

Noise Augmented Double-Stream Graph Convolutional Networks for Image Captioning

Lingxiang Wu^{ID}, Min Xu^{ID}, Member, IEEE, Lei Sang, Ting Yao^{ID}, Member, IEEE,
and Tao Mei^{ID}, Fellow, IEEE

Abstract—Image captioning, aiming at generating natural sentences to describe image contents, has received significant attention with remarkable improvements in recent advances. The problem nevertheless is not trivial for cross-modal training due to the two challenges: 1) image detectors often consider only salient areas in an image and seldom explore the rich background context; 2) the language model is highly vulnerable to small but intentional perturbation attacks. To alleviate these issues, we propose the Noise Augmented Double-stream Graph Convolutional Networks (NADGCN) that novelly exploits the additional background context and enhances the generalization of the language model. Technically, NADGCN capitalizes on grid-stream GCN as a supplementary to the region stream, following the recipe that a rescaled grid graph can encode the relationship across grid areas over the full image rather than salient areas only. Moreover, we devise a noise module and integrate into the double-stream GCN to augment the capability of the basic generator. Such noise module introduces adaptive noise into the Recurrent Neural Networks (RNN) and is learnt through regarding the module as an agent with a stochastic Gaussian policy in Reinforcement Learning (RL). Extensive experiments on MSCOCO validate the design of the grid-stream GCN and the noise agent, and our generator outperforms the comparative baselines clearly.

Index Terms—Captioning, graph convolutional networks, adaptive noise.

I. INTRODUCTION

IMAGE captioning is to automatically generate a descriptive utterance (usually a sentence) that describes the image content, and has emerged as a fundamental task in visual understanding [1]–[7]. Practical applications of image caption generation include helping people with visual impairments by transforming visual signals into understandable information, and benefiting semantic-level photo organization. The typical framework of neural captioning models [8]–[10] is essentially

Manuscript received June 14, 2020; revised September 16, 2020; accepted October 30, 2020. Date of publication November 9, 2020; date of current version August 4, 2021. This article was recommended by Associate Editor M. Shehata. (*Corresponding author: Min Xu*)

Lingxiang Wu, Min Xu, and Lei Sang are with the School of Electrical and Data Engineering, University of Technology Sydney, Ultimo, NSW 2007, Australia (e-mail: lingxiang.wu@student.uts.edu.au; min.xu@student.uts.edu.au; lei.sang@student.uts.edu.au).

Ting Yao and Tao Mei are with JD AI Research, Beijing 100020, China (e-mail: tingyao.ustc@gmail.com; tmei@jd.com).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TCSVT.2020.3036860>.

Digital Object Identifier 10.1109/TCSVT.2020.3036860



Generation: a row of motorcycles parked next to each other

GT: A number of motorbikes parked on an alley

Generation: a man and a woman holding a white snowboard and a black and white photo

GT: A man holding a snowboard next to a man in scary costume

Fig. 1. Suboptimal captions. Left: The generation omits the background *alley*. Right: The generation ends with irrelative phrase *a black and white photo* when we inject noise in the model intentionally.

an encoder-decoder structure. An image is first encoded into one feature vector or a set of region features via Convolutional Neural Network (CNN) or Region-based CNN (R-CNN), and a decoder of Recurrent Neural Network (RNN) is employed to generate a natural sentence. Despite having impressive performances by recent approaches in terms of quantitative scores, qualitative analysis shows that the generated captions still result in undesired effects frequently. The difficulty originates from two aspects: 1) Image encoders [11], [12] tend to emphasize attentive regions or build visual graphs on the detected regions. As such, the methods seldom explore the rich background context. Taking the image shown in the left part of Figure 1 as an example, the generation fails to describe the background information “alley”. 2) The language decoders may suffer from robustness problem since they are highly vulnerable to small but intentional perturbation attacks. As depicted in the right part of Figure 1, the generation ends with an irrelative phrase “a block and white photo” when injecting uniform noise in the RNN decoder.

We propose to mitigate the first issue through double-stream Graph Convolutional Networks (GCN) framework. In addition to the standard GCN on region level, we newly build a grid visual graph in grid-stream GCN. Such grid graph leverages the feature map of the full image and encodes the relationship across visual grids. As a result, the double-stream design takes the advantages of both the full context holistically and salient areas locally. To address the second issue, we devise an additive noise module to avoid skewed

optimization and improve the generalization capability of language model. In the literature, there have been several techniques of introducing noise in neural network training to enhance the robustness of the network. The noise would play a role in the inputs [13], weights [14], [15], gradients [16], and even the activation functions [17]. Nevertheless, the variables of the widely-adopted noise (e.g., naïve Gaussian noise) are often fixed in the approaches. In this case, the noise could be more easily learnt and then masked off by the model. In contrast, we present an adaptive Gaussian noise to dynamically predict the mean and standard deviation, and manipulate the RNN transition states for image captioning.

By consolidating the idea of leveraging the full context of an image and enhancing the capability of language model for image captioning, we present a new Noise Augmented Double-stream Graph Convolutional Networks (NADGCN) framework. Specifically, NADGCN builds two types of graph, i.e., region-stream graph and grid-stream graph. The former is solely based on the detected regions in an image and encodes the relationship between regions. Instead, the latter takes all the grids of the image into account to fully explore the relationship across the context. To normalize the size of the two graphs, we rescale the grid graph via differentiable graph pooling, where we learn a probabilistic assignment matrix that maps the nodes to clusters in the grid graph. Moreover, we integrate an additive Gaussian noise module into the basic RNN caption generator. The noise module dynamically predicts the mean and standard deviation with respect to the current hidden state, visual attended feature and the word embedding at each time step. Technically, the noise module is learnt through regarding the module as an agent with a stochastic Gaussian policy and the noise generation as an action in reinforcement learning. In between, the sampled caption is generated by the basic RNN generator with the noise module while the estimated sentence is greedily decoded without noise. As such, the generator tends to be resistant to the noise and increases the probability of the sampled captions with higher scores at the same time. The whole framework of NADGCN could be jointly learnt and optimized in an end-to-end manner.

The contributions for this paper are as follows.

- We propose a novel double-stream GCN framework, which consists of a region-level graph and a grid-level graph, to leverage the relationship both across attentive regions and over full image context. This solution also leads to the elegant view of how graphs of different scales should be rescaled for fusion, which is a problem not yet fully understood in the literature.
- We devise an adaptive noise addition for caption generator and present a new strategy to approximate the noise policy. The adaptive noise agent is readily pluggable to any conditional recurrent language models.
- Extensive experiments on MSCOCO demonstrate that our methods outperform the comparative models clearly and can achieve promising results compared with the state-of-the-art approaches.

The remaining sections are organized as follows. Section II describes related work on image captioning, GCN and reinforcement learning. Section III presents our double-stream GCN architecture for image captioning, while Section IV formulates the adaptive noise agent. Section V provides empirical evaluation on MSCOCO dataset, followed by the conclusions in Section VI.

