Learning to Collocate Neural Modules for Image Captioning

Xu Yang[1], Hanwang Zhang[1], Jianfei Cai[1,2]
[1]School of Computer Science and Engineering, Nanyang Technological University, Singapore,
[2]Faculty of Information Technology, Monash University, Australia,
s170018@e.ntu.edu.sg,{hanwangzhang@,ASJFCai@}ntu.edu.sg

Abstract

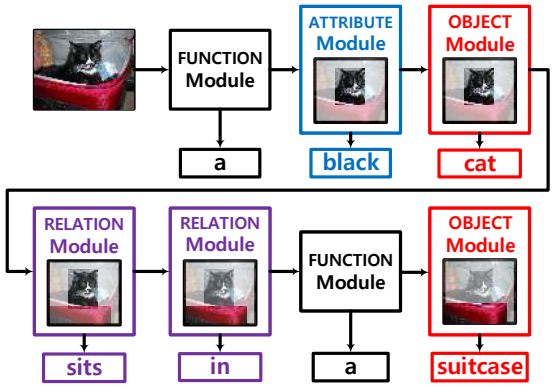
We do not speak word by word from scratch; our brain quickly structures a pattern like STH DO STH AT SOMEPLACE *and then fills in the detailed descriptions. To render existing encoder-decoder image captioners such human-like reasoning, we propose a novel framework: learning to Collocate Neural Modules (CNM), to generate the "inner pattern" connecting visual encoder and language decoder. Unlike the widely-used neural module networks in visual Q&A, where the language (*i.e.*, question) is fully observable, CNM for captioning is more challenging as the language is being generated and thus is partially observable. To this end, we make the following technical contributions for CNM training: 1) compact module design — one for function words and three for visual content words (*e.g.*, noun, adjective, and verb), 2) soft module fusion and multi-step module execution, robustifying the visual reasoning in partial observation, 3) a linguistic loss for module controller being faithful to part-of-speech collocations (*e.g.*, adjective is before noun). Extensive experiments on the challenging MS-COCO image captioning benchmark validate the effectiveness of our CNM image captioner. In particular, CNM achieves a new state-of-the-art 127.9 CIDEr-D on Karpathy split and a single-model 126.0 c40 on the official server. CNM is also robust to few training samples,* e.g.*, by training only one sentence per image, CNM can halve the performance loss compared to a strong baseline.*

1. Introduction

Let's describe the three images in Figure 1a. Most of you will compose sentences varying vastly from image to image. In fact, the ability of using diverse language to describe the colorful visual world is a gift to humans, but a formidable challenge to machines. Although recent advances in visual representation learning [16, 42] and language modeling [18, 50] demonstrate the impressive power of modeling the diversity in their respective modalities, it is still far from being resolved to establish a robust cross-modal connection between them. Indeed, image captioning is not the only model that can easily exploit the dataset bias to captioning even without looking at the image [44], almost all existing models for visual reasoning tasks such as visual Q&A [23, 46, 48] have been spotted mode collapse to certain dataset idiosyncrasies and failed to reproduce the diversity of our world — the more complex the task is, the more severe the collapse will be, such as image paragraph generation [27], scene graph generation [5, 14], and visual dialog [7, 38]. For example, in MS-COCO [32] training set, as the co-occurrence chance of "man" and "standing"

(a) Three diverse images.

(b) Three captions with the same sentence pattern.



(c) The caption generation process of CNM.

Figure 1: The motivation of the proposed learning to Collocate Neural Modules (CNM) for image captioning: neural module collocation imitates the inductive bias — sentence pattern, which regularizes the diverse training effectively.

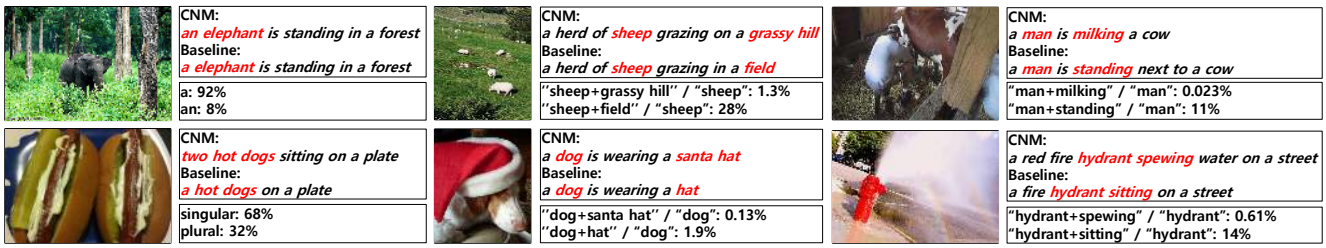


Figure 2: By comparing our CNM with a non-module baseline (an upgraded version of Up-Down [2]), we have three interesting findings in tackling the dataset bias: (a) more accurate grammar. % denotes the frequency of a certain pattern in MS-COCO, (b) more descriptive attributes, and (c) more accurate object interactions. The ratio ./. denotes the percentage of co-occurrence, *e.g.*, "sheep+field"/"sheep" = 28% means that "sheep" and "field" contributes the 28% occurrences of "sheep". We can see that CNM outperforms the baseline even with highly biased training samples.

is 11% high, a state-of-the-art captioner [2] is very likely to generate "man standing", regardless of their actual relationships such as "milking", which is 0.023% rare. We will discuss more biased examples in Figure 2 later.

Alas, unlike a visual concept in ImageNet which has 650 training images on average [8], a specific sentence in MS-COCO has *only one single* image [32], which is extremely scarce in the conventional view of supervised training. However, it is more than enough for us humans — anyone with normal vision (analogous to pre-trained CNN encoder) and language skills (analogous to pre-trained language decoder) does NOT need any training samples to perform captioning. Therefore, even though substantial progress has been made in the past 5 years since Show&Tell [52], there is still a crucial step missing between vision and language in modern image captioners [2, 34, 35]. To see this, given a sentence pattern in Figure 1b, your descriptions for the three images in Figure 1a should be much more constrained. In fact, studies in cognitive science [15, 47] show that do us humans not speak an entire sentence word by word from scratch; instead, we compose a pattern first, then fill in the pattern with concepts, and we repeat this process until the whole sentence is finished. Thus, structuring such patterns is what our human "captioning system" practices every day, and should machines do so. Fortunately, as we expected, for the sentence pattern in Figure 1b, besides those three captions, we have thousands more in MS-COCO.

