Dual Graph Convolutional Networks with Transformer and Curriculum Learning for Image Captioning

Xinzhi Dong
School of Computer Science, Wuhan University
Wuhan, Hubei, China
dongxz97@whu.edu.cn

Wenju Xu
OPPO US Research Center, InnoPeak Technology Inc.
Palo Alto, CA, USA
xuwenju123@gmail.com

Chengjiang Long
JD Finance America Corporation
Mountain View, CA, USA
cfykx@gmail.com

Chunxia Xiao†
School of Computer Science, Wuhan University
Wuhan, Hubei, China
cxxiao@whu.edu.cn

ABSTRACT
Existing image captioning methods just focus on understanding the relationship between objects or instances in a single image, without exploring the contextual correlation existed among contextual image. In this paper, we propose Dual Graph Convolutional Networks (Dual-GCN) with transformer and curriculum learning for image captioning. In particular, we not only use an object-level GCN to capture the object to object spatial relation within a single image, but also adopt an image-level GCN to capture the feature information provided by similar images. With the well-designed Dual-GCN, we can make the linguistic transformer better understand the relationship between different objects in a single image and make full use of similar images as auxiliary information to generate a reasonable caption description for a single image. Meanwhile, with a cross-review strategy introduced to determine difficulty levels, we adopt curriculum learning as the training strategy to increase the robustness and generalization of our proposed model. We conduct extensive experiments on the large-scale MS COCO dataset, and the experimental results powerfully demonstrate that our proposed method outperforms recent state-of-the-art approaches. It achieves a BLEU-1 score of 82.2 and a BLEU-2 score of 67.6. Our source code is available at https://github.com/Unbear430/DGCN-for-image-captioning.

CCS CONCEPTS
• Computing methodologies → Natural language processing; Scene understanding.

KEYWORDS
Image Captioning; Transformer; Graph Neural Network; Curriculum Learning

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MM '21, October 20–24, 2021, Chengdu, China
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8651-7/21/10...$15.00
https://doi.org/10.1145/3474085.3475439

Figure 1: An illustration example of image captioning task, taking Graph Convolutional Networks (GCNs) as the visual encoder and a sequence-to-sequence model as a linguistic decoder. The top represents the visual encoder via a single object-level GCN on a single image, and the bottom indicates the visual encoder through the combination of both the object-level GCN and an image-level GCN. The visual embedding feature extracted in either way can be fed into a linguistic decoder to generate a reasonable text description.

1 INTRODUCTION
Image captioning aims to generate a syntactic and correct description for a given single image, a.k.a., a translation from image to language. One example is illustrated in Figure 1. It requires a clear understanding of different objects and the relation between them, and the grammar in the caption. Otherwise, the model will miss some objects or generate corrections that do not conform to common sense. Many deep learning-based methods and algorithms have been proposed for image captioning [9, 24, 39, 44]. They heavily rely on Convolutional Neural Networks (CNNs) [22, 54, 57, 58] to extract hierarchical image features and take Recurrent Neural Networks (RNNs) like long-short-term memory (LSTM) [4, 25, 32, 43, 48] as the language model to generate descriptive captions. However, these CNN-RNN based methods suffer from the limitations of fixed
captions. Therefore, captions generated by these methods are far from being able to exactly describe the content of the input image.

Graph Convolutional Networks (GCNs) [40, 50, 53, 54] are then exploited on the structured graphs to enrich region representations for image captioning tasks. Recently, transformer [8, 41] has shown promising performance when dealing with serialization information. What’s more, transformer has been recognized as the state of the art in sequence modeling tasks like language understanding and machine translation. Although there exists some efforts [5, 15] of applying transformer to solve the image captioning task and achieve some success, the potential of integrating GCNs with transformer is still largely under-explored.

In this paper, we propose a novel image captioning framework taking Dual Graph Convolutional networks (Dual-GCN) as the visual encoder and take the transformer as the linguistic decoder. As shown in Figure 2, our framework integrates an object-level GCN and an image-level GCN for visual feature encoding. The first GCN focuses on objects in local regions to explore spatial object-level relation [6, 37, 51, 61], while the second one targets on the image-level similarity relation [12, 45] amongst multiple similar images. We calculate the similarity of different images and select the images with high similarity as a meaningful complementary global visual embedding to ensure a more reasonable and accurate text description generation. In this way, our framework can learn the global and local visual embedding with the well-designed Dual-GCN framework. This significantly distinguishes our method from other GCN-based models [53, 54], which only considers the local object regions.

Unlike most existed approaches taking RNNs as the linguistic decoders, we choose transformer as our linguistic decoder instead. Due to the well-designed multi-head self-attentions, the transformer has the advantage of parallelization and has a stronger ability to extract visual features. It shows superior performance in the fusion of visual features and text features. Therefore, taking in the global and local visual embedding as input, our transformer linguistic decoder can generate a syntactic and correct description for the single input image.