II. RELATED WORK

A. Image Captioning

Image captioning methods can be roughly summarized into three categories: retrieval-based methods [18]–[20], template-based methods [21], [22] and sequence generation methods [24]–[26], [71]. Recent research efforts focus on the sentence generation methods, which were mainly based on a CNN-RNN framework. In [71], Vinyals *et al.* proposed to encode the image with CNN and decode the representation into a word sequence with RNN. This method provided a strategy for end-to-end training. In [27], unequal weights are assigned to different words during the training. In [28], image attributes from a separate predictor were used to provide semantic context. On this basis, a bunch of methods [25], [29], [30] tended to integrate attention mechanism into the the vanilla CNN-RNN framework. In [25], Xu *et al.* proposed soft attention and hard attention. In [30], a number of review steps were preformed between the encoder and the attentive decoder. Bottom-up and top-down attention were integrated for image captioning as well as visual question answering in [29]. In [31], an attribute-driven attention model was proposed for image captioning. Besides the CNN-RNN framework, GCN were also applied for captioning [12], [32]. Afterwards, some methods [33]–[35] tended to tackle captioning task with Reinforcement Learning. Some other works focused on special cases such as styled caption generation [36] and dense captioning [37], [38]. In [36], Gan *et al.* proposed to generate humorous/romantic descriptions by updating certain LSTM parameters while training on a second corpus. In dense captioning [37], each region was annotated with a caption. In [2], Yu *et al.* applied a multimodel transformer for image captioning, where the model relies entirely on the attention mechanism instead of the RNN to assess the dependencies between the input and output. In [1], Feng *et al.* proposed to describe an image with both in-domain and out-of-domain vocabulary via a revision mechanism. Different from [1] which focuses on novel objects, our method focus on generating representative description with only in-domain data. Reference [39] exploited the memory network [40] in decoder for visual narrating, especially video captioning and visual storytelling. In [41], Xu *et al.* worked on the multi-model space and generate video captions with a dual-stream RNN framework. Different from all the above methods, we focus on the image caption generation with the double-stream GCN encoder and noise augmented decoder. We construct two types spatial graphs for the image. Moreover, for the first time, we introduce an adaptive noise module in the RNN caption generator and propose a novel RL method to train this module.

B. Graph Convolutional Network

Graph Neural Networks [42] were introduced to combine graph structure data with neural networks. Recently, GCN which extend CNN to aggregate information from graph structure data have received increasing research attention. There are two streams of GCN construction: spectral GCN [43], [44] and spatial GCN [11], [45], [46]. Spectral GCN is based on the spectrum of graph Laplacian. Features of the graph are firstly transformed using Fourier transform and the convolutions are performed in the spectral domain. Spatial GCN approaches define convolutions directly on general graphs, operating on spatially close neighbors. Our method falls into the latter. A number of methods exploited GCN in computer vision applications. In [47], a spatial-temporal GCN was exploited for human action recognition. Brown *et al.* [48] applied GCN for Visual Question Answering (VQA) and learned a graph structure automatically. Zhu *et al.* [49] proposed multi-layer graphs and cross-modal knowledge reasoning for VQA. In [50], a Visual Reasoning and Attention Network (VRANet) was proposed to enhance the visual representations for cross-modal reasoning and retrieval. Yao *et al.* [12] applied a spatial graph and a semantic graph for image captioning. Different from [12], we develop two spatial graphs on region level and grid level, instead of using any external information. Moreover, we rescale the graph before the combination with the graph pooling techniques.

C. Reinforcement Learning

Deep reinforcement learning is used in a wide range of sequential decision making problems [51], [52]. The standard Reinforcement Learning (RL) [53] framework consists of an agent interacting with an environment, executing a series of actions and aiming to maximize the cumulative rewards. Recently, several attempts have been made to apply Reinforcement Learning, especially the policy gradient method, to image captioning and sequence generation tasks. Generally, the recurrent model is viewed as an agent where the action is to predict the next word. The parameters of the generator define a policy and the reward can be any evaluation metrics.

In [54], Ranzato *et al.* first introduced RL into an RNN-based sequence model, which mixed together the cross-entropy loss and the REINFORCE objective in training process. After that, Liu *et al.* [55] proposed to replace the mixed training with Monte Carlo rollouts, and they used the policy gradient method to optimize a combination of two NLP metrics. Rennie *et al.* [33] used the classical REINFORCE [56] algorithm and proposed a novel baseline obtained by the current model. This self-critic training method is efficient while maximizing the evaluation metric CIDEr [57]. Besides, Actor-Critic method was introduced to image captioning in [34], where a policy network and a value network worked collaboratively to generate captions. In this work, we apply the RL techniques to optimize the additive noise module. Different from the existing RL captioning framework, we regard the noise module as an agent and generating noise as the action. Parameters in the noise agent define a policy, while the double-stream GCN generator is viewed as a part of the environment.

To approximate the policy, we introduce a variant of the REINFORCE algorithm [33], [56].

III. DOUBLE-STREAM GRAPH CONVOLUTIONAL NETWORKS

To leverage the full context and salient areas, we develop the double-stream GCN shown in Fig. 2, which consists of a region stream graph and a rescaled grid stream graph.

A. Problem Formulation

Formally, a captioning system receives an image \mathbf{I} as the input and is required to output a sentence \mathbf{S} to describe the image content. \mathbf{S} can be represented as a sequence of words, $\mathbf{S} = \{\mathbf{w}_1, \mathbf{w}_2 \dots \mathbf{w}_N\}$, where \mathbf{w}_t denotes the t -th word. During the training, (\mathbf{I}, \mathbf{S}) is given as a training pair, and the captioning model (parameterized by θ) can be optimized by minimizing the cross entropy loss:

$$L(\theta) = - \sum_{t=1}^N \log p(\mathbf{w}_t | \mathbf{I}, \mathbf{w}_0, \dots, \mathbf{w}_{t-1}; \theta). \quad (1)$$

We formulate our basic model as a double-stream-GCN-RNN structure. We encode the internal image feature by the double-stream spatial GCN, which contains a region-stream graph $G_r = (\mathbf{V}_r, \mathbf{E}_r, \mathbf{X}_r)$ and a grid-stream graph $G_g = (\mathbf{V}_g, \mathbf{E}_g, \mathbf{X}_g)$. \mathbf{V} and \mathbf{E} denote node sets and edge sets respectively, and \mathbf{X} denotes the node feature matrix. After the necessary graph rescaling and graph combination, we feed the integrated visual feature into the RNN sentence generator.

B. Region-Stream Graph

As for G_r , a spatial graph is established to encode relationships among detected regions. We regard each image region as a node and represent the node by the region level feature. Technically, we use Faster R-CNN [58] to detect salient image regions and objects. Region of interest (RoI) pooling is utilized to extract the region level feature for each object. The edges are established according to spatial relationships between every two regions, and the edges are labeled with the manually designed class numbers as [12]. Specifically, class 1 “inside” and class 2 “cover” are established if object o_i is fully covered with object o_j . Class 3 “overlap” is established if the intersection over union (IoU) between o_i and o_j are larger than 0.5. Then we can compute the ratio α_{ij} between the relative distance and the diagonal length of the whole image, and the relative angle δ_{ij} between two bounding boxes. Class 4-11 can be established as $\lceil \delta_{ij}/45^\circ \rceil + 3$ and $IoU < 0.5$. Otherwise, no edge is established.