In this paper, we propose learning to Collocate Neural Modules (CNM) to fill the missing gap in image captioning, where the module collocation imitates the sentence pattern in language generation. As shown in Figure 1c, CNM first uses the FUNCTION module for generating function word "a", and then chooses the ATTRIBUTE module to describe the adjectives like "black" of the "cat", which will be generated by the OBJECT module for nouns, followed by RELATION module for verbs or relationships like "sits in". Therefore, the key of CNM is to learn a dynamic structure that is an inductive bias being faithful to language collocations.

Though using neural module networks is not new in vision-language tasks such as VQA [3], where the question is parsed into a module structure like COLOR(FIND('chair')) for "What color is the chair?"; for image captioning, the case is more challenging as only partially observed sentences are available during captioning and the module structure by parsing is no longer applicable. To this end, we develop the following techniques for effective and robust CNM training. 1) Inspired by the policy network design in partially observed environment reinforcement learning [9], at each generation time step, the output of the four modules will be the fused according to their soft attention, which is based on the current generation context. 2) We adopt multi-step reasoning, *i.e.*, stacking neural modules [19]. These two methods stabilize the CNM training greatly. 3) To further introduce expert knowledge, we impose a linguistic loss for the module soft attention, which should be faithful to part-of-speech collocations, *e.g.*, ATTRIBUTE module should generate words that are ADJ.

Before we delve into the technical details in Section 3, we would like to showcase the power of CNM in tackling the dataset bias in Figure 2. Compared to a strong non-module baseline [2], the observed benefits of CNM include: 1) more accurate grammar like less 'a/an' error and 'singular/plural' error (Figure 2a), thanks to the joint reasoning of FUNCTION and OBJECT module, 2) more descriptive attributes (Figure 2b) due to ATTRIBUTE module, and 3) more accurate interactions (Figure 2c) due to RELATION module. Moreover, we find that when only 1 training sentence of each image is provided, our CNM will suffer less performance deterioration compared with the strong baseline. Extensive discussions and human evaluations are offered in Section 4.2, where we validate the effectiveness of CNM on the challenging MS-COCO image captioning benchmark. Overall, we achieve 127.9 CIDEr-D score on Karpathy split and a single model 126.0 c40 on the official server.

Our contributions are summarized as follows:

- Our CNM is the first module networks for image captioning. This enriches the spectrum of using neural modules

for vision-language tasks.

- We develop several techniques for effective module collocation training in partially observed sentences.
- Experiment results show that significant improvement can be made by using neural modules. CNM is a generic framework that supports potential improvement like more principled module and controller designs.

2. Related Work

Image Captioning. Most early image captioners are template-based models that they first structure sentence patterns and then fill the words into these fixed patterns [29, 30, 37]. However, since the functions used for generating templates and for generating words are not jointly trained, the performances are limited. Compared with them, modern image captioners which achieve superior performances are attention based encoder-decoder methods [53, 52, 43, 6, 60, 34, 2, 55, 35, 36, 56, 41, 12, 13]. However, unlike the template based models, most of the encoder-decoder based models generate word one by one without structure. Our CNM makes full use of the advantages of both template and encoder-decoder based image captioners which can generate captions by structuring patterns and end-to-end training. In particular, from the perspective of module network, several recent works can be reduced to special cases of our CNM. For example, Up-Down [2] only adopts OBJECT module, [17] classifies all the words as visual related (non-FUNCTION module) or unrelated (FUNCTION module), and [10, 45, 57] predict semantic words like object categories (OBJECT module), objects' attributes (ATTRIBUTE module), and objects' actions (RELATION module) and then input these semantic words into the language decoder for captioning.

Neural Module Networks. Recently, the idea of decomposing the network into neural modules is popular in some vision-language tasks such as VQA [3, 20], visual grounding [33, 58], and visual reasoning [46]. In these tasks, high-quality module layout can be obtained by parsing the provided sentences like questions in VQA. Yet in image captioning, only partially observed sentences are available and the module structure by parsing is not applicable anymore. For addressing this challenge, we propose to dynamically collocate neural modules on-the-fly during captioning.

3. Learning to Collocate Neural Modules

Figure 3 shows the encoder-decoder structure of our learning to Collocate Neural Modules (CNM) model. The encoder contains a CNN and four neural modules to generate features for language decoding (cf. Section 3.1). Our decoder has a module controller that softly fuses these features into a single feature for further language decoding by the

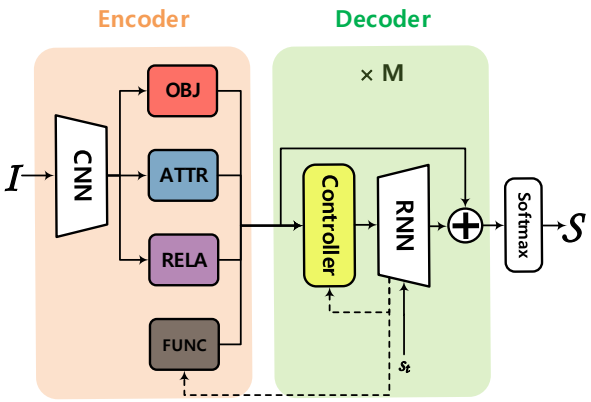


Figure 3: The encoder-decoder pipeline of our learning to Collocate Neural Modules (CNM) image captioner. The dash lines from RNN to FUNCTION module and the module controller mean that both of these sub-networks require the contextual knowledge of partially observed sentences. s_t is the word at t-th time step, which is input to the RNN.

followed RNN (cf. Section 3.2.1). Note that a linguistic loss is imposed for making the module controller more faithful to part-of-speech collocations (cf. Section 3.2.3). Besides the language generation, the RNN would also output the accumulated context of the partially observed sentence as the input to FUNCTION module and controller for linguistic information, which is helpful for these grammar-related modules. For multi-step reasoning, the entire decoder of CNM will repeat this soft fusion and language decoding M times (cf. Section 3.2.2). The residual connections are also implemented for directly transferring knowledge from lower layers to higher ones.

