Supplementary Material for "Causal Attention for Vision-Language Tasks"

Xu Yang[1], Hanwang Zhang[1], Guojun Qi[2], Jianfei Cai[3]
[1]School of Computer Science and Engineering, Nanyang Technological University, Singapore,
[2]Futurewei Technologies
[3]Faculty of Information Technology, Monash University, Australia,
s170018@e.ntu.edu.sg, hanwangzhang@ntu.edu.sg, guojunq@gmail.com, Jianfei.Cai@monash.edu

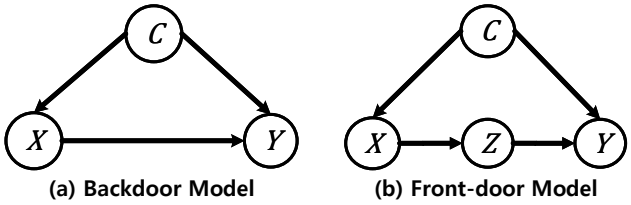

Figure A. Two Structural Causal Models which are (a) a backdoor model and (b) a front-door model.

This supplementary document will further detail the following aspects in the submitted manuscript: A. Causal Preliminaries, B. Formula Derivations, C. More results D. Implementation Details.

A. Causal Preliminaries

A.1. Structural Causal Model

In Causality [7, 8], a Structural Causal Model (SCM) is used to describe the causal relationships. Such a graph connects different variables by directed edges which denote the causal directions. For example, as shown in Figure A(a), $X \rightarrow Y$ denotes that X is the cause of Y. In an SCM, if a variable is the common cause of two variables, it is called the **confounder**. For example, C is the cause of both X and Y, thus it is a confounder which will induce spurious correlation between X and Y to disturb the recognition of the causal effect between them. In particular, such spurious correlation is brought by the **backdoor path** created by the confounder. Formally, a backdoor path between X and Y is defined as **any path from X to Y that starts with an arrow pointing into X**. For example, in Figure A(a), the path $X \leftarrow C \rightarrow Y$ is a backdoor path. Here we use another two examples for helping understand this concept, as in Figure A(b), $X \leftarrow C \rightarrow Y \leftarrow Z$ and $Z \leftarrow X \leftarrow C \rightarrow Y$ are two backdoor paths between X and Z and Z and Y, respectively.

In an SCM, if we want to deconfound two variables X and Y to calculate the true causal effect, we should block every backdoor path between them [8]. For example, in Figure A(a), we should block $X \leftarrow C \rightarrow Y$ to get the causal effect between X and Y.

A.2. Blocking Three Junctions

In an SCM, there are three elemental "junctions" which construct the whole graph and we have some basic rules to block them. In particular, three junctions are given as follows:

1. $X \rightarrow Z \rightarrow Y$. This is called **chain junction**, which constructs a front-door path between X and Y, as shown in Figure A(b). In this junction, once we know the value of the mediator Z, learning about X will not give us any information to raise or lower our belief about Y. Thus, if we know what Z is or directly intervene it as a specific value, we block this chain junction.

2. $X \leftarrow C \rightarrow Y$. This is called **confounding junction** which induces spurious correlation between X and Y, as shown in Figure A(a). In this junction, once we know what the value of C is or directly intervene it to a specific value, there is no spurious correlation between X and Y and thus we block this junction.

3. $Z \rightarrow Y \leftarrow C$. This is called **"collider"** which works in an exactly opposite way from the above chain and confounding junctions. Once we know what the value of Y is, Z and C are correlated. However, if we do not know what Y is or do not intervene it, Z and C are independent and this junction is naturally blocked.

To sum up, if we want to block a path between two variables, we should intervene the middle variables in the chain and confounding junctions and should not intervene in the collider junction. To block a long path, we only need to block a junction of it, *e.g.*, for $X \leftarrow C \rightarrow Y \leftarrow Z$ in Figure A(b), we can block $X \leftarrow C \rightarrow Y$ by intervening C or block $C \rightarrow Y \leftarrow Z$ by not intervening Y.

