Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.038264

ARTICLE

Traffic Scene Captioning with Multi-Stage Feature Enhancement

Dehai Zhang*, **Yu Ma, Qing Liu, Haoxing Wang, Anquan Ren and Jiashu Liang**

School of Software, Yunnan University, Kunming, 650091, China

*Corresponding Author: Dehai Zhang. Email: dhzhang@ynu.edu.cn

Received: 05 December 2022 Accepted: 10 April 2023 Published: 08 October 2023

ABSTRACT

Traffic scene captioning technology automatically generates one or more sentences to describe the content of traffic scenes by analyzing the content of the input traffic scene images, ensuring road safety while providing an important decision-making function for sustainable transportation. In order to provide a comprehensive and reasonable description of complex traffic scenes, a traffic scene semantic captioning model with multi-stage feature enhancement is proposed in this paper. In general, the model follows an encoder-decoder structure. First, multi-level granularity visual features are used for feature enhancement during the encoding process, which enables the model to learn more detailed content in the traffic scene image. Second, the scene knowledge graph is applied to the decoding process, and the semantic features provided by the scene knowledge graph are used to enhance the features learned by the decoder again, so that the model can learn the attributes of objects in the traffic scene and the relationships between objects to generate more reasonable captions. This paper reports extensive experiments on the challenging MS-COCO dataset, evaluated by five standard automatic evaluation metrics, and the results show that the proposed model has improved significantly in all metrics compared with the state-of-the-art methods, especially achieving a score of 129.0 on the CIDEr-D evaluation metric, which also indicates that the proposed model can effectively provide a more reasonable and comprehensive description of the traffic scene.

KEYWORDS

Traffic scene captioning; sustainable transportation; feature enhancement; encoder-decoder structure; multi-level granularity; scene knowledge graph

1 Introduction

Sustainable transportation is "the provision of services and infrastructure for the movement of people and goods in a safe, affordable, convenient, efficient and resilient manner." To ensure sustainable transportation, big data analysis through methods such as [1] and data support through IoT data security transmission technologies such as [2] are used to provide decisions for transportation planning. Deep learning also has an indelible role in this, and natural language description of traffic scenes is important for assisting visually impaired people in their daily lives and in participating in traffic [3,4], as well as generating rich semantic information for drivers, thus assisting in the generation of intelligent decision suggestions, reducing driver decision time, and being important for reducing the risk of accidents [5]. This maintains the resilience of traffic as well as the sustainability of traffic by ensuring road safety. The process of describing traffic scenes in natural language is called traffic scene caption generation; the task is a cross-domain task between computer vision and natural language processing; the goal of this task is to find the most efficient pipeline to process the input scene image, represent its content and convert it into a series of words by generating connections between visual and textual elements while maintaining the fluency of the language [6]. This requires accurate recognition of the objects and targets in the scene and the ability to detect the attributes contained in multiple targets in the scene and their relationships. The most fundamental task is object detection, which is a very important technology in computer vision and has a wide range of applications, not only in the field of autonomous driving [7], but also in medical applications [8]. Based on object detection, it is a challenge to describe the scene accurately and comprehensively in complex traffic scenes.

Existing methods use images as input and generate simple captions for images using visual features, called image captioning, as shown in Fig. 1, left. However, these methods are only capable of simple recognition of the objects involved in the image and simple captioning, but when used in a scene, such as a traffic scene, the generated captions are insufficient to assist in decision generation for the people involved in the traffic and are very uninterpretable. For example, in Fig. 1, the image captioning only generates "a man is riding a motorcycle", and if this is used as an auxiliary decision generation, it is easy to ignore some important information and generate wrong auxiliary decisions, which leads to accidents. In the traffic scene captioning proposed in this paper, the relationships and attributes of the traffic objects are added, and "a man" is "beside" the "stop sign" in the caption generated by the proposed method. Such information is very important for decision generation and can provide stronger interpretability for the traffic scene image. The image captioning method, in which each individual object region can only represent its category and attributes, is unable to represent the relationship between the object regions in the traffic scene, ignoring the prevailing semantic relationship between the regions, which helps to make a more reasonable and rich semantic understanding of the traffic scene, which leads to the generation of a visual-semantic gap. For visual feature processing, most of the existing image captioning methods treat visually salient regions as a collection of individual object regions using bottom-up visual features [9]. In this, compared to directly using the extracted image features as the input to the decoder, a more effective approach is to perform further in-depth processing on the bottom-up features, retaining important information and discarding the unimportant information. However, although these methods retain most of the important information, there is still the problem of losing detailed information by using only the processed coarse-grained features as the input of the decoder.

First, to bridge the visual-semantic gap, a knowledge graph is used to provide prior knowledge for each traffic scene image in our proposed traffic scene captioning method. A knowledge graph describes the concepts and their mutual relations in the physical world in the form of symbols. Its basic units are "entity-relationship-entity" triples and entities and their related attributes-value pairs. Entities are connected through relations to form a net-like knowledge structure, as shown in Fig. 2a, which is the basic structure and a unit of the knowledge graph. Circles, namely nodes, represent entities, and the arrows represent relations. The semantic information represented by this triplet can be expressed as "A man is holding a ball" in natural language. At the same time, the entity represented by each node has some attributes. For example, as shown in Fig. 2b, some basic information about the "ball" is taken as attributes, such as color and size. In this paper, the scene knowledge graph is applied to the decoding process, which is a graph-based representation of an image that encodes objects and the relationships between them. This representation allows a comprehensive understanding of the image [10]. In this paper, a scene knowledge graph fusion module (SKGF) is designed to fuse the scene knowledge graph context-aware embeddings, which assigns different attentional strengths to each of the three semantic information (object, attribute, and relation) through three different attention modules before generating word sequences. The attention mechanism appears in all aspects [11]. Specifically, if the contextual information about the current text unit suggests that the subject or object part should be generated at this stage, then the fusion module assigns more attention to the object information, and if the predicate part should be generated at this stage, then it assigns more attention to the relationship. In this way, it adaptively adjusts the attention to different semantic information in combination with the current context, and then aligns the semantic feature units with the text. The purpose of the integration is to achieve a better fit.

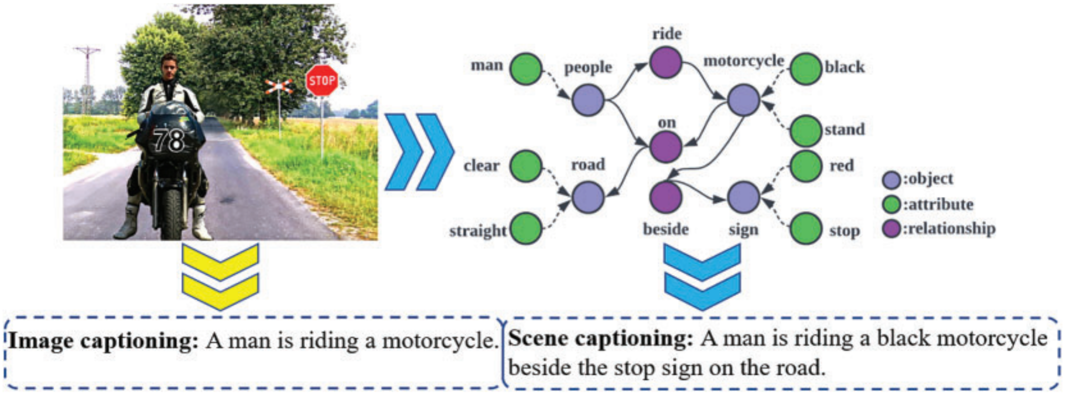
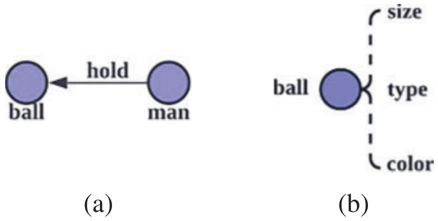


Image captioning: A man is riding a motorcycle.

