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.

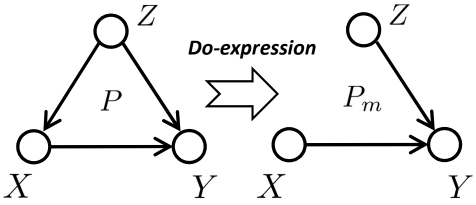


Figure 1. The do expression $P(Y|do(X))$ with a graph surgery. Nodes denote variables and arrows mean the direct causal effects.

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

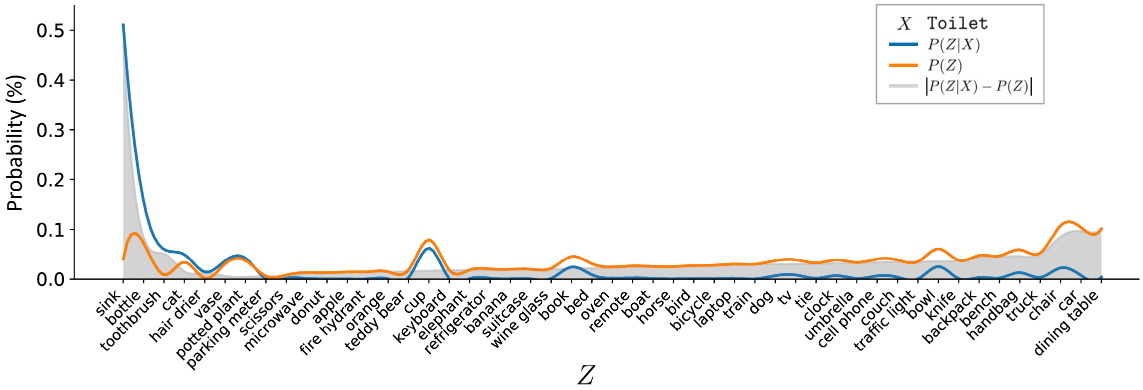


Figure 2. The case study of the differences between $P(z|\texttt{Toilet}$ and $P(z)$ from whole MS-COCO ground-truth object labels. Note that confounders that never appeared with X (*i.e.*, `Toilet`) is not contained.

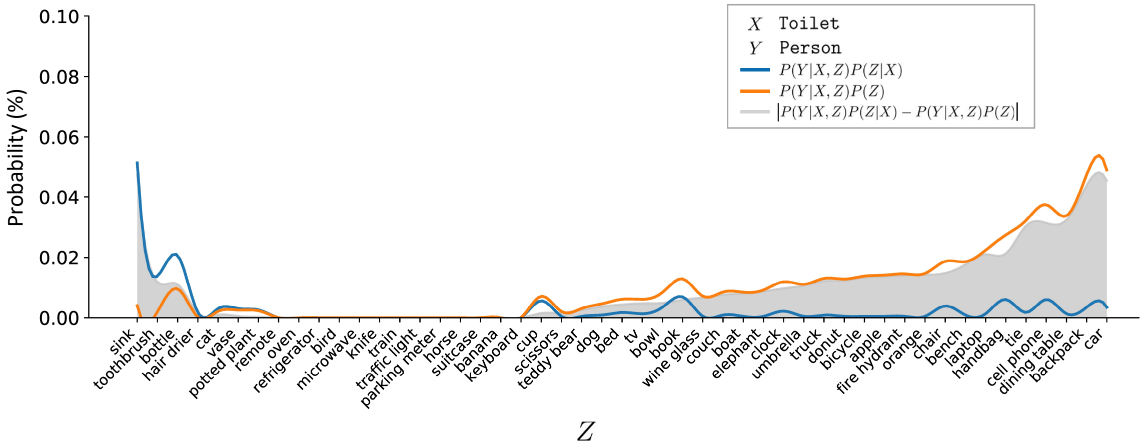


Figure 3. The case study of the differences between $P(\texttt{Person}|\texttt{Toilet}, z)P(z|\texttt{Toilet})$ and $P(\texttt{Person}|\texttt{Toilet}, z)P(z)$ from whole MS-COCO ground-truth object labels. Note that confounders that never appeared with X (*i.e.*, `Toilet`) is not contained.

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.

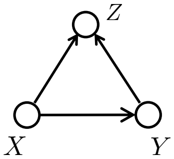


Figure 4. The causal graph structure of the "collider". Nodes denote variables, arrows denote the direct causal effects.