A.3. The Backdoor Adjustment

The backdoor adjustment is the simplest formula to eliminate the spurious correlation by approximating the "physi-

cal intervention". Formally, it calculates the average causal effect of one variable on another at each stratum of the confounder. For example, in Figure A(a), we can calculate the causal effect of X on Y as $P(Y|do(X))$:

$$P(Y|do(X)) = \sum_c P(Y|X, C = c)P(C = c), \quad \text{(A)}$$

where $do(\cdot)$ signifies that we are dealing with an active intervention rather than a passive observation. The role of Eq. (A) is to guarantee that in each stratum c, X is not affected by C and thus the causal effect can be estimated stratum by stratum from the data.

A.4. The Front-door Adjustment

From Eq. (A), we find that to use the backdoor adjustment, we need to know the details of the confounder for splitting it into various strata. However, in our case, we have no idea about what constructs the hidden confounders in the dataset, thus we are unable to deploy the backdoor adjustment. Fortunately, the front-door adjustment [6] does not require any knowledge on the confounder and can also calculate the causal effect between X and Y in a front-door SCM as in Figure A(b).

In Section 3.1 of the submitted manuscript, we have shown the derivation of the front-door adjustment from the attention mechanism perspective. Here we demonstrate a more formally derivation. The front-door adjustment calculates $P(Y|do(X))$ in the front-door $X \rightarrow Z \rightarrow Y$ by chaining together two partially causal effects $P(Z|do(X))$ and $P(Y|do(Z))$:

$$P(Y|do(X)) = \sum_z P(Z = z|do(X))P(Y|do(Z = z)). \quad \text{(B)}$$

To calculate $P(Z = z|do(X))$, we should block the backdoor path $X \leftarrow C \rightarrow Y \leftarrow Z$ between X and Z. As we discussed in Section A.2 that a collider junction is naturally blocked and here $C \rightarrow Y \leftarrow Z$ is a collider, thus this path is already blocked and we have:

$$P(Z = z|do(X)) = P(Z = z|X). \quad \text{(C)}$$

For $P(Y|do(Z))$, we need to block the backdoor path $Z \leftarrow X \leftarrow C \rightarrow Y$ between Z and Y. Since we do not know the details about the confounder C, we can not use Eq. (A) to deconfound C. Thus we have to block this path by intervening X:

$$P(Y|do(Z = z)) = \sum_x P(Y|Z = z, X = x)P(X = x). \quad \text{(D)}$$

At last, by bringing Eq. (C) and (D) into Eq. (B), we have:

$$P(Y|do(X)) = \sum_z P(Z = z|X) \sum_x P(X = x)[P(Y|Z = z, X = x)], \quad \text{(E)}$$

which is the front-door adjustment given in Eq. (3) of the submitted manuscript.

B. Formula Derivations

Here we show how to use Normalized Weighted Geometric Mean (NWGM) approximation [14, 11] to absorb the sampling into the network for deriving Eq. (5) in the submitted manuscript. Before introducing NWGM, we first revisit the calculation of a function $y(x)$'s expectation according to the distribution $P(x)$:

$$\mathbb{E}_x[y(x)] = \sum_x y(x)P(x), \quad \text{(F)}$$

which is the weighted arithmetic mean of $y(x)$ with $P(x)$ as the weights.

Correspondingly, the weighted geometric mean (WGM) of $y(x)$ with $P(x)$ as the weights is:

$$\text{WGM}(y(x)) = \prod_x y(x)^{P(x)}, \quad \text{(G)}$$

where the weights $P(x)$ are put into the exponential terms. If $y(x)$ is an exponential function that $y(x) = \exp[g(x)]$, we have:

$$\begin{aligned} &\text{WGM}(y(x)) = \prod_x y(x)^{P(x)} \\ &= \prod_x \exp[g(x)]^{P(x)} = \prod_x \exp[g(x)P(x)] \\ &= \exp[\sum_x g(x)P(x)] = \exp\{\mathbb{E}_{\boldsymbol{x}}[g(x)]\}, \end{aligned} \quad \text{(H)}$$

where the expectation $\mathbb{E}_x$ is absorbed into the exponential term. Based on this observation, researchers approximate the expectation of a function as the WGM of this function in the deep network whose last layer is a Softmax layer [14, 11]:

$$\mathbb{E}_x[y(x)] \approx \text{WGM}(y(x)) = \exp\{\mathbb{E}_x[g(x)]\}, \quad \text{(I)}$$

where $y(x) = \exp[g(x)]$.