Scene captioning: A man is riding a black motorcycle beside the stop sign on the road.

Figure 1: The difference between the image captioning and our traffic scene captioning



(a) (b)

Figure 2: Basic structure of the knowledge graph

Second, to address the problem of losing detailed information when processing visual features, in this paper, a multi-level granularity fusion module (MLGF) is proposed to fuse the features at different levels of granularity processed by encoders at different depth levels, to enhance the visual features of the encoder output. Specifically, the fine-grained features generated by low-level encoder processing can effectively preserve the detailed part of the traffic scene, the coarse-grained features generated by high-level encoder processing can preserve the overall features, and the new features generated by fusing the features at different levels can effectively combine the overall and detailed features to ensure the generation of more comprehensive traffic scene captioning.

In this paper, the model is evaluated by quantitative and qualitative results on the MS-COCO dataset. The experimental results show that our method achieves better results in various evaluation metrics and is effective in generating comprehensive and reasonable captions for the traffic scenes. The contributions of this paper are outlined below:

1. This paper proposes a multi-stage feature enhancement method, which includes the following:
   - This paper proposes a multi-level granularity feature fusion module (MLGF) that enhances the visual features learned by the encoder to help generate more comprehensive traffic scene captions.
   - This paper proposes a scene knowledge graph fusion module (SKGF) to enhance the features learned by the decoder to help generate richer and more reasonable traffic scene captions.
2. Extensive experiments on the MS-COCO dataset validate the effectiveness of our method, and in quantitative analysis, the model proposed in this paper achieves strong performance, especially in qualitative analysis, which fully demonstrates the superiority of our model by comparing the generated captions.

2 Related Work

The comparison between the related work and the model proposed in this paper is shown in Table 1. This section will write about the related work from the following two aspects:

Visual features only. Some early approaches [12,13] using only visual features as the source of information have achieved good results by encoding the images with convolutional neural networks (CNNs) and decoding them with recurrent neural networks (RNNs) to generate sentences. After that, references [14,15] pioneered the introduction of an attention mechanism to allow the decoder to extract image features better. Reference [16] proposed a backtracking method to integrate the attention weights of the previous time step into the attention measure of the current time step, which is more suitable for human visual consistency. In recent years, reference [9] proposed a combined bottom-up and top-down visual attention mechanism called Up-Down. This bottom-up mechanism utilizes a set of salient image regions extracted by Faster R-CNN [17], each represented by an ensemble of convolutional feature vectors. Some approaches have achieved good results with model improvements based on this bottom-up feature [18,19]. For visual feature extraction, some methods were improved on the basis of Faster R-CNN. Reference [20] proposed an end-to-end scale invariant head detection model to count the flow of people in high-density population, and achieved good results. In addition, some methods use a combination of long short-term memory (LSTM) language models and self-attention mechanisms to pursue high accuracy. Among them, reference [21] expanded the LSTM by using the attention-on-attention module, which computes another step of attention based on visual self-attention, and [22] improved the visual encoding and language models by enhancing self-attention through second-order interactions. They both employ gated linear units (GLUs) [23] to filter out valid information, and similar to them, our approach also employs GLUs to control the information transfer. In [24], a learning configuration neural module is proposed to generate "internal patterns" connecting the visual coder and the language decoder. These methods, which only use visual features as input, can achieve good results with relatively high accuracy, but the generated sentences lack richness and rationality because they do not take into account the relationship between objects.

Graphs based. Inspired by research representations in the graphics domain, studies such as [25] are devoted to generating graphs to represent scenes, and some recent approaches have introduced the use of graphs based on visual features, hoping to improve the generated sentences by encoding object attributes and relationships. Among them, reference [26] treated relations as edges in a graph and improves region-level properties by using object relations and graph convolutional networks (GCNs). Reference [27] achieved good results by fusing language-induced bias and graphs into the decoder. Reference [28] introduced a hierarchical attention-based module to learn the distinguishing features at each time step for word generation. Reference [29] treated relations as nodes in a graph and combines both semantic and geometric graphs. Reference [30] decomposed the graph into a set of subgraphs, each subgraph capturing one semantic component of the input image. In addition, some works, such as [31], which use only graph features without utilizing visual features, reduce the computational cost while obtaining good results by bridging the semantic gap between the image scene graph and the caption scene graph. Similar to most of the approaches that use graphs, GCNs [32,33] are commonly used to integrate semantic information in graphs and then further decode the sentences using the features aggregated over the whole graph.

Table 1: Comparison of related models and proposed model

Models	Features	Advantage	Disadvantage
Related models	Visual features only Graphs based	High accuracy, high speed Rich semantics	Not rich semantics Slightly lower accuracy
Proposed model	Visual features + Graphs	Advantage: Ensures accuracy while improving sentence richness and rationality	

3 Approach

3.1 Approach Overview

The entire flow of our model for implementing traffic scene captioning is shown in Fig. 3. The model generally follows the encoder-decoder architecture to generate comprehensive and reasonable traffic scene descriptions by multi-level feature enhancement. First, visual and semantic features of the scene are obtained: (1) for the visual features, bottom-up features extracted based on Faster R-CNN are used. Bottom-up features are the relevant feature vectors of the region where each element in the scene is located. Instead of simply dividing the image into regions of the same size, bottom-up features focus on content more in line with human observation habits, which helps generate more reasonable traffic scene captions; (2) for semantic features, provided by the scene knowledge graph constructed from the traffic scene images. The scene knowledge graph contains the object, the attribute of the object, and the relationship between the objects, and this paper represents both the relationship and attribute as nodes similar to the object and embed the object node, the relationship node, and the attribute node in the scene knowledge graph by the scene knowledge graph embedding method. The relationship node and the attribute node are represented by embedding the scene knowledge graph. Second, the visual features are encoded by the encoder and then fused with multi-level granularity visual features to generate new visual features and input to the decoder, which decodes them by fusing the semantic features provided by the scene knowledge graph and the new visual contextual features to finally generate the semantic description of the traffic scene.

3.2 Encoder

The encoder is used to refine the visual features extracted by the visual feature extractor. Because of the lack of interaction between the object features extracted directly by the visual feature extractor, compared to feeding the acquired bottom-up feature vectors directly to the decoder, as done in [9], the method of this paper uses an encoder consisting of multiple layers of the same encoder layer, similar to the transformer, to further process the feature vectors to deeply capture the relationships between the visual regions of the traffic scene images.