It is worth mentioning that existing caption models [1, 5, 9, 21, 21, 43, 48, 50, 52–56] are trained in a straightforward manner, in which all the training data are fed into the model evenly and equally, ignoring the fact that different training images may contain information at different levels. We argue that a difficult image capturing a complex scene usually needs a long caption to describe the objects and their spatial positioning. Therefore, an under-trained model could be misled by the wrong gradients when trained on difficult images. Inspired by the human nature learning process in an easy-to-hard manner, we adopt the curriculum learning [47] as a training strategy. A cross-review mechanism is then introduced to distinguish the difficulty of the training images. In particular, we partition the entire training dataset evenly into multiple subsets and train an image caption model on each subset. For any training example, its difficulty level is determined across all the other learned models, each of which never takes it as the training example. With the difficulty levels assigned on all the training examples, we can train our proposed model on easy images at the beginning. Then we keep adding difficult images as the training examples. Obviously, the newly added information dramatically enhances our model’s robustness and generalization ability in terms of understanding different objects and the relation between them. To summarize, the main contributions of this paper are three-fold:

- We propose a novel framework that takes Dual-GCN as a powerful visual encoder and transformer as the linguistic decoder for image captioning. The first object-level GCN captures the region to region relation in a single image, and the second image-level GCN captures the image’s similarity relation. This treatment enables both global and local visual information well encoded and fed into the transformer to generate better captioning descriptions.
- To our best knowledge, we are the first one to apply the Curriculum Learning as the training strategy on image captioning model in an easy-to-hard manner. Note that the difficulty level is determined by our well-designed cross-review mechanism.
- We conduct extensive experiments on the MS COCO dataset. The results strongly demonstrate that our proposed approach outperforms state-of-the-art methods. It achieves a BLEU-1 score of 82.2 and a BLEU-2 score of 67.6.

2 RELATED WORK

Image Captioning is a cross-modal task combining computer vision and natural language processing. It has been widely studied as a deep learning task [28] in recent years. The dominant paradigm in modern image captioning are sequence learning methods [9, 43, 48, 52, 55, 56]. Vinyls et al. [43] used a deep CNN to extract the visual features of the image and adopted LSTM [53] as the language model to guide the language sequence generation. Instead of learning a visual attention map on the input image, [36] proposed an encoder-decoder model that learns an adaptive attention map to combine the visual feature and language embedding. [1] proposed bottom-up attention and a top-down attention mechanism for caption generation. Chen et al. [3] extended the spatial-wise and channel-wise attention mechanism to extract both spatial features and semantic features from an image. Unlike these mentioned models, Yao et al. [53] adopted the graph neural network to integrate spatial and semantic relationships between different objects. In recent years, transformer-based methods [50] show great potential in image captioning. Herdade et al. [16] used the transformer architecture for image captioning. Then Li et al. [29] took an external tagger to enhance the performance of the transformer model. As reverse process of image captioning, and different from video generating [17], Hu et al. [19] [18] used diverse conditional GAN [49] for image generation based on text representation. Different from generating and editing images using scene graph [11] or illumination processing [60] [59], Fang et al. [10] analyzed the text content and produced corresponding images.

Graph Neural Networks have been widely used in many applications to build up the relations between different features, such as nature language processing and computer vision tasks [20, 26, 33–35, 53]. In particular, [26] solves the computing bottleneck by learning the Laplacian polynomials of graphs. When combining with the LSTM [53], GCN can be used for image captioning task. It is used to extract relations between different objects, which achieved great results in the final caption generation. However, these GCN based
methods only explore the object relations within one single image. Instead, we propose a Dual-GCN module to aggregate information extracted from both the input image and other similar images.

Transformer is first proposed by [41]. It is a combination of multi-layer encoder and decoder. The encoder consists of a stack of self-attention and feedforward layers, while the decoder uses self-attention on the text and the cross-attention to the last encoder layer. Herda et al. [16] then used the transformer in image captioning. They combined geometric relationships between different input objects. [5] used a priori knowledge module to improve the transformer encoder and also used mesh connection between multi-layer encoder layer and decoder layer to improve the hierarchical diversity of feature extraction. Li et al. [29] used the transformer to make use of visual information and additional semantic knowledge given by an external tagger. In this paper, we take transformer as the linguistic model to generate a reasonable text description based on the extracted visual embedding feature.

Curriculum Learning is first proposed by [2] in the machine learning area. This method first classifies the sub-dataset according to the difficulty, such as finding examples that are easy to implement in the data set. Then it sets up an easy-to-hard curriculum for the learning procedure. Such a human-like learning strategy considerably improves the model’s capability. Curriculum learning has been widely used in the computer vision area [13, 23, 46]. With the continuous expansion of transformer in computer vision, we propose to use a curriculum learning strategy aiming at improving our Dual-GCN model’s performance on image captioning task.

3 PROPOSED METHOD

Our method aims to generate a caption that correctly describes the content of an input image. It comprises three main parts: Dual-GCN encoder, transformer decoder, and curriculum learning strategy. In this section, we will introduce each framework and our algorithm in detail. Figure 2 shows our main framework.