3.1. Neural Modules

Four distinguishable and compact neural modules are designed based on different principles for predicting the orthogonal knowledge from the image, *e.g.*, OBJECT module focuses on the object categories while ATTRIBUTE module focuses on the visual attributes. **In this way, the visual reasoning can be robusified because the captions are generated from the appeared elements of the visual sence, not merely from the language context which is more likely overfitted to dataset bias**. For example, the more accurate description "bird-perch-tree" will be reduced to "bird-fly" without using RELATION module, due to the high co-occurrence of "bird" and "fly" in the dataset.

OBJECT Module. It is designed to transform the CNN features to a feature set $\mathcal{V}_O$ containing the knowledge on object categories, *i.e.*, the feature set $\mathcal{V}_O$ facilitates the prediction of nouns like "person" or "dog". The input of this module is $\mathcal{R}_O$, which is an $N \times d_r$ feature set of N RoI features extracted by a ResNet-101 Faster R-CNN [42]. This ResNet

is pre-trained on object detection task by using the object annotations of VG dataset [28]. Formally, this module can be formulated as:

$$\begin{aligned} \textbf{Input:} \quad & \mathcal{R}_O, \\ \textbf{Output:} \quad & \mathcal{V}_O = \text{LeakyReLU}(\text{FC}(\mathcal{R}_O)), \end{aligned} \tag{1}$$

where $\mathcal{V}_O$ is the $N \times d_v$ output feature set.

ATTRIBUTE Module. It is designed to transform the CNN features to a feature set $\mathcal{V}_A$ on attribute knowledge, for generating adjectives like "black" and "dirty". The input of this module is an $N \times d_r$ feature set extracted by a ResNet-101 Faster R-CNN, and the network used here is pre-trained on attribute classification task by using the attribute annotations of VG dataset. Formally, this module can be written as:

$$\begin{aligned} \textbf{Input:} \quad & \mathcal{R}_A, \\ \textbf{Output:} \quad & \mathcal{V}_A = \text{LeakyReLU}(\text{FC}(\mathcal{R}_A)), \end{aligned} \tag{2}$$

where $\mathcal{V}_A$ is the $N \times d_v$ feature set output from this module.

RELATION Module. It transforms the CNN features to a feature set $\mathcal{V}_R$ representing potential interactions between two objects. This transferred feature set $\mathcal{V}_R$ would help to generate verbs like "ride", prepositions like "on", or quantifiers like "two". This module is built based on the multi-head self-attention mechanism [50], which automatically seeks the interactions among the input features. Here, we use $\mathcal{R}_O$ in Eq. (1) as the input because these kinds of features are widely applied as the input for successful relationship detection [61, 59]. This module is formulated as:

$$\begin{aligned} \textbf{Input:} \quad & \mathcal{R}_O, \\ \textbf{Multi-Head:} \quad & \mathcal{M} = \text{MultiHead}(\mathcal{R}_O), \\ \textbf{Output:} \quad & \mathcal{V}_R = \text{LeakyReLU}(\text{MLP}(\mathcal{M})), \end{aligned} \tag{3}$$

where MultiHead$(\cdot)$ means the multi-head self-attention mechanism, MLP$(\cdot)$ is a feed-forward network containing two fully connected layers with a ReLU activation layer in between [50], and $\mathcal{V}_R$ is the $N \times d_v$ feature set output from this module. Specifically, we use the following steps to compute the multi-head self-attention. We first use scaled dot-product to compute k self-attention head matrices as:

$$\textbf{head}_i = \text{Softmax}(\frac{\mathcal{R}_O \boldsymbol{W}_i^1 (\mathcal{R}_O \boldsymbol{W}_i^2)^T}{\sqrt{d_k}}) \mathcal{R}_O \boldsymbol{W}_i^3, \tag{4}$$

where $\boldsymbol{W}_i^1, \boldsymbol{W}_i^2, \boldsymbol{W}_i^3$ are all $d_r \times d_k$ trainable matrices, $d_k = d_r/k$ is the dimension of each head vector, and k is the number of head matrices. Then these k heads are concatenated and linearly projected to the final feature set $\mathcal{M}$:

$$\mathcal{M} = \text{Concat}(\textbf{head}_1, ..., \textbf{head}_k) \boldsymbol{W}_C, \tag{5}$$

where $\boldsymbol{W}_C$ is a $d_r \times d_r$ trainable matrix, $\mathcal{M}$ is the $N \times d_r$ feature set.

FUNCTION Module. It is designed to produce a single feature $\hat{\boldsymbol{v}}_F$ for generating function words like "a" or "and".

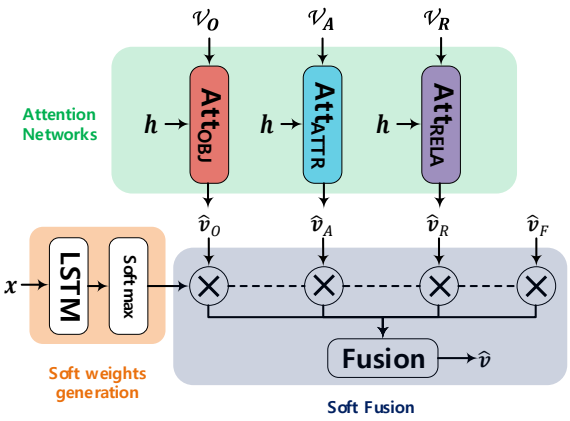


Figure 4: The detailed structure of our module controller. This controller will generate four soft weights by an LSTM for softly fusing attended features of four modules into a single fused feature $\hat{\boldsymbol{v}}$.