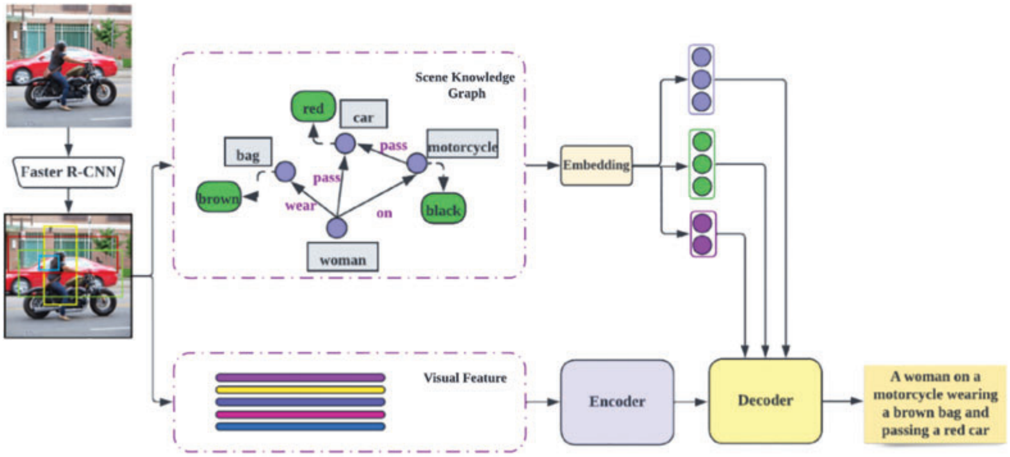


Figure 3: Overall framework of our proposed model

Our encoder is shown in Fig. 4. Through a large number of experiments and verifications, and in order to control the model scale, the encoder is composed of six layers of the same encoder layer, if the number of encoder layers is greater than six layers, it will lead to the model size is too large, and the number of encoder layers is less than six layers, which will lead to a decrease in accuracy. The encoder uses the output of each layer of the encoder layer to be used as the decoder input after feature fusion through the multi-level granularity feature fusion module.

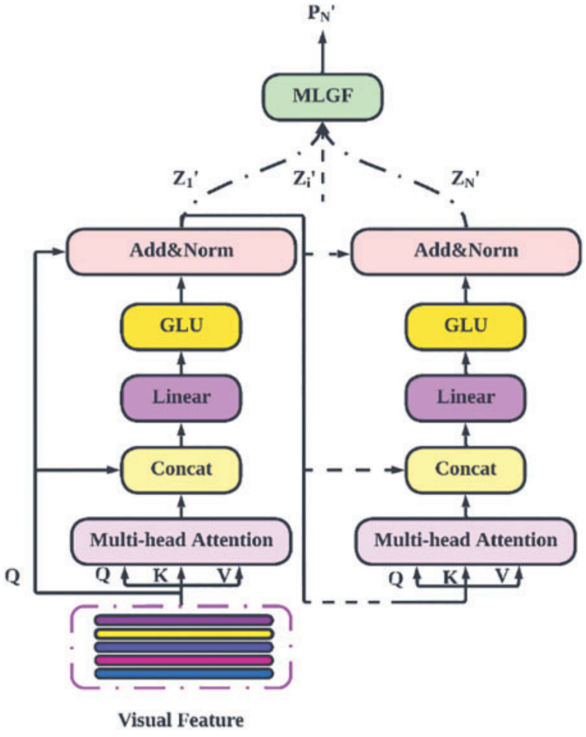


Figure 4: The framework of the encoder. The framework shows only two encoder layers. In fact, our encoder consists of six identical encoder layers

3.2.1 Encoder Layer

First, after inputting the traffic scene images, the regions of interest in the traffic scene are detected by the pre-trained Faster R-CNN, the detected image regions are subjected to feature extraction, and a set of visual feature vectors are represented as $Z = \{z_1, z_2, z_3, \ldots, z_n\}$, where $Z \in \mathbb{R}^{n\times d}$, d is the dimensionality of each of the n vectors. The feature vectors are linearly projected into queries Q, keys K and values V, and then fed into the multi-head attention mechanism, which is an integration of h identical self-attention. The adopted multi-head attention mechanism is formulated as follows, where W_i^Q, W_i^K, W_i^V, W^O is the trainable weight matrix, here, $i \in (1, h)$, and h is the number of heads of the multi-head attention mechanism. In Eq. (1), the *MultiHead* represents multi-head attention, and in Eq. (2), the *Attention* represents self-attention operation:

$$MultiHead\ (Q, K, V) = \ Concat\ (head_{\ 1}, \ldots,\ head_{\ h})\ W^O \tag{1}$$

$$Attention\,(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) V \tag{2}$$

$$Q_i = QW_i^Q, K_i = KW_i^K, V_i = VW_i^V \tag{3}$$

$$head_{\mathrm{i}} = Attention\,(Q_i, K_i, V_i) \tag{4}$$

After obtaining the output of the multi-headed attention mechanism, the input query Q is concatenated with the output of the multi-head attention mechanism and sent to the gated linear unit (GLU) through the linear layer control, which can filter the output of the multi-head attention and the invalid features in the query Q, and output the useful feature information. Finally, the encoder layer is output by the Add & Norm layer. One of the encoder layers is formulated as follows, where $i \in (1, N)$, N is the number of encoder layers in the encoder, LN denotes the LayerNorm operation and *Cat* denotes the concatenation operation:

$$f_{En}\,(Q_i, k_i, V_i) = LN\,(Q_i + GLU\,(Cat\,(MultiHead\,(Q_i, k_i, V_i)\,, Q_i))) \tag{5}$$

3.2.2 Encoder with Multi-Level Granularity Feature Fusion Module

The encoder is composed of multiple encoder layers with the same structure. In the conventional transformer, the decoder always only takes the output of the last layer of the encoder as its input:

$$Z_i' = f_{En}\left(Z_{i-1}'\right) \tag{6}$$

In this paper, the output of the i-th encoder layer is denoted as Z_i', and the i-th encoder layer takes the output of the $(i-1)$-th encoder layer Z_{i-1}' as input, $i \in (1, N)$, N is the number of encoder layers in the encoder, and the input of the decoder in the traditional transformer structure is Z_N' at this time, which is formulated as follows, where Y is the semantic sequence of the decoder output:

$$Y = f_{De}\left(Z_N'\right) \tag{7}$$

However, the output of the last layer of the encoder can only represent the global features of the traffic scene, i.e., coarse-grained features, and lacks the finer and more precise details of the traffic scene, i.e., fine-grained features, and using only the output of the last layer of the encoder as the input of the decoder will inevitably cause the loss of details, thus leading to incomplete traffic scene captioning. Therefore, to enable the decoder to understand the more detailed and precise parts of the traffic scene, this paper proposes the multi-level granularity feature fusion module (MLGF), which can focus not only on the global features of the traffic scene, i.e., coarse-grained features, but also on the features of different levels of granularity of the traffic scene. In particular, the first encoder layer focuses on the finest granularity features in the traffic scene, and the MLGF fuses the features at different levels of granularity. The MLGF consists of an element-level summation unit and a LayerNorm unit, which takes the output of each encoder layer as input and generates new visual features for the encoder that reflect the different levels of granularity of the original traffic scene. The fused features are then used as the input of the decoder. At this time, the input of the decoder is $P_N^{'}$:

$$P_N' = LN\left(ElementwiseSum\left(Z_1', Z_2', \ldots, Z_N'\right)\right) \tag{8}$$

$$Y = f_{De}\left(P_N'\right) \tag{9}$$

3.3 Decoder

The decoder fuses the visual features provided by the encoder and the semantic relation information provided by the scene knowledge graph to generate a text representation of the traffic scene image, that is, to generate traffic scene captions.