In our case, we treat $P(Y|X, Z)$ (Eq. (3) of the submitted manuscript) as a predictive function and parameterize it by a network with a Softmax layer as the last layer:

$$P(Y|X, Z) = \text{Softmax}[g(X, Z)] \propto \exp[g(X, Z)]. \quad \text{(J)}$$

Following Eq. (3) of the manuscript and Eq. (I), we have:

$$\begin{aligned} &P(Y|do(X)) \\ &= \sum_z P(Z = z|X) \sum_x P(X = x)[P(Y|Z = z, X = x)] \\ &= \mathbb{E}_{[Z|X]}\mathbb{E}_{[X]}[P(Y|Z, X)] \approx \text{WGM}(P(Y|Z, X)) \\ &\approx \exp\{[g(\mathbb{E}_{[Z|X]}[Z], \mathbb{E}_{[X]}[X])]\}. \end{aligned} \quad \text{(K)}$$

Note that, as in Eq. (J), $P(Y|Z, X)$ is only proportional to $\exp[g(Z, X)]$ instead of strictly equalling to, we only have $\text{WGM}(P(Y|Z, X)) \approx \exp\{[g(\mathbb{E}_{[Z|X]}[Z], \mathbb{E}_{[X]}[X])]\}$ in Eq. (K) instead of equalling to. Furthermore, to guarantee the sum of $P(Y|do(X))$ to be 1, we use a Softmax layer to normalize these exponential units:

$$P(Y|do(X)) \approx \text{Softmax}(g(\mathbb{E}_{[Z|X]}[Z], \mathbb{E}_{[X]}[X])), \quad \text{(L)}$$

where the first part $\mathbb{E}_{[Z|X]}[Z]$ is In-Sample Sampling (IS-Sampling) and the second part $\mathbb{E}_{[X]}[X]$ is CS-Sample Sampling (CS-Sampling). Since the Softmax layer normalizes these exponential terms, this is called the normalized

Table A. The performances of various single models on the online MS-COCO test server.

Model	B@4		M		R-L		C-D	
Metric	c5	c40	c5	c40	c5	c40	c5	c40
BUTD [1]	36.9	68.5	27.6	36.7	57.1	72.4	117.9	120.5
CAVP [5]	37.9	69.0	28.1	37.0	58.2	73.1	121.6	123.8
RFNet [3]	38.0	69.2	28.2	37.2	58.2	73.1	122.9	125.1
SGAE [15]	37.8	68.7	28.1	37.0	58.2	73.1	122.7	125.5
CNM [16]	37.9	68.4	28.1	36.9	58.3	72.9	123.0	125.3
AoANet† [2]	37.3	68.1	28.3	37.2	57.9	72.8	124.0	126.2
Transformer	37.9	69.2	28.7	37.7	58.3	73.3	124.1	126.7
Transformer+CATT	**38.8**	**70.6**	**28.9**	**38.2**	**58.7**	**73.9**	**126.3**	**128.8**

weighted geometric mean (NWGM) approximation.

In a network, the variables X and Z are represented by the embedding vectors and thus we use $\boldsymbol{x}$ and $\boldsymbol{z}$ to denote them. Following the convention in attention research where the attended vectors are usually represented in the matrix form, we also pack the estimated IS-Sampling and CS-Sampling vectors to $\hat{\boldsymbol{X}}$, $\hat{\boldsymbol{Z}}$. In this way, we have:

$$P(Y|do(X)) \approx \text{Softmax}[g(\hat{\boldsymbol{Z}}, \hat{\boldsymbol{X}})], \tag{M}$$

which is given in Eq. (5) of the submitted manuscript.

