

Deconfounded Image Captioning: A Causal Retrospect

Xu Yang¹, Hanwang Zhang¹, Jianfei Cai²

`s170018@e.ntu.edu.sg`, `hanwangzhang@ntu.edu.sg`, `jianfei.cai@monash.edu`

¹School of Computer Science and Engineering, Nanyang Technological University,

²Faculty of Information Technology, Monash University,

Abstract. The dataset bias in vision-language tasks is becoming one of the main problems that hinder the progress of our community. However, recent studies lack a principled analysis of the bias. In this paper, we present a novel perspective: Deconfounded Image Captioning (DIC), to find out the cause of the bias in image captioning, then retrospect modern neural image captioners, and finally propose a DIC framework: DICv1.0. DIC is based on causal inference, whose two principles: the backdoor and front-door adjustments, help us to review previous works and design the effective models. In particular, we showcase that DICv1.0 can strengthen two prevailing captioning models and achieves a single-model 130.7 CIDEr-D and 128.4 c40 CIDEr-D on Karpathy split and online split of the challenging MS-COCO dataset, respectively. Last but not least, DICv1.0 is merely a natural derivation from our causal retrospect, which opens a promising direction for image captioning.

Keywords: Image Captioning; Causal Reasoning; Deconfounding; Back-door and Front-door Adjustments

1 Introduction: Finding the Devil

Recently, our vision-language community has paid more and more attention to dataset bias. On one hand, it shows that we have already achieved impressive or even super-human performances on many benchmark leader-boards. On the other hand, it also shows that we do not actually believe in these “super-human” systems due to the obvious dataset bias, *e.g.*, image captioners are likely to discriminate what they see [16] (*e.g.*, generating “man” instead of “woman” when it is actually a woman snowboards) and hallucinate what they do not see [43] (*e.g.*, generating “sitting on bench” when only seeing people talking on the phone); VQA agents usually answer “Yes” when asked “Are the apples red?” even without a look at the image [14]. Our efforts on confronting the bias have evolved from diagnosis — collecting new datasets [20,14,1,45] — to treatment — designing unbiased models [16,9].

Unfortunately, none of us has ever asked the question “Who caused the bias?”; or, we take for granted to blame annotation workers; or, we are thus

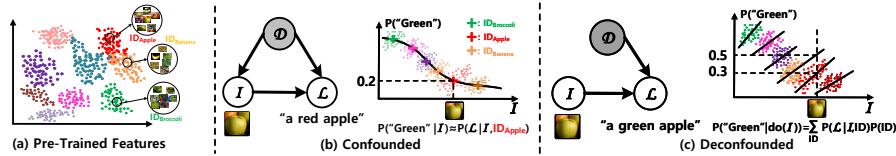


Fig. 1: A causal look at the dataset bias in image captioning. (a) By pre-training, image features are grouped into semantic clusters. (b) The conventional training objective $P(\mathcal{L}|\mathcal{I})$ is confounded by \mathcal{D} , which is brought by pre-training. The clusters collapse into the semantic points and the learnt probability manifold crosses these points. (c) If \mathcal{D} is observed, we can deconfound the training by using a new objective $P(\mathcal{L}|do(\mathcal{I}))$, which sums the probability of each data stratum, where each probability is learnt by the data in the corresponding stratum.

trapped in the “make a dataset”–“it’s biased”–“make a new one” loop¹. However, we seriously believe that the real devil is someone else hidden somewhere, because we human ourselves are living in a biased nature, and our biological vision-language system works well regardless of any bias, *e.g.*, long-tailed concept distributions, reporting bias [34], and language bias [8]. So shouldn’t we blame the datasets. The goal of this paper is trying to pursue the cause of the bias and propose a principled solution to end the loop. In particular, we use Image Captioning (IC) as the case study because, among all the vision-language tasks [33, 49, 4, 11, 18], it has the longest history (since Show&Tell [49] in early deep learning era) and the simplest cross-modal objective.

The devil we believe is in the pre-training dataset. This conjecture may sound shocking at first and it won’t be after we delve into the story depicted in Fig. 1. Indeed, modern computer vision systems are almost all built upon backbone deep neural networks (*e.g.*, ResNet [15] or Faster R-CNN [41]) pre-trained on large-scale datasets (*e.g.*, ImageNet [44] and MS-COCO [26]). The pre-training not only speeds up the training, but also provides a powerful feature extractor for down-stream tasks. As shown in Fig. 1(a), the backbone network will represent the image features into groups, such as ID_{Apple} , ID_{Banana} , and $ID_{Broccoli}$, which are inherited from the semantic labels in the discarded pre-training data. This meets our expectation because good feature representations should be semantic. Once pre-trained, all of us will keep the network but discard the dataset, and it is this behaviour that turns the “angel” pre-training into a “devil” confounder. A confounder yields *spurious correlation* between two independent events [37]. For example, we usually observe that many accepted CV papers are colorful, but we cannot conclude that **Colorful Paper** → **Accept** using $P(\text{Accept}|\text{Colorful Paper})$, because there is probably a plausible confounder **High Quality** such that **High Quality** → **Colorful Paper** and **High Quality** → **Accept**.

¹ by Alexei Efros@CVPR2019 Computer Vision After 5 Years workshop (<https://futurecv.github.io/schedule.html>)

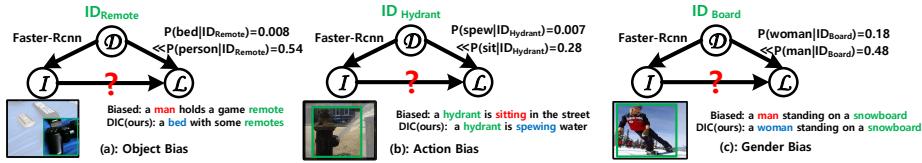


Fig. 2: Three examples show how the confounder \mathcal{D} causes the spurious correlation $\mathcal{I} \leftarrow \mathcal{D} \rightarrow \mathcal{L}$ to mislead us from the true objective $\mathcal{I} \rightarrow \mathcal{L}$. The probability denotes the percentage of co-occurrence of two words in the training set, e.g., $P(\text{person}|ID_{\text{Remote}})$ means that “person” and “remote” contributes the 54% occurrences of “remote”. Red/blue denote the wrong/right words, respectively.

When we attach the pre-trained network to the captioning model, two causal effects happen. As shown in Fig. 1(b), $\mathcal{D} \rightarrow \mathcal{I}$ denotes that pre-training dataset \mathcal{D} has a causal effect on image \mathcal{I} , implying that the visual encoder transforms the input image into the above mentioned visual features; while $\mathcal{D} \rightarrow \mathcal{L}$ denotes that \mathcal{D} also has a causal effect on caption \mathcal{L} . This is because IC is trained with image-caption pairs, which essentially maps the visual concepts inherited from the pre-training dataset into words, e.g., ID_{Human} maps to word “man” or “woman”. This is how the causal link $\mathcal{D} \rightarrow \mathcal{L}$ is built. Thus, \mathcal{D} **becomes a confounder** that influences both \mathcal{I} and \mathcal{L} and causes a spurious correlation $\mathcal{I} \leftarrow \mathcal{D} \rightarrow \mathcal{L}$ when the image content has nothing to do with the captions. Fig. 2 revisits some typical biases in the new point of view. For example, since there are much more “remote-person” than “remote-bed” in the captioning training set, the captioning models incorrectly exploit such co-occurrence from the spurious correlation to generate the captioning. Fig. 1(b) illustrates a more general scenario for confounded IC. Due to the existence of \mathcal{D} , $P(\mathcal{L}|\mathcal{I})$ will inevitably learn a prediction function (the curve in Fig. 1(b)) by catering to the dominating confounding factors such as ID_{Apple} is usually paired with **Red Apple** sentences. Thus, once the IC discovers the feature cluster ID_{Apple} in a “green apple” image, $P(\mathcal{L}|\mathcal{I})$ will force $P(\text{“Green”}|\mathcal{I}) \approx P(\text{“Green”}|\mathcal{I}, ID_{\text{Apple}})$, which is low.

So far, we show that *pre-training* dataset \mathcal{D} is the confounder who introduces the spurious correlation $\mathcal{I} \leftarrow \mathcal{D} \rightarrow \mathcal{L}$ that causes the bias by using the conventional training objective $P(\mathcal{L}|\mathcal{I})$. The rest is how to deconfound it — *Deconfounded Image Captioning (DIC)* — which is our goal in this paper. A principled solution is to pursue a new training objective: $P(\mathcal{L}|do(\mathcal{I}))$, which is fundamentally different from $P(\mathcal{L}|\mathcal{I})$ in causal inference [37,36]. As illustrated in Fig. 1(c), the *do*-operation promotes the posterior probability from passive *observation* to active *intervention*, which includes the following two steps: 1) cut off the link $\mathcal{D} \rightarrow \mathcal{I}$ because we hope that the input \mathcal{I} will never relate to the confounder \mathcal{D} by assuming \mathcal{I} could be any visual concept, and 2) we are safe to calculate the likelihood $P(\mathcal{L}|\mathcal{I}, d)$ within each stratum d of \mathcal{D} , where d denotes any one of the visual concepts in \mathcal{D} . The two steps are also known as the backdoor adjustment [37], which will be formally detailed in Section 2.1.

However, do not forget that \mathcal{D} is **no longer observed** after pre-training, especially when we use any of the 3rd-party feature extractors [13] — we have no access to the data. Hence, in this paper, our technical contribution for DIC is to provide another resolution that avoids the explicit intervention for \mathcal{D} . Particularly, in Section 4, we propose a novel DIC framework called **DICv1.0** based on both the backdoor and front-door adjustments [37] (see Section 2.1 and 2.2). We apply it in two prevailing models: the classic Up-Down [3] and the state-of-the-art AoANet [17], and help both of them boost the CIDEr-D scores from 126.4 to 129.5 and from 128.7 to 130.7, respectively, where the latter is submitted to the MS-COCO Caption test server and achieves a 128.4 CIDEr c40.

We would like to highlight that the above mentioned DIC framework is not a deliberate idea, but merely a natural derivation from our Causal Retrospect (see Section 3 and Fig. 4), which *per se* is a brief history for our 5-year-old IC community, enlightened by the causal view of bias in Fig. 1. More promisingly, as we will discuss in Section 6, this retrospect points us to a whole new future.

2 Preliminaries: Causal Intervention

Deconfounding seeks the true causal effect of one variable on another, and it is appealing for the objective of image captioning (IC): given \mathcal{I} , we hope the model’s caption \mathcal{L} being faithful only to the content of \mathcal{I} . In this section, we review two main deconfounding techniques in causal inference [37,36].

2.1 Backdoor Adjustment

We first use the example in Fig. 1(b) to see through, mathematically, why passive correlation $P(\mathcal{L}|\mathcal{I})$ brings biases into IC models if the dataset is unbalanced. Using Bayes rule, we have

$$P(\mathcal{L}|\mathcal{I}) = \sum_d P(\mathcal{L}|\mathcal{I}, d)P(d|\mathcal{I}), \quad (1)$$

where d is a concept ID in \mathcal{D} . If the pre-training is perfectly good, when \mathcal{I} contains an apple, we have $P(ID_{Apple}|\mathcal{I}) \approx 1$, which indicates that the “fruitful” apple becomes a “dry” ID_{Apple} . So, $P(\mathcal{L}|\mathcal{I})$ degrades to $P(\mathcal{L}|\mathcal{I}, ID_{Apple})$, which we are actually training! Once the samples `Red Apple` are dominating in the unbalanced data, the IC tends to build strong connections between `Red` and ID_{Apple} even without seeing the color of the apple. In this way, IC is contaminated by the spurious correlation: $\mathcal{I} \leftarrow \mathcal{D} \rightarrow \mathcal{L}$.

The backdoor adjustment [37] computes an active intervention posterior $P(\mathcal{L}|do(\mathcal{I}))$ as:

$$P(\mathcal{L}|do(\mathcal{I})) = \sum_d P(\mathcal{L}|\mathcal{I}, d)P(d). \quad (2)$$

Compared with Eq. (1), we see the key difference: the adjustment weight $P(d|\mathcal{I})$ is changed to $P(d)$ because \mathcal{D} is no longer dependent on \mathcal{I} , *i.e.*, $P(d|\mathcal{I}) = P(d)$, after the intervened cut-off (Fig. 1(c)). This encourages IC to maximize $P(\mathcal{L}|\mathcal{I}, d)$ for every stratum d , only subject to a prior $P(d)$ listening to no one, and hence the IC is deconfounded.

