



Learning Cooperative Neural Modules for Stylized Image Captioning

Xinxiao Wu¹ · Wentian Zhao¹ · Jiebo Luo²

Received: 12 July 2021 / Accepted: 26 May 2022 / Published online: 22 July 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

Recent progress in stylized image captioning has been achieved through the encoder-decoder framework that generates a sentence in one-pass decoding process. However, it remains difficult for such a decoding process to simultaneously capture the syntactic structure, infer the semantic concepts and express the linguistic styles. Research in psycholinguistics has revealed that the language production process of humans involves multiple stages, starting with several rough concepts and ending with fluent sentences. With this in mind, we propose a novel stylized image captioning approach that generates stylized sentences in a multi-pass decoding process by training three cooperative neural modules under the reinforcement learning paradigm. A low-level neural module called *syntax module* first generates the overall syntactic structure of the stylized sentence. Next, two high-level neural modules, namely *concept module* and *style module*, incorporate the words that describe factual content and the words that express linguistic style, respectively. Since the three modules contribute to different aspects of the stylized sentence, i.e. the fluency, the relevancy of the factual content and the style accuracy, we encourage the modules to specialize in their own tasks by designing different rewards for different actions. We also design an attention mechanism to facilitate the communication between the high-level and low-level modules. With the help of the attention mechanism, the high-level modules are able to take the global structure of the sentence into consideration and maintain the consistency between the factual content and the linguistic style. Evaluations on several public benchmark datasets demonstrate that our method outperforms the existing one-pass decoding methods in terms of multiple different evaluation metrics.

Keywords Stylized image captioning · Cooperative modular networks · Reinforcement learning · Multi-pass decoding

1 Introduction

The task of stylized image captioning requires incorporating the linguistic style into natural language descriptions of images. It has attracted growing research interest due

Communicated by Svetlana Lazebnik.

This work was supported in part by the Natural Science Foundation of China (NSFC) under Grant No 62072041.

✉ Xinxiao Wu
wuxinxiao@bit.edu.cn

Wentian Zhao
wentian_zhao@bit.edu.cn

Jiebo Luo
jluo@cs.rochester.edu

¹ Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing, China

² Department of Computer Science, University of Rochester, Rochester, NY 14627, USA

to its wide applications, such as generating attractive titles for photos in social media and automatically writing lyrics or poems according to images. Most existing methods of image captioning employ an encoder-decoder framework (Andrew Shin and Harada, 2016; Chen et al., 2019, 2018; Gan et al., 2017; Guo et al., 2019; Mathews et al., 2016; Xu et al., 2019; Wu et al., 2019) where a convolutional neural network serves as the encoder to encode the input image and a recurrent neural network serves as the decoder to generate the output sentence. Such a decoding process is referred to as one-pass decoding (Xia et al., 2017), as illustrated in Fig. 1a. Although the encoder-decoder framework has achieved great success, it remains difficult for the one-pass decoding process in stylized image captioning to simultaneously capture the syntactic structure, infer the semantic concepts and express the linguistic styles.

Psycholinguistic research (Slevc, 2011) has revealed that humans produce a sentence by first constructing a coarse pattern and then filling in details, rather than directly organizing a word sequence from scratch. Motivated by the language

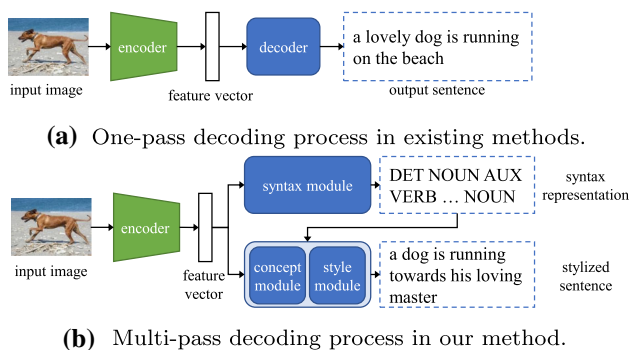


Fig. 1 Difference between one-pass decoding process in the existing methods (a) and multi-pass decoding process in our method (b)

production process of humans, we propose a novel stylized image captioning method that formulates the stylized sentence generation as a multi-pass decoding process, as shown in Fig. 1b. Our method first constructs a syntax pattern, and then generates words that describe the factual content and reflect the linguistic style. The proposed model is able to effectively capture the syntactic structure, infer semantic concepts and express linguistic styles in different decoding stages. We design multiple cooperative neural modules to perform the multi-pass decoding process and formulate the training of the neural modules as a reinforcement learning problem. Specifically, in the first decoding pass, a low-level *syntax module* constructs the overall syntactic structure of the stylized sentence. In the second decoding pass, two high-level modules, namely *concept module* and *style module*, generate the words in the sentence with the guidance of the previously generated syntactic structure. The concept module and the style module alternately predict the words that are related to the content of the image and the words that reflect the linguistic style.

In the existing reinforcement learning methods (Wang et al., 2018; Huang et al., 2019), the captioning model receives a sentence-level reward that reflects the quality of the entire sentence. However, in stylized image captioning, the quality of a stylized sentence is determined by multiple factors, including the fluency of the sentence, the relevancy of the factual content and the style accuracy. In this case, it is difficult for the sentence-level reward to distinguish the contribution of each word, referred to as credit assignment problem (Panait and Luke, 2005). For instance, the sentence-level reward for a sentence that accurately describes the image content but reflects no linguistic style and the sentence-level reward for another sentence that is stylized but irrelevant to the image might be similar. However, the contributions of the concept module and the style module to the two sentences are actually different. To alleviate this problem, we design different rewards for the outputs generated by different neural modules in multiple decoding passes, and thus to encourage the neural modules to specialize in their indi-

vidual tasks. Specifically, the reward for the factual words generated by the concept module is determined by the relevancy between the image and the generated sentence. The reward for the stylized words generated by the style module is determined by a sentence style classifier that evaluates whether the generated sentence expresses the linguistic style and a statistic language model that evaluates the fluency of the sentence.

Since the neural modules generate stylized sentences in a cooperative manner, the information passing between them is crucial. We design an attention mechanism that enables the low-level syntax module to share the overall syntactic structure of the sentence with the high-level modules in a simple but efficient way. In each decoding step, the high-level modules fuse the overall syntactic structure of the sentence using an attention module. In this way, the high-level modules are able to consider whether a word fits in the whole sentence from a global perspective, and ensure the consistency between the words that describe the image content and the words that reflect the linguistic style.

In summary, the main contributions of this paper are as follows:

- We make the first attempt to formulate the stylized image captioning as a multi-pass decoding process that imitates the language production process of humans, and design multiple cooperative neural modules to implement the multi-pass decoding process.
- We propose to optimize the neural modules using a reinforcement learning method, where different rewards are designed for different types of actions to encourage each module to focus on its own task.
- Experiments on SentiCap, FlickrStyle10K and stylized paragraphs comprehensively demonstrate the superiority of our method over the existing state-of-the-art methods.

2 Related Work

2.1 Stylized Image Captioning

Stylized image captioning has attracted growing attention in recent years. SentiCap (Mathews et al., 2016) is the first stylized image captioning method to generate image captions in positive or negative sentiments with the proposed switching RNN, where additional word-level sentiment annotations are also applied to guide the training process. Chen et al. (2018) propose a style-factual LSTM to generate sentences in positive, negative, romantic and humorous styles. The style-factual LSTM is trained using an adaptive learning strategy that employs a reference model learned from a stylized corpus to provide factual knowledge.

To alleviate the burden of annotating stylized sentences for images, several methods are proposed for training stylized image captioning models using unpaired stylized sentences. StyleNet (Gan et al., 2017) is proposed to factorize the parameters in LSTM and learn the knowledge about factual and stylized sentences with different parameters. Mathews et al. (2018) use semantic terms to represent the semantic information in images and sentences, which enables training using unpaired stylized corpus. Chen et al. (2019) propose to learn the knowledge about linguistic style using a domain layer norm, thus generating stylized sentences in a more flexible manner. MSCap (Guo et al., 2019) is the first method that attempts to generate image descriptions in multiple styles with a single model. Zhao et al. (2020) propose a style memory module to memorize the knowledge about style-related words or phrases.

Most of the above methods generate stylized image descriptions in a single decoding process. In contrast, our method generates stylized sentences in two decoding passes using multiple neural modules, where the first pass constructs the syntactic structure of the sentence using a low-level module, and the second pass generates the words using two high-level modules.

2.2 Multi-Pass Decoding for Natural Language Generation

Since the one-pass decoding process in the encoder-decoder framework has some deficiencies, e.g., the prediction of one word only relying on the previous words (Xia et al., 2017) and the error accumulation problem (Liu et al., 2019), some multi-pass decoding methods are proposed to generate natural language. Deliberation networks (Xia et al., 2017) with two levels of decoders are designed to perform neural machine translation and text summarization. The first decoder takes the output of the encoder and generates a coarse sentence, and the second decoder refines the sentence generated by the first decoder. Gu et al. (2018) propose a multi-pass decoding method that involves three LSTM decoders trained in a reinforcement learning manner. The first LSTM generates coarse sentences and the subsequent LSTM networks refine the output of the previous LSTM. Guo et al. (2019) propose a ruminant decoding framework that contains two decoders to generate image captions, where the first decoder generates a raw sentence and the second decoder utilizes the global information in the raw sentence for caption refinement. Xu et al. (2020) propose reinforcement learning polishing networks to refine the raw captions generated by any captioning model. Specifically, two networks with the same structure are designed to correct the word errors and the grammar errors in the raw captions, respectively.