The input of this module is a d_c dimensional context vector $\boldsymbol{c}$ provided by the RNN, as the dashed line drawn in Figure 3. We use $\boldsymbol{c}$ as the input because it contains rich language context knowledge of the partially generated captions and such knowledge is suitable for generating function words, like "a" or "and", which require few visual knowledge. This module is formulated as:

$$\begin{aligned} \textbf{Input:} \quad & \boldsymbol{c}, \\ \textbf{Output:} \quad & \hat{\boldsymbol{v}}_F = \text{LeakyReLU}(\text{FC}(\boldsymbol{c})), \end{aligned} \tag{6}$$

where $\hat{\boldsymbol{v}}_F$ is the d_v dimensional output feature.

3.2. Controller

Figure 4 shows the detailed design of the module controller, which contains three attention networks, and one LSTM for soft weights generation. The output of this controller is a single fused feature vector $\hat{\boldsymbol{v}}$ which would be used for the next step reasoning by the followed RNN as in Figure 3. Next, we describe our module controller.

3.2.1 Soft Fusion

Yet, it is still an open question on how to define a complete set of neural modules for visual reasoning [58, 3]. However, we believe that a combination of simple neural modules can approximate to accomplish a variety of complex tasks [19]. Before the soft fusion, three additive attention networks are used to respectively transform feature sets output from three visual modules into three more informative features:

$$\begin{aligned} \textbf{Object Attention:} \quad & \hat{\boldsymbol{v}}_O = \text{Att}_{Obj}(\mathcal{V}_O, \boldsymbol{h}), \\ \textbf{Attribute Attention:} \quad & \hat{\boldsymbol{v}}_A = \text{Att}_{Attr}(\mathcal{V}_A, \boldsymbol{h}), \\ \textbf{Relation Attention:} \quad & \hat{\boldsymbol{v}}_R = \text{Att}_{Rela}(\mathcal{V}_R, \boldsymbol{h}), \end{aligned} \tag{7}$$

where $\hat{\boldsymbol{v}}_O$, $\hat{\boldsymbol{v}}_A$, and $\hat{\boldsymbol{v}}_R$ are the d_v dimensional transformed features of $\mathcal{V}_O$, $\mathcal{V}_A$, and $\mathcal{V}_R$ produced by three visual modules (cf. Section 3.1), respectively; $\boldsymbol{h}$ is the d_c dimensional query vector produced by an LSTM (specified in Section 3.3); and the three attention networks own the same structure as that in [2] while the parameters are not shared.

After getting the three transformed features, $\hat{\boldsymbol{v}}_O$, $\hat{\boldsymbol{v}}_A$, and $\hat{\boldsymbol{v}}_R$ from Eq. (7) and the output $\hat{\boldsymbol{v}}_F$ from FUNCTION module, the controller generates four soft weights for them. The process of generating soft weights is formulated as:

$$\begin{aligned} \textbf{Input:} \quad & \boldsymbol{x} = \text{Concat}(\hat{\boldsymbol{v}}_O, \hat{\boldsymbol{v}}_A, \hat{\boldsymbol{v}}_R, \boldsymbol{c}), \\ \textbf{Soft Vector:} \quad & \boldsymbol{w} = \text{Softmax}(\text{LSTM}(\boldsymbol{x})), \\ \textbf{Output:} \quad & \hat{\boldsymbol{v}} = \text{Concat}(w_O\hat{\boldsymbol{v}}_O, w_A\hat{\boldsymbol{v}}_A, w_R\hat{\boldsymbol{v}}_R, w_F\hat{\boldsymbol{v}}_F), \end{aligned} \tag{8}$$

where the input $\boldsymbol{x}$ is the concatenation of three visual embedding vectors and the context vector accumulated in the RNN used in Eq.(6); $\boldsymbol{w} = \{w_O, w_A, w_R, w_F\}$ is a four-dimensional soft attention vector; and the output vector $\hat{\boldsymbol{v}}$ will be fed into the RNN for the subsequent language decoding.

We use $\boldsymbol{x}$ for generating soft weights because both visual clues ($\hat{\boldsymbol{v}}_O, \hat{\boldsymbol{v}}_A, \hat{\boldsymbol{v}}_R$) and the language context knowledge $\boldsymbol{c}$ of partially generated captions are all indispensable for achieving satisfied module collocation. Also, since the layouts of modules at a new time step are highly related to the previous ones, an LSTM is applied here to accumulate such knowledge for generating new soft weights.

3.2.2 Multi-Step Reasoning

Different from many sentence-provided visual tasks like VQA where approximately perfect module layout can be parsed by the fully observed sentences, our module layout is still noisy because only partially observed sentences are available. To robustify the visual reasoning, we repeat the soft fusion and language decoding M times as in [50, 40, 25]. In this way, the generated captions are usually more relevant to the images by observing more visual clues. For example, as the experiment results shown in Section 4.2, when multi-step reasoning is implemented, more accurate quantifiers are generated because the visual patterns of the objects with the same category can be accumulated. In addition, residual connections (cf. Figure 3) are used for directly transferring knowledge from lower layers to higher ones when such knowledge is already sufficient for word generation.

3.2.3 Linguistic Loss

For ensuring each module to learn the orthogonal and non-trivial knowledge from the image, *e.g.*, OBJECT module focuses more on object categories instead of visual attributes, even it owns the same structure as ATTRIBUTE module, we design a linguistic loss which is imposed on the module controller for further distinguishing these neural modules.

We build this loss by extracting the words' lexical categories (*e.g.*, adjectives, nouns, or verbs) from ground-truth captions by the Part-Of-Speech Tagger tool [49]. According to these lexical categories, we assign each word a 4-dimensional one hot vector $\boldsymbol{w}^*$, indicating which module should be chosen for generating this word. In particular, we assign OBJECT module to nouns (NN like "bus"), ATTRIBUTE module to adjectives (ADJ like "green"), RELATION module to verbs (VB like "drive"), prepositions (PREP like "on") and quantifiers (CD like "three"), and FUNCTION module to the other words (CC like "and").

By providing these expert-guided module layout $\boldsymbol{w}^*$, the cross-entropy value between $\boldsymbol{w}^*$ and soft weights $\boldsymbol{w}$ in Eq.(8) is imposed to train the module controller:

$$L_{lin} = -\sum_{i=1}^{4} w_i^* \log w_i. \tag{9}$$

Note that this linguistic loss is imposed on all the M module controllers in the language decoder (cf. Section 3.2.2).