3.3.1 Scene Knowledge Graph Representation

Following [27], the scene knowledge graph is extracted by the image parser, which consists of three components: the object detector Faster R-CNN, the relationship detector MOTIFNET [34], and the attribute classifier consisting of FC-ReLU-FC-Softmax. The extracted scene knowledge graph can be represented as $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes, treating attributes and relationships as nodes similar to objects, so $\mathcal{N}$ contains object node $O = \{o_i\}$, attribute node $A = \left\{a_{i,k}\right\}$ and relationship node $R = \{r_{ij}\}$, where o_i denotes the i-th object, $a_{i,k}$ is the k-th attribute of o_i, r_{ij} is the relationship between o_i and o_j, $\mathcal{E}$ is the set of edges, and the triple $< o_i - r_{ij} - o_j >$ denotes the triple $< subject - predicate - object >$.

3.3.2 Scene Knowledge Graph Embedding with GCN

For the scene knowledge graph, there are three types of nodes that need to learn their semantic context-aware embeddings, i.e., to learn the semantic context-aware embeddings of the o_i, $a_{i,k}$ and r_{ij} nodes in the scene knowledge graph, and for each set of nodes $\mathcal{N}$, they can be represented by d-dimensional vectors, i.e., e_{o_i}, $e_{a_{i,k}}$ and $e_{r_{ij}}$. Similar to the work in [29], for the object node, contextual embedding cannot be learned using only the object node representation provided by the scene knowledge graph. We fuse the visual features v_{o_i} detected by the Faster R-CNN with the object node features to obtain the object node feature representation, and the object node context-aware embedding is calculated as follows:

$$x_{o_i} = g_o\left(f_o\left(Cat\left(v_{o_i}, e_{o_i}\right)\right)\right) + v_{o_i} \tag{10}$$

For the attribute node, since the attribute is the attribute of the object, it is relative to the object, so the attribute is integrated with its object context. The context-aware embedding of the attribute node is as follows:

$$x_{a_i} = g_a\left(x_{o_i}, f_a\left(e_{a_{i,k}}\right)\right) + f_a\left(e_{a_{i,k}}\right) \tag{11}$$

For the relationship node, similar to the attribute node, it is relative to the subject object node o_i and the object node o_j, so integrating it with its object node, the context-aware embedding of the relationship node is as follows:

$$x_{r_{ij}} = g_r\left(x_{o_i}, f_r\left(e_{r_{ij}}\right), x_{o_j}\right) + f_r\left(e_{r_{ij}}\right) \tag{12}$$

where g_o, g_{a} and g_r are spatial graph convolutions with independent parameters but with the same structure, using the FC-ReLU-Dropout layer structure. f_o, f_{a} and f_r are feature projection layers; similarly, vectors are input to the fully connected layer, followed by a RELU. x_{o_i}, x_{a_i}, and $x_{r_{ij}}$ are learned semantic context-aware embeddings.

3.3.3 Decoder with Scene Knowledge Graph Fusion Module

The decoder uses the encoder-enhanced visual features $P_N^{'}$ to generate sentences describing the traffic scene. To accurately describe the relationships and attributes of the objects in the traffic scene, this paper proposes the scene knowledge graph fusion module (SKGF). The semantic features x_{o_i}, x_{a_i} and $x_{r_{ij}}$ provided by the scene knowledge graph are fused to generate a more reasonable caption of the traffic scene. In a sentence describing a traffic scene, a word usually corresponds to a semantic unit in the image. For example, the subject and object parts correspond to the object unit in the semantic unit, and the predicate part corresponds to the relationship unit in the semantic unit. Therefore, when fusing semantic features, it is necessary to consider the alignment of text words with semantic units, and use SKGF to implement the alignment of text with object units, attribute units, and relationship units to fuse semantic features in a suitable way.

Specifically, given the LSTM output h_t from step t of the decoder, we compute the resulting context vectors of h_t with object features x'_{o_i}, attribute features x'_{a_i}, and relationship features $x'_{r_{ij}}$ as follows:

$$x'_{o_i} = MHA^O\left(h_t, x_{o_i}\right) \tag{13}$$

$$x'_{a_i} = MHA^A\left(h_t, x_{a_i}\right) \tag{14}$$

$$x'_{r_{ij}} = MHA^R\left(h_t, x_{r_{ij}}\right) \tag{15}$$

where MHA^O, MHA^A and MHA^R are multi-head attention mechanisms. With the following multi-head attention mechanisms, the input is no longer the linear projection query Q, the key K and the value V of the feature vectors, but the output h_t of the LSTM and the semantic features x_{o_i}, x_{a_i} and $x_{r_{ij}}$, calculated as follows:

$$MHA^O = MultiHead\ \left(h_t, x_{o_i}, x_{o_i}\right) \tag{16}$$

$$MHA^A = MultiHead\ \left(h_t, x_{a_i}, x_{a_i}\right) \tag{17}$$

$$MHA^R = MultiHead\ \left(h_t, x_{r_{ij}}, x_{r_{ij}}\right) \tag{18}$$

The details of the self-attention mechanisms used in the multi-head attention mechanisms are calculated as follows, where d is the vector dimension of h_t and the semantic features x_{o_i}, x_{a_i}, $x_{r_{ij}}$:

$$Attention\left(h_t, x_{o_i}, x_{o_i}\right) = softmax\left(\frac{h_t {x_{o_i}}^T}{\sqrt{d}}\right) x_{o_i} \tag{19}$$

$$Attention\left(h_t, x_{a_i}, x_{a_i}\right) = softmax\left(\frac{h_t {x_{a_i}}^T}{\sqrt{d}}\right) x_{a_i} \tag{20}$$

$$Attention\left(h_t, x_{r_{ij}}, x_{r_{ij}}\right) = softmax\left(\frac{h_t {x_{r_{ij}}}^T}{\sqrt{d}}\right) x_{r_{ij}} \tag{21}$$

Our decoder is shown in Fig. 5. In particular, at decoding time step t, the input to the LSTM is set to the current input word W_t, the visual features provided by the average pooled encoder $\overline{\mathrm{P}'_{\mathrm{N}}}$ plus the previously saved context vector $\boldsymbol{c}_{t-1}$, and the previous LSTM hidden state h_{t-1}, and the current LSTM output is calculated as follows:

$$h_t = LSTM\left(\left[W_e\Pi_t, \overline{P'_N} + \boldsymbol{c}_{t-1}\right], h_{t-1}\right) \tag{22}$$

where $W_e \in \mathbb{R}^{E\times\Sigma}$ is a word embedding matrix for a vocabulary of size Σ, and Π_t is the one-hot encoding of the input word at time step t. h_t is used as a query vector to compute x'_{o_i}, x'_{a_i} and $x'_{r_{ij}}$.