The aforementioned methods focus on generating factual sentences, where the first decoding pass generates a coarse

sentence and the second decoding pass generates refined sentences with more details. In contrast, our method generates stylized sentences by constructing the syntactic structure of the sentence in the first decoding pass and then generating the specific words in the second decoding pass.

2.3 Reinforcement Learning

Recent years have witnessed the success of reinforcement learning in many fields, including object detection (Kong et al., 2017), text generation (Dethlefs and Cuayáhuitl, 2010), and dialog systems (Peng et al., 2017). In the field of vision and language, Rennie et al. (2017) first use the REINFORCE algorithm (Williams, 1992) with a baseline to optimize image captioning models. Hierarchical reinforcement learning has been used for video captioning (Wang et al., 2018) and visual storytelling (Huang et al., 2019). Wang et al. (2018) propose a video captioning model that involves a manager and a worker. The manager first generates a sub-goal and the worker then fulfills the sub-goal by generating a small text segment. This process is repeated until the entire sentence is finished. Similarly, Huang et al. (2019) propose to generate a coherent story for multiple images with a two-level hierarchical decoder. The manager generates a topic distribution for each image and the worker generates a sentence according to the image and the corresponding topic distribution. The above methods use sentence-level CIDEr score as the reward in the training process and it is difficult to distinguish the contribution of each individual agent. To tackle this problem, Guo et al. (2020) propose to train a non-autoregressive image captioning model using multi-agent reinforcement learning, where each position in the sentence is regarded as an agent. Each agent receives a word-level reward that measures its individual contribution to the sentence generation.

The existing methods generate factual sentences, and the rewards are calculated by comparing the generated sentences and the ground-truth sentences. Our method focuses on generating stylized captions for images, which is a more challenging task. So the rewards used in our method measure multiple aspects of the generated sentences, including the fluency, the relevancy of the factual content and the stylishness.

3 Our Approach

3.1 Overview

A stylized image captioning model is expected to generate a stylized description of an image that preserves the visual content of the image and expresses certain linguistic styles simultaneously. To train the captioning model, we are given a set of training images and their corresponding factual descriptions, denoted as $D_f = \{(x_i, Y_i^f)|_i\}$, where x_i

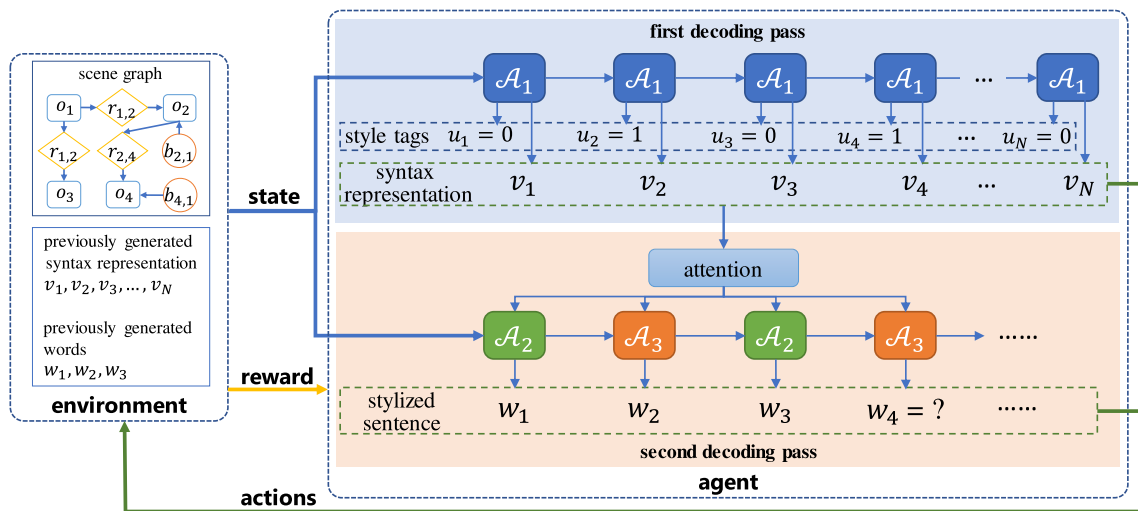


Fig. 2 Overview of the proposed method. \mathcal{A}_1 , \mathcal{A}_2 and \mathcal{A}_3 denote the syntax module, the concept module and the style module, respectively. In the first decoding pass, the syntax module generates the part-of-speech tags and the style tags of the words in the sentence. The part-of-speech tags represent the grammatical component of each word and the style tags indicate whether the words are related to fac-

tual content or linguistic style. In the second decoding pass, the concept module and the style module alternatively generate the words in the stylized sentence. The concept module generates the words that describe the content of the image and the style module generates the words that reflect the desired linguistic style. Best viewed in color

represents the i th image and Y_i^f represents the corresponding factual sentence. For each linguistic style s , we are also given a training corpus $D_s = \{Y_i^s | i\}$, where Y_i^s denotes the i th sentence with the style s .

Our method mainly consists of a low-level module called syntax module \mathcal{A}_1 and two high-level modules, namely the concept module \mathcal{A}_2 and style module \mathcal{A}_3 . For an input image x , firstly it is converted into a scene graph G^x that encodes the information of objects, relationships and attributes in a structured manner and provides rich semantic information. Then, the three modules take the scene graph as input and generate the stylized sentence in two decoding passes. In the first decoding pass, the syntax module constructs the overall structure of the stylized sentence by generating a part-of-speech (POS) tag sequence $\hat{V} = \{\hat{v}_1, \hat{v}_2, \dots, \hat{v}_N\}$ and a style tag sequence $\hat{U} = \{\hat{u}_1, \hat{u}_2, \dots, \hat{u}_N\}$. The POS tag $\hat{v}_i \in S_{pos}$ represents the grammatical component of the i th word and S_{pos} denotes the set of all possible POS tags. The style tag $\hat{u}_i \in \{0, 1\}$ indicates whether the i th word reflects the linguistic style. In the second decoding pass, the concept module and the style module generate the stylized sentence $\hat{Y}^s = \{\hat{w}_1, \hat{w}_2, \dots, \hat{w}_N\}$ by predicting the words that are related to factual content and the words that are related to linguistic style, respectively. The information passing between the low-level module and the high-level modules is facilitated by an attention mechanism that enables the high-level modules to utilize the syntactic structure constructed by the low-level module. The framework of the proposed method is illustrated in Fig. 2.

3.2 Scene Graph Generation

Scene graph characterizes the objects, relationships and attributes in images and sentences in a structured form that contains rich semantic information. It has been proved to be beneficial to many vision-and-language tasks, such as image captioning (Li and Jiang, 2019; Yang et al., 2019; Zhao et al., 2020) and cross-modal retrieval (Johnson et al., 2015). Since the scene graph serves as an ideal intermediate form between images and sentences, we transform both the images and the sentences into scene graphs to enable training our captioning model using unpaired stylized corpus.

Formally, a scene graph can be denoted as $G = (V, E)$ where V and E represent the node set and the edge set in the scene graph, respectively. The node set contains object nodes, relationship nodes and attribute nodes, and the edge set comprises directed edges that connect the three types of nodes. For each object node o_i and its k th attribute node $b_{i,k}$, there is a directed edge $o_i \rightarrow b_{i,k}$ in the edge set. The relationship between the i th object and the j th object is denoted as r_{ij} . The nodes o_i , r_{ij} and o_j are the subject, predicate and object of the relationship, respectively. We use two directed edges $o_i \rightarrow r_{ij}$ and $r_{ij} \rightarrow o_j$ to represent the relationship between o_i and o_j . We denote the scene graph of image x and the scene graph of sentence Y as G^x and G^Y , respectively.

We use a pre-trained scene-graph generator (Zellers et al., 2018) to generate the scene graph G^x of the image x . For a factual sentence Y^f , we generate its scene graph G^{Y^f} following the method proposed in (Anderson et al., 2016) where the sentence is first parsed into a dependency tree and then

the dependency parse is converted into a scene graph via a rule-based tree transformation method. For a stylized sentence Y^s , we first use the sentence decomposition algorithm in Sect. 3.3 to remove the words that reflect the linguistic style. Then we convert the remaining factual content of the sentence, denoted as \bar{Y}^s , to the sentence scene graph G^{Y^s} .

3.3 Stylized Sentence Decomposition

Since the concept module generates the words that describe the factual content while the style module generates the words that reflect the linguistic style, distinguishing the two types of words in the ground-truth stylized sentences is necessary for training the two modules effectively. In this section, we introduce a stylized sentence decomposition algorithm that outputs a style tag $u_i \in \{0, 1\}$ for each word w_i in the stylized sentence Y^s using an attention-based text classifier. The text classifier calculates an attention weight for each word, and distinguishes whether the input sentence is stylized according to the weighted sum of the encoded representations of the words. Since the words with higher attention weights are more important to the linguistic style of a stylized sentence, we estimate the style tag of each word using the attention weights of the text classifier.