Given an image I, we take a textual caption to describe its content, denoted by S. First, we take the Faster-RCNN [39] as a backbone to produce a set of detected objects \( V_{obj} = \{v_i\}_{i=1}^{O} \) where \( O \) denotes different object regions in an image. \( v_i \) denotes a \( C \)-dimensional feature representation of the \( i \)-th object region.

3.1 Dual-GCN Encoder

To learn a better latent representation of the input image, we propose a Dual-GCN encoder, which consists of an object-level GCN and a image-level GCN. Based on the region feature from the Faster-RCNN detector, our Dual-GCN encoder produces better features by seeking the spatial relation within the same image and the similarity among different images.

Object-Level GCN. We take the object-level GCN to build a spatial relation between different objects within the same image. To this end, we consider each feature extracted from one single object region as a vertex and consider the relative position amongst different objects as a directed line segment. Specifically, we build a spatial graph \( G_{obj} = (V_{obj}, E_{obj}) \), where \( E_{obj} = \{(v_i, v_j)\} \) is the set of spatial relation edges between region vertices, and the edge \((v_i, v_j)\) represents the relative geometry relationship between the \( i \)-th and the \( j \)-th objects. More specifically, following [53], we use the IOU, relative distance, and the relative angle between two objects as the relative geometry relationship to determine the edges, these relationships can be divided into "inside", "cover", "overlap" and "no" relation and so on. Therefore, each vertex \( v_i \) is encoded via a modified GCN as:

\[
a_{v_{obj}}^i = \sigma(\sum_{v_j\in N(v_i)} W_{v_j\rightarrow v_i} a_j + b_{v_j\rightarrow v_i}),
\]

where each \( a_{v_j} \) is a real-valued vector and it represents the information of the nodes in the graph. \( \sigma \) is an activation function, and \( N(v_i) \) denotes a set of neighbor vertices of \( v_i \). \( v_j \rightarrow v_i \) indicates the direction from \( v_j \) to \( v_i \). \( W_{v_j\rightarrow v_i} \) and \( b_{v_j\rightarrow v_i} \) denote the transformation matrix and the bias vector, respectively.

Image-Level GCN. The information extracted from a single image is very limited. As a remedy, we propose a novel image-level GCN, which enhances the information flow in the graph by considering other relevant images. In image feature extraction, we observe that
images sharing a certain degree of similarity to the input image can be an auxiliary information source for the input image captioning.

Let $\delta_j$ stands for the latent representation of $j$-th image, which contains a set of objects $V_{obj}^j = \{v_{ij}^{obj}\}_{i=1}^O$. We define:

$$\delta_j = \frac{1}{O} \sum_{i=1}^O v_{ij}^{obj}, \quad (2)$$

Based on the latent representation, we seek similar images $\delta_k \in \Omega(\delta_j)$ according to the $L_2$ distance between the features of the input image and other images from the same dataset. This is formulated as:

$$\delta_k \in \Omega(\delta_j) : = \left\{ \delta_k : \sum_{i=1}^C \left| v_{ij}^{obj} - \bar{v}_{ij} \right|^2 \right\}_{K}, \quad (3)$$

where $C$ is the dimension of the feature; $\cdot_K$ is to select $K$ images with lowest distance score. Once we obtain the top-$K$ images, we establish an image-level graph $G_{img} = (V_{img}, E_{img})$ by considering the image feature as a node $V_{img}$ and the similarity between paired images as an edge $E_{img}$. Then we encode the features represented by each image node in the graph:

$$U_j^{img} = \sigma \left( \sum_{\delta_k \in \Omega(\delta_j)} W\delta_k + b \right), \quad (4)$$

where $\Omega(\delta_j)$ denotes the set of similar images of $\delta_j$. Note that in our image-level graph $U_j^{img}$ represents the feature of an image and its $K$ nearest neighbor images.

**Global and Local Visual Embedding.** Since we only need to generate the caption of a given image, we choose to extract the feature vector of GCN encoded in the node of the original image. Then we concatenate the feature vectors of GCN recoded on image-level similarity with the original image encoded on object-level as the input of a transformer as the language model for text prediction:

$$U_j = \text{Concat}(V_{obj}^j, U_j^{img})$$

$$= [\text{concat}(v_{ij}^{obj}, u_j^{img}), \ldots, \text{concat}(v_{iO}^{obj}, u_j^{img})]. \quad (5)$$

In this way, we combine the information extracted from the input image and other similar images. The final visual embedding representation integrates the characteristics of each object in the input image locally and the contextual object information from multiple similar images globally. Then, the global and local visual embedding information is fed into the linguistic decoder to ensure a reasonable caption generation.

### 3.2 Transformer as Linguistic Decoder

As shown in Figure 2, after we get the global and local embedding $U = \{U_j\}_{j = 1, \ldots, O}$ of the input image, we take a transformer decoder to generate a caption correctly describing the content of the image.