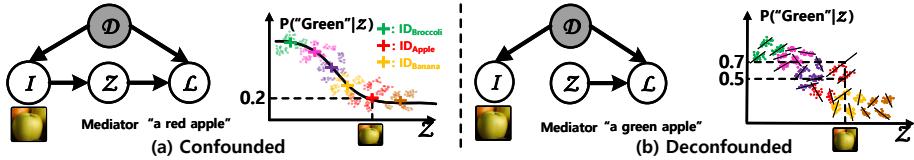


Fig. 3: The front-door IC whose captioning process is $\mathcal{I} \rightarrow \mathcal{Z} \rightarrow \mathcal{L}$, where \mathcal{Z} groups representations into smaller clusters. (a) If this IC is trained by the passive correlation $P(\mathcal{L}|\mathcal{I})$ (Eq. (3)), \mathcal{Z} is still implicitly affected by \mathcal{D} through the path $\mathcal{D} \rightarrow \mathcal{I} \rightarrow \mathcal{Z}$. Then the learnt $P("Green"|\mathcal{Z})$ still crosses the collapsed semantic points as in Fig. 1(b). (b) When we train this IC by the active intervention $P(\mathcal{L}|do(\mathcal{I}))$ (Eq. (4)), the link $\mathcal{I} \rightarrow \mathcal{Z}$ is cut off and this IC is deconfounded. Then each smaller stratum learns a corresponding probability and $P("Green"|do(\mathcal{Z}))$ is got by summing them as in Fig. 1(c).

2.2 Front-door Adjustment

Since \mathcal{D} is no longer accessible after pre-training, we cannot deploy the backdoor adjustment to calculate the intervention $P(\mathcal{L}|do(\mathcal{I}))$. Fortunately, we have the front-door adjustment [37]. Please refer to Supplementary Material for a mathematical proof. Here, we only sketch its key idea.

As shown in Fig. 3(a), a mediator \mathcal{Z} is used to transfer knowledge from \mathcal{I} to \mathcal{L} (*e.g.*, the prevailing visual attention [50]). We wish to use \mathcal{Z} as a better representation than \mathcal{I} , *e.g.*, it groups features into more fine-grained clusters. Then the caption is generated from $P(\mathcal{L}|\mathcal{Z} = \mathbf{z})$, where \mathbf{z} is drawn from $P(\mathbf{z}|\mathcal{I})$. So, by the Bayes' rule, we have:

$$P(\mathcal{L}|\mathcal{I}) = \sum_{\mathbf{z}} P(\mathbf{z}|\mathcal{I})P(\mathcal{L}|\mathcal{Z} = \mathbf{z}). \quad (3)$$

Fortunately, $P(\mathbf{z}|\mathcal{I}) = P(\mathbf{z}|do(\mathcal{I}))$ because \mathcal{L} is a collider that blocks any information through the path $\mathcal{I} \leftarrow \mathcal{D} \rightarrow \mathcal{L} \leftarrow \mathcal{Z}$, which means the path $\mathcal{I} \rightarrow \mathcal{Z}$ is already deconfounded. However, the path $\mathcal{Z} \rightarrow \mathcal{L}$ is still confounded by \mathcal{D} via the backdoor $\mathcal{Z} \leftarrow \mathcal{I} \leftarrow \mathcal{D} \rightarrow \mathcal{L}$, *e.g.*, fine-grained clusters still compose some bigger clusters indexed by the same color, which are implicitly affected by \mathcal{D} . Therefore, similar to the above backdoor analysis, IC with a mediator is still confounded, *i.e.*, $P("Green"|\mathcal{I})$ is low.

As illustrated in Fig. 3(b), where the link $\mathcal{I} \rightarrow \mathcal{Z}$ is cut off, the front-door adjustment intervenes \mathcal{Z} by calculating the likelihood at each stratum \mathbf{x} of \mathcal{I} , where \mathbf{x} represents the visual feature. Though the diversity of them is extremely large, they are after all observable. Similar to Eq. (2), we have $P(\mathcal{L}|do(\mathcal{Z} = \mathbf{z})) = \sum_{\mathbf{x}} P(\mathcal{L}|\mathbf{z}, \mathbf{x})P(\mathbf{x})$. Overall, by replacing $P(\mathcal{L}|\mathcal{Z} = \mathbf{z})$ in Eq. (3) with $P(\mathcal{L}|do(\mathcal{Z} = \mathbf{z}))$, we have the front-door adjustment as:

$$P(\mathcal{L}|do(\mathcal{I})) = \sum_{\mathbf{z}} P(\mathbf{z}|\mathcal{I}) \sum_{\mathbf{x}} P(\mathcal{L}|\mathbf{z}, \mathbf{x})p(\mathbf{x}). \quad (4)$$

When we use $P(\mathcal{L}|do(\mathcal{I}))$ to train a front-door IC model, this model is not affected by unbalanced training because both $P(\mathbf{z}|\mathcal{I})$ and $\sum_{\mathbf{x}} P(\mathcal{L}|\mathbf{z}, \mathbf{x})p(\mathbf{x})$

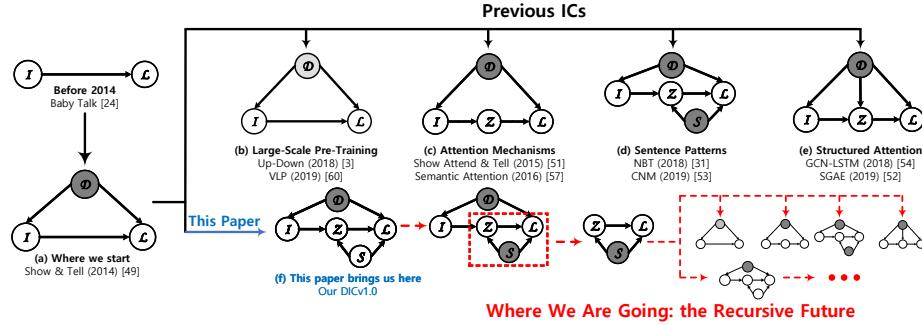


Fig. 4: The causal retrospect of some major IC models. The past/present/future are colored by black/blue/red, respectively. Shaded \mathcal{D} denotes that confounder is not observable. We will look ahead the recursive future in Section 6.

are deconfounded. Therefore, similar to the above backdoor analysis, this IC is deconfounded, which means that $P(\text{"Green"} | \mathcal{I})$ is high.

3 Related Work: A Causal Retrospect

We follow Figure 4 to retrospect the image captioning models (ICs) proposed in recent 5 years from the causal view: deconfounded IC (DIC). For space limit, we mainly review IC in the deep learning era. We will see that even though the researchers have improved IC in many ways, they might be unaware of the underlying reasons for their contributions.

Show & Tell (S&T) [49] (Figure 4(a)). Compared with earlier template-based IC like Baby Talk [24], S&T was the first modern IC which is pre-trained on large-scale ImageNet [44]. Though S&T gained large improvement from pre-training, we now know that the pre-training also releases the devil to cause bias.

Large-scale Pre-training (Figure 4(b)). A straightforward way to deconfounding is to turn the confounder into non-confounder. We can approximate this by scaling up the pre-training data. If it is infinite, it will include every possible visual-semantic pair, thus the pre-trained model will perfectly parse the image into self-contained semantic labels and then the semantic gap disappears between vision and language. Many works can fall into this causal graph, *e.g.*, Up-Down [3] exploited Visual Genome [23] with a larger label space, HIP [54] used more fine-grained segmentation annotations, and vision-language BERT frameworks [29,58] used 3 million image-caption pairs provided by Conceptual Captions [45]. Though their performances are boosted, these ICs are still confounded since these pre-train data are far from complete.

Attention Mechanisms (Figure 4(c)). SAT [50] was the first work using attention mechanism, and then this mechanism was used in every follow-up IC systems [56,30,55,3,57,17]. This mechanism allows an IC to scan over all the visual concepts of an image and then select suitable ones conditioned on the lan-

guage context, *e.g.*, spatial attention [50,3] and semantic attention [56] selected the most informative visual features and semantic labels to generate the captions. In fact, the attention mechanism is a hybrid approximation of the backdoor and the front-door adjustments. Compared with the backdoor adjustment (Eq. (2)), where $P(\mathcal{L}|do(\mathcal{I}))$ is averaged over all the visual concepts of \mathcal{D} , attention only averages over all the appeared visual concepts of the given image, which is only a small subset of \mathcal{D} . Thus, \mathcal{I} is only partially intervened. As for front-door, attention treats the visual regions as \mathcal{Z} and then generates captions from the selected positions. However, like we discussed in Section 2.2, by only adding a front-door mediator but without intervention, these ICs are still confounded.

Sentence Patterns (Figure 4(d)). Researchers also designed ICs which imitate humans to dynamically structure sentence patterns for captioning. In particular, they used diverse modules for different patterns, *e.g.*, NBT [31] designed two modules for nouns and other words, and CNM [52] applied four fine-grained modules for nouns, adjectives, relation words, and function words. During captioning, sentence pattern is learnt dynamically for selecting suitable modules to generate the corresponding words. At first glance, this framework looks like a front-door IC where \mathcal{Z} is sentence pattern. However, if we look closer, it can be discovered that both \mathcal{Z} and \mathcal{L} are confounded by language resource \mathcal{S} , since both sentence patterns and captions are learnt from the language resource. Thus, the causal graph is given in Figure 4(d), where the two confounders \mathcal{D} and \mathcal{S} exist. **Structured Attention** (Figure 4(e)). When we humans describe an image, we first build a semantic structure (*e.g.*, scene graph) about this image and then turn the structure into the final caption. Inspired by this, researchers [53,51] proposed to learn a scene graph from the image first and applied structured attention to select the sub-structure for captioning. This also belongs to the front-door graph where \mathcal{Z} is the structured attention. However, they learn the structure generator from the pre-training dataset \mathcal{D} , which in fact causes a link $\mathcal{D} \rightarrow \mathcal{Z}$ to further confound the IC, as shown in Figure 4(e).

4 Deconfounded Image Captioning

In this section, we discuss how to derive our DICv1.0 from the causal retrospect in Fig. 4 and how to implement it into the prevailing encoder-decoder framework.

4.1 Choosing \mathcal{Z}

Since \mathcal{D} is not available after pre-training, we have to deconfound the IC by the front-door adjustment (Fig. 3(a)). To achieve this, the first challenge is the selection of the mediator \mathcal{Z} . Based on the retrospect, we know three candidates of \mathcal{Z} which are spatial position (in attention mechanism), sentence pattern, and structured attention. For spatial position (Fig. 4(c)), it only provides the visual concepts in the given image, which is a small subset of \mathcal{D} , thus it is not a good candidate. For sentence pattern (Fig. 4(d)), the calculation of its probability is

almost impossible since it is decided by the whole caption, which is not available during the caption generation, thus sentence pattern is also not a good candidate. For structured attention (Fig. 4(e)), it is designed to be decided by the image only, while researchers [53,51] learn the structure generator from \mathcal{D} , which adds the link $\mathcal{D} \rightarrow \mathcal{Z}$ into the causal graph, thus this IC is confounded. However, if we only associate the structure attention with the image, we have a causal graph which cuts off the link $\mathcal{D} \rightarrow \mathcal{Z}$ as Fig. 3(a).