Specifically, we train the binary text classifier proposed in (Sun and Lu, 2020). The sentences in the stylized corpus D^s are used as positive samples, and the sentences in factual data D^f are used as negative samples. The text classifier first encodes the word embeddings $\{e_{w_1}, e_{w_2}, \dots, e_{w_N}\}$ using a bidirectional GRU network and the encoded representations are denoted as $\{c_1, c_2, \dots, c_N\}$, where $e_{w_i} \in \mathbb{R}^d$, $c_i \in \mathbb{R}^d$ and d denotes the dimension of word embedding vectors. The attention score β_i of the i th word and the classifier’s output probability p are then calculated by

$$\begin{aligned} \hat{\beta}_i &= c_i^\top m_1, \\ \beta_i &= \frac{\exp(\hat{\beta}_i)}{\sum_j \exp(\hat{\beta}_j)}, \\ c &= \sum_i \beta_i c_i, \\ p &= \sigma(c^\top m_2), \end{aligned} \tag{1}$$

where $m_1 \in \mathbb{R}^d$ and $m_2 \in \mathbb{R}^d$ are learnable parameters, and σ denotes the sigmoid function. Given the attention weights of the words in a sentence Y , the threshold β' for discriminating the words that reflect linguistic style is the maximum value that satisfies $\sum_{\beta_i \geq \beta'} \beta_i > 0.8$. The value of the style tag u_i for the word w_i is 1 if the attention score β_i is no less than the threshold β' , and 0 otherwise. The factual part \bar{Y}^s of a stylized sentence Y^s consists of all the words whose corresponding style tag is 0.

3.4 Modular Captioning Model

3.4.1 Scene Graph Encoding

The object node o_i , attribute node $b_{i,k}$ and relationship node r_{ij} in the scene graph are represented by d_e -dimensional embedding vectors, denoted as e_{o_i} , $e_{b_{i,k}}$ and $e_{r_{ij}}$, respectively. The embedding vector of each node is calculated using the word embeddings of the corresponding class label. We further encode the object nodes and relationship nodes together with their neighbouring nodes to gather the context-aware information. The context-aware embeddings of the object o_i and relationship r_{ij} are calculated by

$$\begin{aligned} u_{o_i} &= \frac{1}{N_{o_i} + 1} \left(\sum_{k=1}^{N_{o_i}} W_o [e_{o_i}; e_{b_{i,k}}] + e_{o_i} \right), \\ u_{r_{ij}} &= W_r [e_{o_i}; e_{r_{ij}}; e_{o_j}], \end{aligned} \tag{2}$$

where N_{o_i} denotes the total number of attributes that belong to o_i , $W_o \in \mathbb{R}^{d_e \times 2d_e}$ and $W_r \in \mathbb{R}^{d_e \times 3d_e}$ are learnable parameters, and $[\]$ denotes the vector concatenation operation. The encoding of the scene graph G is represented by

$$u_G = \frac{1}{N_o} \sum_{i=1}^{N_o} u_{o_i} + \frac{1}{N_r} \sum_{r_{ij} \in G} u_{r_{ij}}, \tag{3}$$

where N_o and N_r are the total numbers of objects and relationships in G , respectively.

3.4.2 Multi-Pass Decoding

In this section, we illustrate the three cooperative neural modules that form the decoder in detail. Since all the modules are RNN-based, we first revisit the process of generating sequences using RNN. We then introduce the structure of the low-level syntax module. Finally, we present the attention mechanism that facilitates the information passing between the low-level module and the high-level modules, and show the structure of the high-level modules.

RNN-based language model At each time step t , an RNN-based module takes a vector x_t and the previous hidden state $h_{t-1} \in \mathbb{R}^{d_h}$ as input and outputs a hidden state $h_t \in \mathbb{R}^{d_h}$. We denote this process as $h_t = \mathcal{A}(h_{t-1}, x_t)$, where the module \mathcal{A} can be the syntax module \mathcal{A}_1 , the concept module \mathcal{A}_2 or the style module \mathcal{A}_3 . The parameters of \mathcal{A}_1 , \mathcal{A}_2 and \mathcal{A}_3 are denoted as θ_1 , θ_2 and θ_3 , respectively.

Low-level module In the first decoding pass, the low-level syntax module \mathcal{A}_1 summarizes the content of the image and learns the pattern of the sentence by generating a POS tag sequence \hat{V} and a style tag sequence \hat{U} . It takes the embedding of the scene graph G as input and generates the POS

tags and the style tags sequentially. The t th step in the first decoding pass can be formulated as

$$\begin{aligned}
 \mathbf{x}_t^{low} &= \begin{cases} \mathbf{W}_{in}^{low}[\mathbf{u}_G; \mathbf{e}_s] & \text{for } t = 0, \\ \mathbf{e}_{\hat{v}_{t-1}} & \text{for } t > 0, \end{cases} \\
 \mathbf{h}_t^{low} &= \mathcal{A}_1(\mathbf{h}_{t-1}^{low}, \mathbf{x}_t^{low}), \\
 \mathbf{p}_t^v &= \text{softmax}(\mathbf{W}_v \mathbf{h}_t^{low}), \\
 \mathbf{p}_t^u &= \text{softmax}(\mathbf{W}_u \mathbf{h}_t^{low}),
 \end{aligned} \tag{4}$$

where $\mathbf{u}_G \in \mathbb{R}^{d_e}$ is the encoding of the scene graph defined in Eq. (3), $\mathbf{e}_s \in \mathbb{R}^{d_e}$ denotes the embedding of the desired linguistic style s , and $\mathbf{e}_{\hat{v}_{t-1}} \in \mathbb{R}^{d_e}$ denotes the embedding of the POS tag generated at the previous step. $\mathbf{p}_t^v \in \mathbb{R}^{|S_{pos}|}$ and $\mathbf{p}_t^u \in \mathbb{R}^2$ denotes the probability distribution for generating the t -th POS tag \hat{v}_t and the probability distribution for generating the t -th style tag \hat{u}_t , respectively, where S_{pos} denotes the set containing all the POS tags. The matrices $\mathbf{W}_{in}^{low} \in \mathbb{R}^{d_e \times 2d_e}$, $\mathbf{W}_v \in \mathbb{R}^{|S_{pos}| \times d_h}$ and $\mathbf{W}_u \in \mathbb{R}^{2 \times d_h}$ are learnable parameters.

Attention Module Both the concept module and the style module leverage the syntactic structure generated by the syntax module via the attention module. At the t th step in the second decoding process, the attention module encodes the syntactic structure of the sentence by calculating a weighted sum of the syntax module’s hidden states according to the hidden state \mathbf{h}_t^{high} of the high-level module. By denoting the hidden states of the low-level syntax module as $\mathbf{H}^{low} = [\mathbf{h}_1^{low}; \mathbf{h}_2^{low}; \dots; \mathbf{h}_N^{low}]$, we have

$$\begin{aligned}
 \hat{\alpha}_i &= \mathbf{w}_a^T \tanh(\mathbf{W}_q \mathbf{h}_t^{high} + \mathbf{W}_k \mathbf{h}_i^{low}), \\
 \alpha_i &= \frac{\exp(\hat{\alpha}_i)}{\sum_j \exp(\hat{\alpha}_j)}, \\
 \text{attention}(\mathbf{h}_t^{high}, \mathbf{H}^{low}) &= \mathbf{H}^{low} \boldsymbol{\alpha},
 \end{aligned} \tag{5}$$

where α_i is the i th dimension of $\boldsymbol{\alpha}$, and $\boldsymbol{\alpha} \in \mathbb{R}^N$ denotes the weights for the N hidden states of the low-level module. $\mathbf{w}_a \in \mathbb{R}^{d_h}$ and the matrices $\mathbf{W}_q \in \mathbb{R}^{d_h \times d_h}$ and $\mathbf{W}_k \in \mathbb{R}^{d_h \times d_h}$ are learnable parameters.

High-level modules Given the syntactic structure of the sentence predicted by the low-level syntax module \mathcal{A}_1 , the high-level concept module \mathcal{A}_2 and style module \mathcal{A}_3 generate the stylized sentence by alternatively predicting the words that describe the content of the image and the words that reflect the linguistic style in the second decoding pass. The word \hat{w}_t is predicted by \mathcal{A}_2 when $\hat{u}_t = 0$, and is predicted by \mathcal{A}_3 when $\hat{u}_t = 1$. The concept module and the style module have the same structure but the parameters are not shared. Formally, the t th step in the second decoding pass can be

written as

$$\begin{aligned}
 \mathbf{x}_0^{high} &= \mathbf{W}_{in}^{high}[\mathbf{u}_G; \mathbf{e}_s] \\
 \mathbf{c}_t^{high} &= \text{attention}(\mathbf{h}_{t-1}^{high}, \mathbf{H}^{low}), \\
 \mathbf{x}_t^{high} &= \mathbf{W}_d^{high}[\mathbf{e}_{\hat{w}_{t-1}}; \mathbf{e}_{\hat{v}_{t-1}}; \mathbf{c}_t^{high}], t > 0, \\
 \mathbf{h}_t^{high} &= \begin{cases} \mathcal{A}_2(\mathbf{h}_{t-1}^{high}, \mathbf{x}_t^{high}) & \text{for } \hat{u}_t = 0, \\ \mathcal{A}_3(\mathbf{h}_{t-1}^{high}, \mathbf{x}_t^{high}) & \text{for } \hat{u}_t = 1, \end{cases} \\
 \mathbf{p}_t^w &= \begin{cases} \mathbf{W}_{wf} \mathbf{h}_t^{high} & \text{for } \hat{u}_t = 0, \\ \mathbf{W}_{ws} \mathbf{h}_t^{high} & \text{for } \hat{u}_t = 1, \end{cases}
 \end{aligned} \tag{6}$$

where \mathbf{e}_s denotes the embedding of the desired linguistic style s , $\mathbf{e}_{\hat{w}_{t-1}} \in \mathbb{R}^{d_e}$ denotes the embedding of the previously generated word \hat{w}_{t-1} , $\mathbf{p}_t^w \in \mathbb{R}^{|S_{word}|}$ denotes the probability distribution for generating the word \hat{w}_t and S_{word} denotes the set containing all the possible words in the sentences. The matrices $\mathbf{W}_{in}^{high} \in \mathbb{R}^{d_e \times 2d_e}$, $\mathbf{W}_d^{high} \in \mathbb{R}^{d_e \times 3d_e}$, $\mathbf{W}_{wf} \in \mathbb{R}^{|S_{word}| \times d_h}$ and $\mathbf{W}_{ws} \in \mathbb{R}^{|S_{word}| \times d_h}$ are learnable parameters.