C. Grid-Stream Graph

Another spatial graph, G_g , is established to encode relationships among grids over the full image. The grid graph is constructed over the full image’s CNN feature map. We regard each instance on the feature map as a node and represent the node with the instance’s feature vector. We establish edges within each node’s 8-connected neighborhood, and edges are labeled as class number 1 ~ 9 as shown in Fig. 2.

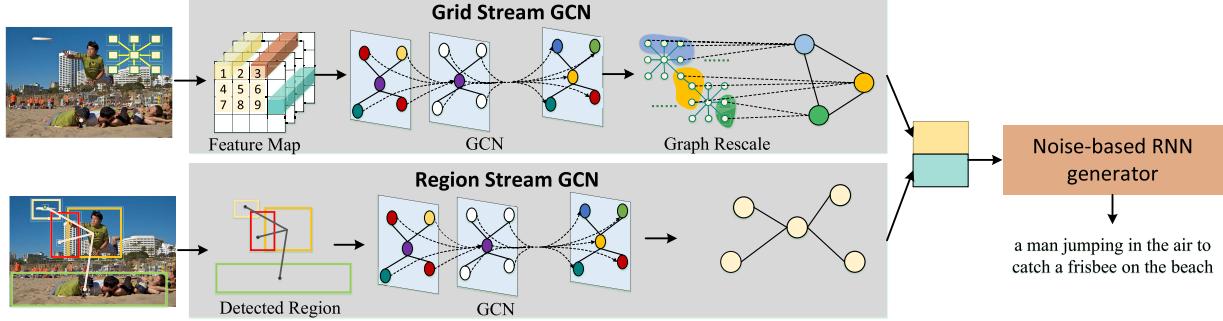


Fig. 2. The double-stream GCN consist of a region graph and a rescaled grid graph.

As each instance in the feature map can correspond to a visual receptive field in the raw image, this design can connect the neighboring parts in the full image. The effect of the grid graph is experimentally demonstrated in Section V-B.1.

D. GCN for Directed Labeled Graphs

With the constructed graph, we propagate information across edges of each graph via GCN. In this way, we get a high-order node representation \mathbf{X}^{k+1} by considering the connection among graph nodes:

$$\mathbf{X}^{k+1} = \text{GCN}_{\text{emd}}(\mathbf{X}^k, \mathbf{A}), \quad (2)$$

where k is the layer index, and \mathbf{A} indicates the graph adjacency matrix.

In this part, we exploit the GCN for directed labeled graphs [59], so as to consider more dedicate information in the graph:

$$\mathbf{x}_i^{k+1} = \text{ReLU} \left(\sum_{j \in \mathcal{N}(i)} g_{i,j}^k \left(\mathbf{W}_{\text{dir}(i,j)}^k \mathbf{x}_j^k + \mathbf{b}_{\text{lab}(i,j)}^k \right) \right). \quad (3)$$

\mathbf{x}_i^k is the feature vector of node v_i before embedding, and $\mathcal{N}(i)$ represents the set of neighbors of node v_i . $\text{dir}(i, j)$ includes three types of edges, i.e., the edge from v_i to v_j , the edge from v_j to v_i and the self loop edge. The edge labels are explicitly encoded in the bias vector $\mathbf{b}_{\text{lab}(i,j)}$, where $\text{lab}(i, j)$ indicates the edge label as describe in Section III-B and Section III-C. Parameters are not shared among different edges and labels. $g_{i,j}^k$ is a scalar gating mechanism:

$$g_{i,j}^k = \sigma \left(\hat{\mathbf{W}}_{\text{dir}(i,j)}^k \mathbf{x}_j^k + \hat{\mathbf{b}}_{\text{lab}(i,j)}^k \right), \quad (4)$$

where σ is the sigmoid function. $\hat{\mathbf{W}}_{\text{dir}(i,j)}$ and $\hat{\mathbf{b}}_{\text{lab}(i,j)}$ are the weight and bias for the gate. In this way, the new node representation \mathbf{x}_i^{k+1} can integrate information from neighboring nodes as well as labeled edges.

E. Grid Graph Rescaling

A challenge comes when we combine the two-stream graphs, as the grid graph contains much more nodes than the region one. To avoid an unbalanced combination, we rescale the grid graph via differentiable graph pooling. Inspired by the pooling strategy in graph classification [60], [61], we map the nodes in the grid graph into clusters via an assignment matrix.

The assignment matrix can be learned automatically given the \mathbf{X}_g^n and \mathbf{A}_g :

$$S = \text{GCN}_{\text{cluster}} \left(\mathbf{X}_g^n, \mathbf{A}_g \right). \quad (5)$$

The $\text{GCN}_{\text{cluster}}$ refers to:

$$s_i = \text{softmax} \left(\sum_{j \in \mathcal{N}(i)} (\mathbf{W}_{\text{cluster}} \mathbf{x}_j + \mathbf{b}_{\text{cluster}}) \right), \quad (6)$$

where the softmax function is applied in a row-wise fashion.

Then, we get a rescaled graph containing cluster nodes via:

$$\mathbf{X}_g^{n'} = \mathbf{S}^T \mathbf{X}_g^n, \quad (7)$$

$$\mathbf{A}_g' = \mathbf{S}^T \mathbf{A}_g \mathbf{S}. \quad (8)$$

Afterwards, we concatenate the region graph and the rescaled grid graph as $\mathbf{U} \in \mathbb{R}^{m \times d}$, and then fed the visual representation into the RNN generator.

IV. ADAPTIVE NOISE AGENT

In this section, we introduce the additive Gaussian noise module which is integrated into the basic RNN caption generator as shown in Fig. 3.

A. Basic RNN Generator

To introduce the basic RNN generator, we take the top-down attention generator [29] as an example, which contains an attention LSTM (A-LSTM) and a language LSTM (L-LSTM) at each step. The input for the A-LSTM consists of the previous output of the L-LSTM, the mean-pooled visual feature $\bar{\mathbf{u}} = \frac{1}{m} \sum_m \mathbf{u}_i$ and the previously generated word \mathbf{w}_t :

$$\mathbf{h}_t^1 = \text{lstm} \left([\mathbf{h}_{t-1}^2, \bar{\mathbf{u}}, \mathbf{W}_E \mathbf{w}_t] \right), \quad (9)$$

where \mathbf{W}_E is a word embedding matrix.

Given the output \mathbf{h}_t^1 , the visual feature with attention is calculated as:

$$\begin{aligned} a_{i,t} &= \mathbf{W}_a \tanh \left(\mathbf{W}_{va} \mathbf{u}_i^k + \mathbf{W}_{ha} \mathbf{h}_t^1 \right), \\ \lambda_t &= \text{soft max}(\mathbf{a}_t), \\ \mathbf{e}_t &= \sum_{i=1}^m \lambda_{i,t} \mathbf{u}_i^k. \end{aligned} \quad (10)$$

Then, we can get the output of the L-LSTM:

$$\mathbf{h}_t^2 = \text{lstm} \left([\mathbf{e}_t, \mathbf{h}_t^1] \right). \quad (11)$$

We get the predicted word \mathbf{w}_{t+1} through \mathbf{h}_t^2 followed by a fully connected layer and a softmax layer.

