observe that with our VC feature, model can generate more fruitful descriptions with more accurate attention. For example, in Figure 8 bottom with our VC feature, model focuses on birds and gives the accurate and fruitful descriptions: "two birds perched" rather than "a bird sitting" generated by the baseline model. Furthermore, we can also see that our VC feature can help to overcome the language bias efficiently. Other than giving the common collections, the model can generate reasonable captions according to the image content. For example in the middle of Figure 8, "cat" appearing with "bed" ("cat+bed"/"cat"=6.7%) is quite more often than "cat" with "blanket" ("cat+blanket"/"cat"=1.4%) in the training text, leading to a "hallucination" to generate "bed" without seeing the bed.

E.3. VQA

We presented the comparison of Faster R-CNN feature (left) and our VC feature (right) in VQA in Figure 9 & 10 based on the Up-Down model. We can see that in VQA task the most serious problem is the incorrect attention even with the correct answer, which means the model actually NOT understand the question and make inference combining the vision and language. As we described in Introduction of the main paper, the dataset co-occurring bias may lead to the incorrect attention. For example in the middle of Figure 9 the model attend to the horse rather than human since horse and person co-occur too many times. Thanks to our VC feature, the attention becomes better and more accurate with alleviating the correlation bias by our proposed intervention.

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*, pages 6077–6086, 2018. 1, 4

[2] Pierre Baldi and Peter Sadowski. The dropout learning algorithm. *Artificial intelligence*, 210:78–122, 2014. 2

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009. 4

[4] Felix Elwert and Christopher Winship. Endogenous selection bias: The problem of conditioning on a collider variable. *Annual review of sociology*, 40:31–53, 2014. 3

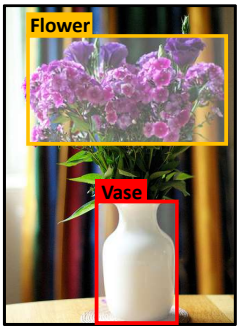
[5] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123(1):32–73, 2017. 4

[6] David Lopez-Paz, Robert Nishihara, Soumith Chintala, Bernhard Scholkopf, and Léon Bottou. Discovering causal signals in images. In *CVPR*, pages 6979–6987, 2017. 3

[7] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958, 2014. 2

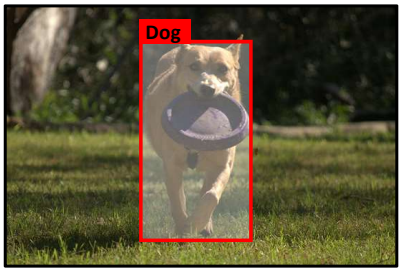
[8] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, pages 3156–3164, 2015. 2

A vase of flowers sitting on top of a table.



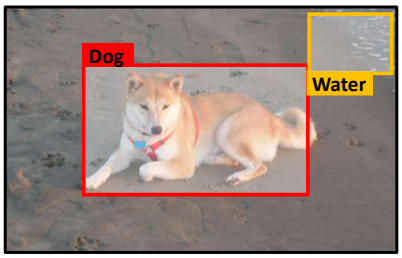
A white vase filled with purple flowers on top of a table.

A dog holding a frisbee in his mouth.



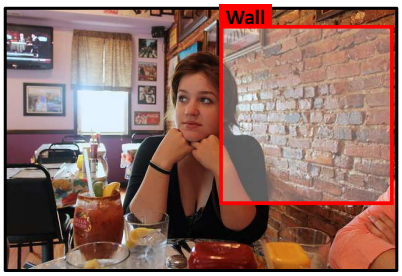
A dog is running with a frisbee in his mouth.

A couple of dog lying on the beach.



A dog lying on the beach next to the water.

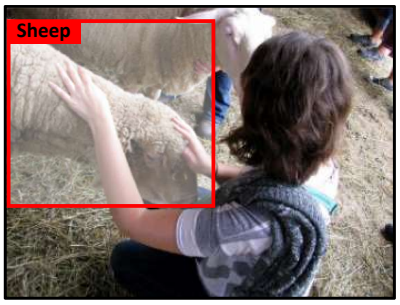
A woman sitting at a table with a table.



A woman sitting at a table in a restaurant.

Figure 7. Qualitative visualizations in Image Captioning with utilizing Faster R-CNN feature (left) and our VC feature (right). Boxes in image represent the attention region when generating words with the same color.

A girl standing next to a sheep.



A women petting a sheep in a field.

A black and white cat sitting on a bed.

A black and white cat laying on a green blanket.

A plane is flying in the sky.

An airplane is flying in the sky over a tree.

A bird sitting on top of a tree.

Two birds perched on top of a tree.

Figure 8. Qualitative visualizations in Image Captioning with utilizing Faster R-CNN feature (left) and our VC feature (right). Boxes in image represent the attention region when generating words with the same color.

Q: Is the man wearing a scarf?
A: Yes

Q: Is the man wearing a scarf?
A: Yes

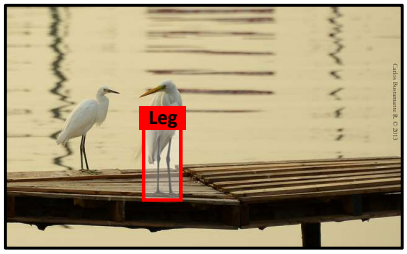
Q: How many wheels does the vehicle behind the man have?
A: 2

Q: How many wheels does the vehicle behind the man have?
A: 2

Q: Is the rider a child or an adult?
A: Adult

Q: Is the rider a child or an adult?
A: Adult

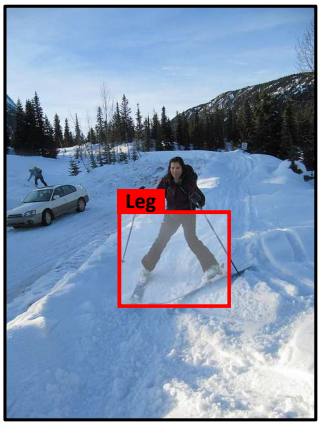
Q: Are the birds legs touching the water?
A: Yes



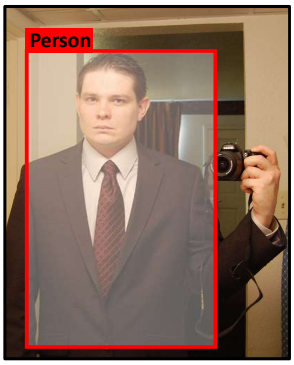
Q: Are the birds legs touching the water?
A: No

Figure 9. The qualitative results of Visual Question Answering by using the Faster R-CNN feature (left) and concatenated with our VC feature (right). Boxes denote the attended region when answering.

Q: Is this woman legs stuck?
A: No



Q: Is this woman legs stuck?
A: No



Q: Is there a camera?
A: Yes.

Q: Is there a camera?
A: Yes

Q: What is in the sink?
A: nothing

Q: What is in the sink?
A: nothing

Figure 10. The qualitative results of Visual Question Answering by using the Faster R-CNN feature (left) and concatenated with our VC feature (right). Boxes denote the attended region when answering.