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 $y_{ij} = f_i\left(x_{ij}\right) + v_{ij}e_{ij}$ 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.

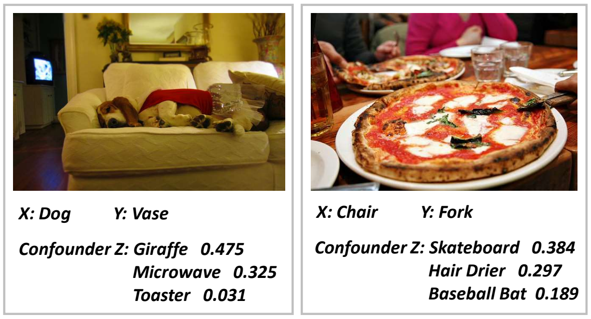


Figure 5. The examples of the failure case about confounder finding in VC R-CNN.

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.

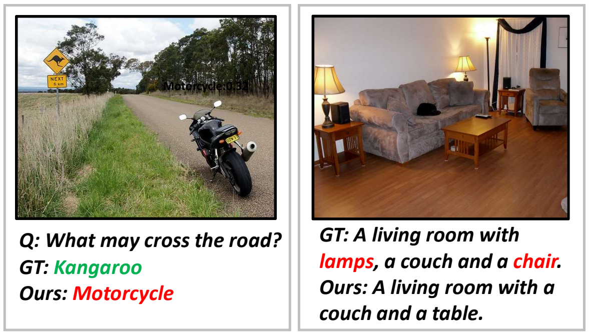

Q: What may cross the road?
GT: Kangaroo
Ours: Motorcycle
GT: A living room with lamps, a couch and a chair.
Ours: A living room with a couch and a table.

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

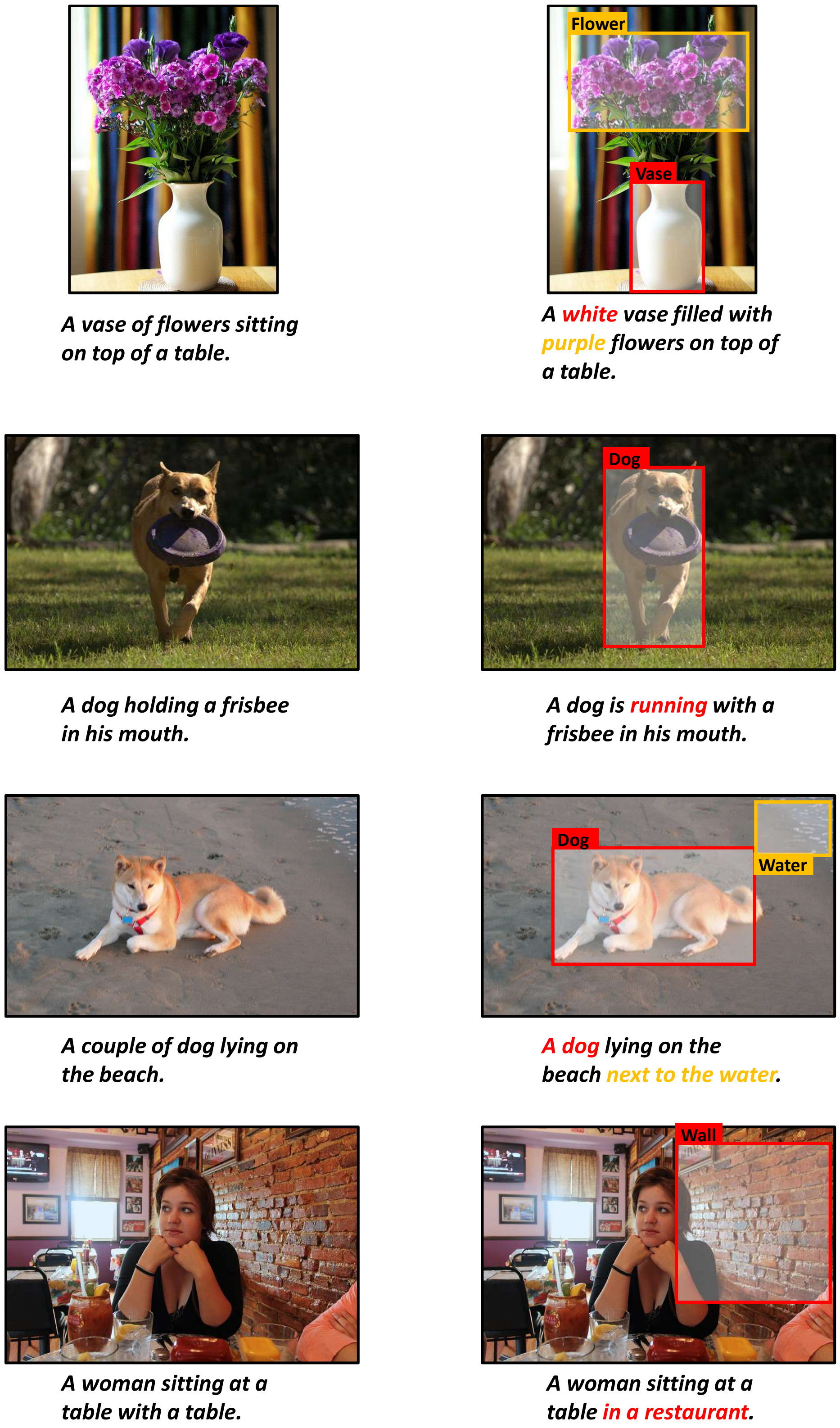
[4] Felix Elwert and Christopher Winship. Endogenous selection bias: The problem of conditioning on a collider variable. *Annual review of sociology*, 40:31–53, 2014. 3