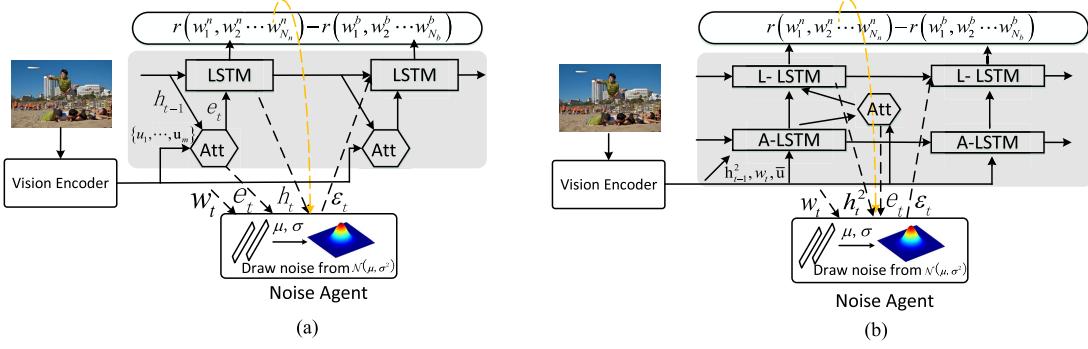


Fig. 3. Noise agent on two types of baselines. (a) The noise agent is added in the one layer soft attention generator. (b) The noise agent is added in the Top-Down generator.

B. Gaussian Noise Agent

In the literature, noise was introduced into the parameter space or the transition hidden states in the RNN, through which the robustness of the network can be enhanced. While, the variables of the widely-adopted noise are often fixed and can rarely be adaptive with respect to variant states. In a recent work, translations with a higher log-probability can be found by injecting unstructured noise into RNN hidden states [62].

In this work, we present an adaptive Gaussian noise module which can predict the mean and standard deviation, and manipulate the RNN transition states for image captioning. Our caption generator contains a basic RNN generator and a noise module. In Fig. 3, we present the details in two cases. With the Top-Down generator as an example, the noise module generates noise with respect to the RNN hidden state \mathbf{h}_t^2 , attention \mathbf{e}_t and the previous word \mathbf{w}_t at each step. The noise module outputs a noise vector ε_t which can be added to the transition state \mathbf{h}_t^2 as follows:

$$\hat{\mathbf{h}}_t^2 = \mathbf{h}_t^2 + \varepsilon_t, \quad (12)$$

$$\pi_\phi : \varepsilon_t \sim \mathcal{N}(\mu, \sigma^2), \quad (13)$$

$$\mu = \tanh \left(\mathbf{W}_\mu \left[\mathbf{h}_t^2, \mathbf{e}_t, \mathbf{W}_E \mathbf{w}_t \right] + \mathbf{b}_\mu \right), \quad (14)$$

$$\sigma = \text{sigmoid} \left(\mathbf{W}_\sigma \left[\mathbf{h}_t^2, \mathbf{e}_t, \mathbf{W}_E \mathbf{w}_t \right] + \mathbf{b}_\sigma \right). \quad (15)$$

μ and σ denotes the mean and standard deviation of the distribution. In this way, μ and σ can be modeled by the MLP given the inputs aforementioned, where tanh and sigmoid are utilized as the activation function respectively.

With reference to [62], uncertainty is often greatest when predicting earlier symbols and gradually decreases as more and more context becomes available, and in [16] adding annealed Gaussian noise performs better than using fixed Gaussian noise. We adopt a strategy where we anneal the noise along the decoding process:

$$\hat{\mathbf{h}}_t^2 = \mathbf{h}_t^2 + \frac{\varepsilon_t}{t}. \quad (16)$$

Additionally, we clamp noises into the range $(-\gamma, \gamma)$ to avoid inferior perturbation.

C. Noise Critic Training

In order to train the noise module, we propose a novel RL framework. The noise module is viewed as an agent with a stochastic gaussian policy, whose action is to generate noise from the Gaussian distribution. The other networks such as the basic generator are fixed and regarded as parts of the environment. We approximate the Gaussian policy π_ϕ with the parametric function in Equation 14 and Equation 15. The evaluation metric, e.g., CIDEr, is used as the episode reward. The training goal is to minimize the negative expected rewards:

$$L(\phi) = -\mathbb{E}_{S \sim \pi_\phi} \left[\sum r_{\varepsilon_t} \right]. \quad (17)$$

Policy gradient algorithms are generally exploited to approximate the policy. REINFORCE [56] uses Monte Carlo sample to get the episode reward, i.e. play out the whole episode to compute the total reward r_S . With a baseline for variance reduction, the gradient can be computed as:

$$\Delta_\phi L(\phi) \approx -(r_S - b) \Delta_\phi \log p_\phi(\varepsilon). \quad (18)$$

The baseline can be any function, even a random variable. In this way, the gradients encourage the parameters increase in the direction proportional to the reward, which makes actions with high rewards more likely.

We propose a noise-critic training following SCST [33], where we use the basic generation (non-noise output) as the baseline. The gradients in the noise module can be calculated as:

$$\Delta_\phi L(\phi) \approx -(r_{S_n} - r_{S_b}) \Delta_\phi \log p_\phi(\varepsilon), \quad (19)$$

where r_{S_n} denotes the CIDEr reward for noise-augmented generation, r_{S_b} is the reward for basic (noiseless) generation. As a result, the sampled noise that get a higher rewards than the basic output can be encouraged, but inferior noise can be suppressed. In Gaussian distribution, the log probability of the generated noise is:

$$\log p_\phi(\varepsilon) = -\frac{(\varepsilon - \mu)^2}{2\sigma^2} - \log \sigma - \log \sqrt{2\pi}. \quad (20)$$

In this design, the generator increases the probability of the sampled captions with higher rewards. We avoid learning a baseline network by regarding the non-noise generation as

TABLE I
PERFORMANCE OF DOUBLE-STREAM GCN AND VARIANT MODELS ON MSCOCO KARPATHY TEST SPLIT

Model variants	Cross entropy loss					Self critic training				
	C↑	M↑	B4↑	B1↑	R↑	C↑	M↑	B4↑	B1↑	R↑
Resnet	107.8	26.8	33.9	74.8	55.3	120.1	27.6	36.1	78.8	57.0
GCN-g	112.9	27.5	35.2	76.3	56.2	124.4	28.2	37.2	79.8	57.8
GCN-r	115.3	27.8	36.7	76.9	56.9	126.0	28.2	37.9	80.1	58.1
GCN-r-g	114.4	27.6	35.9	76.8	56.7	125.3	28.1	37.6	79.9	57.9
Double-stream GCN	115.6	27.8	36.2	76.8	56.8	126.4	28.5	38.2	80.4	58.2

a baseline. The adaptive noise module is flexible and pluggable to any conditional recurrent language models.