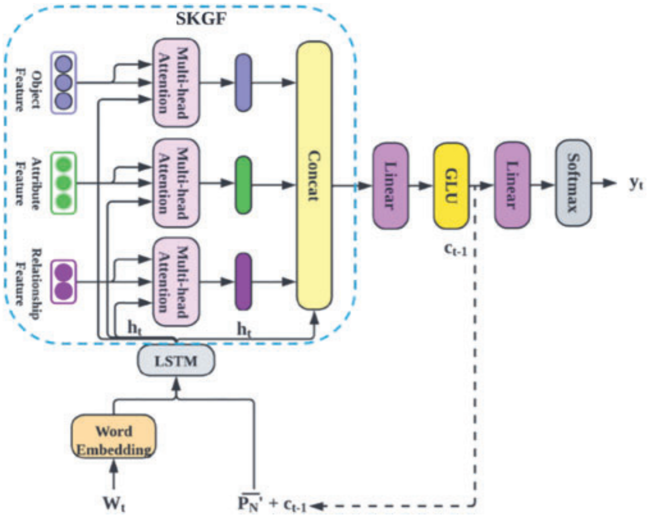


Figure 5: The framework of the decoder, where $\boldsymbol{c}_{t-1}$ is initialized to 0 at the beginning, and the SKGF module is the scene knowledge graph fusion module. Three different multi-head attention mechanisms in SKGF are used to adaptively adjust the attention to different semantic information according to the current context, so as to align the semantic features and text features

The output h_t of the LSTM is then fused with the aligned context vectors x'_{o_i}, x'_{a_i}, $x'_{r_{ij}}$ of the three semantic units, and then filtered using the gating unit to learn which semantic unit the current word is about and decide what the decoder should pay more attention to at the moment. The final filtered context vector is used as input to the generator, which contains a linear layer as well as a softmax layer for generating probability scores to predict the next word:

$$p\left(y_t|y_{1:t-1}\right) = softmax\left(W_p\boldsymbol{c}_t + \boldsymbol{b}_p\right) \tag{23}$$

where W_p is the learning weight and $\boldsymbol{b}_p \in \mathbb{R}^{\Sigma}$ is the learning bias.

4 Experiments

4.1 Dataset

This paper has carried out experiments on the popular dataset MS-COCO [35], which contains 123,287 images, containing 82,783 training images and 40,504 validation images, each containing five sentences describing the image. In this paper, the Karpathy data split [13] was used for evaluation, of which 113,287 were training images, 5000 were validation images, and 5000 were testing images. For the caption text, this paper converts all words to lowercase, remove rare words that occur less than five times, and modify the maximum length of each sentence to 16.

4.2 Evaluation Metrics

In this paper, five standard automatic evaluation metrics are used to evaluate the proposed method, namely, BLEU [36], METEOR [37], ROUGE-L [38], CIDEr-D [39], and SPICE [40].

4.3 Objective

For a given sequence of ground truth captions, we train our model by minimizing the following cross-entropy loss, where θ is the model parameter and T is the word sequence length:

$$L_{XE}(\theta) = -\sum\nolimits_{t=1}^{T} \log\left(p_\theta\left(y_t^* | y_{1:t-1}^*\right)\right) \tag{24}$$

then, the following negative expectations were minimized using the CIDEr-D [39] score as a bonus:

$$L_R(\theta) = -\boldsymbol{E}_{y_{1:T}\sim p_0}\left[r\left(y_{1:T}\right)\right] \tag{25}$$

where r is the scoring function, and the gradient of this loss is approximated according to the self-critical sequence training (SCST) [41] as:

$$\nabla_\theta L_R(\theta) \approx -\left(r\left(y_{1:T}^s\right) - r\left(\hat{y}_{1:T}\right)\right)\nabla_\theta \log p_\theta\left(y_{1:T}^s\right) \tag{26}$$

4.4 Implementation Details

Visual Genome (VG) [42] has rich scene graph annotations such as classes of objects, attributes of objects, and pairwise relationships, so this paper used the Visual Genome (VG) dataset to train the image parser, including object detector, attribute classifier and relationship detector, and after filtering the dataset by objects, attributes and relationships, 305 objects, 103 attributes, and 64 relationships were left for training.

For the input word, object, attribute, and relationship categories, this paper uses word embeddings of size 1024, 128, 128, and 128, respectively. For the feature projection layer (f_o, f_{a}, f_r) and GCN (g_o, g_{a}, g_r), we set the output dimension to 1024. Similarly, for visual features, set its dimension as the input to the encoder to 1024, which is also the hidden size of the LSTM in the decoder. In this paper, the number of encoder layers $N = 6$ and multiple attention mechanism $h = 8$ are set. During training, the Adam [43] optimizer is used to train 35 epochs under cross-entropy loss, set the batch size to 10, initialize the learning rate to 2×10^{-4}, and decay 0.8 every three epochs. Then, by optimizing the CIDEr-D reward, reinforcement learning [41] is used to train the model to 50 epochs, and the initial learning rate is set as 3×10^{-5}. In the inference stage, a beam search with a beam size of 2 is used.

In this paper, the training and inference process are completed on the server equipped with RTX A5000. Under the training of cross-entropy loss, it takes an average of 0.156 s to train an iteration, and an epoch contains 11,328 iterations, and the total time to train an epoch averages 30 min, counting the simple inference tests done during training. Under the training of CIDEr-D score optimization, it takes an average of 0.280 s to train an iteration, and an epoch also contains 11,328 iterations, and the total time to train an epoch is an average of 54 min. In the inference stage, 5000 images are used for inference, and the average inference time for one image is 0.046 s, and the inference time for 5000 images is 230 s.

4.5 *Performance Comparison and Analysis*

4.5.1 Comparison Methods

This paper compares the proposed traffic scene captioning model with the following image captioning models, and for a fair comparison, all methods are experimented and validated on the same COCO Karpathy test split: (1) SCST [41] proposes a reinforcement learning method based on the idea of self-critical to train sequence generation models; (2) Up-Down [9] proposes a visual attention mechanism combining bottom-up and top-down attention with a two-layer LSTM as the core model for sequence generation, specifically, SCST and Up-Down are two powerful baselines for using more advanced self-critical rewards and visual features; (3) Hierarchical attention network (HAN) [44] uses hierarchical features to predict different words; (4) CIDErBtw [45] proposes a new training strategy to encourage the uniqueness of each image-generated caption; (5) Human consensus-oriented image captioning (HCO-IC) [46] explicitly uses human consensus to measure in advance the quality of ground truth captions and directly encourages the model to learn high quality captions with high priority; (6) Recurrent fusion network (RFNet) [47] uses a recurrent fusion network to fuse different source features and exploit complementary information from multiple encoders; (7) Spatio-temporal memory attention (STMA) [48] uses an attention model to learn the spatio-temporal relationships of image captions; (8) SubGC [30] decomposes the graph into a series of subgraphs by that capture meaningful objects; (9) Graph convolutional networks plus long short-term memory (GCN-LSTM) [26] treats visual relations as edges in a graph to help refine region-level features; (10) Scene graph auto-encoder (SGAE) [27] proposes to introduce self-coding of graphs into the model; (11) Visual semantic units alignment (VSUA) [29] fuses semantic and geometric (geometrical) graphs; (12) SG2Caps [31] uses only graph labels to bridge the semantic gap between image scene graphs and caption scene graphs; (13) Divergent-convergent attention (DCA) [49] proposes a new divergence-convergence attention to focus on fine-grained semantic information.