To achieve this, we sample a semantic structure set from ConceptNet [28] and treat it as the mediator \mathcal{Z} , *e.g.*, **Car AtLocation Road** and **Apple Is Red**. After collecting \mathcal{Z} (Section 5.1), we transfer those discrete structures to continuous representations \mathbf{z} by averaging the word embeddings of the words in the semantic structure, which are naturally better than the original visual features since both semantic structures and captions belong to the semantic domain. When this IC generates the caption, it will first use the visual features of the image to retrieve the related representations of structures from \mathcal{Z} ($\mathcal{I} \rightarrow \mathcal{Z}$ of Fig. 3(a)) and compose them into the final caption ($\mathcal{Z} \rightarrow \mathcal{L}$ of Fig. 3(a)), where the implementation details are given in Section 4.4. For example, given an image contains **A Green Apple**, we wish the IC to retrieve the related structures about **Apple** or **Green**, *e.g.*, **Apple Is Red** or **Grass Is Green** (Fig. 5(b)), and then the IC generates words from these structures. Strictly speaking, this strategy also introduces the language resource \mathcal{S} as the confounder since we use the words from \mathcal{S} as the keys for sampling related semantic structures ($\mathcal{S} \rightarrow \mathcal{Z}$ of Fig. 4(f)) and those words also affect the caption generation ($\mathcal{S} \rightarrow \mathcal{L}$ of Fig. 4(f)). Fortunately, compared with the inaccessible \mathcal{D} , \mathcal{S} is available since we know what exactly these key words are. Based on the above analysis, we derive our **DICv1.0**, whose causal graph is sketched in Fig. 4(f), which has two confounders \mathcal{D} and \mathcal{S} .

Given this causal graph, we exploit both the backdoor and front-door adjustments to calculate the corresponding intervention distribution as:

$$\begin{aligned} P(\mathcal{L}|do(\mathcal{I})) &= \sum_{\mathbf{s}} P(\mathbf{s}) \sum_{\mathbf{x}} P(\mathbf{x}) \sum_{\mathbf{z}} P(\mathbf{z}|\mathcal{I}) [P(\mathcal{L}|\mathbf{s}, \mathbf{x}, \mathbf{z})] \\ &= \mathbb{E}_{\mathbf{s}} \mathbb{E}_{\mathbf{x}} \mathbb{E}_{[\mathbf{z}|\mathcal{I}]} [P(\mathcal{L}|\mathbf{s}, \mathbf{x}, \mathbf{z})], \end{aligned} \quad (5)$$

which is the expected value of $P(\mathcal{L}|\mathbf{s}, \mathbf{x}, \mathbf{z})$ according to three variables \mathbf{s} , \mathbf{x} , and \mathbf{z} , which denote the word embeddings of key words, the visual features of the image \mathcal{I} , and the embeddings of the semantic structures, respectively. The derivation of Eq. (5) is given in the Supplementary Material.

To implement our DICv1.0 into the encoder-decoder framework, we parameterize $p(\mathcal{L}|\mathbf{s}, \mathbf{x}, \mathbf{z})$ by a network. The last layer of this network is a Softmax layer that implements $P(\mathcal{L}|\mathbf{s}, \mathbf{x}, \mathbf{z})$ as:

$$P(\mathcal{L}|\mathbf{s}, \mathbf{x}, \mathbf{z}) = \text{Softmax}[g(\mathbf{s}, \mathbf{x}, \mathbf{z})], \quad (6)$$

where $g(\cdot)$ is the embedding layer before the Softmax. However, this brings one challenge that in order to compute $P(\mathcal{L}|do(\mathcal{I}))$ in Eq. (5), we need a huge number of outputs sampled from this network. To solve this challenge, we propose a two-step approximation which allows us to forward the network only once for an

estimation of $P(\mathcal{L}|do(\mathcal{I}))$. The first step is called Normalized Weighted Geometric Mean (NWGM) approximation [50, 46, 6] which absorbs the expectations into the network (see Section 4.2). The second step is to sample finite values from \mathcal{S} , \mathcal{X} , and \mathcal{Z} for estimating the expectations in Eq. (5) (see Section 4.3).

4.2 Normalized Weighted Geometric Mean

By NWGM approximation [50], the expectation of a Softmax unit is approximated as the Softmax of the expectation:

$$P(\mathcal{L}|do(\mathcal{I})) = \mathbb{E}_{\mathbf{s}} \mathbb{E}_{\mathbf{x}} \mathbb{E}_{[\mathbf{z}|\mathcal{I}]} \{\text{Softmax}[g(\mathbf{s}, \mathbf{x}, \mathbf{z})]\} \approx \text{Softmax}\{\mathbb{E}_{\mathbf{s}} \mathbb{E}_{\mathbf{x}} \mathbb{E}_{[\mathbf{z}|\mathcal{I}]}[g(\mathbf{s}, \mathbf{x}, \mathbf{z})]\}. \quad (7)$$

Furthermore, if $g(\cdot)$ is a fully connected layer, we have:

$$P(\mathcal{L}|do(\mathcal{I})) \approx \text{Softmax}\{g(\mathbb{E}_{\mathbf{s}}[\mathbf{s}], \mathbb{E}_{\mathbf{x}}[\mathbf{x}], \mathbb{E}_{[\mathbf{z}|\mathcal{I}]}[\mathbf{z}])\} \quad (8)$$

Here we can put the expectation into the fully connected layer $g(\cdot)$ because the linear projection of the expectation of one variable equals to the expectation of the linear projection of that variable. More details about the derivations of Eq. (7) and (8) by NWGM approximation are given in Supplementary Material.

4.3 Sampling for Expectations

When we compute the intervention distribution at word level, it is also conditioned on a context vector \mathbf{h} which accumulates the knowledge of previous generated words. By modifying Eq. (5), the word distribution is:

$$\begin{aligned} P(\mathcal{L}|do(\mathcal{I}), \mathbf{h}) &= \sum_{\mathbf{s}} P(\mathbf{s}|\mathbf{h}) \sum_{\mathbf{x}} P(\mathbf{x}|\mathbf{h}) \sum_{\mathbf{z}} P(\mathbf{z}|\mathcal{I}, \mathbf{h}) [P(\mathcal{L}|\mathbf{s}, \mathbf{x}, \mathbf{z}, \mathbf{h})] \\ &= \mathbb{E}_{[\mathbf{s}|\mathbf{h}]} \mathbb{E}_{[\mathbf{x}|\mathbf{h}]} \mathbb{E}_{[\mathbf{z}|\mathcal{I}, \mathbf{h}]} [P(\mathcal{L}|\mathbf{s}, \mathbf{x}, \mathbf{z}, \mathbf{h})]. \end{aligned} \quad (9)$$

By Eq. (7) and (8), we can put the expectations into the Softmax, what we should do next is to calculate: $\mathbb{E}_{[\mathbf{s}|\mathbf{h}]}[\mathbf{s}]$, $\mathbb{E}_{[\mathbf{x}|\mathbf{h}]}[\mathbf{x}]$, and $\mathbb{E}_{[\mathbf{z}|\mathcal{I}, \mathbf{h}]}[\mathbf{z}]$.

Here we use $\mathbb{E}_{[\mathbf{x}|\mathbf{h}]}[\mathbf{x}]$ as the example to show how to estimate these expectations. The challenge is that it is time-prohibitive to compute $\mathbb{E}_{[\mathbf{x}|\mathbf{h}]}[\mathbf{x}] = \sum_{\mathbf{x}} p(\mathbf{x}|\mathbf{h}) \mathbf{x}$ by sampling all the possible visual feature \mathbf{x} . To solve this, we first learn K samples from the visual features of the whole training set by a dictionary learning algorithm [32] and group them as a dictionary $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_K\}$. Then, we compute the expectation of these K samples as an estimation of $\mathbb{E}_{[\mathbf{x}|\mathbf{h}]}[\mathbf{x}]$. Specifically, we define an **EXPT** module to estimate this expectation:

$$\begin{aligned} \text{Input: } \mathbf{X} &= \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_K\}, \mathbf{h} \\ \text{Probability: } P(\mathbf{x}_k|\mathbf{h}) &= \text{Softmax}(\mathbf{x}_k^T \mathbf{h}) \\ \text{Output: } \mathbb{E}_{[\mathbf{x}|\mathbf{h}]}[\mathbf{x}] &\approx \sum_k P(\mathbf{x}_k|\mathbf{h}) \mathbf{x}_k, \end{aligned} \quad (10)$$

where \mathbf{h} is the context vector. Similarly, to estimate $\mathbb{E}_{[\mathbf{z}|\mathcal{I}, \mathbf{h}]}[\mathbf{z}]$, we sample a semantic structure set \mathbf{Z} from ConceptNet [28] (see Section 5.1). When we sample

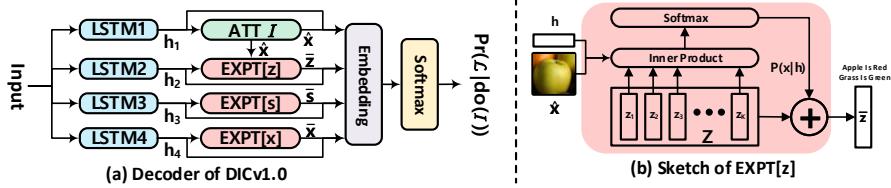


Fig. 5: (a) The sketch of our DICv1.0’s decoder, where ATT/EXPT represent attention/expectation modules. (b) The sketch of EXPT $[z]$ (Eq. (10)).

Z , we use the nouns, verbs, and adjectives of language resource as the key words for searching the related triplets from ConceptNet, thus these key words act as the confounder S and we group the word embeddings of them as the dictionary S for estimating $E_{[s|h]}[s]$.

4.4 Implementation Details

We incorporate our DICv1.0 into two models: Up-Down [3] and AoANet [17] and name them as **UD-DICv1.0** and **AoA-DICv1.0**, respectively. In both models, the visual encoder is a ResNet-101 Faster R-CNN [41] pre-trained on Visual Genome [23] as in Up-Down [3]. The decoders of two models have a similar architecture, which is sketched in Fig. 5. The input of this decoder concatenates three terms: the mean pooling of the image feature set I , the previous generated word’s embedding, and the previous embedding layer’s output. ATT I represents an attention module which computes the attended vector \hat{x} from I . This \hat{x} is used to retrieve the related semantic structures and is input to the embedding layer since it is also a sample of the visual features. EXPT $[z]$, EXPT $[s]$, and EXPT $[x]$ are used to estimate $\bar{z} = E_{[z|I,h]}[z]$, $\bar{s} = E_{[s|h]}[s]$, and $\bar{x} = E_{[x|h]}[x]$, respectively.

When UD-DICv1.0 or AoA-DICv1.0 is deployed, ATT I is Top-Down attention or Attention on Attention; the embedding layer is an LSTM or a GLU [12], respectively. Note that though the embedding layer is not a fully connected layer, we still observe less bias and better performances compared with the original models (see Section 5.2). We train both models 35 epochs by cross-entropy loss and another 65 epochs by self-critical with CIDEr-D rewards [42,48]. When inference, we use beam search with a beam size of 5.

5 Experiments

5.1 Datasets and Metrics

MS-COCO [10]. We validated our models on MS-COCO IC dataset. In particular, our models were tested on two different splits: Karpfathy split [21] and the official online test split, which divide the whole dataset into 113,287/5,000/5,000 and 82,783/40,504/40,775 images for training/validation/test, respectively. We

Table 1: The performances of various ablative studies on MS-COCO Karpathy split. The metrics: B@4, M, R, C, S, CHs, CHi, A@Gen, A@Attr, and A@Act denote BLEU@4, METEOR, ROUGE-L, CIDEr-D, SPICE, CHAIRs, CHAIRi, the accuracy of gender, attribute, and action words. The symbols \uparrow and \downarrow mean the higher the better and the lower the better, respectively.

Models	B@4 \uparrow	M \uparrow	R \uparrow	C \uparrow	S \uparrow	CHs \downarrow	CHi \downarrow	A@Gen \uparrow	A@Attr \uparrow	A@Act \uparrow
UD	37.2	27.5	57.3	125.3	20.7	13.7	8.9	0.81	0.41	0.52
UD-BD	38.2	28.2	58.0	126.9	21.3	11.2	7.6	0.87	0.50	0.56
UD-FD/Cor	38.0	28.1	58.0	126.5	21.1	12.3	8.3	0.83	0.46	0.54
UD-FD	38.5	28.4	58.7	127.6	21.8	10.5	7.0	0.89	0.55	0.58
UD-DICv1.0	38.7	28.4	58.8	128.2	21.9	10.2	6.7	0.90	0.57	0.59

followed previous researches to pre-process our captions [3,51]. At last we trimmed each caption to a maximum of 16 words and had a vocabulary of 10,369 words by removing the words which appear less than 5 times.