V. EXPERIMENTS

A. Experimental Setup

1) *Dataset*: We train and evaluate our model on MSCOCO [63], which is the most popular captioning benchmark used for Microsoft COCO caption challenge. MSCOCO contains 82,783 training images, 40,504 validation images and 40,775 unlabeled testing images. Each image in the training set and validation set is annotated with five English descriptive sentences. We adopt a widely used Karpathy’s data split [30], [33] for the offline evaluation, where 5K images are used for validation, 5K for testing and 113,287 for training. Following [12], [33], we convert all descriptions to lower case and map words that occur less than five times to *<UNK>* tokens.

2) *Evaluation Metrics*: To evaluate the quality of the generated description, objective evaluation metrics are utilized given the human annotated ground truth. Results are reported with metrics: BLEU [64], METEOR [65], ROUGE-L [66], CIDEr [57] and SPICE [67]. The evaluation metrics are provided by MSCOCO caption evaluation tool.¹ Besides cross entropy training, we also apply the self-critic training [33], where CIDEr score is used as the reward. In the validation stage, we also use CIDEr to choose the best model.

3) *Implementation Details*: In the visual encoder, we apply faster R-CNN [58] in cooperating with ResNet101 [68]. For the region level GCN, 36 regions with top detection confidences are utilized. Each region is represented as a 2,048-dimension feature vector. We use the feature trained with bottom-up attention [29]. For the grid level GCN, we encode the image with the final convolutional layer in ResNet101 and apply the spatial adaptive pooling, which results in a $14 \times 14 \times 2048$ feature map. We construct the grid graph on the feature map. Thus, the grid level graph contains 196 nodes, and each node is represented as a 2,048-dimension feature vector. The node representation after GCN is still set as 2,048. After graph pooling, we set the rescaled graph to contain 36 nodes.

In the RNN captioning model, we set the hidden state size in LSTM as 1,024 and the word encoding as 512. The hidden size for measuring attention distribution is set as 512. Dropout is set as 0.5 experimentally. The MLP in noise agent has a hidden layer sized 32. Noise threshold γ is set

¹<https://github.com/tylin/coco-caption>

TABLE II
COMPARISON OF GRID GCN ITERATIONS ON
MSCOCO KARPATHY TEST SPLIT

Model variants	C↑	M↑	B4↑	B1↑	R↑
GCN-g-itrl	109.3	27.0	34.3	75.0	55.2
GCN-g-itr2	112.9	27.5	35.2	76.3	56.2
GCN-g-itr3	108.6	26.8	33.8	75.6	55.4

to 1 experimentally. We start training the basic captioning model using Adam optimizer with an initial learning rate of $5 \times e^{-4}$. Scheduled sampling and learning rate decay start from the beginning. For a stable learning process, we train the noise module for 30 epochs with initial learning rate at $1 \times e^{-4}$ after the basic generator has been pretrained. To train the basic generator, we start the self-critic training [33] after 30-epoch cross entropy training and stop at the 70-th epoch. In the inference stage, 5 parallel decoding processes are conducted. We run the experiments on a NVIDIA TITAN X GPU. The models in Table I, Table II and Table III are constructed or reimplemented by ourselves via PyTorch. The performance of the latest models in Table IV and Table V are extracted from the corresponding papers.

B. Experiments on the Double-Stream GCN

Experiment results about the Double-stream GCN and its variants are listed in Table I. All the model variants use the Top-Down generator with beam size as 2. In Table I, **Resnet** refers to the model using Resnet101 as the visual encoder. **GCN-g** is the proposed grid GCN, which is constructed on the Resnet feature. **GCN-r** refers to the region GCN. **GCN-r-g** is the model where we directly combine the regional graph and the grid graph without rescaling. **Double-stream GCN** is the model we proposed in Section III, where we combine the region graph and the rescaled grid graph. Overall, our proposed models perform an obvious improvement over the corresponding baselines. All the improvements are highlighted in bold. In the following, we will analyse the quantitative results in details.

1) *Grid GCN*: As the grid graph is constructed on the Resnet feature map, we compare the Resnet model with GCN-g to demonstrate the grid graph’s efficiency. It is obvious that GCN-g outperforms the Resnet model by a large margin. Optimized by cross entropy loss, the Grid-g achieves 112.9 in CIDEr whereas Resnet model only achieves 107.8, which results in 4.7% improvement. With the self-critic training, our GCN-g gains a 3.6% improvement on CIDEr over the Resnet baseline (124.4 vs. 120.1).

TABLE III
PERFORMANCE OF THE NOISE MODEL AND THE BASELINES ON MSCOCO KARPATHY 5K SET

Model	Cross entropy loss												Self critic training												
	Greedy sample						Beam search						Greedy sample						Beam search						
	C ↑	M ↑	B4 ↑	B1 ↑	R ↑	C ↑	M ↑	B4 ↑	B1 ↑	R ↑	C ↑	M ↑	B4 ↑	B1 ↑	R ↑	C ↑	M ↑	B4 ↑	B1 ↑	R ↑	C ↑	M ↑	B4 ↑	B1 ↑	R ↑
BottomUp-SoftAttention baseline	107.9	26.6	33.8	76.5	55.7	112.0	27.3	36.0	77.0	56.7	121.5	27.7	36.7	79.1	57.5	121.7	27.7	36.8	79.2	57.5	-	-	-	-	-
BottomUp-SoftAttention + naive noise	105.9	26.5	32.9	75.3	55.4	112.0	27.3	36.0	77.1	56.7	121.2	27.8	36.8	79.2	57.5	122.3	27.8	36.8	79.3	57.6	-	-	-	-	-
BottomUp-SoftAttention + noise agent	111.0	27.1	34.6	76.5	56.3	112.8	27.4	36.5	76.5	56.7	123.2	28.0	37.0	79.4	57.7	123.7	27.9	37.0	79.4	57.7	-	-	-	-	-
DoubleStreamGCN-TopDown baseline	114.3	27.4	34.9	77.0	56.6	115.6	27.8	36.2	76.8	56.8	125.8	28.5	38.0	80.1	58.1	126.4	28.5	38.2	80.4	58.2	-	-	-	-	-
DoubleStreamGCN-TopDown + naive noise	113.4	27.4	34.6	76.2	56.5	115.5	27.7	36.2	76.8	56.8	125.7	28.4	38.2	80.4	58.2	126.3	28.5	38.3	80.4	58.2	-	-	-	-	-
DoubleStreamGCN-TopDown + noise agent	115.3	27.9	35.1	76.1	56.9	116.1	27.8	36.1	77.0	57.0	126.0	28.5	38.2	80.1	58.2	126.4	28.5	38.3	80.4	58.2	-	-	-	-	-

We also present comparative experiments to see the best iteration for the grid GCN. The results are listed in Table II, where the models are trained with cross entropy loss and decoded with beam size 2. From the experiment results we can see that the grid GCN with two iterations can achieve the superior performance. Thus, in this paper, we adopt 2 GCN iterations for the grid graph.