3.3. Training and Inference

By assembling the neural modules, module controller, ResNet-101 [16] as CNN, and the top-down LSTM [2] as RNN, our CNM image captioner can be trained end-to-end. More specifically, at time step t, the query vector $\boldsymbol{h}$ in Eq. (7) is the output of the first LSTM in the top-down structure at the same time step, and the context vector $\boldsymbol{c}$ in Eq. (6) and Eq. (8) is the output of the second LSTM in the top-down structure at time step $t-1$. The previous word is used as a part of the input to the first LSTM language decoder. The model architecture is detailed in Section "Network Architecture" of the supplementary material.

Given a ground-truth caption $\mathcal{S}^* = \{s^*_{1:T}\}$ with its extracted part-of-speech tags $\boldsymbol{w}^*$, we can end-to-end train our CNM by minimizing the linguistic loss proposed in Eq. (9) and the language loss between the generated captions and the ground-truth captions. Suppose that the probability of word s predicted by the language decoder of our CNM model is $P(s)$, we can define the language loss L_{lan} as the cross-entropy loss:

$$L_{lan} = L_{XE} = -\sum_{t=1}^{T} \log P(s_t^*), \tag{10}$$

or the negative reinforcement learning (RL) based reward [43]:

$$L_{lan} = L_{RL} = -\mathbb{E}_{s_t^s \sim P(s)}[r(s^s_{1:T}; s^*_{1:T})], \tag{11}$$

where r is a sentence-level metric for the sampled sentence $\mathcal{S}^s = \{s^s_{1:T}\}$ and the ground-truth $\mathcal{S}^* = \{s^*_{1:T}\}$, *e.g.*, the

CIDEr-D [51] metric. Given the linguistic loss and language loss, the total loss is:

$$L = L_{lan} + \lambda L_{lin}, \tag{12}$$

where λ is a trade-off weight. During inference stage, we adopt the beam search strategy [43] with a beam size of 5.

4. Experiments

4.1. Datasets, Settings, and Metrics

MS-COCO [32]. This dataset provides one official split: 82,783, 40,504 and 40,775 images for training, validation and test respectively. The 3rd-party Karpathy split [24] was also used for the off-line test, which has 113,287, 5,000, 5,000 images for training, validation and test respectively.

Visual Genome [28] (VG). We followed Up-Down [2] to use object and attribute annotations provided by this dataset to pre-train CNN. We filtered this noisy dataset by keeping the labels which appear more than $2,000$ times in the training set. After filtering, 305 objects and 103 attributes remain. Importantly, since some images co-exist in both VG and COCO, we also filtered out the annotations of VG which also appear in COCO test set.

Settings. The captions of COCO were addressed by the following steps: the texts were first tokenized on white spaces; all the letters were changed to lowercase; the words were removed if they appear less than 5 times; each caption was trimmed to a maximum of 16 words. At last, the vocabulary included $10,369$ words in all.

In Eq. (1), d_r and d_v were set to 2,048 and 1,000 respectively; and in Eq. (6), d_c was set to 1,000. The number of head vectors k in Eq. (5) was 8. At training time, Adam optimizer [26] was used. In addition, the learning rate was initialized to $5e^{-4}$ and was decayed by 0.8 for every 5 epochs. The cross-entropy loss Eq. (10) and the RL-based loss Eq. (11) were in turn used to train our CNM 35 epochs and 100 epochs respectively. The batch size was set to 100. In our experiments, we found that the performance is non-sensitive to λ in Eq. (12). By default, we set the trade-off weight $\lambda = 1$ and $\lambda = 0.5$ when the cross-entropy loss and RL-based loss were used as language loss, respectively.

Metrics. Five standard metrics were applied for evaluating the performances of the proposed method: CIDEr-D [51], BLEU [39], METEOR[4], ROUGE [31], and SPICE [1].

4.2. Ablative Studies

We conducted extensive ablations for CNM, including architecture and fewer training sentences.

Architecture. We will investigate the effectiveness of designed modules, soft module fusion, linguistic loss, and deeper decoder structure in terms of proposing research questions (**Q**) and empirical answers (**A**).

Q1: Will each module generate more accurate module-specific words, *e.g.*, will OBJECT module generate more accurate nouns? We deployed a single visual module as the encoder and the top-down attention LSTM [2] as the decoder. When OBJECT, ATTRIBUTE, and RELATION modules were used, the baselines are denoted as **Module/O**, **Module/A**, and **Module/R**, respectively. In particular, baseline Module/O is the upgraded version of Up-Down [2].

Q2: Will the qualities of the generated captions be improved when the modules are fused? We designed three strategies for fusing modules by using three kinds of fusion weights. Specifically, when we set all the fusion weights as 1, the baseline is called **Col/1**; when soft fusion weights were used, the baseline is called **Col/S**; and when Gumbel-Softmax layer [21] was used for hard selection, the baseline is called **Col/H**.

Q3: Will the expert knowledge of part-of-speech collocations provided by the linguistic loss benefit the model? We added the linguistic loss to baselines Col/H and Col/S to get baselines **Col/S+L** and **Col/H+L**, respectively. Noteworthy, linguistic loss can not be used to Col/1 since we do not need module controller here.

Q4: Will better captions be generated when a deeper language decoder is implemented? We stacked the language decoder of baseline Col/S+L M times to get baseline **CNM#M**. Also, we designed **Module/O#M** by stacking M times of the top-down LSTM of baseline Module/O to check whether the performances can be improved when only the deeper decoder is used.

Evaluation Metrics. For comprehensively validating the effectiveness of our CNM, we not only computed five standard metrics (cf. Section 4.1), but also conducted human evaluation and calculated the recalls of five part-of-speech words. Specifically, we invited 20 workers for human evaluation. We exhibited 100 images sampled from the test set for each worker and asked them to pairwise compare the captions generated from three models: Module/O, Col/S+L, and CNM#3. The captions are compared from two aspects: 1) **the fluency and descriptiveness of the generated captions** (the top three pie charts in Figure 5); 2) **the relevance of the generated captions to images** (the bottom three pie charts in Figure 5). For calculating the recalls of five part-of-speech words, we counted the ratio of the words in predicted captions to the words in ground-truth captions. Such results are reported in Table 2.