4.5.2 Quantitative Analysis

This paper compares the performance of our proposed traffic scene captioning model with the image captioning models on the MS-COCO dataset trained with cross-entropy loss, and the results are summarized in Table 2. The top part of Table 2 shows the models that use only visual features as the feature source, the middle part shows the models that employ the semantic information of the graph, and the bottom part shows our proposed model that utilizes both visual features and the semantic information provided by the scene knowledge graph. From the results, it can be seen that with cross-entropy loss training, our approach improves significantly over the model using only visual features and outperforms the models using semantic information in all evaluation metrics. This shows the advancement and effectiveness of our model that fuses visual features as well as semantic features, while incorporating encoder multi-level granularity features.

To fairly compare the performance of our model, this paper also reports the performance of the image captioning models trained under CIDEr-D Score optimization compared to our proposed traffic scene captioning model, as shown in Table 3. It can be seen from the reported results that the model trained with CIDEr-D Score optimization shows a significant improvement in all metrics relative to cross-entropy loss, in particular, the CIDEr-D score improves from 119.3 to 129.0, a 9.7% improvement, and 0.6% to 2.9% improvement in other metrics. Meanwhile, our model outperformed other models in most evaluation metrics.

Table 2: Performance comparison with the existing methods on the MS-COCO Karpathy test split. All models are trained with cross-entropy loss. All values are reported as percentages (%), where B@N, M, R, C, and S are short for the BLEU@N, METEOR, ROUGE-L, CIDEr-D, and SPICE scores. "-" indicates that the value is not mentioned in the published work

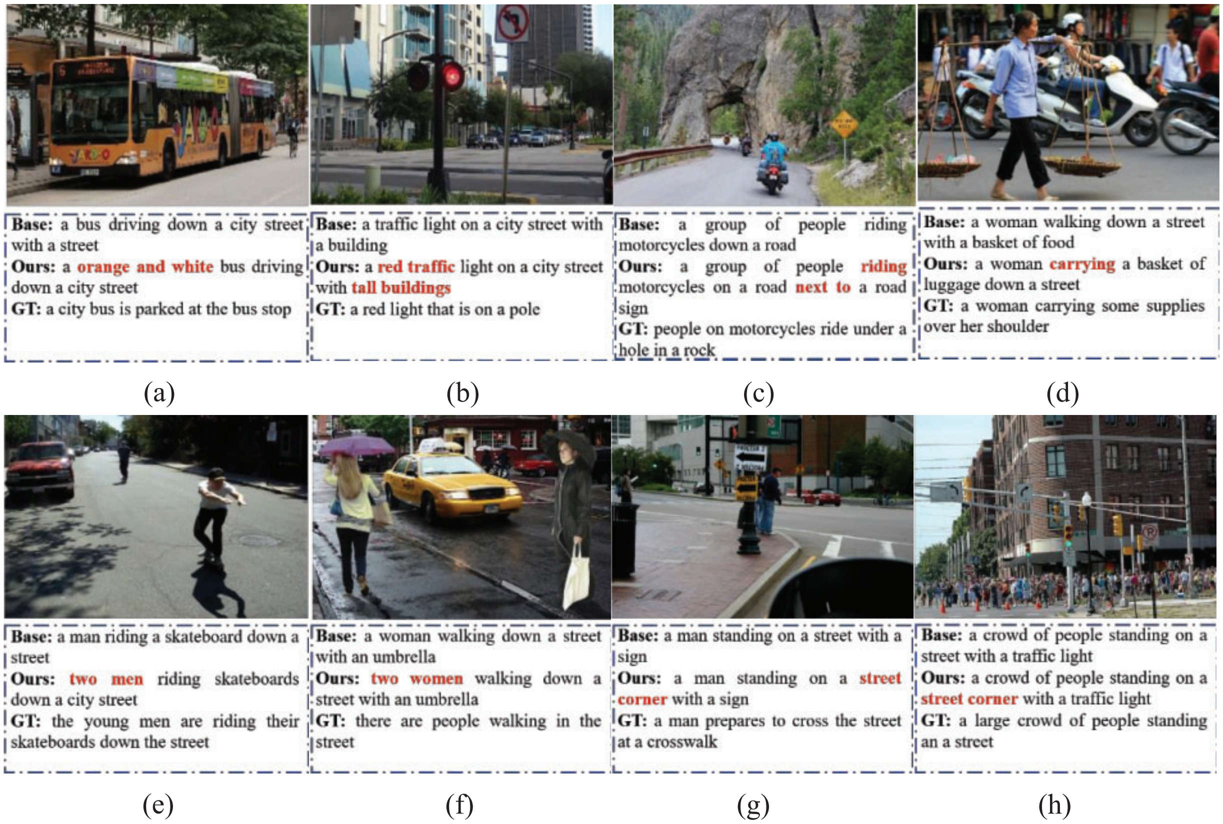
Model	B@1	B@2	B@3	B@4	M	R	C	S
SCST:Att2all [41]	-	-	-	30.0	25.9	53.4	99.4	-
Up-Down [9]	77.2	-	-	36.2	27.0	56.4	113.5	20.3
HAN [44]	77.2	61.2	47.7	36.2	27.5	56.6	114.8	20.6
CIDErBtw [45]	-	-	45.4	35.0	27.6	55.7	112.4	20.7
HCO-IC [46]	-	-	-	35.7	27.7	56.4	114.8	-
RFNet [47]	76.4	60.4	46.6	35.8	27.4	56.5	112.5	20.5
STMA [48]	77.4	61.5	47.6	36.5	27.4	56.8	114.4	20.5
SubGC [30]	76.8	-	-	36.2	27.7	56.6	115.3	20.7
GCN-LSTM [26]	77.3	-	-	36.8	27.9	57.0	116.3	20.9
SG2Caps [31]	-	-	-	32.0	26.2	54.9	104.4	19.5
Ours	77.8	62.2	48.5	37.6	28.2	57.4	119.3	21.5

Table 3: Performance comparison with the existing methods on the MS-COCO Karpathy test split. All models are trained with CIDEr-D Score Optimization. All values are reported as percentages (%), where B@N, M, R, C, and S are short for the BLEU@N, METEOR, ROUGE-L, CIDEr-D, and SPICE scores. "-" indicates that the value is not mentioned in the published work