**Encoding Layer.** The encoding layer is based on multi-head attention layer, which can be formulated as:

$$U_{att} = \text{softmax} \left( \frac{W_q^T(W_k U)^T}{\sqrt{d}} \right) W_k U,$$  
\[6\]

where $W_q$, $W_k$ and $W_o$ are the learnable parameters used to create the query, keys and values respectively; $d$ is a constant used for normalization. Besides, a residual structure is adopted to stabilize the attention module. This ensures that the transformer can extract the spatial information of different objects to a greater extent in the self-attention part. Thus, we achieve:

$$U = \text{AddNorm}(U \oplus U_{att}), \quad (7)$$

where $\oplus$ is an element-wise addition operation. Based on the above structure, multiple encoding layers are stacked in sequence so that each encoder layer receives the output calculated by the previous encoder layer. This is equivalent to creating a multi-layer high-level coding of the relationship between image regions, in which the higher coding layer can utilize and refine the previously recognized relationship and finally produce the output.

**Decoding Layer.** We take a similar multi-head attention module to process text and image feature. The decoder layer also contains feed-forward layer, and all components are encapsulated in addnorm operator. The structure of decoder layer can be described as:

$$Q = \text{AddNorm}(U, Y), \quad (8)$$

where $Y$ is the input sequence. Like the encoder layer, our decoder layer takes a similar self-attention and residual structure. And multiple decoder layers are also stacked in sequence. The purpose is to extract higher-level text and image features and generate more accurate word distribution probability:

$$\hat{Q} = \text{AddNorm}(Q \oplus Q_{att}), \quad (9)$$

where $Q_{att}$ stands for the output of a multi-head attention module in the decoding layer. Finally, by inputting the generated results into the softmax layer, the word probability distribution of the currently generated words is obtained:

$$P = \text{Softmax}(\hat{Q}). \quad (10)$$

After getting the word probability distribution through the transformer, we get the probability distribution $P(S'|S, I)$ for the given image. Where $S$ is the ground-truth caption embedding matrix, $I$ is the image feature we used, and $S'$ is the currently generated captions. Through the algorithm of conditional probability, our model gets the whole caption probability for the given image.

### 3.3 Curriculum Learning

To adopt the curriculum learning for our image captioning task, we have to determine the difficulty level from easy to hard.

**Objective Function.** Let $D_{data}$ be the MSCOCO training dataset, and $\Theta$ be our Dual-GCN+Transformer caption model. In order to obtain a more comparable and stable difficulty score, we first uniformly split our training dataset $D_{data}$ into $M$ parts. Each sub-dataset is denoted by $D_i$ ($i = 1, 2, 3, .. M$). Then we train our Dual-GCN+Transformer caption models $\Theta_i$ on them respectively. Each model can only use $1/K$ of the training dataset for training. The parameters of these models can be learned by the following optimization:

$$\Theta_i = \arg \min_{\Theta_i} \sum_{(I,S) \in D_i} L_{ce} (\Theta_i(I), S), \quad (11)$$

where \((I, S)\) is the image-to-text pair in the \(i\)-th training dataset. \(L_{ce}\) defines the cross entropy loss function between the probability prediction by \(\Theta_i(I)\) and the corresponding ground-truth caption \(S\).

**Difficulty Assessment.** Most of the traditional ideas are based on word frequency. The difficulty is defined according to the number of times each word appears in the dataset. For example, in the MSCOCO training set, “person” appeared in 64,115 images and had 262,465 tag boxes, while “bus” only appeared in 3,952 images and had 6,069 tag boxes. Obviously, “person” is more accessible to generate than “bus”. However, since image caption is a cross-modal matching task from image to text, the difficulty evaluation should be consistent with the corresponding evaluation metrics like BLEU-1 and BLEU-2. Inspired by [47], we adopt a cross review mechanism to determine the difficulty level for all the training examples, as shown in Figure 3.

After we train our caption models on \(M\) sub-datasets, we evaluate the difficulty level for each training example. Note that each image-to-text example \((I, S) \in D_i\) has been seen by model \(\Theta_i\) during training. Hence we use another model \(\Theta_k\) to evaluate the difficulty of \((I, S)\):

\[
E_k(I, S) = 1 - \text{Metric}(\Theta_k(I), S),
\]

where \(E_k(I, S)\) is the difficulty score of the text-image pair \((I, S)\). \text{Metric} stands for a formula, which could be one of the image captioning metrics such as BLEU-1, BLEU-2, BLEU-3 and BLEU-4 [38]. In this paper, we take the average of BLEU-1 score to indicate the difficulty. This is formulated as the sum of the evaluation scores of other caption models:

\[
DS((I, S)) = \frac{1}{M-1} \sum_{(I,S) \in D_k, k \neq i} E_k(I, S),
\]

where \(DS((I, S))\) is the difficulty score of the text-image pair \((I, S)\) in sub dataset \(D_i\).

**Curriculum Learning Arrangement Part.** We first sort all text-image pairs according to their difficulty scores \(DS\), and then divide them into \(M\) sub datasets: \(U_i\). The training dataset is then arranged from \(U_1\) (the easiest) to \(U_M\) (the hardest). The numbers of example in each categories are defined as \([U_1], [U_2], \ldots, [U_M]\). We take \(M\) stages to train our model, it can be defined as \(C_i (i = 1, \ldots, M)\). Notice that at each stage \(C_i\), the image-to-text example is still shuffled to keep local stochastics, and each example from different stages does not overlap in order to prevent overfitting. For each learning stage \(C_i\), we choose example according to a certain proportion from the aforementioned categories according to difficulty. The specific quantity is as follows:

\[
\frac{|U_1|}{M} : \frac{|U_2|}{M} : \ldots : \frac{|U_M|}{M}.
\]

When the training stages are reached on \(C_M\), we think the model should be ready to train on the whole MS COCO dataset. So we add another learning stage \(C_{M+1}\), which is the entire MS COCO dataset.