A1. From Table 2, we can observe that each single module prefers to generate more accurate module-specific words, *e.g.*, the recall of nouns generated by Module/O is much higher than Module/A. Such observation validates that each module can indeed learn the knowledge of the corresponding module-specific words.

A2. As shown in Table 1, when modules are fused, the performances can be improved. Also, by comparing Col/1,

Table 1: The performances of various methods on Karpathy split. The metrics: B@N, M, R, C, and S denote BLEU@N, METEOR, ROUGE-L, CIDEr-D, and SPICE, respectively.

Models	B@1	B@4	M	R	C	S
SCST [43]	–	34.2	26.7	55.7	114.0	–
StackCap [11]	78.6	36.1	27.4	–	120.4	–
Up-Down [2]	79.8	36.3	27.7	56.9	120.1	21.4
RFNet [22]	80.4	37.9	28.3	58.3	125.7	21.7
CAVP [60]	–	38.6	28.3	58.5	126.3	21.6
SGAE [54]	**80.8**	38.4	**28.4**	58.6	127.8	**22.1**
Module/O	79.6	37.5	27.7	57.5	123.1	21.0
Module/A	79.4	37.3	27.4	57.1	121.9	20.9
Module/R	79.7	37.9	27.8	57.8	123.8	21.2
Module/O#3	79.9	38.0	27.9	57.5	124.3	21.3
Col/1	80.2	38.2	27.9	58.1	125.3	21.3
Col/H	80.1	38.1	27.8	58.1	124.7	21.2
Col/H+L	80.2	38.3	27.9	58.4	125.4	21.4
Col/S	80.2	38.2	28.0	58.4	125.7	21.4
Col/S+L (CNM#1)	80.3	38.5	28.2	58.6	126.4	21.5
CNM#2	80.5	38.5	28.2	58.7	127.0	21.7
CNM#3	80.6	38.7	**28.4**	58.7	127.4	21.8
CNM#3+SGAE	**80.8**	**38.9**	**28.4**	**58.8**	**127.9**	22.0

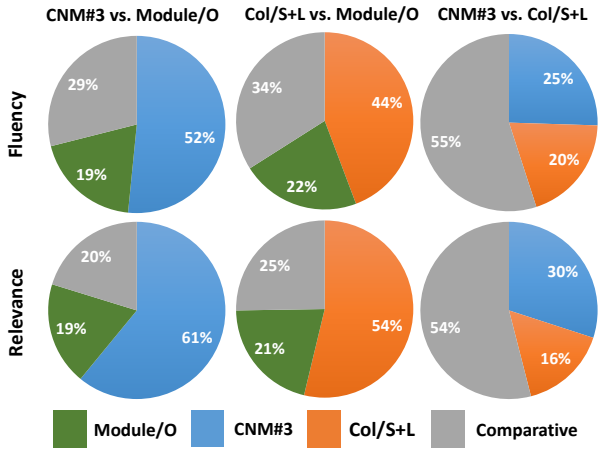


Figure 5: The pie charts each comparing the two methods in human evaluation.

Table 2: The recalls (%) of five part-of-speech words.

Models	nouns	adjectives	verbs	prepositions	quantifiers
Module/A	42.4	12.4	20.2	41.7	14.3
Module/O	44.5	11.5	21.8	42.6	17.1
Module/R	44.3	11.3	22.8	43.5	22.3
Col/S	45.2	13.1	23.1	43.6	24.1
Col/S+L	45.9	14.3	23.5	43.9	25.4
CNM#3	**47.3**	**16.1**	**24.3**	**44.8**	**30.5**

Col/S, and Col/H, we can find that Col/S achieves the highest performance. This is reasonable since compared with Col/1, Col/S can make word generation ground to the specific module. Compared with Col/H, Col/S can exploit more knowledge from all the modules when the modules are not correctly collocated.

A3. As shown in Table 1 and 2, we can find that the performances of Col/S+L are better than Col/S. Such observations validate that the expert supervision can indeed benefit the caption generation. In addition, from the results shown in Figure 5, we can find that when soft module fusion and linguistic loss are deployed, the generated captions have higher qualities evaluated by humans.

Table 3: The CIDEr-D loss (CIDEr-D) of using fewer training sentences.

X	5	4	3	2	1
CNM&X	0(127.4)	**0.4**(127.0)	**1.2**(126.2)	**2.3**(125.1)	**3.6**(123.8)
Module-O&X	0(123.1)	0.9(122.2)	2.3(120.8)	4.1(119.0)	6.8(116.3)

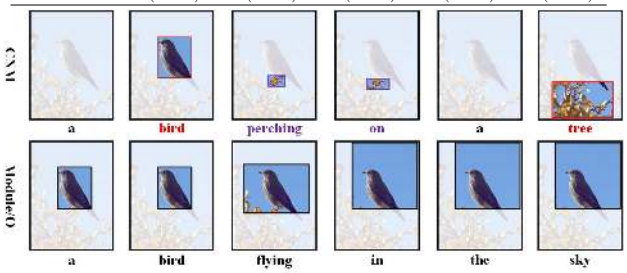


Figure 6: The visualizations of the caption generation process of CNM#3 and Module/O. Different colours refer to different modules, *i.e.*, red for OBJECT module, purple for RELATION module, and black for FUNCTION module. For simplicity, we only visualize the module layout generated by the last module controller of the deeper decoder.

A4. By inspecting the standard evaluation scores in Table 1, the recalls of words in Table 2, and the human evaluations in Figure 5, we can find that when a deeper decoder is used, *e.g.*, CNM#3 vs. CNM#1, the qualities of the generated captions can be improved. Also, by comparing Module/O#3 with CNM#3, we can find that only using a deeper decoder is not enough for generating high qualities captions.

Fewer Training Samples. To test the robustness of our CNM in the situation where only fewer training sentences are available (cf. Section 1), we randomly assigned X sentences among all the annotated captions to one image for training models CNM#3 and Module-O to get baselines **CNM&X** and **Module-O&X**. The results are reported in Table 3, where the values mean the losses of CIDEr-D compared with the model trained by all sentences, and the values in the bracket are the CIDEr-D scores.