Model	B@1	B@2	B@3	B@4	M	R	C	S
SCST:Att2all [41]	-	-	-	34.2	26.7	55.7	114.0	-
Up-Down [9]	79.8	-	-	36.3	27.7	56.9	120.1	21.4
HAN [44]	**80.9**	64.6	49.8	37.6	27.8	58.1	121.7	21.5
HCO-IC [46]	-	-	-	37.1	28.2	57.7	126.1	-
RFNet [47]	79.1	63.1	48.4	36.5	27.7	57.3	121.9	21.2
STMA [48]	80.2	64.4	49.7	37.7	28.2	58.1	125.9	21.7
GCN-LSTM [26]	80.5	-	-	38.2	28.5	58.3	127.6	22.0
SGAE [27]	80.8	-	-	38.4	28.4	58.6	127.8	22.1
VSUA [29]	-	-	-	38.4	28.5	58.4	128.6	22.0
SG2Caps [31]	-	-	-	33.0	26.2	55.6	112.3	19.4
DCA [49]	80.4	-	-	38.4	28.7	58.5	128.0	22.2
Ours	80.5	**65.1**	**50.6**	**38.7**	**28.8**	**58.7**	**129.0**	**22.4**

4.5.3 Qualitative Analysis

To qualitatively evaluate the validity of our proposed model, Fig. 6 shows some examples of generated traffic scene captions. This paper adopted the strong baseline model Up-Down as the baseline. In particular, we reimplemented the Up-Down model, and our reimplemented Up-Down model achieved better performance than the official Up-Down model and took this as our baseline, as shown in Table 4, where "*" represents the result of the re-implementation. The "Base", "Ours", and "GT" in Fig. 6 represent the captions from the baseline, our model, and the ground truth.


Base: a bus driving down a city street with a street
Ours: a **orange and white** bus driving down a city street
GT: a city bus is parked at the bus stop

(a)

Base: a traffic light on a city street with a building
Ours: a **red traffic** light on a city street with **tall buildings**
GT: a red light that is on a pole

(b)

Base: a group of people riding motorcycles down a road
Ours: a group of people **riding** motorcycles on a road **next to** a road sign
GT: people on motorcycles ride under a hole in a rock

(c)

Base: a woman walking down a street with a basket of food
Ours: a woman **carrying** a basket of luggage down a street
GT: a woman carrying some supplies over her shoulder

(d)

Base: a man riding a skateboard down a street
Ours: **two men** riding skateboards down a city street
GT: the young men are riding their skateboards down the street

(e)

Base: a woman walking down a street with an umbrella
Ours: **two women** walking down a street with an umbrella
GT: there are people walking in the street

(f)

Base: a man standing on a street with a sign
Ours: a man standing on a **street corner** with a sign
GT: a man prepares to cross the street at a crosswalk

(g)

Base: a crowd of people standing on a street with a traffic light
Ours: a crowd of people standing on a **street corner** with a traffic light
GT: a large crowd of people standing an a street

(h)

Figure 6: Example results of the captions generated by our model, Up-Down baseline, and ground truth

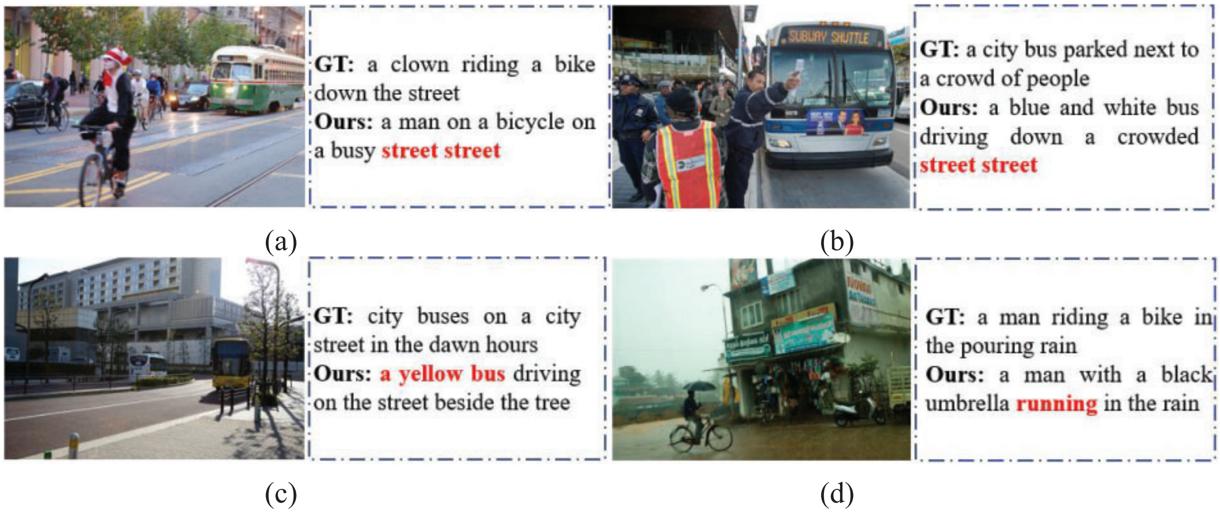
Table 4: Comparison between the performance of our reimplemented Up-Down model and the Up-Down model in the published work

Model	B@1	B@2	B@3	B@4	M	R	C	S
Up-Down [9]	79.8	-	-	36.3	27.7	56.9	120.1	21.4
Up-Down∗	79.2	**63.4**	**48.7**	**36.9**	**27.8**	**57.5**	**122.3**	21.3

In general, the eight examples in Fig. 6 show that the baseline can generate smooth and relatively traffic scene-appropriate captions, but our model can generate more reasonable and comprehensive traffic scene captions, i.e., rich and reasonable traffic scene captions through object attributes, objects and relationships, and more fine-grained feature processing. Specifically, Figs. 6a and 6b provide more detailed descriptions of the object attributes in the traffic scene, e.g., in Fig. 6a, our approach describes the attribute of "traffic light" as "red" and the attribute of "buildings" as "red". The baseline can only identify the objects of "traffic light" and "buildings", but cannot describe their attributes. Figs. 6c and 6d describe the relationship between objects in the traffic scene. In particular, for example, in Fig. 6c, our method generates a traffic scene caption in which the relationship between "people" and "motorcycles" is "riding", and the relationship between "motorcycles" and "road sign" is "next to", which the baseline cannot express. Figs. 6e and 6f show that our model can describe more detailed and comprehensive information in the traffic scene. For example, in Fig. 6e, the baseline can only focus on "a man" in the traffic scene, but our model can focus on another object in the traffic scene, that is, "two men". The "street corner" in Figs. 6g and 6h shows that our model can also identify and describe objects in the traffic scene more accurately. In summary, our model has the following advantages: (1) it can make the identification and description of objects in the traffic scene more accurate; (2) it can describe the attributes of objects in detail; (3) the relationships between objects are clearer and more accurate; and (4) it can pay attention to more fine-grained information in the traffic scene.