### 4 EXPERIMENTS

#### 4.1 Datasets and Metrics

**Microsoft COCO Dataset** [31] (MS COCO) is the dataset most commonly used in image captioning tasks. This dataset consists of about 82,700 training images and 40,500 validation images. There are five human-annotated descriptions for each image. Following the settings used by [25], we take 5,000 images for validation, 5,000 images for testing and almost 110,000 images for training.

**Visual Genome** [27] is a large dataset containing almost 108,000 images with densely annotated objects, attributes and relationships. We take almost 100,000 images for training, 5,000 images for validation and 5,000 images for testing as done in [1] and [53]. We observe that there are approximately 51,000 images in the visual genome are also found in the MS COCO dataset. To avoid contamination of our MS COCO validation and test sets, we make sure that...
Then we select the top $O = 36$ regions with the highest confidences to represent the image. In the transformer decoder part, we use a $C = 2048$-dimensional vector to encode each region. Based on the Visual Genome dataset, we pretrain our object detector which is built upon the Faster-R-CNN [39] and enhanced by replacing its backbone with a resnet-101 [14]. For each input image, we firstly apply this detector to detect the object regions. Each region is represented by a 2048-dimensional vector $C = 2048$. Then we select the top $O = 36$ regions with the highest confidences to represent the image. In the transformer decoder part, we use a 1000-dimensional word embedding to represent each word instead of using one-hot vectors. The number of transformer encoder layers and decoder layers are both set to 6. Similar to [41], the dimensionality of the layer and the number of the multi-head attention are set to 512 and 8, respectively. The number of similar images $K$ is set to 6. Toward the curriculum learning strategy, we calculate the average of BLEU-1 and BLEU-4 as the final score which turns out to be a more stable and realistic difficulty evaluation. We choose $M = 8$ as the number of the datasets.

**4.2 Implementation Detail**

Based on the Visual Genome dataset, we pretrain our object detector which is built upon the Faster-R-CNN [39] and enhanced by replacing its backbone with a resnet-101 [14]. For each input image, we firstly apply this detector to detect the object regions. Each region is represented by a 2048-dimensional vector $C = 2048$. Then we select the top $O = 36$ regions with the highest confidences to represent the image. In the transformer decoder part, we use a 1000-dimensional word embedding to represent each word instead of using one-hot vectors. The number of transformer encoder layers and decoder layers are both set to 6. Similar to [41], the dimensionality of the layer and the number of the multi-head attention are set to 512 and 8, respectively. The number of similar images $K$ is set to 6. Toward the curriculum learning strategy, we calculate the average of BLEU-1 and BLEU-4 as the final score which turns out to be a more stable and realistic difficulty evaluation. We choose $M = 8$ as the number of the datasets.

**4.3 Comparisons with State-of-the-Art**

**Compared Methods.** We compare our proposed DGCN with six state-of-the-art methods including (1) two attention based methods, i.e., Up-down [1] and AOA Net [21], (2) three graph based methods, i.e., GCN-LSTM [53], GCN-LSTM+HIP [54] and SGAE [50], and (3) one transformer based method, $M^2$ Transformer [5].

**Quantitative Comparison.** Table 2 shows the testing result of different models on the MS COCO dataset for image captioning task. Our proposed Dual-GCN + Transformer + CL model achieves the best results in all seven different evaluation metrics. Notably, it produces an 82.2 BLEU-1 score, which significantly outperforms other state-of-the-art models. This score is 2.0 higher than the traditional up-down method [1]. Compared with the latest AOA method [21], our model produces better BLEU-1 and CIDEr scores, which are 1.2 and 2.3 higher than AOA, respectively. We also observe that the

Table 2: Comparison with the state of the art on the Microsoft COCO dataset, in single-model setting.