ConceptNet [28]. ConceptNet has structures denoted as **Subject Relation Object**. We used the nouns, verbs, and adjectives which appear more than 20 times in MS-COCO IC set to search for the related structures. We removed the structure if it contains a word out of the caption vocabulary or its weight is lower than 2.5. Finally, we had a semantic structure set Z with 9,590 elements.

Metrics. We not only followed previous researches to use the following five metrics: CIDEr-D [48], BLEU [35], METEOR[7], ROUGE [25], and SPICE [2], but also used CHAIRs and CHAIRi [43] to measure the bias degree.

5.2 Ablative Studies

We used Up-Down as the backbone to design various ablative studies to validate the importance of the backdoor adjustment (Section 2.1), the front-door adjustment (Section 2.2), and our DICv1.0 (Section 4.4).

Comparing Methods. **UD:** We re-implemented Up-Down [3] as our baseline, where only ATT \mathcal{I} exists in the decoder. **UD-BD:** Compared with UD, we followed the backdoor adjustment (Eq. (2)) and estimated $\mathbb{E}_{[d|h]}[d]$ by EXPT module (Eq. (10)). The input dictionary D contained the word embeddings of 80 visual concepts in MS-COCO. This baseline was designed to confirm the utility of the backdoor adjustment. **UD-FD/Cor:** Compared with UD, we added EXPT $[z]$ into the decoder. This equals to train a front-door IC by passive correlation $P(\mathcal{L}|\mathcal{I})$. **UD-FD:** Compared with UD-FD/Cor, we added EXPT $[x]$ into the decoder. This equals to train a front-door IC by active intervention $P(\mathcal{L}|do(\mathcal{I}))$ while neglecting the confounder S . This baseline was used to confirm the utility of the front-door adjustment. **UD-DICv1.0:** Compared with UD-FD, we added EXPT $[s]$ into the decoder to get the integral decoder of our DICv1.0 as in Fig. 5.

Results and Analysis. Table 1 reports the performances of our UD-DICv1.0 and the baselines. Compared with the original UD, UD-DICv1.0 improves CIDEr-D from 125.3 to 128.2, which means that *UD-DICv1.0 generates the most similar*

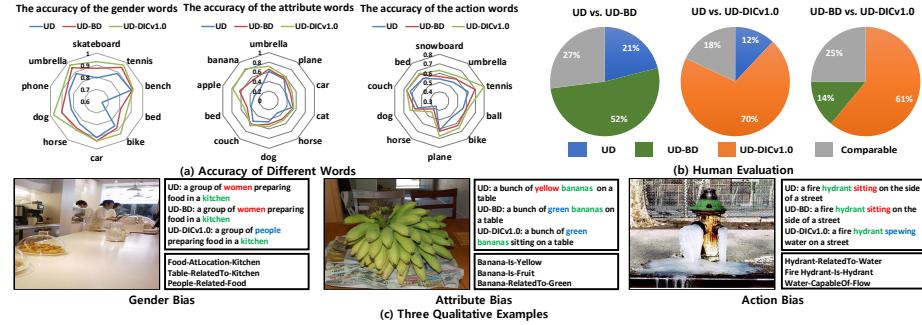


Fig. 6: (a) The accuracy of the gender, attribute, and action words when some specific visual concepts appear. (b) The pie charts each comparing two ICs. (c) Three examples show that our UD-DICv1.0 generates better captions. The visual concepts which may cause bias are colored by green. The red/blue mean the inconsistent/consistent words, respectively. The bottom blocks show some retrieved semantic structures when UD-DICv1.0 generates the blue words.

captions as the ground-truth. More importantly, UD-DICv1.0 lowers CHs/CHi from 13.7/8.9 to 10.2/6.7, which confirms that *UD-DICv1.0 generates the least biases*. By comparing UD-BD with UD, we observe that UD-BD achieves higher CIDEr-D and lower CHs&CHi than UD, which confirms the utility of the backdoor adjustment. We can observe a similar result when comparing UD-FD with UD-FD/Cor, which confirms the utility of the front-door adjustment. Interestingly, compared with UD-BD, UD-FD has better performances, which means that *the approximation of the backdoor adjustment is less effective than the front-door adjustment*. Such observation coincides with the discussion in Introduction that the front-door adjustment is a better choice when the confounder \mathcal{D} is not observed. Importantly, UD-DICv1.0 performs better than UD-FD, which reflects that *simply using the additional resource is not enough for generating the best captions unless we discover the hidden confounder and deconfound it*.

Analysis of the Bias. Apart from using CHs&CHi to measure the object bias, we also analyzed more specific biases: gender bias, action bias, and attribute bias. We tested these biases by calculating the accuracy of these words when a visual concept appears in the generated captions. For example, for gender bias, we calculated whether the gender word is consistent between the ground truth caption with the generated caption when a visual concept, *e.g.*, skateboard, appears. Table 1 shows the mean accuracy of these specific words when some visual concepts appear. Noteworthy, we consider the balance of the words here that we separately calculate the accuracy of each word and average the results to obtain the mean accuracy. We can observe that when better deconfounding techniques are used, the accuracy increases, *e.g.*, UD-BD performs better than UD and UD-FD outperforms UD-FD/Cor. Importantly, our UD-DICv1.0 achieves the highest accuracy of all the gender, attribute, and action words, which means

Table 2: The performances on Karpathy split. The left and right parts report the performances trained by CIDEr-D computed from 5 captions and the whole training set, respectively. “Group” shows each IC’s category according to Fig. 4. The symbol \dagger means the re-implemented model.

Models	Group	B@4	M	R	C	S
Up-Down [3]	b	36.3	27.7	56.9	120.1	21.4
Up-Down \dagger [3]	b	37.2	27.5	57.3	125.3	20.7
RFNet [19]	c	37.9	28.3	58.3	125.7	21.7
CAVP [57]	c	38.6	28.3	58.5	126.3	21.6
LBPF [39]	c	38.3	28.5	58.4	127.6	22.0
CNM [52]	d	38.7	28.4	58.7	127.4	21.8
GCN-LSTM [53]	e	38.2	28.5	58.3	127.6	22.0
SGAE [51]	e	38.4	28.4	58.6	127.8	22.1
UD-DICv1.0	f	38.7	28.4	58.8	128.2	21.9

Models	Group	B@4	M	R	C	S
Up-Down \dagger [3]	b	37.7	28.2	58.1	126.4	21.8
UD-HIP [54]	b	38.2	28.4	58.3	127.2	21.9
VLP [58]	b	39.5	—	—	129.3	23.2
AoANet [17]	c	38.9	29.2	58.8	129.8	22.4
AoANet \dagger [17]	c	38.9	28.9	58.4	128.7	22.4
UD-DICv1.0	f	38.3	28.5	58.5	129.5	22.0
AoA-DICv1.0	f	39.5	29.5	58.8	130.7	22.6

that *UD-DICv1.0 generates the least biases brought by unbalanced training*. The radar charts in Fig. 6(a) show the accuracy when some specific visual concepts appear. Furthermore, we conducted human evaluation which asks 20 humans to sort the 50 captions, which were generated by UD/UD-BD/UD-DICv1.0, according to their consistencies with the images. The results in Fig. 6(b) demonstrate that humans consider our UD-DICv1.0’s captions more consistent. And Fig. 6(c) visualizes three qualitative examples about three specific biases.

5.3 Comparisons with State-of-The-Art

Comparing Methods. We start our comparison from **Up-Down** [3], whose visual features are most frequently used by the subsequent ICs and so do we. We follow the causal retrospect in Fig. 4 to group the compared state-of-the-art ICs into four groups: **large-scale pre-training**: **Up-Down** [3], **UD-HIP** [54], and **VLP** [58]; **attention mechanisms**: **CAVP** [27], **RFNet** [19], **LBPF** [39], and **AoANet** [17]; **sentence patterns**: **CNM** [52]; and **structured attention**: **GCN-LSTM** [53] and **SGAE** [51]. Noteworthy, two different CIDEr-D are used as the training self-critical rewards in previous ICs. The first one computes Inverse Document Frequency (IDF) from each image’s five captions and the second one computes IDF from the whole training set. For fair comparisons, we report the performances trained by two different CIDEr-D in Table 2, where the left and right parts report the results of the first and second CIDEr-D.

Results and Analysis. From Table 2, we can find that our single-model UD-DICv1.0 and AoA-DICv1.0 achieve the best CIDEr-D: 128.2 and 130.7 with different training CIDEr-D. Compared with UD-HIP which pre-trains their IC by object detection and segmentation, our UD-DICv1.0, which is only pre-trained by object detection, has better performances. Interestingly, compared with VLP which exploits 30 times more samples (3 millions) than ours (0.1 million) to pre-train their model, our UD-DICv1.0 and AoA-DICv1.0 still achieve competitive results. Both comparisons suggest that *our DIC framework is more cost-effective and efficient than large-scale pre-training* in boosting performances.

Table 3: The performances of single methods on the online MS-COCO test server.

Model Metric	B@4		M		R-L		C-D	
	c5	c40	c5	c40	c5	c40	c5	c40
Up-Down [3]	36.9	68.5	27.6	36.7	57.1	72.4	117.9	120.5
CAVP [27]	37.9	69.0	28.1	37.0	58.2	73.1	121.6	123.8
RFNet [19]	38.0	69.2	28.2	37.2	58.2	73.1	122.9	125.1
SGAE [51]	37.8	68.7	28.1	37.0	58.2	73.1	122.7	125.5
CNM [52]	37.9	68.4	28.1	36.9	58.3	72.9	123.0	125.3
AoANet [17]	37.3	68.1	28.3	37.2	57.9	72.8	124.0	126.2
UD-DICv1.0	37.9	69.2	28.7	37.7	58.3	73.3	124.1	126.7
AoA-DICv1.0	38.8	70.5	28.8	38.2	58.6	73.9	126.2	128.4

Our DIC also outperforms the ICs with complex attention mechanisms, *e.g.*, UD-DICv1.0 is better than RFNet, CAVP, and LBPF and AoA-DICv1.0 is better than AoANet though we do not use more advanced training strategy as AoANet [17]. Finally, comparing our UD-DICv1.0 with the approximations of the front-door frameworks: sentence patterns and structured attention, we find that our UD-DICv1.0 is still the best, although we do not use multi-step reasoning as CNM and do not deploy complex graph convolution as GCN-LSTM and SGAE. All of these comparisons confirm the superiority of the proposed DIC framework than the ICs with weaker deconfounding approximations. We also compare our single-model UD-DICv1.0 and AoA-DICv1.0 with the other ICs on MS-COCO online test set. From Table 3 we observe that our UD-DICv1.0 and AoA-DICv1.0 achieve the highest CIDEr-D c5 and c40 scores.

6 Conclusions

We used the causal intervention to offer an in-depth analysis for deconfounded image captioning (DIC), a novel framework that explains why IC is confounded, and we concluded that the confounder is the pre-training dataset. We retrospected the major progress in IC in the DIC framework, and then derived an effective method called DICv1.0. We validated it by using two prevailing models: Up-Down and AoANet, and helped both of them achieve better performances.

For years, alas, our vision-language community has always borrowed the methodologies from other fields, such as NLP for encoder-decoder [47], attention [5], and sentence-level loss [40], and the visual detection community for visual backbones [41]. But our motivation was naive: they succeed in their respective fields, so they should continue in the combination. Yet, we do not have a methodology — of our own — that is from the unique nature of vision-language. From the causal retrospect in Fig. 4, we see a promising future. *DICv1.0 is just a start that is far from an end!* It is derived by assuming that \mathcal{S} is adjustable, while this assumption can be relaxed to unobservable. Then, after re-assigning \mathcal{S} to \mathcal{D} and \mathcal{Z} to \mathcal{I} , we jump back into “where we start” and it is possible to recursively deploy all the previous techniques, including this paper, which will be the “previous IC” in the recursive future, as illustrated in Fig. 4.

Supplementary Material for “Deconfounded Image Captioning: A Causal Retrospect”

Xu Yang¹, Hanwang Zhang¹, Jianfei Cai²

s170018@e.ntu.edu.sg, hanwangzhang@ntu.edu.sg, jianfei.cai@monash.edu

¹School of Computer Science and Engineering, Nanyang Technological University,

²Faculty of Information Technology, Monash University,

This supplementary document will further detail the following aspects in the submitted manuscript: A. Formula Derivations, B. Network Architecture, C. More Results, D. Details of Human Evaluations.