[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.

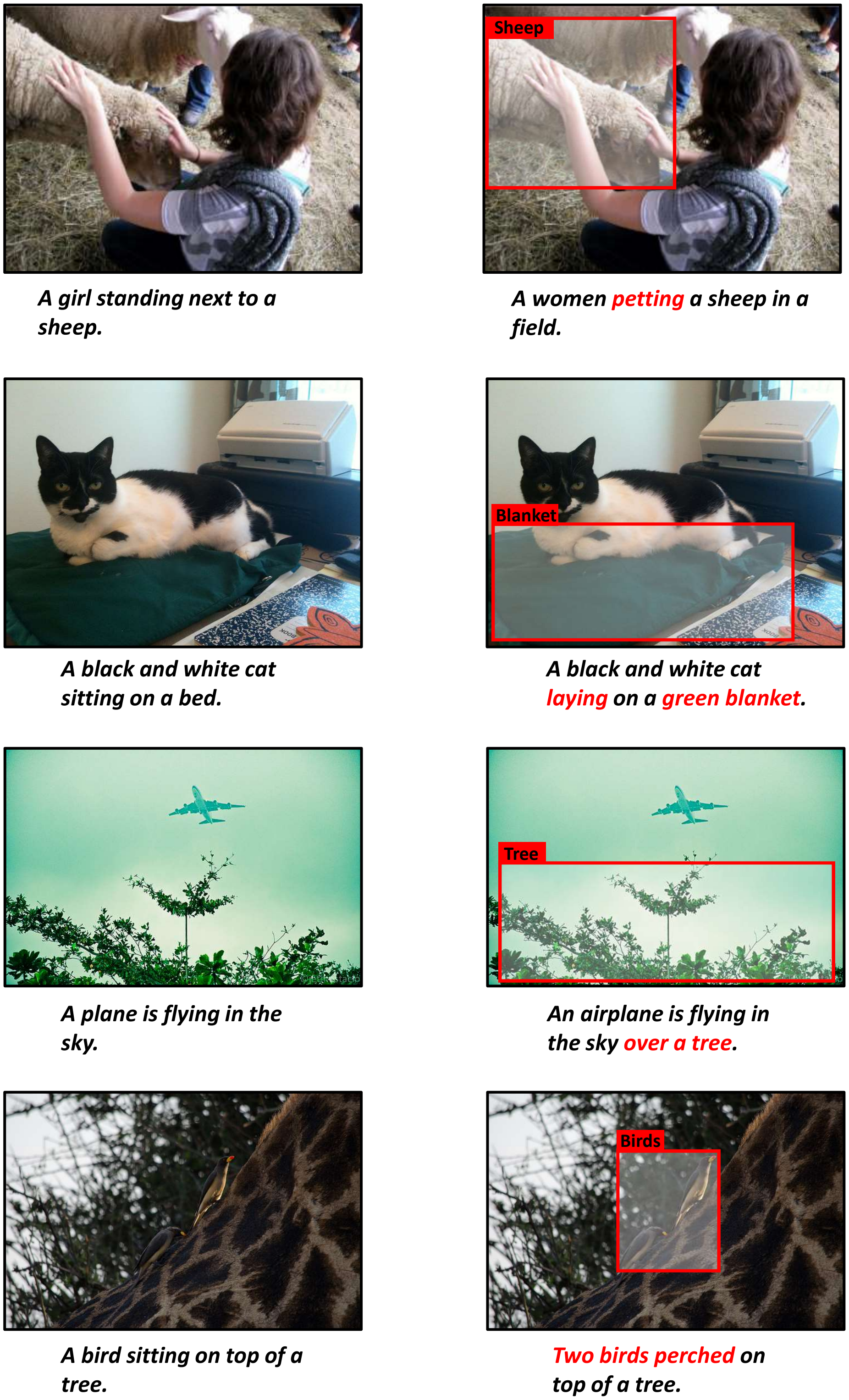
A couple of dog lying on the beach.