<table>
<thead>
<tr>
<th>Method</th>
<th>BLEU-1</th>
<th>BLEU-2</th>
<th>BLEU-3</th>
<th>BLEU-4</th>
<th>METEOR</th>
<th>ROUGE</th>
<th>CIDEr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up-down [1]</td>
<td>80.2</td>
<td>64.1</td>
<td>49.1</td>
<td>36.9</td>
<td>27.6</td>
<td>57.1</td>
<td>117.9</td>
</tr>
<tr>
<td>AOA-NET [21]</td>
<td>81.0</td>
<td>65.8</td>
<td>51.4</td>
<td>39.4</td>
<td>29.1</td>
<td>58.9</td>
<td>126.9</td>
</tr>
<tr>
<td>GCN-LSTM [53]</td>
<td>80.8</td>
<td>65.5</td>
<td>50.8</td>
<td>38.7</td>
<td>28.5</td>
<td>58.5</td>
<td>125.3</td>
</tr>
<tr>
<td>GCN-LSTM+HIP [54]</td>
<td>81.6</td>
<td>66.2</td>
<td>51.5</td>
<td>39.3</td>
<td>28.8</td>
<td>59.0</td>
<td>127.9</td>
</tr>
<tr>
<td>SGAE [50]</td>
<td>81.0</td>
<td>65.6</td>
<td>50.7</td>
<td>38.5</td>
<td>28.2</td>
<td>58.6</td>
<td>123.8</td>
</tr>
<tr>
<td>$M^2$ Transformer [5]</td>
<td>81.6</td>
<td>66.4</td>
<td>51.3</td>
<td>39.7</td>
<td>29.4</td>
<td>59.2</td>
<td>127.9</td>
</tr>
<tr>
<td>Dual-GCN+Transformer+CL</td>
<td>82.2</td>
<td>67.6</td>
<td>52.4</td>
<td>39.7</td>
<td>29.7</td>
<td>59.7</td>
<td>129.2</td>
</tr>
</tbody>
</table>

**Figure 4: Visualization results compared with state-of-the-art methods.** Note that we only plot top-4 images for simplicity.
Table 3: Effectiveness comparison experiment on the MSCOCO dataset. Dual-GCN means the combination of GCN$_{obj}$ and GCN$_{img}$, $F_{obj}$ and $F_{img}$ indicate the input features fed into GCN$_{obj}$ and GCN$_{img}$, respectively. CL means Curriculum learning.

<table>
<thead>
<tr>
<th>Model Configuration</th>
<th>BLEU-1</th>
<th>BLEU-2</th>
<th>BLEU-3</th>
<th>BLEU-4</th>
<th>METEOR</th>
<th>ROUGE</th>
<th>CIDEr</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_{obj}$ + Transformer</td>
<td>71.1</td>
<td>55.9</td>
<td>45.3</td>
<td>28.0</td>
<td>25.2</td>
<td>52.8</td>
<td>114.3</td>
</tr>
<tr>
<td>GCN$_{obj}$ + Transformer</td>
<td>80.5</td>
<td>64.7</td>
<td>49.2</td>
<td>37.6</td>
<td>27.9</td>
<td>57.4</td>
<td>125.8</td>
</tr>
<tr>
<td>$F_{obj}$ + Transformer + CL</td>
<td>72.3</td>
<td>56.8</td>
<td>46.1</td>
<td>28.4</td>
<td>25.4</td>
<td>53.0</td>
<td>114.7</td>
</tr>
<tr>
<td>GCN$_{obj}$ + Transformer + CL</td>
<td>81.6</td>
<td>66.8</td>
<td>51.6</td>
<td>39.3</td>
<td>28.3</td>
<td>58.4</td>
<td>129.4</td>
</tr>
<tr>
<td>$F_{img}$ + Transformer + CL</td>
<td>80.2</td>
<td>64.1</td>
<td>49.0</td>
<td>37.4</td>
<td>27.4</td>
<td>57.0</td>
<td>124.0</td>
</tr>
<tr>
<td>GCN$_{img}$ + Transformer + CL</td>
<td>82.0</td>
<td>66.9</td>
<td>52.2</td>
<td>39.8</td>
<td>29.6</td>
<td>59.4</td>
<td>129.0</td>
</tr>
<tr>
<td>GCN$<em>{img}$&amp;$F</em>{obj}$ + Transformer + CL</td>
<td>81.9</td>
<td>67.3</td>
<td>52.0</td>
<td>39.5</td>
<td>29.4</td>
<td>59.6</td>
<td>129.4</td>
</tr>
<tr>
<td>Dual-GCN + Transformer</td>
<td>82.0</td>
<td>67.3</td>
<td>52.3</td>
<td>39.7</td>
<td>29.8</td>
<td>59.6</td>
<td>129.0</td>
</tr>
<tr>
<td>Dual-GCN + LSTM + CL</td>
<td>81.8</td>
<td>66.2</td>
<td>51.6</td>
<td>39.2</td>
<td>28.7</td>
<td>58.9</td>
<td>128.0</td>
</tr>
<tr>
<td>Dual-GCN + Transformer + CL</td>
<td>81.4</td>
<td>65.8</td>
<td>51.3</td>
<td>38.8</td>
<td>28.0</td>
<td>58.4</td>
<td>127.6</td>
</tr>
<tr>
<td>Dual-GCN + Transformer + CL</td>
<td><strong>82.2</strong></td>
<td><strong>67.6</strong></td>
<td><strong>52.4</strong></td>
<td><strong>39.7</strong></td>
<td><strong>29.7</strong></td>
<td><strong>59.7</strong></td>
<td><strong>129.2</strong></td>
</tr>
</tbody>
</table>

self-attention mechanism in our transformer encoding and decoding process helps to improve the performance as compared to the traditional LSTM decoding used by GCN-LSTM [53]. This is also justified by the fact that the score obtained by GCN-LSTM+HIP [54] is higher than that of the traditional LSTM based method [53]. However, the score of our model is higher than both GCN-LSTM [53] and GCN-LSTM+HIP [54]. Finally, different from $M^2$ transformer [5] which uses memory blocks, our proposed Dual-GCN + Transformer + CL model, which uses the Dual-GCN network, achieves the higher performance in terms of all the metrics. Obviously, these observations strongly demonstrate the efficacy of our proposed method.