1 Formula Derivations

1.1 Causal Graph

Before introducing the backdoor adjustment, it is beneficial to discuss more details of the causal graph. In the framework of causal inference [37,38], a causal graph is represented by a directed graph where the direction of the arrow means whether this link is causal or anticausal. Note that the information can be conveyed in both directions: causal or anticausal. For example, as shown in Fig. 1(a), there are two paths between \mathcal{I} and \mathcal{L} : a causal path $\mathcal{I} \rightarrow \mathcal{L}$ and a spurious correlation path $\mathcal{I} \leftarrow \mathcal{D} \rightarrow \mathcal{L}$. For this spurious correlation path, it is also a backdoor path.

Formally, a backdoor path between \mathcal{I} and \mathcal{L} is defined as: **any path from \mathcal{I} to \mathcal{L} that starts with an arrow pointing into \mathcal{I}** . Here we show another two examples for helping understand this concept. In Fig. 1(b), $\mathcal{I} \rightarrow \mathcal{Z}$ is a causal path between \mathcal{I} and \mathcal{Z} , while the path $\mathcal{I} \leftarrow \mathcal{D} \rightarrow \mathcal{L} \leftarrow \mathcal{Z}$ is a backdoor path between \mathcal{I} and \mathcal{Z} since \mathcal{D} pointing into \mathcal{I} . Another example is the backdoor path $\mathcal{Z} \leftarrow \mathcal{I} \leftarrow \mathcal{D} \rightarrow \mathcal{L}$ between \mathcal{Z} and \mathcal{L} since \mathcal{I} pointing into \mathcal{Z} .

In a causal graph, if we want to **deconfound two variables \mathcal{I} and \mathcal{L} to calculate the causal effect of \mathcal{I} on \mathcal{L} , we only need to block every backdoor path between \mathcal{I} and \mathcal{L}** [38]. For example, if we want to get the causal effect of \mathcal{I} on \mathcal{L} in Fig. 1(a), we only need to block the backdoor path $\mathcal{I} \leftarrow \mathcal{D} \rightarrow \mathcal{L}$.

1.2 Blocking Paths

Here we introduce three rules about how to block a path to stop the flow of information between two variables. In a causal graph, there are three different elemental “junctions” which construct the whole graph. Correspondingly, there are three rules for blocking information flows in these three junctions. Three junctions are given as follows:

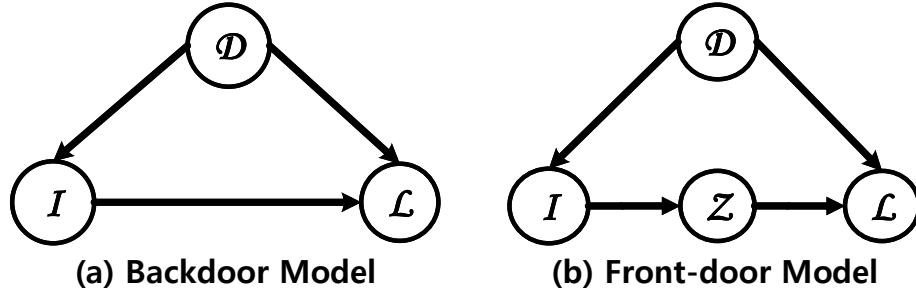


Fig. 1: Two causal graphs which are (a) a backdoor model and (b) a front-door model.

1. $\mathcal{A} \rightarrow \mathcal{B} \rightarrow \mathcal{C}$. This is called **chain junction**, where \mathcal{B} is a mediator transmitting information from \mathcal{A} to \mathcal{C} . In this junction, once we know the value of the mediator \mathcal{B} , learning about \mathcal{A} will not give us any information to raise or lower our belief about \mathcal{C} . Therefore, if we directly control \mathcal{B} to certain value, the information flow from \mathcal{A} to \mathcal{C} is blocked. For example, we know that hard working causes a high quality paper and finally affects the acceptance of this paper: Hard Working \rightarrow High Quality \rightarrow Accept, where High Quality is a mediator. Once we control High Quality to be true, we know the paper is likely to be accepted and we do not need to know any information about Hard Working to lower or raise this belief.

2. $\mathcal{A} \leftarrow \mathcal{B} \rightarrow \mathcal{C}$. This is called **confounding junction** where \mathcal{B} is a confounder of \mathcal{A} and \mathcal{C} . In this junction, once we know what the value of confounder \mathcal{B} is, there is no spurious correlation between \mathcal{A} and \mathcal{C} . Therefore, as in chain junction, if we directly control \mathcal{B} to certain value, the information flow from \mathcal{A} to \mathcal{C} is blocked. We have already met this junction in Introduction of the submitted manuscript where we use Colorful Paper \leftarrow High Quality \rightarrow Accept as the example. In this example, once we control a paper to have High Quality, the spurious correlation between Colorful Paper and Accept is eliminated, which means we deconfound Colorful Paper and Accept.

3. $\mathcal{A} \rightarrow \mathcal{B} \leftarrow \mathcal{C}$. This is called “**collider**” which works in exactly opposite way from the above chain and confounding junctions. In this junction, if we do not know what the value of \mathcal{B} is, \mathcal{A} and \mathcal{C} are independent. However, once we know the value of \mathcal{B} , \mathcal{A} and \mathcal{C} are correlated! We still use paper acceptance as the example, suppose both High Quality and Luck affect Accept, though High Quality and Luck are unrelated. Under this situation, we have Luck \rightarrow Accept \leftarrow High Quality, if we do not know whether a paper is accepted, Luck and High Quality are independent. Once we know a paper is accepted, there is a negative correlation between Luck and High Quality: finding out an accepted paper with High Quality will lower our belief that this paper is accepted due to researchers’ Luck. Therefore, this path is naturally blocked if we do not control \mathcal{B} .

To sum up, if we want to block the information flow between \mathcal{A} and \mathcal{C} , we can directly control \mathcal{B} to certain value in both chain and confounding junctions, and we must not control \mathcal{B} in a “collider”.

For a long pipe with many variables, if a single junction is blocked, then the whole pipe is also blocked. For example, if we want to block the backdoor path $\mathcal{Z} \leftarrow \mathcal{I} \leftarrow \mathcal{D} \rightarrow \mathcal{L}$ between \mathcal{Z} and \mathcal{L} in Fig. 1(b), we can control \mathcal{I} or \mathcal{D} since $\mathcal{Z} \leftarrow \mathcal{I} \leftarrow \mathcal{D}$ or $\mathcal{I} \leftarrow \mathcal{D} \rightarrow \mathcal{L}$ is a chain or confounding junction, respectively. And the backdoor path $\mathcal{I} \leftarrow \mathcal{D} \rightarrow \mathcal{L} \leftarrow \mathcal{Z}$ between \mathcal{I} and \mathcal{Z} in Fig. 1(b) is naturally blocked since there is a collider $\mathcal{D} \rightarrow \mathcal{L} \leftarrow \mathcal{Z}$.

1.3 Backdoor Adjustment

The backdoor adjustment is the simplest formula we can use to deconfound \mathcal{I} and \mathcal{L} by controlling \mathcal{D} to block the backdoor path $\mathcal{I} \leftarrow \mathcal{D} \rightarrow \mathcal{L}$ in Fig. 1(a). In causal inference, “controlling” is achieved by calculating the average causal effect of \mathcal{I} on \mathcal{L} at each stratum d of the deconfounder \mathcal{D} and then computing the weighted average of those strata according to the prior of each stratum $P(d)$. Therefore, we have the intervention distribution:

$$P(\mathcal{L}|do(\mathcal{I})) = \sum_d P(\mathcal{L}|\mathcal{I}, d)P(d), \quad (1)$$

where do -operator signifies that we are dealing with an active intervention rather than a passive observation. The role of Eq. (1) is to guarantee that the causal effect in each stratum d of \mathcal{D} to be the same as the observed trend in this stratum. In this way, the causal effect can be estimated stratum by stratum from the data. For example, when we use Eq. (1) to estimate $P(\text{“Green”}|do(\mathcal{I}))$, it calculates the causal effect of \mathcal{I} on “Green” by using the trend of each stratum, $P(\text{“Green”}|\mathcal{I}, d)$, and final averages them to get the averaged causal effect, as demonstrated in Figure 1(c) of the submitted manuscript. In this way, the image captioning model is deconfounded.

1.4 Front-door Adjustment

However, the backdoor adjustment does not exhaust all ways of estimating the causal effect, there are some graphical patterns where we can not directly apply the backdoor adjustment. For example, in our image captioning model, since pre-train dataset \mathcal{D} is not accessible after pre-training, we can not use the backdoor adjustment to calculate the causal effect at each stratum of \mathcal{D} .

Fortunately, we have the front-door adjustment [37] to calculate the causal effect of \mathcal{I} on \mathcal{L} even when \mathcal{D} is not accessible. Fig. 1(b) shows the front-door model, where a mediator \mathcal{Z} transmits knowledge from \mathcal{I} to \mathcal{L} . To deconfound $\mathcal{I} \rightarrow \mathcal{Z} \rightarrow \mathcal{L}$, we first calculate two partially effects $P(\mathcal{Z}|do(\mathcal{I}))$ and $P(\mathcal{L}|do(\mathcal{Z}))$, then we chain together the two partial effects to get the overall causal effect of \mathcal{I} on \mathcal{L} :

$$P(\mathcal{L}|do(\mathcal{I})) = \sum_z P(\mathcal{Z}=z|do(\mathcal{I}))P(\mathcal{L}|do(\mathcal{Z}=z)). \quad (2)$$

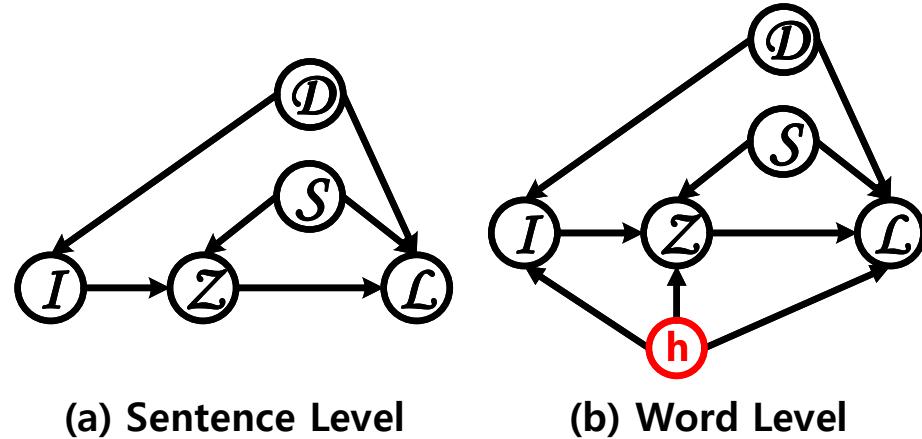


Fig. 2: The causal graphs of our DICv1.0 model, (a) and (b) mean two different perspectives of our model: sentence level and word level, respectively. We denote h as red to mean that the value of this variable is already computed.

To calculate $P(\mathcal{Z} = z|do(\mathcal{I}))$, we should block the backdoor path $\mathcal{I} \leftarrow \mathcal{D} \rightarrow \mathcal{L} \leftarrow \mathcal{Z}$ between \mathcal{I} and \mathcal{Z} . Fortunately, this path is naturally blocked due to the collider $\mathcal{D} \rightarrow \mathcal{L} \leftarrow \mathcal{Z}$ that we do not need to control any variable, thus we have:

$$P(\mathcal{Z} = z|do(\mathcal{I})) = P(\mathcal{Z} = z|\mathcal{I}). \quad (3)$$

For $P(\mathcal{L}|do(\mathcal{Z} = z))$, we need to block the backdoor path $\mathcal{Z} \leftarrow \mathcal{I} \leftarrow \mathcal{D} \rightarrow \mathcal{L}$ between \mathcal{Z} and \mathcal{L} . We can control \mathcal{I} or \mathcal{D} to block this path since $\mathcal{Z} \leftarrow \mathcal{I} \leftarrow \mathcal{D}$ or $\mathcal{I} \leftarrow \mathcal{D} \rightarrow \mathcal{L}$ is a chain or confounding junction. Since \mathcal{D} is not accessible now, we have to control \mathcal{I} to block this path, thus we have:

$$P(\mathcal{L}|do(\mathcal{Z} = z)) = \sum_{\mathbf{x}} P(\mathcal{L}|z, \mathbf{x})P(\mathbf{x}), \quad (4)$$

where \mathbf{x} denotes the visual feature of \mathcal{I} . At last, by Eq. (2), we have:

$$P(\mathcal{L}|do(\mathcal{I})) = \sum_{\mathbf{z}} P(z|\mathcal{I}) \sum_{\mathbf{x}} P(\mathcal{L}|z, \mathbf{x})p(\mathbf{x}), \quad (5)$$

which is Eq. (4) of the submitted manuscript.