Results and Analysis. From Table 3, we can find that both two models will be damaged if fewer training sentences are provided. Interestingly, we can observe that our CNM can halve the performance loss compared to Module/O. Such observations suggest that our CNM is more robust when fewer training samples are provided, compared with the traditional attention-based method.

4.3. Comparisons with State-of-The-Arts

Comparing Methods. Though various captioning models are developed in recent years, for fair comparisons, we only compared our CNM with some encoder-decoder methods due to their superior performances. Specifically, we

Table 4: The performances of various methods on MS-COCO Karpathy split trained by cross-entropy loss.

Models	B@1	B@4	M	R	C	S
SCST [43]	–	30.0	25.9	53.4	99.4	–
StackCap [11]	76.2	35.2	26.5	–	109.1	–
NBT [35]	75.5	34.7	27.1	–	108.9	20.1
Up-Down [2]	77.2	36.2	27.0	56.4	113.5	20.3
RFNet [22]	77.4	37.0	**27.9**	**57.3**	116.3	**20.8**
Col/S+L (CNM#1)	77.3	36.5	27.6	57.0	116.4	20.7
CNM#3	**77.6**	**37.1**	**27.9**	**57.3**	**116.6**	**20.8**

Table 5: The performances of various methods on the online MS-COCO test server.

Model	B@4		M		R-L		C-D	
Metric	c5	c40	c5	c40	c5	c40	c5	c40
SCST [43]	35.2	64.5	27.0	35.5	56.3	70.7	114.7	116.0
StackCap [11]	34.9	64.6	27.0	35.6	56.2	70.6	114.8	118.3
Up-Down [2]	36.9	68.5	27.6	36.7	57.1	72.4	117.9	120.5
CAVP [60]	37.9	69.0	28.1	37.0	58.2	73.1	121.6	123.8
SGAE [54]	37.8	68.7	28.1	37.0	58.2	73.1	122.7	125.5
CNM#3	37.9	68.4	28.1	36.9	58.3	72.9	123.0	125.3
CNM+SGAE	**38.4**	**69.3**	**28.2**	**37.2**	**58.4**	**73.4**	**123.8**	**126.0**

compared our method with **SCST** [43], **StackCap** [11], **Up-Down** [2], **NBT** [35], **CAVP** [60], **RFNet** [22], and **SGAE** [54]. Among these methods, Up-Down and NBT are specific cases of our CNM where only OBJECT modules are deployed. All of StackCap, CAVP, and RFNet use wider encoders or deeper decoders, while they do not design different modules. In addition, we also equipped our CNM a dictionary preserving language bias as in SGAE [54], and this model is denoted as **CNM+SGAE**.

Results. Table 4 and 1 show the performances of various methods trained by cross-entropy loss and RL-based loss, respectively. We can see that our single model CNM+SGAE in Table 1 achieves a new state-of-the-art CIDEr-D score. Specifically, by deploying four compact modules, soft module fusion strategy, and linguistic loss, our CNM can obviously outperform the models, *e.g.*, StackCap, CAVP, and RFNet, which also use deeper decoders or wider encoders. When the dictionary preserving language bias is learned as in SGAE, even the query embeddings do not contain high-level semantic knowledge created by graph convolution network as SGAE, our CNM+SGAE also achieve better performances than SGAE. From the results of the online test in Table 5, we can find that our single model has competitive performances and can achieve the highest CIDEr-D c40 score. In addition, Figure 6 shows the visualizations of the captioning process of our CNM and Module/O (the upgraded version of Up-Down). From this figure, we can observe that our CNM can generate more relevant description "bird perch" and less overfitted to dataset bias of high co-occurrence word combination "bird fly".

4.4. Limitations and Potentials

Though we design three techniques, *e.g.*, soft module fusion, linguistic loss, and multi-step reasoning for robustifying the module collocation, **improper module collocations** still exist since the sentence patterns are structured dynamically without a global "oracle". As a result, inaccurate description will be generated because of the improper module collocations. For example, as shown in Figure 7a top, at time step 4, RELATION module is chosen inaccurately and the verb "standing" is generated, while two cows have different actions; in Figure 7a bottom, at time step 3, it is more suitable to generate the noun "toilet", but FUNCTION module is chosen and inaccurate description "white and black cat" is generated. To tackle this limitation, more advanced techniques like Reinforcement Learning could be exploited for guiding the module collocations.

Figure 7: The limitations of our CNM model.

Another limitation of our CNM is **insufficient commonsense reasoning**. Specifically, many adjectives which require commonsense reasoning can hardly be generated by our model, *e.g.*, "rural", "rusty", or "narrow" are all commonsense adjectives. Figure 7b gives two examples, where the words "rusted" and "rural" cannot be generated. One possible solution is to design a REASON module where a memory network preserving the commonsense knowledge and then the context knowledge can be used as queries for reasoning. The model CNM+SGAE is one preliminary experiment designed for resolving such limitation. From Table 1, we can see that the performance indeed improves. This may shed some light on using more sophisticated modules and commonsense reasoning strategies.

5. Conclusions

We proposed to imitate the humans inductive bias — sentences are composed by structuring patterns — for image captioning. In particular, we presented a novel modular network method: learning to Collocate Neural Modules (CNM), which can generate captions by filling the contents into collocated modules. In this way, the caption generation is expected to be disentangled from dataset bias. We validated our CNM by extensive ablations and comparisons with state-of-the-art models on MS-COCO. In addition, we discussed the model limitations and thus the corresponding potentials are our future work.

Acknowledgements. This work is partially supported by NTU Data Science and Artificial Intelligence Research Center (DSAIR) and Alibaba-NTU JRI.