**Qualitative Analysis.** Figure 4 shows the qualitative results compared with four state-of-the-art methods. We can clearly see that our model produces a caption that correctly describes the content of the input image. Especially, the captions generated by our Dual-GCN + Transformer + CL model are more detailed and authentic.

### 4.4 Ablation Study

We conduct an ablation study to verify the effectiveness of different components in our method. To this end, we introduce several variants of our full model as shown in Table 3: (i) *Transformer* means only using the transformer and the proposed region embeddings to generate image captions. (ii) $F_{obj}$ means the object feature of the original image. (iii) $F_{img}$ means the mean pooling result of the $F_{obj}$. (iv) GCN$_{obj}$ means adopting the GCN to process the region embeddings before feeding them into the transformer. (v) GCN$_{img}$ means adopting the GCN on $F_{img}$ before feeding them into the transformer. (vi) CL means improving the previous model variant by using the Curriculum Learning strategy. (vii) GCN$_{obj}$&$F_{img}$ and GCN$_{img}$&$F_{obj}$ are two variants of our Dual-GCN in which both object-level and image-level visual features are used with only one GCN, either GCN$_{obj}$ or GCN$_{img}$. (viii) LSTM means only using LSTM to replace the transformer to generate image captions. (ix) Dual-GCN + Transformer + CL is our full model and employs the proposed Dual-GCN module with transformer and curriculum learning to improve the quality of the final results further.

**Effectiveness of Dual-GCN.** The proposed Dual-GCN network aims at enhancing the latent embedding. To verify the effectiveness of our Dual-GCN in improving the performance of image captioning, we have conducted a comparative experiment on whether the GCN network is used or not. The quantitative results are listed in Table 3. When we use GCN$_{obj}$&$F_{img}$ + Transformer + CL, it shows that the BLEU-1 and BLEU-2 score reaches 81.9 and 67.3 respectively. Moreover, when we use GCN$_{img}$&$F_{obj}$ + Transformer + CL, the score of BLEU-1 and BLEU-2 reaches 82.0 and 67.3. But until we added Dual-GCN in our model, the result turned out to be better: the BLEU-1 and BLEU-2 scores reach 82.2 and 67.6 respectively, which is higher than the model that only adopts one GCN network. This shows that our Dual-GCN does have a certain improvement effect on the model.

**Effectiveness of Transformer.** The transformer has been used in image captioning only in recent years. To verify the difference between transformer and traditional LSTM, we added the contrast experiment between transformer and LSTM in our experiment. The quantitative results are listed in 3. When using Dual-GCN+LSTM+CL, the BLEU-1 score reaches 81.4 and the CIDEr reaches 127.6. However, when we replace the original LSTM with the transformer, the result changes a lot: The BLEU-1 score reaches 82.2 and the CIDEr score reaches 129.2, which is 0.8 and 1.6 higher than Dual-GCN+LSTM+CL respectively. At the same time, the other of the evaluation scores also have been improved. Therefore, transformer does have a better effect than LSTM in our model.

**Effectiveness of Curriculum Learning.** To verify the effectiveness of our proposed Curriculum Learning strategy, we have conducted a comparative experiment to train the transformer + GCN network either using curriculum learning or traditional training scheme. The experimental results are shown in Table 3. We can find that Curriculum Learning scheme introduces a specific improvement to our model. For example, without curriculum learning, the BLEU-1 score and BLEU-4 score only reached 81.4 and 38.8. When we adopt the curriculum learning to our Dual-GCN+transformer model, it studies from easy to complex and the BLEU-1 and BLEU-4 score reaches 82.2 and 39.7. This shows the effectiveness of our curriculum learning training method.

To make readers better understand our proposed method, we also provide visualization results for comparison in Figure 5. It shows the ground truth caption and different generated captions of our model in ablation experiments. For simplicity, here we only plot top-4 images.
Different Quantities of $K$ Similar Images. In similarity evaluation part, we need to choose how many similar images should be used to generate the caption. To determine the value of parameter $K$, we compare the performance by using different values of parameter $K$. Table 4 shows the quantitative result. We can see that taking $K=6$ produces the best performance. This value is neither the smallest one nor the biggest one. This because smaller $K$ affects the accurate information of a single image, while more images cover more image features. When $K$ is set to 6, the score of BLEU-1 and BLEU-4 can reach the highest 82.2 and 39.7.