To estimate $\hat{\boldsymbol{Z}}$, researchers usually calculate a query set from X: $\boldsymbol{Q}_I = h(X)$ and use it in the Q-K-V operation. Similarly, to estimate $\hat{\boldsymbol{X}}$, we can also calculate a query set as: $\boldsymbol{Q}_C = f(X)$ and use it in the Q-K-V operation. In this way, we have Eq. (5) in the submitted manuscript:

$$\begin{aligned} P(Y|do(X)) \approx\ & \text{Softmax}[g(\hat{\boldsymbol{Z}}, \hat{\boldsymbol{X}})], \\ \textbf{IS-Sampling:}\quad & \hat{\boldsymbol{Z}} = \sum\nolimits_z P(Z=z|h(X))\boldsymbol{z}, \\ \textbf{CS-Sampling:}\quad & \hat{\boldsymbol{X}} = \sum\nolimits_x P(X=x|f(X))\boldsymbol{x}. \end{aligned} \tag{N}$$

Note that although $P(X)$ in CS-Sampling does not condition on any variable, we still require a query in Q-K-V operation, since without a query, the estimated result will degrade into a fixed single vector for each different input X: $\hat{\boldsymbol{x}} = \sum_x P(x)\boldsymbol{x}$, where $P(x)$ is the prior probability. We can also treat it as the strategy to increase the representation power of the whole model.

C. More Results

C.1. Online Captioning Test

We report the performances of the MS COCO online split in Table A. It can be found that our single Transformer+CATT can achieve higher performances than the other state-of-the-art models on this split.

C.2. More Qualitative Examples

Figure B shows more qualitative examples where our CATT helps different models confront the dataset biases. The first two rows show six examples of image captioning and the last two rows show the examples of VQA. For example, in the left example of the first row, after incorporating the CATT module, BUTD [1] generates correctly gender of the person without using the spurious correlation between "woman" with "kitchen" in the dataset.

C.3. Failure Case

Our model may fail in some cases where the correct answer can be hardly inferred from the image but bias, just as the case in Figure C, where the hair dryer may infer the person as female due to their frequent co-occurrence in the training dataset. In this way, exploiting the co-occurrence will help the model to answer the question with the smaller risk. However, since these cases are rare in the whole test set, our CATT based architectures may still achieve better performances.

D. Implementation Details

BUTD + CATT. We deployed this architecture for addressing IC and VQA. In the original BUTD architecture, they only used one attention module and thus we also used one causal attention module as in Figure 4. In this architecture, we set IS-ATT the same as the attention module in BUTD where the probability in Eq. (6) is calculated as:

$$\begin{aligned} a_n &= \boldsymbol{w}^T(\boldsymbol{W}_k\boldsymbol{k}_n + \boldsymbol{W}_q\boldsymbol{q}), \\ \boldsymbol{\alpha} &= \text{Softmax}(\{a_1, ..., a_N\}), \end{aligned} \tag{O}$$

where $\boldsymbol{w}$ is a trainable vector and $\boldsymbol{W}_k$, $\boldsymbol{W}_q$ are two trainable matrices. $\boldsymbol{V}_I$, $\boldsymbol{K}_I$ were both set to the RoI feature set of the current image and $\boldsymbol{q}_I$ was the embedding of the sentence context, *e.g.*, the partially generated caption or the question for IC or VQA, respectively. CS-ATT was set to Eq. (7), $\boldsymbol{q}_C$ was the same as in IS-ATT and $\boldsymbol{V}_C$, $\boldsymbol{K}_C$ were both set to the visual global dictionary. This dictionary was initialized by applying K-means over all the RoI features in the training set to get 1000 cluster centres and was updated during the end-to-end training. The RoI object features were extracted by a Faster-RCNN [9] pre-trained on VG as in [1]. The hidden size of the LSTM layers was set to 1024.

For the IC model, the cross-entropy loss and the self-critical reward [10] were used to train it 35 and 65 epochs, respectively. We used the Adam optimizer [4] and initialized the learning rate as $5e^{-4}$ and decayed it by 0.8 every 5 epochs. The batch size was set to 100. For the VQA model, we followed [12, 1] to use the binary cross-entropy loss and applied the AdaDelta optimizer [18], which does not require to fix the learning rate, to train it 30 epochs. The batch size was set to 512.