3.5 Training

The whole training process of our method involves a pre-training stage and a fine-tuning stage. In the pre-training stage, we use the factual data D_f to train the neural modules. In the fine-tuning stage, we fine-tune the model using the stylized sentences D_s to generate stylized descriptions for images.

3.5.1 Pre-Training Stage

Since the factual data D_f contains images with the corresponding factual sentences rather than stylized sentences, we only train the syntax module and concept module in the pre-training stage. The ground-truth style tags of all the words in the sentences in D_f are set to 0. We acquire the ground-truth POS tag sequences of the sentences in D_f using off-the-shelf POS tagger in the spaCy toolkit (Honnibal et al., 2020). The loss function \mathcal{L}_p in the pre-training stage is formulated as

$$\begin{aligned}
 \mathcal{L}_p &= \mathcal{L}_{ce}^{pos} + \mathcal{L}_{ce}^{word}, \\
 \mathcal{L}_{ce}^{pos} &= -\frac{1}{N} \sum_{t=1}^N \log(p(\hat{v}_t = v_t)), \\
 \mathcal{L}_{ce}^{word} &= -\frac{1}{N} \sum_{t=1}^N \log(p(\hat{w}_t = w_t)),
 \end{aligned} \tag{7}$$

where \mathcal{L}_{ce}^{pos} and \mathcal{L}_{ce}^{word} represent the losses for generating POS tags and words in the sentence, respectively. \hat{v}_t and \hat{w}_t denote the t th POS tag and the t th word generated by the syntax module and concept module, respectively.

3.5.2 Fine-Tuning Stage

In the fine-tuning stage, only the stylized sentences D_s are available. For a stylized description Y^s , a scene graph G^Y is acquired using the method in Sect. 3.2 and G^Y is used as the input of the three modules. For faster convergence, we initialize the parameters θ_3 of the style module \mathcal{A}_3 using the pre-trained parameters θ_2 of the concept module \mathcal{A}_2 . In the first few epochs, we optimize the modules \mathcal{A}_1 , \mathcal{A}_2 and \mathcal{A}_3 using the cross-entropy loss function \mathcal{L}_{ft} :

$$\begin{aligned} \mathcal{L}_{ft} &= \mathcal{L}_{ce}^{pos} + \mathcal{L}_{ce}^{style} + \mathcal{L}_{ce}^{word}, \\ \mathcal{L}_{ce}^{style} &= -\frac{1}{N} \sum_{t=1}^N \log(p(\hat{u}_t = u_t)), \end{aligned} \tag{8}$$

where \mathcal{L}_{ce}^{style} denotes the cross-entropy loss for generating style tags and \hat{u}_t denotes the t th style tag generated by the syntax module.

After fine-tuning using the cross-entropy loss \mathcal{L}_{ce}^{style} , we optimize the captioning model using reinforcement learning. The captioning model that consists of three neural modules can be regarded as an **agent** that interacts with an external **environment**. Here, the environment is represented by the scene graph of the input image and the previously generated part-of-speech tags and words. The caption generation process is formulated as a markov decision process (MDP), denoted as $\mathcal{M} = \langle \mathcal{S}, A, T, R \rangle$, where \mathcal{S} is the state space, A is a set of all possible actions, T denotes the state transition function, and R denotes the reward function. Given the current **state** $s \in \mathcal{S}$ and an **action** $a \in A$, $T(s, a, s')$ represents the probability that the next state is s' , and $R(s, a)$ is the **reward** observed by the agent when an action is performed.

Specifically, at the t th time step, the action a_t can be a part-of-speech tag, a word that describes the factual content or a word that reflects the linguistic style. Thus, the action set is formulated as

$$A = S_{pos} \cup S_{fact} \cup S_{sty}, \tag{9}$$

where S_{pos} , S_{fact} , and S_{sty} denote the set of part-of-speech tags, all the factual words, and all the words that are related to linguistic style, respectively. The action a_t is selected by the agent according to the policy $\pi_\theta(a_t|s_t)$, i.e. a conditional probability distribution parameterized by $\theta = \{\theta_1, \theta_2, \theta_3\}$, where θ_1 , θ_2 and θ_3 denote the parameters of the syntax module, the concept module and the style module, respectively. The state $s_t \in \mathcal{S}$ is a sequence containing the input scene graph G and all the previous actions (i.e. the previously generated part-of-speech tags and words), formulated as

$$s_t = \{G, a_1, a_2, \dots, a_{t-1}\}. \tag{10}$$

When the action a_t is selected, we append the action a_t to the end of s_t to form the next state s_{t+1} . Thus, the probability that the next state is $s_{t+1} = \{G, a_1, \dots, a_t\}$ is 1, and the state transition function is formulated as

$$T(s_t, a_t, s_{t+1} = \{G, a_1, \dots, a_t\}) = 1. \tag{11}$$

The agent observes an immediate reward $R(s_t, a_t)$ after the state is transitioned to the next state s_{t+1} . Since three different neural modules are used to perform the actions, we define different rewards for different actions. For the part-of-speech tags, the words that describe the factual content and the words related to linguistic style, the rewards are defined by

$$R(s_t, a_t) = \begin{cases} \lambda_1(f_p(Y_{1:t}^s) - f_p(Y_{1:t-1}^s)), & \text{for } a_t \in S_{pos}, \\ f_r(Y_{1:t}^s) - f_r(Y_{1:t-1}^s), & \text{for } a_t \in S_{fact}, \\ \lambda_1(f_p(Y_{1:t}^s) - f_p(Y_{1:t-1}^s)) + \lambda_2(f_c(Y_{1:t}^s) - f_c(Y_{1:t-1}^s)), & \text{for } a_t \in S_{sty}, \end{cases} \tag{12}$$

where $Y_{1:t}^s$ denotes the first t tokens of the sentence, $f_p(Y^s)$ is the perplexity of the sentence Y^s and $f_r(Y^s)$ is the CIDER score of Y^s . $f_c(Y^s) \in \{0, 1\}$ is the output of a pre-trained sentence classifier that distinguishes whether the sentence Y^s is a stylized sentence.

The goal of training using reinforcement learning is to maximize the following objective function:

$$J(\theta) = \sum_t^N \mathbb{E}_{a_t \sim \pi_\theta} [R(s_t, a_t)], \tag{13}$$

We compute the expected gradient of the objective function $J(\theta)$ to the parameters θ using the policy gradient theorem (Sutton and Barto, 2018). By using the REINFORCE algorithm, the gradient is approximated using a single Monte-Carlo sampling from the policy π_θ . The estimation of the gradient of $J(\theta)$ is formulated as

$$\begin{aligned} \nabla_\theta J(\theta) &= \sum_t^N \mathbb{E}_{a_t \sim \pi_\theta} [R(s_t, a_t) \nabla_\theta \log \pi_\theta(a_t)], \\ &\approx \sum_t^N R(s_t, a_t) \nabla_\theta \log \pi_\theta(a_t), \end{aligned} \tag{14}$$

where $\pi_\theta(a_t)$ denotes the probability of sampling the action a_t by using the policy π_θ .

To reduce the variance of the estimated gradient, we compute the gradient by subtracting the reward with a baseline b that does not depend on the action a_t :

$$\nabla_\theta J(\theta) \approx \sum_t^N (R(s_t, a_t) - b) \nabla_\theta \log \pi_\theta(a_t). \tag{15}$$

In practice, we use the reward of a greedily-decoded sentence as the baseline. The loss of training using reinforcement learning is formulated as

$$\mathcal{L}_{rl} = - \sum_t^N (R(s_t, a_t) - b) \log \pi_\theta(a_t). \quad (16)$$

4 Experiments

4.1 Datasets

To evaluate the effectiveness of our method, we conduct experiments of both simple stylized sentence generation and complex stylized paragraph generation. For stylized sentence generation, we pre-train our model using the MSCOCO dataset (Lin et al., 2014), and then fine-tune it using the positive and negative sentences in the SentiCap dataset (Mathews et al., 2016) as well as the humorous and romantic sentences in the FlickrStyle10K dataset (Gan et al., 2017). The MSCOCO dataset contains 113,287, 5000 and 5000 images for training, validation and testing, respectively, and each image is annotated with 5 factual captions. The training split of SentiCap contains 2994 positive sentences and 2991 negative sentences, and the test split contains 2019 positive sentences and 1509 negative sentences, respectively. The original FlickrStyle10K dataset contains 10,000 images, and each image is annotated with a humorous sentence and a romantic sentence. Among the 7000 publicly available sentences, we randomly select 6000 of them as the training split and the remaining images are used as the test split, which is consistent with the strategy in (Guo et al., 2019).