2) *Double Stream GCN*: To see the efficiency of the proposed Double-stream GCN, we compare the proposed model with GCN-r and GCN-r-g in Table I. Comparing the Double-stream GCN with GCN-r-g (115.6 Vs. 114.4 on CIDEr), we can see an obvious improvement induced by the rescaling strategy. Overall, the Double-stream GCN generally achieves the superior performance compared with others. It should be noted that we have to reduce the batch size for the double-stream GCN to fit it in the GPU. It is believed that the larger batch size in GCN-r would contribute to its better performance in deep learning.

C. Experiments on the Noise Agent

In order to demonstrate the effect of the noise agent, we add the noise agent in two baselines respectively. (1) **BottomUp-SoftAttention baseline** uses the Bottom Up [29] encoding features and the soft attention language decoder as presented in Fig. 3(a). (2) **DoubleStreamGCN-TopDown baseline** uses the proposed double-stream GCN for the visual encoding and use the Top Down as the language decoder as presented in Fig. 3(b). For each baseline, we add parallel naive noise and the adaptive noise module respectively. The experiment results are listed in Table III. In this table, we present experiment results in two groups and present beam search results as well as greedy sample results.

We can find that the models with naive noise induce fluctuation over the baselines, which is not a stable enhancement. If we look at the first group, we can see that the model with noise agent outperforms the BottomUp-SoftAttention baseline significantly. With greedy sampling, the CIDEr metric is improved from 107.9 to 111.0 with cross entropy loss training, and it is improved from 121.5 to 123.2 with the self-critic training. When looking at the second group, we can also see some improvement (11 out of 20 metrics) induced by the noise agent. All the enhancements over the baseline are highlighted in bold.

As we know, all the image encoders are not perfect, and we believe that the encoded representation may contain unexpected bias/noise. Our adaptive noise module could alleviate this issue. While, on theory, if an encoder is totally

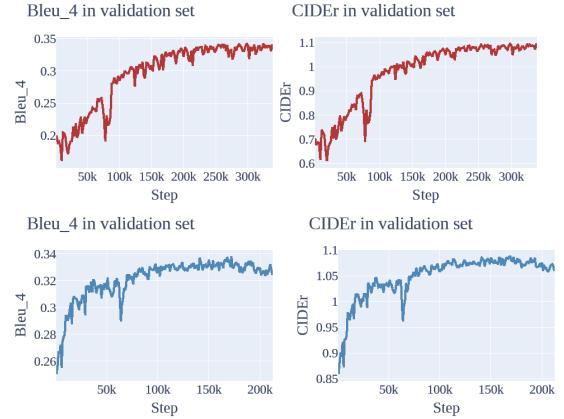


Fig. 4. CIDEr and Bleu-4 performance on the evaluation set along the noise agent training. The upper row: BottomUp-SoftAttention + noise agent. The bottom row: DoubleStreamGCN-TopDown + noise agent.

TABLE IV
COMPARISONS WITH STATE-OF-THE-ART MODELS
ON MSCOCO KARPATHY 5K SET

-	B1 ↑	B4 ↑	M ↑	R ↑	C ↑	S ↑
UP-Down [29]	77.2	36.2	27.0	56.4	113.5	20.3
RFNet [69]	76.4	35.8	27.4	56.5	112.5	20.5
GCN-LSTM _{spa} [12]	77.2	36.5	27.8	56.8	115.6	20.8
NADGCN (ours)	77.0	36.1	27.8	57.0	116.1	21.3
SCST [33]	-	34.2	26.7	55.7	114.0	-
SR-PL [70]	80.1	35.8	27.4	57.0	117.1	21.0
Up-Down-rl [29]	79.8	36.3	27.7	56.9	120.1	21.4
RFNet-rl [69]	79.1	36.5	27.7	57.3	121.9	21.2
NADGCN-rl (ours)	80.4	38.3	28.5	58.2	126.4	21.8

perfect, there wouldn't be any improvement over the baseline. It can be noticed that the *BottomUp-SoftAttention+noise agent* induces more significant improvement over the corresponding baseline than the *DoubleStreamGCN-TopDown+noise agent* does. To further investigate the difference during the noise agent training process, we fix the parameters in the baseline encoder and decoder but only optimize the noise module. Then, a difference on convergence time is noticed as shown in Fig. 4. In Fig. 4, CIDEr and Bleu-4 on the evaluation set along the noise agent training are presented.

D. Comparison With State of the Arts

1) *MSCOCO Offline Comparison*: The offline comparison with state-of-the-art models is presented in Table IV. All comparative models are evaluated in the commonly used Karpathy's data split. We group the results into two categories based on the cross entropy loss training and the

TABLE V
PERFORMANCE ON MSCOCO ONLINE TEST SERVER

Method	B1		B2		B3		B4		M		C		R	
	c5	c40	c5	c40	c5	c40								
Google-NIC [71]	71.3	89.5	54.2	80.2	40.7	69.4	30.9	58.7	25.4	34.6	94.3	94.6	53.0	68.2
Reviewnet [30]	72.0	90.0	55.0	81.2	41.4	70.5	31.3	59.7	25.6	34.7	96.5	96.9	53.3	68.6
Adaptive [72]	74.8	92.0	58.4	84.5	44.4	74.4	33.6	63.7	26.4	35.9	104.2	105.9	55.0	70.5
SCST [33]	78.1	93.7	61.9	86.0	47.0	75.9	35.2	64.5	27.0	35.5	114.7	116.7	56.3	70.7
Up-Down [29]	80.2	95.2	64.1	88.1	49.1	79.4	36.9	68.5	27.6	36.7	117.9	120.5	57.1	72.4
NADGCN (ours)	79.8	94.3	64.0	88.0	49.2	78.5	37.2	67.5	28.0	36.8	121.2	123.4	57.7	72.3



Fig. 5. Generated captions from the Bottom-Up baseline and our NADGCN model.

self-critic training. As we only use the visual information in our model, models utilising external resources like semantic information are excluded in the comparison. The comparative models are introduced as follows. (i) GCN-LSTM_{spa} [12] uses a spatial GCN. (ii) RFNet [69] uses multiple CNN encoders and a fusion network between the encoder and the decoder. (iii) SCST is the first to use self-critic training [33], and is trained with soft attention mechanism [25]. (iv) SR-PL [70] uses self-critic training as well as a text-to-image retrieval reward. (v) Up-Down [29] exploits a combination of bottom-up and top-down attention.

Overall, the evaluation results indicate that our model achieves a promising performance against the state-of-the-art models. Specifically, our model achieves the superior performance among models without external resources. It gains 21.3 on SPICE with the cross entropy loss. In the self-critic training, our model achieves 126.1 on CIDEr and outperforms some latest models such as Up-Down-rl, RFNet-rl, SCST, SR-PL.

2) *MSCOCO Online Comparison*: We also compare our model with the state-of-the-arts using the MSCOCO online test server. Table V reports the performance with five (c5) and forty (c40) reference captions. We include some performing methods that have been officially published. Though not the best one on the MSCOCO Leaderboard, our single model achieves a promising performance compared with many ensemble models.