References

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

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, number 5, page 6, 2018. 2, 3, 5, 6, 7, 8

[3] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 39–48, 2016. 2, 3, 4

[4] Satanjeev Banerjee and Alon Lavie. 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*, pages 65–72, 2005. 6

[5] Long Chen, Hanwang Zhang, Jun Xiao, Xiangnan He, Shiliang Pu, and Shih-Fu Chang. Counterfactual critic multi-agent training for scene graph generation. In *ICCV*, 2019. 1

[6] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *CVPR*, 2017. 3

[7] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. Visual dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 326–335, 2017. 1

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2

[9] Jakob Foerster, Ioannis Alexandros Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2137–2145, 2016. 2

[10] Zhe Gan, Chuang Gan, Xiaodong He, Yunchen Pu, Kenneth Tran, Jianfeng Gao, Lawrence Carin, and Li Deng. Semantic compositional networks for visual captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5630–5639, 2017. 3

[11] Jiuxiang Gu, Jianfei Cai, Gang Wang, and Tsuhan Chen. Stack-captioning: Coarse-to-fine learning for image captioning. *AAAI*, 2017. 7, 8

[12] Jiuxiang Gu, Shafiq Joty, Jianfei Cai, and Gang Wang. Unpaired image captioning by language pivoting. In *ECCV*, 2018. 3

[13] Jiuxiang Gu, Shafiq Joty, Jianfei Cai, Handong Zhao, Xu Yang, and Gang Wang. Unpaired image captioning via scene graph alignments. In *ICCV*, 2019. 3

[14] Jiuxiang Gu, Handong Zhao, Zhe Lin, Sheng Li, Jianfei Cai, and Mingyang Ling. Scene graph generation with external knowledge and image reconstruction. In *CVPR*, 2019. 1

[15] John Hale, Chris Dyer, Adhiguna Kuncoro, and Jonathan R Brennan. Finding syntax in human encephalography with beam search. *arXiv preprint arXiv:1806.04127*, 2018. 2

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 5

[17] Xinwei He, Baoguang Shi, Xiang Bai, Gui-Song Xia, Zhaoxiang Zhang, and Weisheng Dong. Image caption generation with part of speech guidance. *Pattern Recognition Letters*, 2017. 3

[18] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 1

[19] Ronghang Hu, Jacob Andreas, Trevor Darrell, and Kate Saenko. Explainable neural computation via stack neural module networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 53–69, 2018. 2, 4

[20] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 804–813, 2017. 3

[21] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016. 6

[22] 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. 7, 8

[23] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2910, 2017. 1

[24] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015. 6

[25] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear attention networks. In *Advances in Neural Information Processing Systems*, pages 1571–1581, 2018. 5

[26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[27] Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei. A hierarchical approach for generating descriptive image paragraphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 317–325, 2017. 1

[28] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome:

Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017. 4, 6

[29] Girish Kulkarni, Visruth Premraj, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Baby talk: Understanding and generating image descriptions. In *Proceedings of the 24th CVPR*. Citeseer, 2011. 3

[30] Polina Kuznetsova, Vicente Ordonez, Alexander C Berg, Tamara L Berg, and Yejin Choi. Collective generation of natural image descriptions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 359–368. Association for Computational Linguistics, 2012. 3

[31] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 2004. 6

[32] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1, 2, 6

[33] Daqing Liu, Hanwang Zhang, Zheng-Jun Zha, and Feng Wu. Explainability by parsing: Neural module tree networks for natural language visual grounding. *arXiv preprint arXiv:1812.03299*, 2018. 3

[34] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. 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)*, volume 6, page 2, 2017. 2, 3

[35] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Neural baby talk. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7219–7228, 2018. 2, 3, 8

[36] Ruotian Luo, Brian Price, Scott Cohen, and Gregory Shakhnarovich. Discriminability objective for training descriptive captions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6964–6974, 2018. 3

[37] Margaret Mitchell, Xufeng Han, Jesse Dodge, Alyssa Mensch, Amit Goyal, Alex Berg, Kota Yamaguchi, Tamara Berg, Karl Stratos, and Hal Daumé III. Midge: Generating image descriptions from computer vision detections. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 747–756. Association for Computational Linguistics, 2012. 3

[38] Yulei Niu, Hanwang Zhang, Manli Zhang, Jianhong Zhang, Zhiwu Lu, and Ji-Rong Wen. Recursive visual attention in visual dialog. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1

[39] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002. 6

[40] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018. 5

[41] Yu Qin, Jiajun Du, Yonghua Zhang, and Hongtao Lu. Look back and predict forward in image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8367–8375, 2019. 3

[42] 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. 1, 3

[43] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *CVPR*, volume 1, page 3, 2017. 3, 5, 6, 7, 8

[44] Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object hallucination in image captioning. *arXiv preprint arXiv:1809.02156*, 2018. 1

[45] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. Movie description. *International Journal of Computer Vision*, 123(1):94–120, 2017. 3

[46] Jiaxin Shi, Hanwang Zhang, and Juanzi Li. Explainable and explicit visual reasoning over scene graphs. *arXiv preprint arXiv:1812.01855*, 2018. 1, 3

[47] L Robert Slevc. Saying what's on your mind: Working memory effects on sentence production. *Journal of experimental psychology: Learning, memory, and cognition*, 37(6):1503, 2011. 2

[48] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. *arXiv preprint arXiv:1812.01880*, 2018. 1

[49] Kristina Toutanova and Christopher D Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13*, pages 63–70. Association for Computational Linguistics, 2000. 5

[50] 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. 1, 4, 5

[51] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015. 6

[52] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015. 2, 3

[53] 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. 3

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

IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019. 7, 8

[55] Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, and Tao Mei. Boosting image captioning with attributes. In *IEEE International Conference on Computer Vision, ICCV*, pages 22–29, 2017. 3

[56] Guojun Yin, Lu Sheng, Bin Liu, Nenghai Yu, Xiaogang Wang, and Jing Shao. Context and attribute grounded dense captioning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3

[57] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4659, 2016. 3

[58] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. Mattnet: Modular attention network for referring expression comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1307–1315, 2018. 3, 4

[59] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5831–5840, 2018. 4

[60] Zheng-Jun Zha, Daqing Liu, Hanwang Zhang, Yongdong Zhang, and Feng Wu. Context-aware visual policy network for fine-grained image captioning. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 3, 7, 8

[61] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. Visual translation embedding network for visual relation detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5532–5540, 2017. 4