For the stylized paragraph generation, we pre-train our model using the Stanford image-paragraph dataset (Krause et al., 2017) and then fine-tuned using the stylized paragraphs collected by (Chen et al., 2019). The Stanford image-paragraph dataset contains 14,575, 2489 and 2487 images for training, validation and testing, respectively. The stylized paragraphs include the romantic and humorous paragraphs collected from BookCorpus (Zhu et al., 2015) as well as the lyrics and fairy tales collected from the web. We randomly select 50,000 paragraphs for fine-tuning, which is consistent with (Chen et al., 2019). Since the stylized paragraphs does not contain any images, we evaluate our method using the images in the test set of the Stanford image-paragraph dataset.

4.2 Evaluation Metrics

The performance of stylized sentence generation is evaluated in terms of stylishness and fluency of the generated sentences. We use the widely-used metrics for image captioning to evaluate the relevancy, including Bleu (Papineni et al.,

2002), METEOR (Banerjee and Lavie, 2005) and CIDER (Vedantam et al., 2015). We measure the stylishness using the *style accuracy*. Specifically, for each linguistic style s , we use the stylized sentences in D_s as the positive samples and use the factual sentences in D_f as negative samples to train a TextCNN (Kim, 2014) classifier that classifies whether an input sentence is stylized or not. The style accuracy is defined as the percentage of sentences that are classified as stylized. We use the *average perplexity* to measure the fluency of the generated sentences. For each linguistic style s , we train a 3-gram language model using the SRILM toolkit (Stolcke, 2002) to compute the perplexity of each sentence in style s . A lower average perplexity indicates that the generated sentences are more fluent. Specifically, the perplexity of a sentence $Y = \{w_1, w_2, \dots, w_N\}$ is calculated as

$$ppl(Y) = \left(\prod_{i=1}^N \frac{1}{p(w_i | w_{i-2}, w_{i-1})} \right)^{\frac{1}{N}}, \quad (17)$$

where $p(w_i | w_{i-1}, w_{i-2})$ is the probability of word w_i appearing after the words w_{i-2}, w_{i-1} given by the language model.

For the stylized paragraph generation, we use the metrics in (Chen et al., 2019) to evaluate the relevancy and the stylishness of the generated sentences. The relevancy is measured by SPICE (Anderson et al., 2016) and content similarity. The content similarity CS is calculated as follows:

$$\begin{aligned} CS &= \frac{2pr}{p+r}, \\ p &= \frac{|C_T \cap (\hat{C}_S \cup C_S)|}{|C_T|}, \\ r &= \frac{|C_S \cap (\hat{C}_T \cup C_T)|}{|C_S|}, \end{aligned} \quad (18)$$

where C_S and C_T denote the nouns in the ground truth sentence and the generated sentence, respectively. The words in \hat{C}_S and \hat{C}_T are the synonyms of the words in C_S and C_T , respectively. We also report the numerators of p and r , denoted as n_p and n_r . The stylishness is measured by *transfer accuracy* (Fu et al., 2018) that is calculated using a text classifier based on LSTM. We use the stylized paragraphs as positive samples and the paragraphs in the Stanford image-paragraph dataset as negative samples to train the text classifier, and the proportion of sentences that are stylized is reported as the transfer accuracy.

4.3 Implementation Details

The three modules in our method are implemented by GRU and the hidden state dimension d_h is set to 512. The perplexity threshold T_{ppl} in Eq. (12) is set to 20 and the

Table 1 Results of generating stylized sentences

Method	Visual Input	Positive						Negative					
		B-1	B-3	M	C	ppl (↓)	cls	B-1	B-3	M	C	ppl (↓)	cls
StyleNet	ResNet	45.3	12.1	12.1	36.3	24.8	45.2	43.7	10.6	10.9	36.6	25.0	56.6
Ours_single_feat		48.4	17.5	16.0	51.0	11.2	92.2	47.6	16.6	15.5	49.9	11.4	89.5
MemCap_single	Scene	50.8	17.1	16.6	54.4	13.0	99.8	48.7	19.6	15.8	60.6	14.6	93.1
Ours_single	Graph	51.3	18.1	16.8	54.6	13.0	99.2	49.0	19.7	15.9	59.3	14.3	93.8
MSCap	ResNet	46.9	16.2	16.8	55.3	19.6	92.5	45.5	15.4	16.2	51.6	19.2	93.4
Ours_multi_feat		47.5	16.9	16.3	54.6	13.3	98.7	46.7	17.9	16.0	53.2	13.9	95.4
MemCap_multi	Scene	51.1	17.0	16.6	52.8	18.1	96.1	49.2	18.1	15.7	59.4	18.9	98.9
Ours_multi	Graph	52.3	18.2	17.0	54.8	13.2	99.3	49.3	18.4	16.3	55.0	13.0	96.5

Method	Visual Input	Romantic						Humor					
		B-1	B-3	M	C	ppl (↓)	cls	B-1	B-3	M	C	ppl (↓)	cls
StyleNet	ResNet	13.3	1.5	4.5	7.2	52.9	37.8	13.4	0.9	4.3	11.3	48.1	41.9
Ours_single_feat		20.6	4.8	7.9	19.8	11.5	89.3	22.9	5.1	8.8	20.9	15.2	89.8
MemCap_single	Scene	21.2	4.8	8.4	22.4	14.4	98.7	19.9	4.3	7.4	19.4	16.4	98.9
Ours_single	Graph	24.2	5.9	9.4	27.4	13.3	98.4	26.7	5.8	9.3	23.5	15.2	99.0
MSCap	ResNet	17.0	2.0	5.4	10.1	20.4	88.7	16.3	1.9	5.3	15.2	22.7	91.3
Ours_multi_feat		23.2	3.0	6.4	19.5	17.3	90.2	22.0	3.2	6.3	18.7	18.0	92.2
MemCap_multi	Scene	19.7	4.0	7.7	19.7	19.7	91.7	19.8	4.0	7.2	18.5	17.0	97.1
Ours_multi	Graph	25.4	5.7	9.2	24.7	12.3	97.9	27.2	5.9	9.0	22.4	13.5	96.6

The upper part shows the results on SentiCap with positive and negative styles and the lower part shows the results on FlickrStyle10K with romantic and humorous styles. “ResNet” and “Scene Graph” denote ResNet-152 visual feature and scene graph, respectively. B-n, M, C, ppl and cls are abbreviations for Bleu-n, METEOR, CIDEr, average perplexity and style accuracy, respectively. For average perplexity, a lower value is better. The best performances in both single-style setting and multi-style setting are marked in bold

hyper-parameters λ_1 , λ_2 and λ_3 in Eq. (12) are set to 1.0, 1.0 and 0.5, respectively. The Adam optimizer (Kingma and Ba, 2015) is applied in both pre-training stage and fine-tuning stage. In the pre-training stage, the learning rate is set to 5×10^{-4} . In the fine-tuning stage, the cross entropy loss function in Eq. (8) is used in the first 20 epochs and the reinforcement learning is applied in the rest epochs. The learning rate is set to 5×10^{-5} and decays 0.8 times for every 10 epochs.

All experiments are conducted using one NVIDIA RTX 2080Ti GPU. On the SentiCap and FlickrStyle10K datasets, training the model in single-style setting for one epoch takes about 5 min and 10 min using the cross-entropy loss and the reinforcement learning, respectively, and the entire training process takes about 3 h. On the stylized paragraphs dataset, it takes about 10 min and 30 min to train the model for one epoch using cross-entropy loss and reinforcement learning, respectively, and the entire training process on the stylized paragraphs takes about 7 h.

4.4 Results

4.4.1 Results of Generating Stylized Sentences

For the stylized sentence generation, we compare our method with several state-of-the-art methods that use unpaired stylized captions, including StyleNet (Gan et al., 2017), MSCap (Guo et al., 2019) and MemCap (Zhao et al., 2020). StyleNet is a single-style method that trains a separate model for each style. MSCap is a multi-style method that trains a single model to generate sentences in multiple styles. The results of single-style version and multi-style version of MemCap are both reported. For fair comparison, we evaluate both the single-style version and multi-style version of our model, denoted as “Ours_single” and “Ours_multi”. For MSCap and MemCap, the results are directly adopted from their original papers. For StyleNet, we adopt the reproduced results in (Guo et al., 2019). Since StyleNet and MSCap use ResNet-152 visual features, we also conduct experiments that use ResNet-152 features rather than scene graphs as the visual input. In the pre-training stage, the visual features of the images corresponding to the factual sentences are directly used. In the fine-tuning stage, we use a cross-modal retrieval model (Diao et al., 2021) to retrieve the images that are the

most similar to the stylized sentences from the training splits and use the features of the retrieved images as the visual input to the captioning model.

Table 1 shows the results of different methods on generating stylized sentences. It is interesting to observe that our method outperforms MemCap and MSCap on most evaluation metrics in both single-style and multi-style settings, which indicates that our method is capable of generating more fluent and stylized sentences for images, owing to the multi-pass decoding process with the rewards for conducting different types of actions by different neural modules. Particularly, the average perplexity of the generated sentences by our method is much lower than MemCap in romantic and humorous styles. Compared with the positive and negative sentences that use single adjectives or adverbs to express the sentiments, the romantic and humorous sentences include style-related phrases or clauses, which is more complex. So it is obvious that our method achieves better performance when dealing with complex sentences. Moreover, we observe that using scene graphs as the input to the captioning model leads to better performance than using visual features, which validates the effectiveness of the scene graph representation. When using either scene graphs or visual features as visual input, our method also outperforms the existing methods that use the same visual input.