1.5 Derivations of Eq. (5) and Eq. (9)

The causal graphs of our DICv1.0 model are shown in Fig. 2. We will derive Eq. (5) and Eq. (9) of the submitted manuscript based on these two causal graphs. We first show how to calculate Eq. (5), which is the sentence level $P(\mathcal{L}|do(\mathcal{I}))$, from the causal graph in Fig. 2(a). To achieve this, we follow the procedure used in calculating Eq. (5) that we first calculate $P(\mathcal{Z}|do(\mathcal{I}))$ and

$P(\mathcal{L}|do(\mathcal{Z}))$, then we chain them together to get the final $P(\mathcal{L}|do(\mathcal{I}))$. To get $P(\mathcal{Z}|do(\mathcal{I}))$, we find

$$P(\mathcal{Z} = \mathbf{z}|do(\mathcal{I})) = P(\mathcal{Z} = \mathbf{z}|\mathcal{I}). \quad (6)$$

since both the backdoor paths $\mathcal{I} \leftarrow \mathcal{D} \rightarrow \mathcal{L} \leftarrow \mathcal{Z}$ and $\mathcal{I} \leftarrow \mathcal{D} \rightarrow \mathcal{L} \leftarrow \mathcal{S} \rightarrow \mathcal{Z}$ between \mathcal{I} and \mathcal{Z} are naturally blocked due to the colliders $\mathcal{D} \rightarrow \mathcal{L} \leftarrow \mathcal{Z}$ and $\mathcal{D} \rightarrow \mathcal{L} \leftarrow \mathcal{S}$, respectively.

To get $P(\mathcal{L}|do(\mathcal{Z}))$, we need to block two backdoor paths between \mathcal{Z} and \mathcal{L} , which are $\mathcal{Z} \leftarrow \mathcal{S} \rightarrow \mathcal{L}$ and $\mathcal{Z} \leftarrow \mathcal{I} \leftarrow \mathcal{D} \rightarrow \mathcal{L}$. To block the former path, we have to control \mathcal{S} , and to block the latter path, we have to control \mathcal{I} since \mathcal{D} is not accessible. Therefore, we should control two variables \mathcal{S} and \mathcal{I} to block both two backdoor paths. Therefore, we have

$$\begin{aligned} & P(\mathcal{L}|do(\mathcal{Z} = \mathbf{z})) \\ &= \sum_{\mathbf{s}} P(\mathbf{s}) \sum_{\mathbf{x}} P(\mathbf{x}) [P(\mathcal{L}|\mathbf{s}, \mathbf{x}, \mathbf{z})]. \end{aligned} \quad (7)$$

To sum up, after chaining two partial effects together, we have the sentence level $P(\mathcal{L}|do(\mathcal{I}))$ as:

$$\begin{aligned} & P(\mathcal{L}|do(\mathcal{I})) \\ &= \sum_{\mathbf{s}} P(\mathbf{s}) \sum_{\mathbf{x}} P(\mathbf{x}) \sum_{\mathbf{z}} P(\mathbf{z}|\mathcal{I}) [P(\mathcal{L}|\mathbf{s}, \mathbf{x}, \mathbf{z})] \\ &= \mathbb{E}_{\mathbf{s}} \mathbb{E}_{\mathbf{x}} \mathbb{E}_{[\mathbf{z}|\mathcal{I}]} [P(\mathcal{L}|\mathbf{s}, \mathbf{x}, \mathbf{z})], \end{aligned} \quad (8)$$

which is our Eq. (5) in the submitted manuscript.

When we calculate $P(\mathcal{L}|do(\mathcal{I}))$ at word level, as shown in Fig. 2(b), it is also conditioned on the variable \mathcal{H} which denotes the accumulated context knowledge of the partially generated caption. However, at each time step, \mathcal{H} has a computed value \mathbf{h} , which means \mathcal{H} is already controlled to be \mathbf{h} . Note that \mathcal{H} only appears in the confounding junctions: $\mathcal{I} \leftarrow \mathcal{H} \rightarrow \mathcal{Z}$, $\mathcal{I} \leftarrow \mathcal{H} \rightarrow \mathcal{L}$, $\mathcal{Z} \leftarrow \mathcal{H} \rightarrow \mathcal{L}$. Therefore, the paths which go through \mathcal{H} are blocked, e.g., $\mathcal{I} \leftarrow \mathcal{H} \rightarrow \mathcal{Z}$ or $\mathcal{I} \leftarrow \mathcal{D} \rightarrow \mathcal{L} \leftarrow \mathcal{H} \rightarrow \mathcal{Z}$ is already blocked. As a result, we can modify Eq. (6) as:

$$P(\mathcal{Z} = \mathbf{z}|do(\mathcal{I}), \mathbf{h}) = P(\mathcal{Z} = \mathbf{z}|\mathcal{I}, \mathbf{h}), \quad (9)$$

Eq. (7) as:

$$\begin{aligned} & P(\mathcal{L}|do(\mathcal{Z} = \mathbf{z}), \mathbf{h}) \\ &= \sum_{\mathbf{s}} P(\mathbf{s}|\mathbf{h}) \sum_{\mathbf{x}} P(\mathbf{x}|\mathbf{h}) [P(\mathcal{L}|\mathbf{s}, \mathbf{x}, \mathbf{z}, \mathbf{h})], \end{aligned} \quad (10)$$

and Eq. (8) as:

$$\begin{aligned} & P(\mathcal{L}|do(\mathcal{I}), \mathbf{h}) \\ &= \sum_{\mathbf{s}} P(\mathbf{s}|\mathbf{h}) \sum_{\mathbf{x}} P(\mathbf{x}|\mathbf{h}) \sum_{\mathbf{z}} P(\mathbf{z}|\mathcal{I}, \mathbf{h}) [P(\mathcal{L}|\mathbf{s}, \mathbf{x}, \mathbf{z}, \mathbf{h})] \\ &= \mathbb{E}_{[\mathbf{s}|\mathbf{h}]} \mathbb{E}_{[\mathbf{x}|\mathbf{h}]} \mathbb{E}_{[\mathbf{z}|\mathcal{I}, \mathbf{h}]} [P(\mathcal{L}|\mathbf{s}, \mathbf{x}, \mathbf{z}, \mathbf{h})], \end{aligned} \quad (11)$$

which is our Eq. (9) in the submitted manuscript. Note that $P(\mathbf{s}|\mathbf{h})$ should be $P(\mathbf{s})$ since there is no direct link from \mathbf{h} to \mathcal{S} . While in the experiment, we still set \mathcal{S} to be conditioned on \mathbf{h} to increase the representation power of the whole model. And if not, after normalized weighted geometric mean approximation, the expectation of \mathcal{S} will degrade to a fixed vector, as we will show in Section 1.6.

1.6 Derivations of Eq. (7) and Eq. (8)

Here we show how to use Normalized Weighted Geometric Mean (NWGM) approximation [50,46,6] to absorb the expectations into the network for deriving Eq. (7) in the submitted manuscript. Before introducing NWGM, we first revisit the calculation of a function $f(\mathcal{X})$'s expectation according to the distribution $P(\mathcal{X})$:

$$\mathbb{E}_{\mathbf{x}}[f(\mathbf{x})] = \sum_{\mathbf{x}} f(\mathbf{x})P(\mathbf{x}), \quad (12)$$

which is the weighted arithmetic mean of $f(\mathbf{x})$ with $P(\mathbf{x})$ as the weights. Correspondingly, the weighted geometric mean (WGM) of $f(\mathbf{x})$ with $P(\mathbf{x})$ as the weights is:

$$\text{WGM}(f(\mathbf{x})) = \prod_{\mathbf{x}} f(\mathbf{x})^{P(\mathbf{x})}, \quad (13)$$

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

$$\begin{aligned} \text{WGM}(f(\mathbf{x})) &= \prod_{\mathbf{x}} f(\mathbf{x})^{P(\mathbf{x})} \\ &= \prod_{\mathbf{x}} \exp[g(\mathbf{x})]^{P(\mathbf{x})} = \prod_{\mathbf{x}} \exp[g(\mathbf{x})P(\mathbf{x})] \\ &= \exp[\sum_{\mathbf{x}} g(\mathbf{x})P(\mathbf{x})] = \exp\{\mathbb{E}_{\mathbf{x}}[g(\mathbf{x})]\}, \end{aligned} \quad (14)$$

where the expectation $\mathbb{E}_{\mathbf{x}}$ is absorbed into the exponential term. Based on this observation, researchers approximate the expectation of a function by the WGM of this function in the deep network whose last layer is a Softmax layer [50,46,6]:

$$\mathbb{E}_{\mathbf{x}}[f(\mathbf{x})] \approx \text{WGM}(f(\mathbf{x})) = \exp\{\mathbb{E}_{\mathbf{x}}[g(\mathbf{x})]\}, \quad (15)$$

where $f(\mathbf{x}) = \exp[g(\mathbf{x})]$.

In our case, we parameterize $p(\mathcal{L}|\mathbf{s}, \mathbf{x}, \mathbf{z})$ in Eq. (8) by a network with a Softmax layer as the last layer:

$$P(\mathcal{L}|\mathbf{s}, \mathbf{x}, \mathbf{z}) = \text{Softmax}[g(\mathbf{s}, \mathbf{x}, \mathbf{z})] \propto \exp[g(\mathbf{s}, \mathbf{x}, \mathbf{z})]. \quad (16)$$

We follow Eq. (8) and (15) to get:

$$\begin{aligned} P(\mathcal{L}|do(\mathcal{I})) &= \mathbb{E}_{\mathbf{s}} \mathbb{E}_{\mathbf{x}} \mathbb{E}_{[\mathbf{z}|\mathcal{I}]} [P(\mathcal{L}|\mathbf{s}, \mathbf{x}, \mathbf{z})] \\ &\approx \text{WGM}(P(\mathcal{L}|\mathbf{s}, \mathbf{x}, \mathbf{z})) \approx \exp\{\mathbb{E}_{\mathbf{s}} \mathbb{E}_{\mathbf{x}} \mathbb{E}_{[\mathbf{z}|\mathcal{I}]} [g(\mathbf{s}, \mathbf{x}, \mathbf{z})]\}. \end{aligned} \quad (17)$$

Note that, as in Eq. (16), $P(\mathcal{L}|\mathbf{s}, \mathbf{x}, \mathbf{z})$ is only proportional to $\exp[g(\mathbf{s}, \mathbf{x}, \mathbf{z})]$ instead of strictly equalling to, we only have $\text{WGM}(P(\mathcal{L}|\mathbf{s}, \mathbf{x}, \mathbf{z})) \approx \exp\{\mathbb{E}_{\mathbf{s}} \mathbb{E}_{\mathbf{x}}$

$\mathbb{E}_{[\mathbf{z}|\mathcal{I}]}[g(\mathbf{s}, \mathbf{x}, \mathbf{z})]\}$ in Eq. (17) instead of equaling to. Furthermore, to guarantee the sum of $P(\mathcal{L}|do(\mathcal{I}))$ to be 1, we use a Softmax layer to normalize these exponential units:

$$P(\mathcal{L}|do(\mathcal{I})) \approx \text{Softmax}\{\mathbb{E}_{\mathbf{s}}\mathbb{E}_{\mathbf{x}}\mathbb{E}_{[\mathbf{z}|\mathcal{I}]}[g(\mathbf{s}, \mathbf{x}, \mathbf{z})]\}, \quad (18)$$

which is Eq. (7) in the submitted manuscript. Since the Softmax layer normalizes these exponential terms, this is called the normalized weighted geometric mean (NWGM) approximation. In addition, if $g(\cdot)$ is a fully connected layer, we have:

$$P(\mathcal{L}|do(\mathcal{I})) \approx \text{Softmax}\{g(\mathbb{E}_{\mathbf{s}}[\mathbf{s}], \mathbb{E}_{\mathbf{x}}[\mathbf{x}], \mathbb{E}_{[\mathbf{z}|\mathcal{I}]}[\mathbf{z}])\}. \quad (19)$$

In the same vein, we can use NWGM to word level distribution Eq. (11) and get:

$$\begin{aligned} & P(\mathcal{L}|do(\mathcal{I}), \mathbf{h}) \\ & \approx \text{Softmax}\{g(\mathbb{E}_{[\mathbf{s}|\mathbf{h}]}[\mathbf{s}], \mathbb{E}_{[\mathbf{x}|\mathbf{h}]}[\mathbf{x}], \mathbb{E}_{[\mathbf{z}|\mathcal{I}, \mathbf{h}]}[\mathbf{z}])\}. \end{aligned} \quad (20)$$

Then we can use EXPT modules introduced in Section 4.2 (Eq. (10)) of the submitted manuscript to compute these expectations. As discussed in the end of Section 1.5, we set \mathcal{S} to be conditioned on \mathbf{h} to increase the representation power. If not, we find that $\mathbb{E}_{[\mathbf{s}]}[\mathbf{s}]$ will be the same value all the time during the caption generation.