Transformer + CATT. We deployed the architecture in Figure 5 for solving IC and VQA. In this architecture, the Q-K-V operations of all IS-ATT and CS-ATT were imple-

BUTD: a woman and a dog in a kitchen
CATT: a man standing next to a dog in a kitchen

BUTD: a herd of sheep in a field
CATT: a herd of sheep walking down a road

BUTD: a blue and red fire hydrant on a sidewalk
CATT: a blue and yellow fire hydrant on the side of a street

TF: a group of people riding a horse
CATT: a horse drawn carriage on a field with people

BUTD: a desk with four laptops
CATT: two computer monitors and two laptops on a desk

TF: a man feeding a cow
CATT: a man milking a cow with a bottle

What gender is the person holding the frisbee?
TF: male CATT: female

What does it look like the skier is doing?
TF: snowboarding CATT: falling

How many people are shown??
TF: 2 CATT: 3

What sport is being shown on the screen?
LXMERT: dancing CATT: bowling

What the color of the building in the background?
LXMERT: blue CATT: brown

How many elephants are shown?
LXMERT: 2 CATT: 4

Figure B. More examples demonstrate that our CATT helps various models confront dataset biases. Red and blue index the incorrect and correct generated captions and answers, respectively.

Figure C. The failure case of our LXMERT+CATT.

mented by 8-head scaled dot product [13]:

$$\begin{aligned} \textbf{Input:}\quad & \boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} \\ \textbf{Prob:}\quad & \boldsymbol{A}_i = \text{Softmax}(\frac{\boldsymbol{Q}\boldsymbol{W}_i^Q(\boldsymbol{K}\boldsymbol{W}_i^K)^T}{\sqrt{d}}) \\ \textbf{Single-Head}:\quad & \boldsymbol{H}_i = \boldsymbol{A}_i\boldsymbol{V}\boldsymbol{W}_i^V, \\ \textbf{Ouput:}\quad & \hat{\boldsymbol{V}} = \text{Embed}([\boldsymbol{H}_1, ..., \boldsymbol{H}_8]\boldsymbol{W}^H), \end{aligned} \tag{P}$$

where $\boldsymbol{W}_i^*$ and $\boldsymbol{W}^H$ are all trainable matrices; $\boldsymbol{A}_i$ is the soft attention matrix for the i-th head; $[\cdot]$ denotes the concatenation operation, and Embed$(\cdot)$ means the feed-forward network and the residual operation as in [13]. We shared the parameters between IS-ATT and CS-ATT in each CATT to keep the outputs staying in the same feature space. Then compared with the original Transformer, the increments of the trainable parameters only come from the global image and word embedding dictionaries, which were initialized by applying K-means over the RoI and word embeddings of the training set. We set the sizes of both dictionaries to 500 and the hidden size of all the attention modules to 512. The RoI object features were the same as in BUTD+CATT.

For IC, the training included two processes: we first used the cross-entropy loss and then the self-critical reward to train the captioner 15 and 35 epochs, respectively. The learning rates of two processes were initialized as $5e^{-4}$ and $5e^{-5}$ and both of them decayed by 0.8 every 5 epochs. The Adam optimizer was used and the batch size was set to 10. For VQA, we applied the binary cross-entropy loss and the Adam optimizer to train it 13 epochs. We followed [17] to set the learning rate to $\textbf{min}(2.5te^{-5}, 1e^{-4})$, where t is the training epoch and after 10 epochs, the learning rate decayed by 0.2 every 2 epochs. The batch size was set to 64.

References

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018. 3

[2] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *International Conference on Computer Vision*, 2019. 3

[3] Wenhao Jiang, Lin Ma, Yu-Gang Jiang, Wei Liu, and Tong Zhang. Recurrent fusion network for image captioning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 499–515, 2018. 3

[4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3

[5] Daqing Liu, Zheng-Jun Zha, Hanwang Zhang, Yongdong Zhang, and Feng Wu. Context-aware visual policy network for sequence-level image captioning. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 1416–1424. ACM, 2018. 3

[6] Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995. 2

[7] Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016. 1

[8] Judea Pearl and Dana Mackenzie. *The Book of Why*. Basic Books, New York, 2018. 1

[9] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 3

[10] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024, 2017. 3

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

[12] Damien Teney, Peter Anderson, Xiaodong He, and Anton Van Den Hengel. Tips and tricks for visual question answering: Learnings from the 2017 challenge. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4223–4232, 2018. 3

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 5

[14] 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 *International conference on machine learning*, pages 2048–2057, 2015. 2

[15] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *Pro-*

ceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 10685–10694, 2019. 3

[16] Xu Yang, Hanwang Zhang, and Jianfei Cai. Learning to collocate neural modules for image captioning. *arXiv preprint arXiv:1904.08608*, 2019. 3

[17] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6281–6290, 2019. 5

[18] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012. 3