4.4.2 Results of Generating Stylized Paragraphs

We compare our method with Neural Story Teller(NST) (Kiros et al., 2015), StyleNet (Gan et al., 2017) and DLN (Chen et al., 2019) when generating stylized paragraphs. Neural Story Teller first generates factual descriptions and then transfers the factual descriptions into stylized descriptions. StyleNet decomposes the parameter in the LSTM model into two groups and uses different parameters to learn the knowledge about the factual descriptions and stylized linguistic patterns, respectively. DLN assumes that a latent space exists and the images, factual descriptions and stylized descriptions can be represented in the latent space. By learning the mapping from images to the latent space and the mapping from the latent space to stylized descriptions, DLN is able to generate stylized descriptions for images.

The results of experiments on stylized paragraphs are shown in Table 2. We observe that our method generally outperforms DLN in terms of content similarity, SPICE as well as p and r , verifying the superiority of our method on generating sentences that better preserve the semantic information in the image. Our method also works better than DLN on capturing the linguistic styles, which shows the benefit of using multiple neural modules for stylized captioning.

4.5 Ablation Studies on Model Components

To evaluate the contribution of each component in our proposed method, we conduct ablation studies on generating stylized sentences under the single-style setting. The following variants of our full model are evaluated:

- **w/o rl**: To evaluate the contribution of reinforcement learning, we only fine-tune the three modules using the cross-entropy loss defined in Eq. (8) and the reinforcement learning is not used.
- **same reward**: To validate the effect of using different rewards for different types of actions, we replace the reward for generating part-of-speech tags, the reward for generating factual words and the reward for generating stylized words in Eq. (12) using the same reward $R(s_t, a_t) = \lambda_1(f_p(Y_{1:t}^s) - f_p(Y_{1:t-1}^s)) + (f_r(Y_{1:t}^s) - f_r(Y_{1:t-1}^s)) + \lambda_2 f_c(Y_{1:t}^s) + f_c(Y_{1:t-1}^s)$ during fine-tuning.
- **w/o syn,con**: To evaluate the advantage of using multiple neural modules for multi-pass decoding, a single module (i.e., the style module) is used to generate stylized sentences in a one decoding pass with the reward $r_g(a_t) = \lambda_1(f_p(Y_{1:t}^s) - f_p(Y_{1:t-1}^s)) + (f_r(Y_{1:t}^s) - f_r(Y_{1:t-1}^s)) + \lambda_2 f_c(Y_{1:t}^s) + f_c(Y_{1:t-1}^s)$.
- **w/o syn**: To evaluate the effect of the syntax module, the syntax module is removed and the captioning model performs single-pass decoding using the concept module and the style module. At each decoding step, the style module predicts the style tag. The word is predicted by the concept module if the style tag is 0, and by the style module if the style tag is 1.
- **w/o con**: To evaluate the effect of the concept module, we remove the concept module and only use the style module in the second decoding pass. During fine-tuning, the reward for the part-of-speech tags remains unchanged, and the reward for the words generated by the style module is $r_g(a_t) = \lambda_1(f_p(Y_{1:t}^s) - f_p(Y_{1:t-1}^s)) + (f_r(Y_{1:t}^s) - f_r(Y_{1:t-1}^s)) + \lambda_2 f_c(Y_{1:t}^s) + f_c(Y_{1:t-1}^s)$. The style tags generated by the syntax module are no longer used, since the style module generates both factual words and stylized words in the second decoding pass without distinguishing them.
- **w/o con +tag**: Compared to “w/o con”, the style module takes the embedding of the style tag as an additional input. Specifically, we use $\mathbf{x}_t^{high} = \mathbf{W}_d^{high} [e_{\hat{w}_{t-1}}; e_{\hat{v}_{t-1}}; \mathbf{c}_t^2; e_{\hat{u}_{t-1}}]$ as the input to the style module, where $e_{\hat{u}_{t-1}}$ denotes the embedding of the style tag.
- **POS template**: To validate the necessity of generating the sequence of the style tags and part-of-speech tags, we use pre-defined templates of style tags and part-of-speech tags to replace the prediction results of the syntax module. Each template is a sequence of POS tags with their

Table 2 Results of generating stylized paragraphs

Method	Visual Input	Style	CS	S	T	p	r	n_p	n_r
NST(Kiros et al., 2015)	ResNet	Lyrics	0.037	0.016	100.00%	0.041	0.044	0.680	0.750
StyleNet(Gan et al., 2017)			0.033	0.014	100.00%	0.038	0.038	0.570	0.670
DLN(Chen et al., 2019)			0.083	0.033	99.20%	0.080	0.115	1.250	1.920
Ours_feat			0.103	0.042	96.0%	0.290	0.109	1.250	1.697
Ours	Scene Graph		0.117	0.050	98.90%	0.107	0.114	1.283	1.937
NST(Kiros et al., 2015)	ResNet	Romance	0.088	0.039	100.00%	0.087	0.113	1.570	1.900
StyleNet(Gan et al., 2017)			0.012	0.005	100.00%	0.032	0.001	0.110	0.140
DLN(Chen et al., 2019)			0.151	0.058	95.40%	0.193	0.148	1.560	2.430
Ours_feat			0.147	0.064	92.0%	0.173	0.142	1.620	2.340
Ours	Scene Graph		0.152	0.073	93.90%	0.195	0.153	1.620	2.438
NST(Kiros et al., 2015)	ResNet	Humor	0.103	0.041	99.70%	0.097	0.143	2.220	2.440
StyleNet(Gan et al., 2017)			0.010	0.005	99.80%	0.024	0.001	0.120	0.150
DLN(Chen et al., 2019)			0.173	0.065	70.00%	0.205	0.182	2.320	2.990
Ours_feat			0.154	0.063	83.0%	0.197	0.143	1.970	2.320
Ours	Scene Graph		0.180	0.071	93.80%	0.213	0.110	2.305	3.021
NST(Kiros et al., 2015)	ResNet	Fairy tale	0.116	0.044	99.80%	0.116	0.145	2.470	2.440
StyleNet(Gan et al., 2017)			0.028	0.013	99.80%	0.045	0.026	0.340	0.460
DLN(Chen et al., 2019)			0.135	0.050	93.70%	0.194	0.125	1.290	2.060
Ours_feat			0.164	0.075	92.0%	0.290	0.133	1.160	1.990
Ours	Scene Graph		0.176	0.080	93.90%	0.312	0.136	1.287	2.100

The metrics CS, S and T stand for content similarity defined in (Chen et al., 2019), SPICE and transfer accuracy, respectively. The metrics p , r , n_p and n_r , defined in (Chen et al., 2019) measure the relevancy of the generated sentences to the image

associated style tags. We regard each template as a class, and use a template classifier that takes the embedding of the scene graph as input to predict the template. On the SentiCap dataset and FlickrStyle10K dataset, we use about 800 and 500 different templates from the training split, respectively.

- **w/o comm:** To evaluate the contribution of the information passing between the neural modules, we remove the attention module defined in Eq. (5). The output of the attention mechanism, i.e. c_t^{high} in Eq. (6) is replaced by a zero vector.

The results of the ablation studies are reported in Table 3. From these results, we have the following observations:

- The model performs worse on all the metrics when reinforcement learning is removed, which indicates that the reinforcement learning paradigm is capable of improving the performance of our model.
- When the rewards designed for conducting different types of actions are the same, the performance degrades, showing that using different rewards for different actions are beneficial for generating stylized captions by encouraging the modules to focus on their own tasks.

- By removing either or both of the concept module and the syntax module, the performance drops on both relevancy (measured by Bleu-3 and CIDEr), style accuracy (measured by cls) and fluency (measured by ppl). Though “w/o con +tag” uses the style tag as an additional input to the style module, “Ours” still achieves better performance than “w/o con+tag”, demonstrating the advantage of using two modules to generate the stylized sentence.
- “POS template” performs better than “Ours” in terms of perplexity, since the templates of the part-of-speech tags and the associating style tags are from the training splits of the datasets. However, “Ours” outperforms “POS template” in terms of both relevancy and stylishness, which indicates the advantage of using the style module to predict the part-of-speech tag sequences and style tag sequences.
- The perplexity on all four style increases when the attention module used for information passing between the low-level module and the high-level modules is removed (“w/o comm”), validating the importance of information passing between neural modules in generating fluent sentences.