2 Network Architecture

In this section, we will detail the network architectures of UD-DICv1.0 and AoA-DICv1.0 proposed in Section 4.4 of the submitted manuscript.

2.1 EXPT Module

In Eq. (10) of the submitted manuscript, we show how EXPT module works. The detail structure of this module is listed in Table 2.

2.2 ATT Module

Two ATT modules are respectively deployed in UD-DICv1.0 and AoA-DICv1.0, which are Top-Down Attention and Attention on Attention. The details of two modules are given in Table 3 and Table 4, respectively. Self attention in Table 4(c) is computed as follows:

$$\begin{aligned} & \text{Input: } \mathcal{I}, \mathbf{h} \\ & \text{Head: } \mathbf{head}_i = \text{Softmax}\left(\frac{\mathbf{h}\mathbf{W}_i^1(\mathcal{I}\mathbf{W}_i^2)^T}{\sqrt{d_k}}\right)\mathcal{I}\mathbf{W}_i^3, \\ & \text{Multihead: } \mathcal{M} = \text{Concat}(\mathbf{head}_1, \dots, \mathbf{head}_s)\mathbf{W}_C, \\ & \text{Output: } \hat{\mathbf{x}} = \text{LeakyReLU}(\text{MLP}(\mathcal{M})), \end{aligned} \quad (21)$$

2.3 Common Structure of the Decoder

The common structure of the two decoders of UD-DICv1.0 and AoA-DICv1.0 is given in Table 5. When UD-DICv1.0 or AoA-Dicv1.0 is implemented, ATT module in (14) is Top-Down attention (Table 3) or Attention on Attention (Table 4), and $g(\cdot)$ in (18) is an LSTM layer or a GLU layer.

2.4 Implementation Details

In the beginning, we trained both the models by the cross-entropy loss 35 epochs:

$$L_{XE} = -\log P(\mathcal{L}^*|do(\mathcal{I})), \quad (22)$$

where \mathcal{L}^* denotes the ground-truth caption. After that, we used RL-based loss to train both models another 65 epochs:

$$L_{RL} = -\mathbb{E}_{\mathcal{L}^s \sim P(\mathcal{L}^s|do(\mathcal{I}))}[r(\mathcal{L}^s; \mathcal{L}^*)], \quad (23)$$

where r is a sentence-level metric between the sampled sentence \mathcal{L}^s and the ground-truth \mathcal{L}^* , *e.g.*, the CIDEr-D [48] metric. We used Adam optimizer [22] to train both models and the learning rate was initialized to $5e^{-4}$ and was decayed by 0.8 for every 5 epochs. Importantly, the learning rate of all the EXPT modules were set 10 times smaller than the other layers. The batch size in UD-DICv1.0 and AoA-DICv1.0 were set to 100 and 10, respectively.

3 More Results

We show more quantitative results and qualitative examples in this section.

3.1 More Quantitative Results

We report the performances of our DIC models and the compared state-of-the-art image captioning models trained by cross entropy loss (Eq. (22)) in Table 1. We can find that both UD-DICv1.0 and AoA-DICv1.0 have higher CIDEr-D scores than the original Up-Down and AoANet. Particular, our AoA-DICv1.0 achieve the highest CIDEr-D scores compared with the other state-of-the-art models.

3.2 More Qualitative Examples

Fig. 3 shows more comparisons between captions generated by UD, UD-BD, and UD-DICv1.0. It can be find that compared with UD and UD-BD, our UD-DICv1.0 generate more consistent captions, which demonstrates that our UD-DICv1.0 commits less bias.

Table 1: The performances of various methods on MS-COCO Karpathy split trained by cross-entropy loss.

Models	B@4	M	R	C	S
Up-Down [3]	36.2	27.0	56.4	113.5	20.3
Up-Down [†] [3]	36.5	27.1	56.7	114.1	20.3
RFNet [19]	37.0	27.9	57.3	116.3	20.8
LBPF [39]	37.4	28.1	57.5	116.4	21.2
CNM [52]	37.1	27.9	57.3	116.6	20.8
GCN-LSTM [53]	36.8	27.9	57.0	116.3	20.9
SGAE [51]	36.9	27.7	57.2	116.7	20.9
UD-HIP [54]	37.0	28.1	57.1	116.6	21.2
AoANet [17]	37.2	28.4	57.5	119.8	21.3
AoANet [†] [17]	36.6	28.1	57.0	116.9	20.5
UD-DICv1.0	37.0	28.2	57.2	117.1	21.0
AoA-DICv1.0	37.4	28.3	57.4	120.1	21.5

Table 2: The details of EXPT module.

Index	Input	Operation	Output	Trainable Parameters
(1)	context vector	-	\mathbf{h} (1,000)	-
(2)	Dictionary \mathbf{X}	-	\mathbf{X} (1,000 \times 10,000)	-
(3)	(1),(2)	inner product $\mathbf{X}^T \mathbf{h}$	\mathbf{p} (10,000)	\mathbf{X} (1,000 \times 10,000)
(4)	(3)	Softmax	P (10,000)	-
(5)	(4)	weighted sum $\mathbf{X}P$	$\bar{\mathbf{x}}$ (1,000)	-

4 Details of Human Evaluations

When we deployed human evaluation, we invited 20 humans to ask them to sort the captions according to the consistencies with the given images. When these humans considered that two captions are similar, they will sort them with the same rank. After that, we pairwise compared the sort results to compute the pie chart shown in Figure 7 of the submitted manuscript. Fig. 4 shows one example of the interface of our human evaluation.

Table 3: The details of Top-Down Attention.

Index	Input	Operation	Output	Trainable Parameters
(1)	-	feature set	\mathcal{I} ($1,000 \times M$)	-
(2)	-	context vector	\mathbf{h} (1,000)	-
(3)	(1),(2)	attention weights $\mathbf{w}_a \tanh(\mathbf{W}_v \mathbf{x}_m + \mathbf{W}_h \mathbf{h})$	$\boldsymbol{\alpha}$ (M)	\mathbf{w}_a (512), \mathbf{W}_v ($512 \times 1,000$) \mathbf{W}_h ($512 \times 1,000$)
(4)	(3)	Softmax	$\boldsymbol{\alpha}$ (M)	-
(5)	(1),(4)	weighted sum $\mathcal{I}\boldsymbol{\alpha}$	$\hat{\mathbf{x}}$ (1,000)	-

Table 4: The details of Attention on Attention.

Index	Input	Operation	Output	Trainable Parameters
(1)	-	feature set	\mathcal{I} ($1,000 \times M$)	-
(2)	-	context vector	\mathbf{h} (1,000)	-
(3)	(1),(2)	self attention	$\hat{\mathbf{x}}$ (1,000)	-
(4)	(1),(2),(3)	information vector $\mathbf{W}_q^i \mathbf{h} + \mathbf{W}_x^i \hat{\mathbf{x}} + \mathbf{b}^i$	\mathbf{v}_i (1,000)	\mathbf{W}_q^i ($1,000 \times 1,000$) \mathbf{W}_x^i ($1,000 \times 1,000$), \mathbf{b}^i (1000)
(5)	(1),(2),(3)	attention gate $\sigma(\mathbf{W}_g^i \mathbf{h} + \mathbf{W}_x^i \hat{\mathbf{x}} + \mathbf{b}^g)$	\mathbf{v}_g (1,000)	\mathbf{W}_g^i ($1,000 \times 1,000$) \mathbf{W}_x^i ($1,000 \times 1,000$), \mathbf{b}^g (1000)
(6)	(4),(5)	element-wise multiplication $\mathbf{v}_i \cdot * \mathbf{v}_g$	$\hat{\mathbf{v}}$ (1,000)	-

Table 5: The details of the common structure of the two decoders.

Index	Input	Operation	Output	Trainable Parameters
(1)	-	word label	\mathbf{w}_{t-1} (10,369)	-
(2)	-	word generator's output at $t-1$	\mathbf{o}^{t-1} (1,000)	-
(3)	-	image feature set \mathcal{I}	\mathcal{I} ($1,000 \times M$)	-
(4)	-	Dictionary \mathbf{Z}	\mathcal{I} ($1,000 \times 9,590$)	-
(5)	-	Dictionary \mathbf{S}	\mathcal{I} ($1,000 \times 1,342$)	-
(6)	-	Dictionary \mathbf{X}	\mathcal{I} ($1,000 \times 10,000$)	-
(7)	(1)	word embedding $\mathbf{W}_\Sigma \mathbf{w}_{t-1}$	\mathbf{e}_{t-1} (1,000)	\mathbf{W}_Σ ($1,000 \times 10,369$)
(8)	(3)	mean pooling	\mathbf{i} (1,000)	-
(9)	(2),(7),(8)	concatenate	\mathbf{u}_t (3,000)	-
(10)	(9)	LSTM ₁ ($\mathbf{u}_t; \mathbf{h}_{t-1}^1$)	\mathbf{h}_t^1 (1,000)	LSTM ₁ (3,000 \rightarrow 1,000)
(11)	(9)	LSTM ₂ ($\mathbf{u}_t; \mathbf{h}_{t-1}^2$)	\mathbf{h}_t^2 (1,000)	LSTM ₂ (3,000 \rightarrow 1,000)
(12)	(9)	LSTM ₃ ($\mathbf{u}_t; \mathbf{h}_{t-1}^3$)	\mathbf{h}_t^3 (1,000)	LSTM ₃ (3,000 \rightarrow 1,000)
(13)	(9)	LSTM ₄ ($\mathbf{u}_t; \mathbf{h}_{t-1}^4$)	\mathbf{h}_t^4 (1,000)	LSTM ₄ (3,000 \rightarrow 1,000)
(14)	(3),(10)	ATT \mathcal{I}	$\hat{\mathbf{x}}$ (1,000)	ATT
(15)	(4),(11),(14)	EXPT [\mathbf{z}]	$\bar{\mathbf{z}}$ (1,000)	EXPT
(16)	(5),(12)	EXPT [\mathbf{s}]	$\bar{\mathbf{s}}$ (1,000)	EXPT
(17)	(6),(13)	EXPT [\mathbf{x}]	$\bar{\mathbf{x}}$ (1,000)	EXPT
(18)	(14),(15),(16),(17)	$g(\hat{\mathbf{x}}, \bar{\mathbf{z}}, \bar{\mathbf{s}}, \bar{\mathbf{x}})$	\mathbf{o}_t (10,369)	$g(\cdot)$
(19)	(18)	Softmax	P_t (10,369)	-