E. Qualitative Analysis

Fig. 5 shows some generated examples. Captions produced by the baseline model and our NADGCN model are presented. From the visualised results, it is easy to see that our NADGCN

captions tend to exhibit more diverse sentence structures. We highlight the more descriptive fragments in NADGCN captions with the red color. These fragments somehow make the captions accurate via enriching description details.

VI. CONCLUSION

In this paper, we have proposed a novel model, NADGCN generator, to leverage the full context of an image and enhance the generalization of the language model. To extract rich visual context, we develop double-stream GCN which contain a region graph and a rescaled grid graph. Moreover, unlike existing CNN-RNN framework, our model contains an additive noise module that can introduce adaptive noise and manipulate the transition hidden states in the RNN decoder. We learn the module through regarding it as an agent with a stochastic Gaussian policy and regarding the non-noise generation as a baseline. Exhaustive experiments conducted on MSCOCO indicate that our model outperforms comparative baselines clearly and achieves a promising performance overall.

REFERENCES

- [1] Q. Feng, Y. Wu, H. Fan, C. Yan, M. Xu, and Y. Yang, "Cascaded revision network for novel object captioning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 10, pp. 3413–3421, Oct. 2020.
- [2] J. Yu, J. Li, Z. Yu, and Q. Huang, "Multimodal transformer with multi-view visual representation for image captioning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 12, pp. 4467–4480, Dec. 2020.
- [3] Z. Zhang, D. Xu, W. Ouyang, and C. Tan, "Show, tell and summarize: Dense video captioning using visual cue aided sentence summarization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 9, pp. 3130–3139, Sep. 2020.
- [4] F. Guo, W. Wang, Z. Shen, J. Shen, L. Shao, and D. Tao, "Motion-aware rapid video saliency detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 12, pp. 4887–4898, Dec. 2020.
- [5] Z. Zheng, W. Wang, S. Qi, and S.-C. Zhu, "Reasoning visual dialogs with structural and partial observations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 6669–6678.

- [6] L. Sang, M. Xu, S. Qian, M. Martin, P. Li, and X. Wu, "Context-dependent propagating based video recommendation in multimodal heterogeneous information networks," *IEEE Trans. Multimedia*, early access, Jul. 8, 2020, doi: [10.1109/tmm.2020.3007330](https://doi.org/10.1109/tmm.2020.3007330).
- [7] Y. Li, T. Yao, Y. Pan, H. Chao, and T. Mei, "Jointly localizing and describing events for dense video captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7492–7500.
- [8] S. Ye, J. Han, and N. Liu, "Attentive linear transformation for image captioning," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5514–5524, Nov. 2018.
- [9] M. Zhang, Y. Yang, H. Zhang, Y. Ji, H. T. Shen, and T.-S. Chua, "More is better: Precise and detailed image captioning using online positive recall and missing concepts mining," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 32–44, Jan. 2019.
- [10] L. Zhou, Y. Zhang, Y.-G. Jiang, T. Zhang, and W. Fan, "Re-caption: Saliency-enhanced image captioning through two-phase learning," *IEEE Trans. Image Process.*, vol. 29, pp. 694–709, 2020.
- [11] X. Chen, L.-J. Li, L. Fei-Fei, and A. Gupta, "Iterative visual reasoning beyond convolutions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7239–7248.
- [12] T. Yao, Y. Pan, Y. Li, and T. Mei, "Exploring visual relationship for image captioning," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 684–699.
- [13] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [14] M. Plappert *et al.*, "Parameter space noise for exploration," in *Proc. ICLR*, 2018, pp. 2–18.
- [15] K.-C. Jim, C. L. Giles, and B. G. Horne, "An analysis of noise in recurrent neural networks: Convergence and generalization," *IEEE Trans. Neural Netw.*, vol. 7, no. 6, pp. 1424–1438, Nov. 1996.
- [16] A. Neelakantan *et al.*, "Adding gradient noise improves learning for very deep networks," 2015, [arXiv:1511.06807](https://arxiv.org/abs/1511.06807). [Online]. Available: <http://arxiv.org/abs/1511.06807>
- [17] C. Gulcehre, M. Moczulski, M. Denil, and Y. Bengio, "Noisy activation functions," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 3059–3068.
- [18] V. Ordonez, G. Kulkarni, and T. L. Berg, "Im2text: Describing images using 1 million captioned photographs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 1143–1151.
- [19] M. Hodosh, P. Young, and J. Hockenmaier, "Framing image description as a ranking task: Data, models and evaluation metrics," *J. Artif. Intell. Res.*, vol. 47, pp. 853–899, Aug. 2013.
- [20] A. Farhadi *et al.*, "Every picture tells a story Generating sentences from images," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2010, pp. 15–29.
- [21] S. Li, G. Kulkarni, T. L. Berg, A. C. Berg, and Y. Choi, "Composing simple image descriptions using web-scale N-Grams," in *Proc. 15th Conf. Comput. Natural Language Learn.*, 2011, pp. 220–228.
- [22] Y. Ushiku, M. Yamaguchi, Y. Mukuta, and T. Harada, "Common subspace for model and similarity: Phrase learning for caption generation from images," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 2668–2676.
- [23] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3156–3164.
- [24] L. Wu, M. Xu, J. Wang, and S. Perry, "Recall what you see continually using gridlstm in image captioning," *IEEE Trans. Multimedia*, vol. 22, no. 3, pp. 808–818, Mar. 2019.
- [25] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.
- [26] T. Yao, Y. Pan, Y. Li, Z. Qiu, and T. Mei, "Boosting image captioning with attributes," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 4894–4902.
- [27] G. Ding, M. Chen, S. Zhao, H. Chen, J. Han, and Q. Liu, "Neural image caption generation with weighted training and reference," *Cogn. Comput.*, vol. 11, no. 6, pp. 763–777, 2019.
- [28] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4651–4659.
- [29] P. Anderson *et al.*, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6077–6086.
- [30] Z. Yang, Y. Yuan, Y. Wu, W. W. Cohen, and R. R. Salakhutdinov, "Review networks for caption generation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2361–2369.
- [31] H. Chen, G. Ding, Z. Lin, S. Zhao, and J. Han, "Show, observe and tell: Attribute-driven attention model for image captioning," in *Proc. IJCAI*, 2018, pp. 606–612.
- [32] X. Yang, K. Tang, H. Zhang, and J. Cai, "Auto-encoding scene graphs for image captioning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10685–10694.
- [33] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, "Self-critical sequence training for image captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7008–7024.
- [34] Z. Ren, X. Wang, N. Zhang, X. Lv, and L.-J. Li, "Deep reinforcement learning-based image captioning with embedding reward," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 290–298.
- [35] H. Chen, G. Ding, S. Zhao, and J. Han, "Temporal-difference learning with sampling baseline for image captioning," in *Proc. AAAI*, Apr. 2018, pp. 1–8.
- [36] C. Gan, Z. Gan, X. He, J. Gao, and L. Deng, "StyleNet: Generating attractive visual captions with styles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3137–3146.
- [37] J. Johnson, A. Karpathy, and L. Fei-Fei, "Densecap: Fully convolutional localization networks for dense captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4565–4574.
- [38] X. Li, S. Jiang, and J. Han, "Learning object context for dense captioning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 8650–8657.
- [39] A. Wu, Y. Han, Z. Zhao, and Y. Yang, "Hierarchical memory decoder for visual narrating," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Sep. 1, 2020, doi: [10.1109/tcsvt.2020.3020877](https://doi.org/10.1109/tcsvt.2020.3020877).
- [40] J. Weston, S. Chopra, and A. Bordes, "Memory networks," in *Proc. ICLR*, 2015, pp. 1–31.
- [41] N. Xu, A.-A. Liu, Y. Wong, Y. Zhang, W. Nie, Y. Su, and M. Kankanhalli, "Dual-stream recurrent neural network for video captioning," *IEEE Trans. Circuits Syst. Vide Technol.*, vol. 29, no. 8, pp. 2482–2493, Aug. 2018.
- [42] F. Scarselli, M. Gori, A. Chung Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.
- [43] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. ICLR*, 2017, pp. 1–13.
- [44] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.
- [45] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5115–5124.
- [46] D. Boscaini, J. Masci, E. Rodolà, and M. Bronstein, "Learning shape correspondence with anisotropic convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3189–3197.
- [47] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–10.
- [48] W. Norcliffe-Brown, S. Vafeias, and S. Parisot, "Learning conditioned graph structures for interpretable visual question answering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 8344–8353.
- [49] Z. Zhu, J. Yu, Y. Wang, Y. Sun, Y. Hu, and Q. Wu, "Mucko: Multi-layer cross-modal knowledge reasoning for fact-based visualquestion answering," 2020, [arXiv:2006.09073](https://arxiv.org/abs/2006.09073). [Online]. Available: <https://arxiv.org/abs/2006.09073>
- [50] J. Yu, W. Zhang, Y. Lu, Z. Qin, Y. Hu, J. Tan, and Q. Wu, "Reasoning on the relation: Enhancing visual representation for visual question answering and cross-modal retrieval," *IEEE Trans. Multimedia*, vol. 22, no. 12, pp. 3196–3209, Dec. 2020.
- [51] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.
- [52] D. Silver *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, p. 484, 2016.
- [53] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. Cambridge, MA, USA: MIT Press, 2018.
- [54] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, "Sequence level training with recurrent neural networks," in *Proc. ICLR*, 2016, pp. 1–15.
- [55] S. Liu, Z. Zhu, N. Ye, S. Guadarrama, and K. Murphy, "Improved image captioning via policy gradient optimization of SPIDER," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 873–881.
- [56] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 229–256, May 1992.