Table 3 Results of ablation studies on SentiCap and FlickrStyle10K using single-style setting

Method	Positive				Negative			
	B-3	C	ppl (↓)	cls	B-3	C	ppl (↓)	cls
w/o rl	17.5	51.9	23.8	82.3	16.9	50.1	24.0	78.0
same reward	17.2	52.0	19.6	91.4	17.2	50.9	19.7	91.3
w/o syn, con	16.5	50.2	18.3	94.3	16.6	49.9	19.2	93.5
w/o syn	17.9	53.4	22.8	91.2	16.9	50.3	22.1	90.1
w/o con	17.9	52.3	15.4	95.3	17.1	58.9	16.7	93.9
POS template	17.9	54.3	12.7	97.5	17.1	59.0	13.2	92.4
w/o con +tag	17.7	52.2	12.9	98.4	17.2	58.9	15.6	93.6
w/o comm	18.0	52.9	13.8	98.7	16.8	58.2	15.7	93.2
Ours_single	18.1	54.6	13.0	99.2	19.7	59.3	14.3	93.8

Method	Romantic				Humorous			
	B-3	C	ppl (↓)	cls	B-3	C	ppl (↓)	cls
w/o rl	4.8	21.7	11.6	70.9	4.6	19.4	10.8	69.2
same reward	5.2	23.8	10.9	87.3	5.3	20.8	11.7	88.1
w/o syn, con	4.6	17.4	11.4	90.9	4.4	18.3	11.8	90.0
w/o syn	5.4	24.9	10.3	93.4	5.1	20.1	10.9	94.0
w/o con	5.3	25.6	10.7	94.6	5.2	21.3	10.3	94.0
POS template	5.6	26.8	12.6	95.3	5.4	22.0	10.9	92.7
w/o con +tag	5.3	27.2	13.7	94.5	5.5	21.9	11.0	94.5
w/o comm	5.4	27.3	13.9	97.3	5.8	23.4	15.4	96.8
Ours_single	5.9	27.4	13.3	97.9	5.8	23.5	15.2	99.0

“w/o rl” and “same reward” denote the model that is not trained using reinforcement learning and the model that is trained using the same reward for all actions, respectively. “w/o syn, con” denotes the model that only uses the style module to generate sentences. “w/o syn” and “w/o con” denote the model that removed the syntax module and the model that removed the concept module, respectively. “POS template” is the model that uses pre-defined POS templates to replace the POS sequences generated by the syntax module, and “w/o con+tag” denotes the model that removes the concept module and feeds the embedding of style tags to the style module. Please refer to Section 4.5 for more details

Table 4 Results of fine-tuning with different rewards on the positive style of the SentiCap dataset and the romantic style of the FlickrStyle10K dataset using single-style setting

r_{rel}	r_{cls}, r_{ppl}	Positive				Romantic					
		B-3	M	C	ppl (↓)	cls	B-3	M	C	ppl (↓)	cls
Bleu-3	✓	18.9	16.0	52.3	14.2	98.9	6.5	9.0	26.5	12.3	94.3
METEOR	✓	17.9	17.0	52.4	13.3	99.0	5.8	9.0	26.2	14.7	95.3
SPICE	✓	18.0	16.5	54.2	13.2	98.5	5.8	9.1	26.7	13.3	97.3
CIDEr	×	18.2	17.0	55.3	14.5	88.4	5.9	9.3	25.7	15.3	85.8
CIDEr	✓	18.1	16.8	54.6	13.0	99.2	5.9	9.4	27.4	13.3	97.9

4.6 Ablation Studies on Rewards

We conduct ablation studies on the rewards defined in Eq. (12) to evaluate the contribution of each reward. To validate the effect of the evaluation metric used to calculate r_{rel} , we replace the CIDEr score with other evaluation metrics, including Bleu-4, ROUGE-L and SPICE. The results are shown in lines 1–3 of Table 4. We observe that compared to other evaluation metrics, using CIDEr score as r_{rel} leads to better relevancy.

To evaluate the contributions of r_{cls} and r_{ppl} , we also conduct experiments that remove the two rewards and only use the CIDEr reward to optimize the three neural modules. From these results, we observe that though only using the CIDEr reward leads to better relevancy, the performance of the model drops significantly in terms of stylishness and fluency, which indicate that the stylishness reward and fluency reward are necessary for generating stylized sentences.

Table 5 Parameter analysis results of λ_1 and λ_2 on the positive style of the SentiCap dataset using single-style setting

(a) Bleu-3				(b) CIDEr				(c) ppl (\downarrow)				(d) cls			
$\lambda_1\lambda_2$	0.5	0.7	1.0	$\lambda_1\lambda_2$	0.5	0.7	1.0	$\lambda_1\lambda_2$	0.5	0.7	1.0	$\lambda_1\lambda_2$	0.5	0.7	1.0
0.5	18.1	17.9	18.1	0.5	53.2	53.8	54.6	0.5	13.3	13.2	13.0	0.5	85.3	90.3	99.2
0.7	17.7	17.3	16.8	0.7	52.3	52.6	46.2	0.7	11.7	12.3	11.4	0.7	84.5	95.4	97.2
1.0	17.3	16.7	16.3	1.0	52.0	52.3	45.6	1.0	9.5	10.4	10.2	1.0	84.4	92.0	97.6

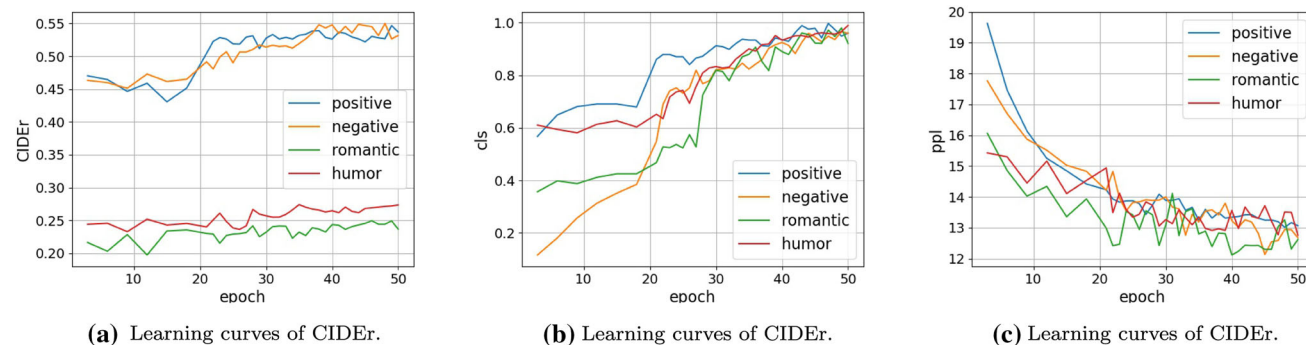


Fig. 3 The learning curves of our method in the multi-style setting on the SentiCap dataset and the FlickrStyle10K dataset

Table 6 Human evaluation results of our method on relevancy and stylishness of all the styles on both SentiCap and FlickrStyle10K. “Rel.” and “Sty.” are the abbreviations for relevancy and stylishness, respectively

Method	Style	Rel.	Sty.
StyleNet(Gan et al., 2017)	Positive	1.80	1.34
MemCap(Zhao et al., 2020)		2.23	1.82
Ours_single		2.40	1.87
StyleNet(Gan et al., 2017)	Negative	1.69	1.49
MemCap(Zhao et al., 2020)		2.29	1.70
Ours_single		2.34	1.71
StyleNet(Gan et al., 2017)	Romantic	1.77	1.40
MemCap(Zhao et al., 2020)		2.23	1.62
Ours_single		2.28	1.66
StyleNet(Gan et al., 2017)	Humorous	1.71	1.45
MemCap(Zhao et al., 2020)		2.10	1.57
Ours_single		2.13	1.61

4.7 Parameter Analysis

To analyze the effect of the hyper-parameters λ_1 and λ_2 defined in Eq. (12), we conduct additional parameter analysis by varying the values of λ_1 and λ_2 in {0.5, 0.7, 1.0} on the positive style of the SentiCap dataset. The results are shown in Table 5. From these results, we observe that though the perplexity of the sentences improve when the value of λ_1 increases to 0.7 or 1.0, the performance in terms of relevancy and stylishness significantly drops. When the value of λ_2 decreases, the model generates less stylized sentences.

To sum up, the optimal values of λ_1 and λ_2 are 0.5 and 1.0, respectively.

4.8 Convergence Analysis

To better understand the reinforcement learning process, we visualize the training process of our method in the multi-style setting. The model is trained with the cross-entropy loss in Eq. (8) in the first 20 epochs, and then trained with the self-critical loss \mathcal{L}_{rl} defined in Eq. (16). Figure 3 shows the learning curves of CIDEr, cls and ppl. We observe that the performance of our model increases when fine-tuning the model with reinforcement learning and the model converges after about 40 epochs, which demonstrate the effectiveness of the multi-module reinforcement learning paradigm.

4.9 Human Evaluation

We conduct human evaluation to further assess the performance of our method in generating stylized sentences. For all four linguistic styles, we randomly select 100 images from the test splits of SentiCap and FlickrStyle10K, and generate stylized sentences for each image using the single style version of our model. We also generate stylized sentences for the same images using StyleNet and MemCap, resulting in a total of 1200 stylized sentences to be evaluated. We evaluate the sentences generated by different methods in the same condition, and the human annotators are asked to assess each sentence in terms of relevancy and stylishness when they see the image and the sentence. The relevancy is scored from 0 (the sentence is totally unrelated to the image) to 3


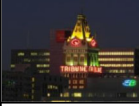


image	positive	negative
	gt: A delicious lunch of healthy food sits waiting on the table. w/o rl: A plate of <u>tasty</u> food on a table. Ours: A <u>great</u> bowl filled with <u>healthy</u> food is sitting on a table.	gt: Two containers one with a salad and another with some sort of disgusting food in it w/o rl: A group of food with food on a plate Ours: A white bowl of some <u>rotten</u> food on a <u>dirty</u> white plate
	gt: A magnificent view of the night sky and a lit clock tower. w/o rl: A clock tower is sitting in front of a <u>beautiful</u> building. Ours: A <u>great</u> image of a <u>beautiful</u> building with a clock tower.	gt: A clock tower is lit in the night ominous sky. w/o rl: A <u>lonely</u> street is in front of a street. Ours: A clock tower sits in front of a <u>lonely</u> building .
image	humorous	romantic
	gt: Black dog with red collar splashing in water like a dolphin . w/o rl: A black and white dog is running in the water . Ours: A black dog is running in the water <u>for fun</u> .	gt: A black dog with a red collar is jumping in the water to reach his lover . w/o rl: A black dog is swimming through the water. Ours: A <u>playful</u> dog is swimming in the water.
	gt: A person climbing a rock face like a lizard . w/o rl: A man is standing on a rock climbing on a rock <u>to finish the race</u> . Ours: A man is climbing a rock wall <u>looking for treasure</u> .	gt: A lone man climbing the side of a large rock wall looks up to the top . w/o rl: A man is riding a large rock <u>enjoying the mountain</u> . Ours: A man is climbing on a rock <u>to meet his lover</u> .