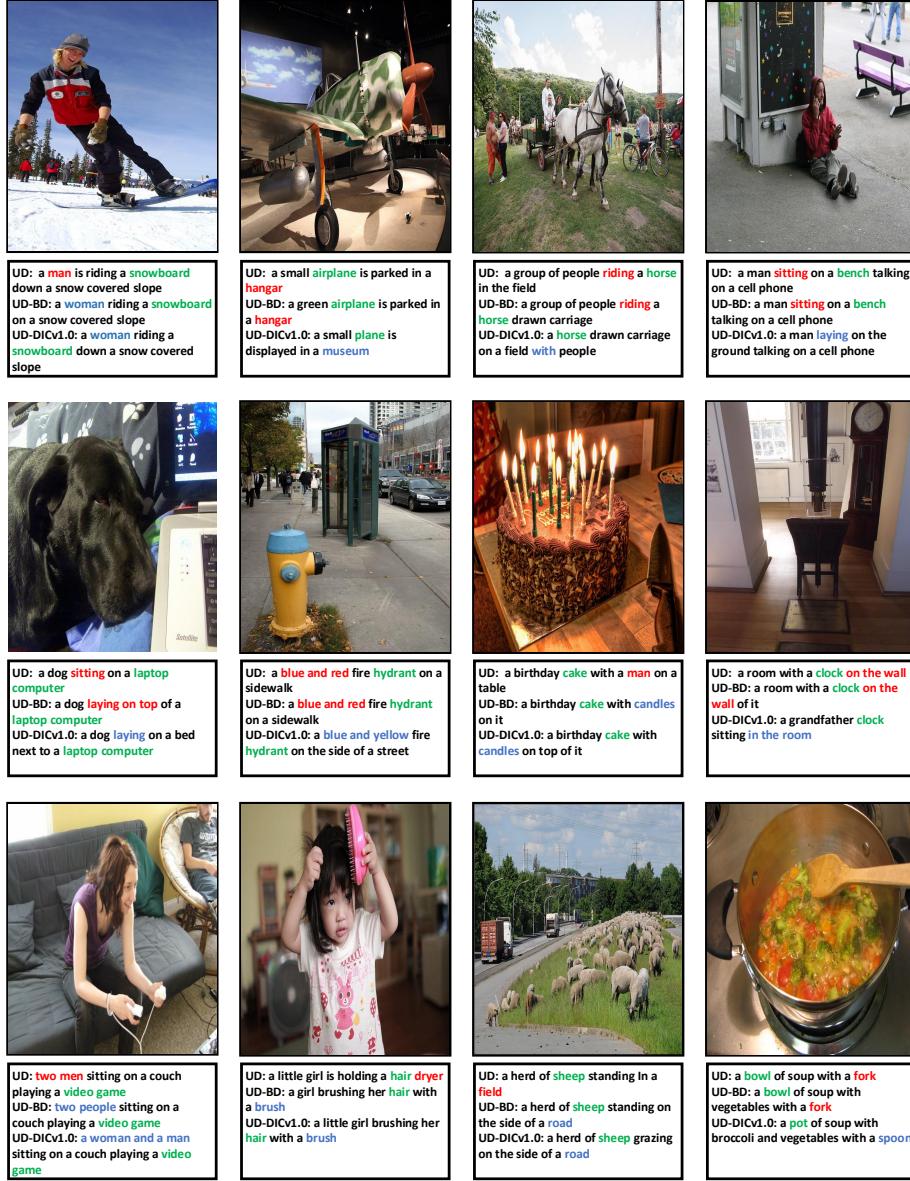


Fig. 3: Some examples show that our UD-DICv1.0 generates the most consistent captions. The visual concepts which may cause bias are colored by green. The red and blue words represent the inconsistent and consistent words, respectively.

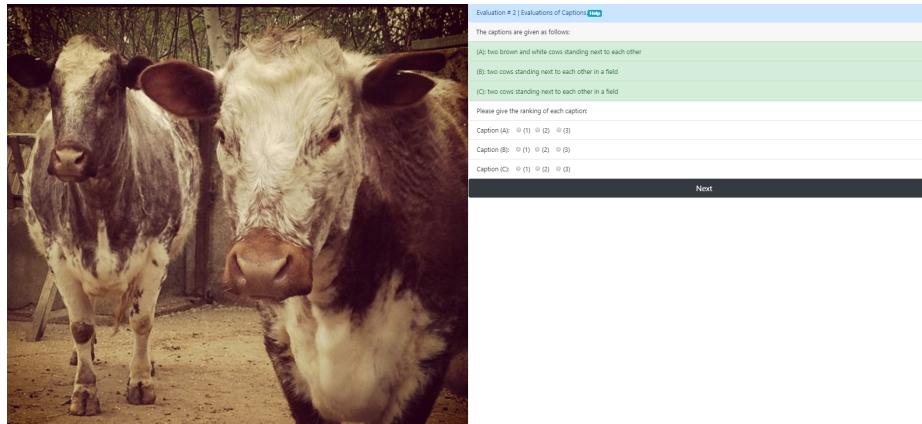


Fig. 4: The evaluation interface for comparing captions generated by different models.

References

1. Agrawal, A., Batra, D., Parikh, D., Kembhavi, A.: Don't just assume; look and answer: Overcoming priors for visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4971–4980 (2018)
2. Anderson, P., Fernando, B., Johnson, M., Gould, S.: Spice: Semantic propositional image caption evaluation. In: European Conference on Computer Vision. pp. 382–398. Springer (2016)
3. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and visual question answering. In: CVPR (2018)
4. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., Parikh, D.: Vqa: Visual question answering. In: Proceedings of the IEEE international conference on computer vision. pp. 2425–2433 (2015)
5. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
6. Baldi, P., Sadowski, P.: The dropout learning algorithm. Artificial intelligence **210**, 78–122 (2014)
7. Banerjee, S., Lavie, A.: Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In: Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization. pp. 65–72 (2005)
8. Bolukbasi, T., Chang, K.W., Zou, J.Y., Saligrama, V., Kalai, A.T.: Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In: Advances in neural information processing systems. pp. 4349–4357 (2016)
9. Cadene, R., Dancette, C., Ben-younes, H., Cord, M., Parikh, D.: Rubi: Reducing unimodal biases in visual question answering. arXiv preprint arXiv:1906.10169 (2019)
10. Chen, X., Fang, H., Lin, T.Y., Vedantam, R., Gupta, S., Dollár, P., Zitnick, C.L.: Microsoft coco captions: Data collection and evaluation server. arXiv preprint arXiv:1504.00325 (2015)
11. Das, A., Kottur, S., Gupta, K., Singh, A., Yadav, D., Moura, J.M., Parikh, D., Batra, D.: Visual dialog. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 326–335 (2017)
12. Dauphin, Y.N., Fan, A., Auli, M., Grangier, D.: Language modeling with gated convolutional networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 933–941. JMLR. org (2017)
13. Girshick, R., Radosavovic, I., Gkioxari, G., Dollár, P., He, K.: Detectron. <https://github.com/facebookresearch/detectron> (2018)
14. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6904–6913 (2017)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
16. Hendricks, L.A., Burns, K., Saenko, K., Darrell, T., Rohrbach, A.: Women also snowboard: Overcoming bias in captioning models. In: European Conference on Computer Vision. pp. 793–811. Springer (2018)

17. Huang, L., Wang, W., Chen, J., Wei, X.Y.: Attention on attention for image captioning. In: International Conference on Computer Vision (2019)
18. Hudson, D.A., Manning, C.D.: Gqa: A new dataset for real-world visual reasoning and compositional question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6700–6709 (2019)
19. Jiang, W., Ma, L., Jiang, Y.G., Liu, W., Zhang, T.: Recurrent fusion network for image captioning. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 499–515 (2018)
20. Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C.L., Girshick, R.: Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In: Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on. pp. 1988–1997. IEEE (2017)
21. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3128–3137 (2015)
22. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
23. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., et al.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. International Journal of Computer Vision **123**(1), 32–73 (2017)
24. Kulkarni, G., Premraj, V., Ordonez, V., Dhar, S., Li, S., Choi, Y., Berg, A.C., Berg, T.L.: Babytalk: Understanding and generating simple image descriptions. In: CVPR (2011)
25. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. Text Summarization Branches Out (2004)
26. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
27. Liu, D., Zha, Z.J., Zhang, H., Zhang, Y., Wu, F.: Context-aware visual policy network for sequence-level image captioning. In: 2018 ACM Multimedia Conference on Multimedia Conference. pp. 1416–1424. ACM (2018)
28. Liu, H., Singh, P.: Conceptnet-a practical commonsense reasoning tool-kit. BT technology journal **22**(4), 211–226 (2004)
29. Lu, J., Batra, D., Parikh, D., Lee, S.: Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. arXiv preprint arXiv:1908.02265 (2019)
30. Lu, J., Xiong, C., Parikh, D., Socher, R.: Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). vol. 6, p. 2 (2017)
31. Lu, J., Yang, J., Batra, D., Parikh, D.: Neural baby talk. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
32. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online dictionary learning for sparse coding. In: Proceedings of the 26th annual international conference on machine learning. pp. 689–696. ACM (2009)
33. Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A.L., Murphy, K.: Generation and comprehension of unambiguous object descriptions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 11–20 (2016)
34. Misra, I., Lawrence Zitnick, C., Mitchell, M., Girshick, R.: Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels. In: Pro-

- ceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2930–2939 (2016)
35. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting on association for computational linguistics. pp. 311–318. Association for Computational Linguistics (2002)
 36. Pearl, J.: Causality: models, reasoning and inference, vol. 29. Springer (2000)
 37. Pearl, J., Glymour, M., Jewell, N.P.: Causal inference in statistics: A primer. John Wiley & Sons (2016)
 38. Pearl, J., Mackenzie, D.: The Book of Why. Basic Books, New York (2018)
 39. Qin, Y., Du, J., Zhang, Y., Lu, H.: Look back and predict forward in image captioning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8367–8375 (2019)
 40. Ranzato, M., Chopra, S., Auli, M., Zaremba, W.: Sequence level training with recurrent neural networks (2015)
 41. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
 42. Rennie, S.J., Marcheret, E., Mroueh, Y., Ross, J., Goel, V.: Self-critical sequence training for image captioning. In: CVPR. vol. 1, p. 3 (2017)
 43. Rohrbach, A., Hendricks, L.A., Burns, K., Darrell, T., Saenko, K.: Object hallucination in image captioning. arXiv preprint arXiv:1809.02156 (2018)
 44. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision **115**(3), 211–252 (2015)
 45. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2556–2565 (2018)
 46. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research **15**(1), 1929–1958 (2014)
 47. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in neural information processing systems. pp. 3104–3112 (2014)
 48. Vedantam, R., Lawrence Zitnick, C., Parikh, D.: Cider: Consensus-based image description evaluation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4566–4575 (2015)
 49. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: A neural image caption generator. In: CVPR (2015)
 50. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: International conference on machine learning. pp. 2048–2057 (2015)
 51. Yang, X., Tang, K., Zhang, H., Cai, J.: Auto-encoding scene graphs for image captioning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10685–10694 (2019)
 52. Yang, X., Zhang, H., Cai, J.: Learning to collocate neural modules for image captioning. arXiv preprint arXiv:1904.08608 (2019)
 53. Yao, T., Pan, Y., Li, Y., Mei, T.: Exploring visual relationship for image captioning. In: Computer Vision–ECCV 2018, pp. 711–727. Springer (2018)

54. Yao, T., Pan, Y., Li, Y., Mei, T.: Hierarchy parsing for image captioning. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2621–2629 (2019)
55. Yao, T., Pan, Y., Li, Y., Qiu, Z., Mei, T.: Boosting image captioning with attributes. In: IEEE International Conference on Computer Vision, ICCV. pp. 22–29 (2017)
56. You, Q., Jin, H., Wang, Z., Fang, C., Luo, J.: Image captioning with semantic attention. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4651–4659 (2016)
57. Zha, Z.J., Liu, D., Zhang, H., Zhang, Y., Wu, F.: Context-aware visual policy network for fine-grained image captioning. IEEE transactions on pattern analysis and machine intelligence (2019)
58. Zhou, L., Palangi, H., Zhang, L., Hu, H., Corso, J.J., Gao, J.: Unified vision-language pre-training for image captioning and vqa. arXiv preprint arXiv:1909.11059 (2019)