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.

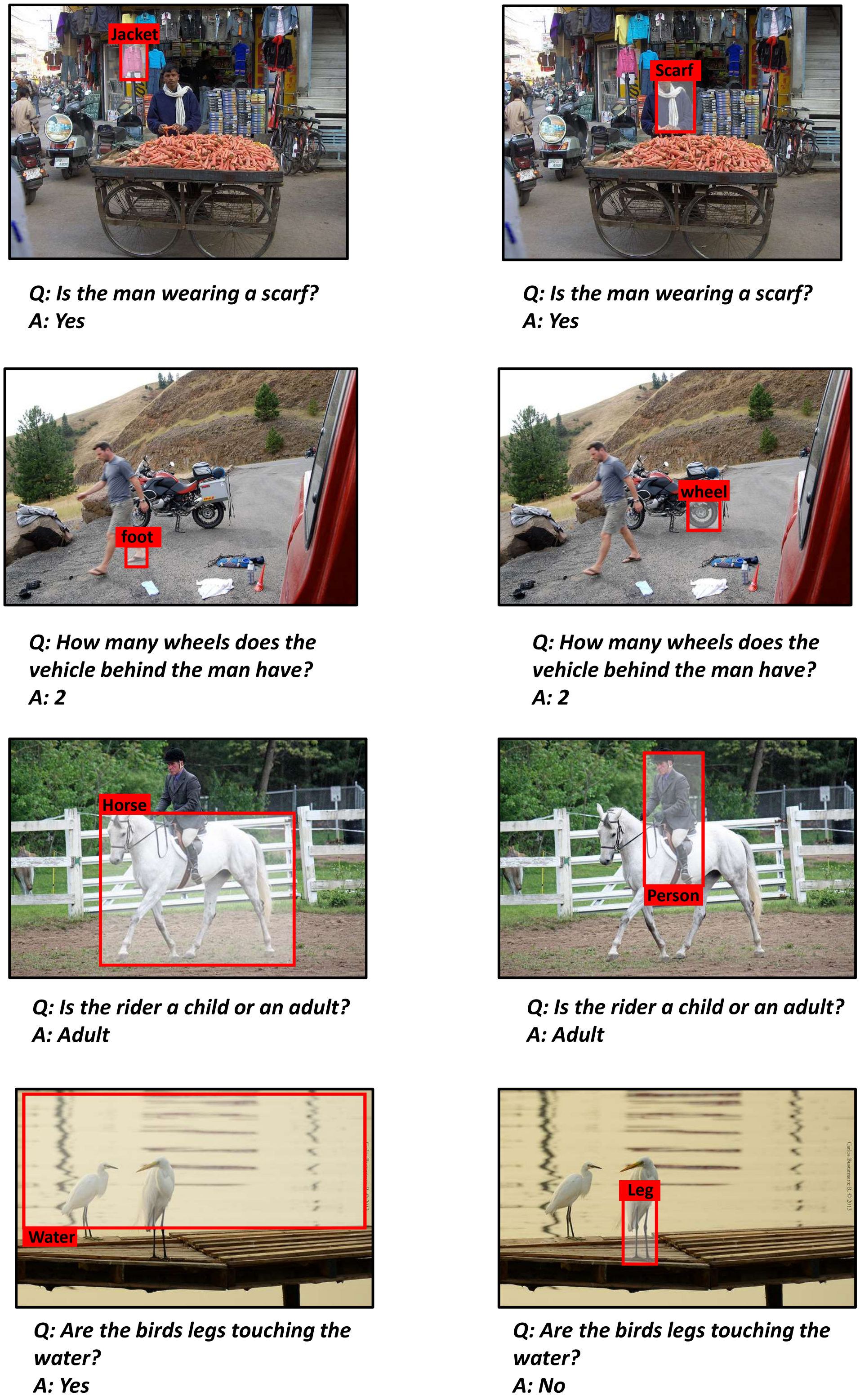
A plane is flying in the sky.