Fig. 4 Examples of stylized captions generated by the proposed method. “gt” denotes the ground-truth caption. “w/o rl” and “Ours” denote the captions generated by the model fine-tuned without reinforcement and our full model, respectively. The words and phrases that reflect the linguistic style are underlined

(the sentence well describes the image), and the stylishness is scored from 0 (the sentence has no linguistic style) to 2 (the sentence reflects the desired style appropriately). Table 6 reports the average scores of each style on the SentiCap and FlickrStyle10K datasets. Our method scores between 2.13 and 2.40 in terms of relevancy and scores between 1.61 and 1.87 in terms of stylishness, which shows that our method is able to generate satisfactory stylized captions. Our method outperforms StyleNet on both relevancy and stylishness, further demonstrating the advantage of multi-pass decoding process via multiple cooperative modules for stylized image captioning.

4.10 Qualitative Results

We show some exemplars of the stylized sentences generated by the model that is only fine-tuned using cross-entropy loss (“w/o rl”) and the full model (“Ours”) in Fig. 4. As illustrated in Fig. 4, the descriptions generated by our method well describe the content of the images. Compared with “w/o rl”, the sentences generated by our full model are more fluent and reflect the linguistic style more appropriately.

5 Conclusion and Future Work

We have presented a novel method that learns multiple cooperative neural modules using reinforcement learning scheme for stylized image captioning. Through the multi-pass decoding process implemented by the multiple neural modules, our method significantly improves the relevancy, stylishness and fluency of the generated sentences. Thanks to the information passing between neural modules and the rewards designed for

different types of actions, our method is able to effectively learn the syntactic structure, infer the semantic concepts and express the desired linguistic style. Extensive experiments with different stylized corpora have demonstrated the effectiveness of the proposed method. In the future, we are going to design more effective architectures that facilitate the information passing between different neural modules.

Acknowledgements If you’d like to thank anyone, place your comments here and remove the percent signs.

References

- Anderson, P., Fernando, B., Johnson, M., & Gould, S. (2016). Spice: Semantic propositional image caption evaluation. In *European conference on computer vision*, (pp. 382–398), Springer
- Andrew Shin, Y.U., & Harada, T. (2016). Image captioning with sentiment terms via weakly-supervised sentiment dataset. In C. Richard, E.R.H. Wilson, W.A.P. Smith (eds). *Proceedings of the british machine vision conference (BMVC)*, (pp 53.1–53.12), BMVA Press
- Banerjee, S., & Lavie, A. (2005). 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).
- Chen, C. K., Pan, Z., Liu, M. Y., & Sun, M. (2019). Unsupervised stylish image description generation via domain layer norm. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33, 8151–8158.
- Chen, T., Zhang, Z., You, Q., Fang, C., Wang, Z., Jin, H., & Luo, J. (2018). “factual” or “emotional”: Stylized image captioning with adaptive learning and attention. In *Proceedings of the european conference on computer vision (ECCV)*, (pp. 519–535).
- Dethlefs, N., & Cuayáhuil, H. (2010). Hierarchical reinforcement learning for adaptive text generation. In *Proceedings of the 6th international natural language generation conference, association for computational linguistics*, (pp. 37–45).

- Diao, H., Zhang, Y., Ma, L., & Lu, H. (2021). *Similarity reasoning and filtration for image-text matching*. Technical Report
- Fu, Z., Tan, X., Peng, N., Zhao, D., & Yan, R. (2018). Style transfer in text: exploration and evaluation. In *Thirty-second AAAI conference on artificial intelligence*, (pp. 663–670).
- Gan, C., Gan, Z., He, X., Gao, J., & Deng, L. (2017). Stylenet: Generating attractive visual captions with styles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 3137–3146).
- Gu, J., Cai, J., Wang, G., & Chen, T. (2018). Stack-captioning: Coarse-to-fine learning for image captioning. In *Thirty-second AAAI conference on artificial intelligence*, (pp. 6837–6844).
- Guo, L., Liu, J., Lu, S., & Lu, H. (2019). Show, tell and polish: Ruminant decoding for image captioning. *IEEE Transactions on Multimedia*, 22(8), 2149–2162.
- Guo, L., Liu, J., Yao, P., Li, J., & Lu, H. (2019). Mscap: Multi-style image captioning with unpaired stylized text. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 4204–4213).
- Guo, L., Liu, J., Zhu, X., He, X., Jiang, J., & Lu, H. (2020). Non-autoregressive image captioning with counterfactuals-critical multi-agent learning. In *Proceedings of the twenty-ninth international joint conference on artificial intelligence*, (pp. 767–773).
- Honnibal, M., Montani, I., Van Landeghem, S., & Boyd, A. (2020). spaCy: Industrial-strength natural language processing in python. <https://doi.org/10.5281/zenodo.1212303>
- Huang, Q., Gan, Z., Celikyilmaz, A., Wu, D., Wang, J., & He, X. (2019). Hierarchically structured reinforcement learning for topically coherent visual story generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33, 8465–8472.
- Johnson, J., Krishna, R., Stark, M., Li, L.J., Shamma, D., Bernstein, M., & Fei-Fei, L. (2015). Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 3668–3678).
- Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), association for computational linguistics*, (pp. 1746–1751), Doha, Qatar, <https://doi.org/10.3115/v1/D14-1181>
- Kingma, D.P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *3rd international conference on learning representations*
- Kiros, R., Zhu, Y., Salakhutdinov, R.R., Zemel, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Skip-thought vectors. In *Advances in neural information processing systems*, (pp. 3294–3302).
- Kong, X., Xin, B., Wang, Y., & Hua, G. (2017). Collaborative deep reinforcement learning for joint object search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 1695–1704).
- Krause, J., Johnson, J., Krishna, R., & Fei-Fei, L. (2017). A hierarchical approach for generating descriptive image paragraphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 317–325).
- Li, X., & Jiang, S. (2019). Know more say less: Image captioning based on scene graphs. *IEEE Transactions on Multimedia*, 21(8), 2117–2130.
- Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C.L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, (pp. 740–755), Springer.
- Liu, C., He, S., Liu, K., & Zhao, J. (2019). Vocabulary pyramid network: Multi-pass encoding and decoding with multi-level vocabularies for response generation. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, (pp. 3774–3783).
- Mathews, A., Xie, L., & He, X. (2018). Semstyle: Learning to generate stylized image captions using unaligned text. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 8591–8600).
- Mathews, A.P., Xie, L., & He, X. (2016). Senticap: Generating image descriptions with sentiments. In *Thirtieth AAAI conference on artificial intelligence*, (pp. 3574–3580).
- Panait, L., & Luke, S. (2005). Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-agent Systems*, 11(3), 387–434.
- Papineni, K., Roukos, S., Ward, T., & Zhu, W.J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics, Association for Computational Linguistics*, (pp. 311–318).
- Peng, B., Li, X., Li, L., Gao, J., Celikyilmaz, A., Lee, S., & Wong, K.F. (2017). Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, (pp. 2231–2240).
- Rennie, S.J., Marcheret, E., Mroueh, Y., Ross, J., & Goel, V. (2017). Self-critical sequence training for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 7008–7024).
- Slevc, L. R. (2011). Saying what's on your mind: Working memory effects on sentence production. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 37(6), 1503.
- Stolcke, A. (2002). Srlm—an extensible language modeling toolkit. In *Proceedings of ICSLP*, (pp. 901–904).
- Sun, X., Lu, W. (2020). Understanding attention for text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics*, (pp. 3418–3428).
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. Cambridge: MIT Press.
- Vedantam, R., Lawrence Zitnick, C., Parikh, D. (2015). Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 4566–4575).
- Wang, X., Chen, W., Wu, J., Wang, Y.F., Yang Wang, W. (2018). Video captioning via hierarchical reinforcement learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 4213–4222).
- Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3), 229–256.
- Wu, L., Xu, M., Wang, J., & Perry, S. (2019). Recall what you see continually using gridlstm in image captioning. *IEEE Transactions on Multimedia*, 22(3), 808–818.
- Xia, Y., Tian, F., Wu, L., Lin, J., Qin, T., Yu, N., & Liu, T.Y. (2017). Deliberation networks: Sequence generation beyond one-pass decoding. In *Advances in neural information processing systems*, (pp. 1784–1794).
- Xu, N., Zhang, H., Liu, A. A., Nie, W., Su, Y., Nie, J., & Zhang, Y. (2019). Multi-level policy and reward-based deep reinforcement learning framework for image captioning. *IEEE Transactions on Multimedia*, 22(5), 1372–1383.
- Xu, W., Yu, J., Miao, Z., Wan, L., Tian, Y., Ji, Q. (2020). Deep reinforcement polishing network for video captioning. *IEEE Transactions on Multimedia*, 23, 1772–1784.
- Yang, X., Tang, K., Zhang, H., Cai, J. (2019). Auto-encoding scene graphs for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 10685–10694).
- Zellers, R., Yatskar, M., Thomson, S., Choi, Y. (2018). Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 5831–5840).

- Zhao, W., Wu, X., & Zhang, X. (2020). Memcap: Memorizing style knowledge for image captioning. In *The thirty-fourth AAAI conference on artificial intelligence*, (pp. 12984–12992).
- Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, (pp. 19–27).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.