### Table 4: The influence of different values of $K$.

<table>
<thead>
<tr>
<th>$K$</th>
<th>B-1</th>
<th>B-2</th>
<th>B-3</th>
<th>B-4</th>
<th>METEOR</th>
<th>ROUGE</th>
<th>CIDER</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>80.2</td>
<td>64.8</td>
<td>49.4</td>
<td>37.2</td>
<td>27.4</td>
<td>57.0</td>
<td>124.7</td>
</tr>
<tr>
<td>4</td>
<td>80.7</td>
<td>65.0</td>
<td>50.2</td>
<td>37.4</td>
<td>27.9</td>
<td>57.4</td>
<td>124.9</td>
</tr>
<tr>
<td>5</td>
<td>81.6</td>
<td>66.2</td>
<td>51.3</td>
<td>38.6</td>
<td>29.0</td>
<td>59.0</td>
<td>126.8</td>
</tr>
<tr>
<td>6</td>
<td><strong>82.2</strong></td>
<td><strong>67.6</strong></td>
<td><strong>52.4</strong></td>
<td><strong>39.7</strong></td>
<td><strong>29.7</strong></td>
<td><strong>59.7</strong></td>
<td><strong>129.2</strong></td>
</tr>
<tr>
<td>7</td>
<td>82.0</td>
<td>67.1</td>
<td>52.2</td>
<td>39.4</td>
<td>29.4</td>
<td>59.7</td>
<td>128.9</td>
</tr>
<tr>
<td>8</td>
<td>81.8</td>
<td>66.7</td>
<td>51.7</td>
<td>38.9</td>
<td>29.2</td>
<td>59.2</td>
<td>127.6</td>
</tr>
<tr>
<td>9</td>
<td>82.0</td>
<td>67.0</td>
<td>51.9</td>
<td>39.1</td>
<td>29.4</td>
<td><strong>59.7</strong></td>
<td>128.4</td>
</tr>
</tbody>
</table>

Different Quantities of $M$ Sub-datasets. To make sure the specific influence of different parameter $M$ on the experimental results, we have tested our model using different integer values. Table 5 shows the quantitative result. The experimental results show the effect of increasing $M$ on the performance of our model. In a specific range, the scores obtained by our model increase with the increase of $M$. But the scores go down when the value of $M$ is bigger than a certain value. And our model obtains the best performance with $M=8$. We take $M=8$ in all other experiments.

### Table 5: The influence of different values of $M$.

<table>
<thead>
<tr>
<th>$M$</th>
<th>B-1</th>
<th>B-2</th>
<th>B-3</th>
<th>B-4</th>
<th>METEOR</th>
<th>ROUGE</th>
<th>CIDER</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>81.7</td>
<td>66.5</td>
<td>51.4</td>
<td>38.7</td>
<td>28.9</td>
<td>59.0</td>
<td>128.2</td>
</tr>
<tr>
<td>6</td>
<td>82.1</td>
<td>67.0</td>
<td>52.3</td>
<td>39.4</td>
<td>29.4</td>
<td>59.6</td>
<td>128.9</td>
</tr>
<tr>
<td>7</td>
<td>81.8</td>
<td>66.6</td>
<td>51.6</td>
<td>39.0</td>
<td>29.0</td>
<td>59.1</td>
<td>129.2</td>
</tr>
<tr>
<td>8</td>
<td><strong>82.2</strong></td>
<td><strong>67.6</strong></td>
<td><strong>52.4</strong></td>
<td><strong>39.7</strong></td>
<td><strong>29.7</strong></td>
<td><strong>59.7</strong></td>
<td><strong>129.2</strong></td>
</tr>
<tr>
<td>9</td>
<td>81.6</td>
<td>66.4</td>
<td>51.2</td>
<td>38.4</td>
<td>28.7</td>
<td>58.7</td>
<td>127.4</td>
</tr>
</tbody>
</table>

Limitation and Failure Cases. To clarify, the measurement of similarity between different images largely depends on the $L_2$ distance of the image feature vector. Therefore, when our algorithm wrongly selects images to guide the image captioning of the original image, it will have some negative effects. Figure 6 shows the visualization of negative examples.

4.5 Discussion

**Different Quantities of $K$ Similar Images.** Table 5 shows the influence of different values of $M$ on the performance of our model. In a specific range, the scores obtained by our model increase with the increase of $M$. But the scores go down when the value of $M$ is bigger than a certain value. And our model obtains the best performance with $M=8$. We take $M=8$ in all other experiments.

**Limitation and Failure Cases.** Figure 6 shows the visualization of negative examples.

5 CONCLUSION

We propose a novel image captioning model in a combination of Dual-GCN and Transformer with Curriculum Learning as training strategy. The visual feature encoded by an object-level GCN and an image-level GCN is designed to incorporate both local and global visual encoding. The transformer decoder is able to understand the extracted visual features to generate reasonable caption results. With a cross-review mechanism to determine the difficulty of the dataset, we take curriculum learning as the training strategy to ensure that our proposed model is trained in a manner from easy to hard. Experiments conducted on the MS COCO dataset have strongly confirmed the efficacy of our proposed method.

ACKNOWLEDGEMENT

This work was partly supported by the Key Technological Innovation Projects of Hubei Province (2018AA062), NSFC (NO. 61972298), and CAAI-Huawei MindSpore Open Fund. We use MindSpore to partly verify our DGCN network such like Faster-R-CNN.