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

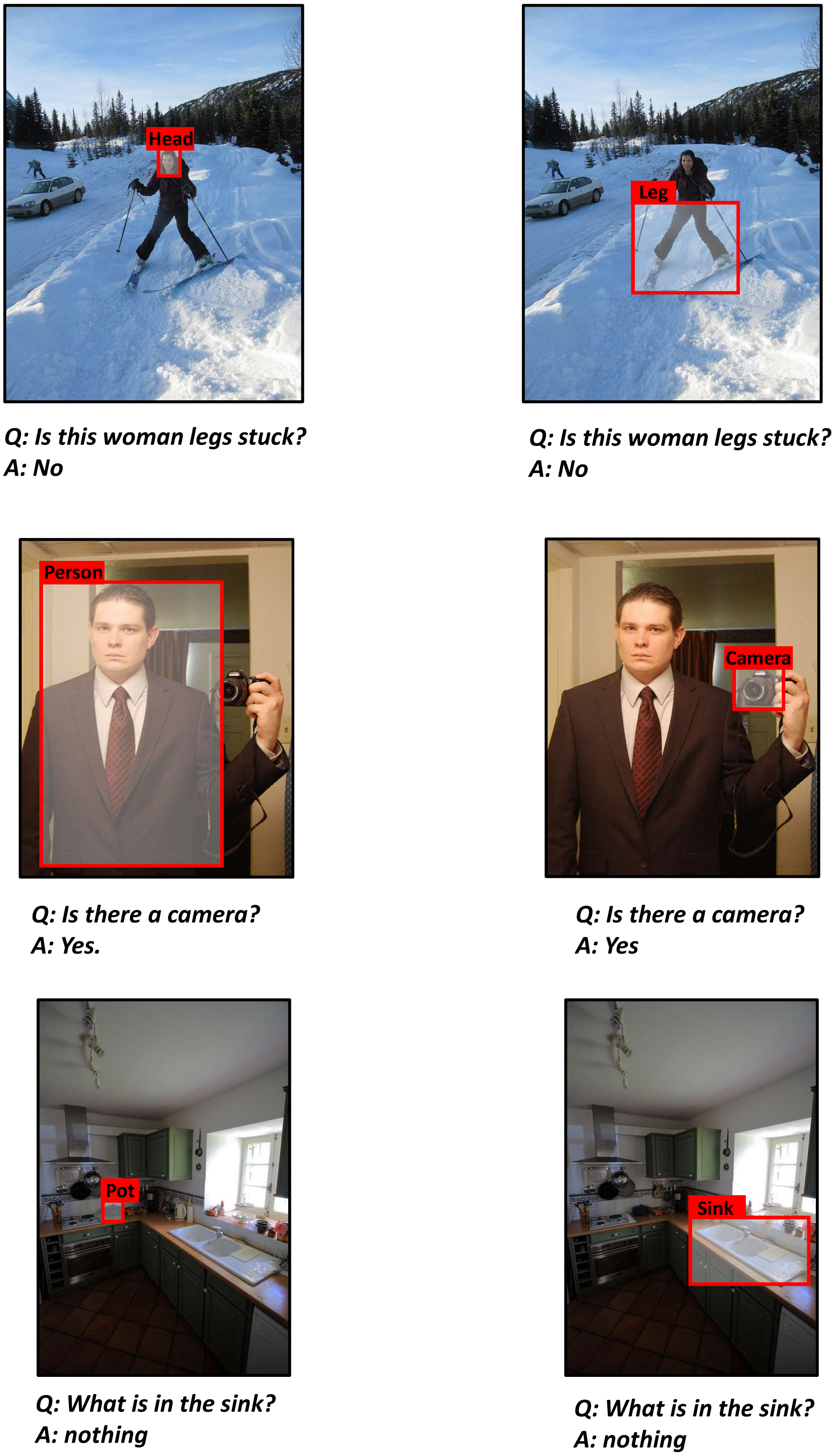
Q: Is the rider a child or an adult?
A: Adult

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.