- [57] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4566–4575.
- [58] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
- [59] D. Marcheggiani and I. Titov, "Encoding sentences with graph convolutional networks for semantic role labeling," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 1506–1515.
- [60] H. Dai, B. Dai, and L. Song, "Discriminative embeddings of latent variable models for structured data," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 2702–2711.
- [61] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 4800–4810.
- [62] K. Cho, "Noisy parallel approximate decoding for conditional recurrent language model," 2016, *arXiv:1605.03835*. [Online]. Available: <http://arxiv.org/abs/1605.03835>
- [63] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick, "Microsoft coco captions: Data collection and evaluation server," 2015, *arXiv:1504.00325*. [Online]. Available: <https://arxiv.org/abs/1504.00325>
- [64] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics*, 2002, pp. 311–318.
- [65] M. Denkowski and A. Lavie, "Meteor universal: Language specific translation evaluation for any target language," in *Proc. 9th Workshop Stat. Mach. Transl.*, 2014, p. 376.
- [66] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Proc. Text Summarization Branches Out ACL Workshop*, vol. 8. Barcelona, Spain, 2004, pp. 1–8.
- [67] P. Anderson *et al.*, "Spice: Semantic propositional image caption evaluation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 382–398.
- [68] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [69] W. Jiang, L. Ma, Y.-G. Jiang, W. Liu, and T. Zhang, "Recurrent fusion network for image captioning," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 499–515.
- [70] X. Liu, H. Li, J. Shao, D. Chen, and X. Wang, "Show, tell and discriminate: Image captioning by self-retrieval with partially labeled data," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 338–354.
- [71] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3156–3164.
- [72] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing when to look: Adaptive attention via a visual sentinel for image captioning," 2016, *arXiv:1612.01887*. [Online]. Available: <http://arxiv.org/abs/1612.01887>



Lingxiang Wu received the B.E. degree from the Computer Science and Engineering Department, Beijing Institute of Technology, in 2015. She is currently pursuing the Ph.D. degree with the School of Electrical and Data Engineering, University of Technology Sydney, Australia. In 2015, she did internship at the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China. Her research interests include computer vision, machine learning, and deep learning.



Min Xu (Member, IEEE) received the B.E. degree from the University of Science and Technology of China, the M.S. degree from the National University of Singapore, and the Ph.D. degree from The University of Newcastle, Australia. She is currently an Associate Professor with the School of Electrical and Data Engineering, University of Technology Sydney. She has published over 150 research papers in high quality international journals and conferences. Her research interests include multimedia data analytics, pattern recognition, and computer vision.



Lei Sang is currently pursuing the Ph.D. degree with the Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW, Australia. His current research interests include natural language processing and recommender systems.



Ting Yao (Member, IEEE) is currently a Principal Researcher with the Vision and Multimedia Lab, JD AI Research, Beijing, China. His research interests include video understanding, large-scale multimedia search, and deep learning. Prior to joining JD AI Research, he was a Researcher with Microsoft Research Asia, Beijing. He is also the Principal Designer of several top-performing multimedia analytic systems in international benchmark competitions, such as ActivityNet Large Scale Activity Recognition Challenge 2019–2016, Visual Domain Adaptation Challenge 2018 and 2017, and COCO Image Captioning Challenge. He is the Leader Organizer of MSR Video to Language Challenge in ACM Multimedia 2017 and 2016, and built MSR-VTT, a large-scale video to text dataset that is widely used worldwide. His works have also led to many awards, including the ACM SIGMM Outstanding Ph.D. Thesis Award 2015 and the ACM SIGMM Rising Star Award 2019.



Tao Mei (Fellow, IEEE) is currently the Technical Vice President with JD.com and the Deputy Managing Director of JD AI Research, where he also serves as the Director of the Computer Vision and Multimedia Lab. Prior to joining JD.com in 2018, he was a Senior Research Manager with Microsoft Research Asia, Beijing, China, where he contributed 20 inventions and technologies to Microsoft's products and services. He has authored or coauthored more than 200 publications (with 11 best paper awards) and holds 20 U.S. granted patents. He was elected as a fellow of IAPR and a Distinguished Scientist of ACM in 2016, for his contributions to large-scale multimedia analysis and applications. He is also a distinguished industry speaker of the IEEE Signal Processing Society. He is the General Co-Chair of IEEE ICME 2019, the Program Co-Chair of the ACM Multimedia 2018, IEEE ICME 2015, and IEEE MMSP 2015. He was or has been an Editorial Board Member of the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON MULTIMEDIA, the ACM Transactions on Multimedia Computing, Communications, and Applications, and the ACM Transactions on Intelligent Systems and Technology.