4.5.4 Failure Cases Analysis

In order to comprehensively analyze the model proposed in this paper, the failure cases are analyzed. Fig. 7 shows some examples of generated scene captions. The "Ours" and "GT" in Fig. 7 respectively represent the captions generated from the model proposed in this paper and the ground truth.



GT: a clown riding a bike down the street
Ours: a man on a bicycle on a busy **street street**

(a)

GT: a city bus parked next to a crowd of people
Ours: a blue and white bus driving down a crowded **street street**

(b)

GT: city buses on a city street in the dawn hours
Ours: **a yellow bus** driving on the street beside the tree

(c)

GT: a man riding a bike in the pouring rain
Ours: a man with a black umbrella **running** in the rain

(d)

Figure 7: Example results of the captions generated by our model and ground truth

In Fig. 7a, although the model proposed in this paper recognizes "busy street", the word "street" appears twice, and the same problem also appears in Fig. 7b. Through a large number of experimental studies, it is found that this is because "street" appears more frequently in the scene knowledge graph, which leads to "street" being introduced into the model as noise. Therefore, the same word will be generated twice during the generation of captions, resulting in the sentence is not smooth and even grammar errors. In Fig. 7c, although the model in this paper describes the color of "bus", only the color of one "bus" is recognized. In Fig. 7d, due to the influence of noise, the model of this paper incorrectly identifies the cyclist as the person running in the rain. From the above examples, it can be concluded that while using the semantic features provided by the scene knowledge graph to describe the semantic relations, the introduction of noise is an inevitable and urgent problem. However, the frequency of such problems is low, which has little impact on the performance of the model.

4.5.5 Ablative Analysis

To demonstrate the effectiveness of the two core modules (MLGF and SKGF) of our model for traffic scene captioning, this paper quantifies the contribution of our proposed module through an ablation study. All the models participated in the ablation study used the same hyperparameters and were evaluated through cross-entropy loss training. The results of our ablation study are shown in Table 5.

Table 5: Ablations of our method, evaluated on the MS-COCO Karpathy split. All models are trained with cross-entropy loss

Model			Evaluation metrics							
ID	MLGF (En)	SKGF (De)	B@1	B@2	B@3	B@4	M	R	C	S
1	×	×	75.8	59.8	45.9	35.0	27.1	56.0	110.4	20.3
2	✓	×	76.4	60.2	46.2	35.2	27.1	56.2	111.1	20.3
3	×	✓	77.7	61.9	48.1	37.1	28.2	57.5	118.2	21.4
4	✓	✓	77.8	62.2	48.5	37.6	28.2	57.4	119.3	21.5

In model 1, the encoder does not have the multi-level granularity feature fusion (MLGF) module and the decoder does not have the scene knowledge graph fusion (SKGF) module. Specifically, for the encoder, after deleting the MLGF module, the output of the last layer of the encoder is directly used as the input of the decoder, as shown in Fig. 8b. For the decoder, after deleting the SKGF module, the output of the LSTM is directly used as the input of the linear layer, as shown in Fig. 8c. Compared with model 1, model 2 adds the MLGF module to the encoder. Similarly, compared with model 3, model 4 adds the MLGF module to the encoder. From the results in Table 5, it can be seen that model 2 and model 4 have some improvement in most metrics compared with model 1 and model 3 without the encoder MLGF module. Thus, it can be seen that our encoder MLGF module focuses not only on the global information of the traffic scene, but also for the fine-grained information on the traffic scene, which helps to improve the performance of the model. Compared with model 1, model 3 uses SKGF to fuse the scene knowledge graph information in the decoder. Similarly, compared with model 2, model 4 also fuses the scene knowledge graph information in the decoder. As seen from the results in Table 5, compared with model 1 and model 2 without fusion scene knowledge graph in the decoder, model 3 and model 4 have improved by 1.1%∼7.8% and 1.1%∼8.2%, respectively, in all metrics, which indicates that the semantic information of objects, attributes, and relationships provided by the scene knowledge graph plays a very important role in traffic scene captioning.

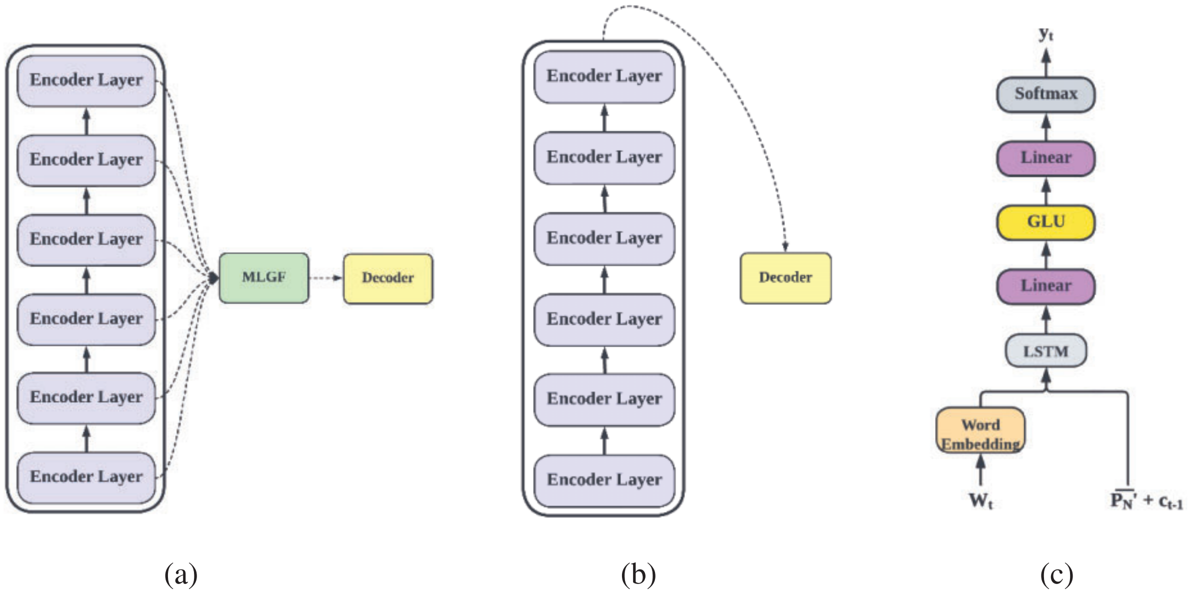


Figure 8: Frameworks for ablative analysis: (a) Encoder with the Multi-level granularity feature fusion module (MLGF); (b) Encoder without the Multi-level granularity feature fusion module (MLGF); (c) Decoder without the scene knowledge graph fusion module (SKGF)

5 Conclusions

This paper proposes a multi-stage feature enhancement approach, i.e., a deeper refinement of scene information through visual and semantic feature enhancement in the encoding and decoding stages, which helps to generate traffic scene captions more comprehensively and rationally. Experimental results show that our model achieves better performance in most metrics, especially in CIDEr-D and SPICE evaluation metrics, which have achieved scores of 129.0 and 22.4, respectively, which is a great improvement over other methods that only use visual features or scene graphs. However, the model proposed in this paper is still too large, the total number of model parameters is about 104 M, which may lead to a decrease in efficiency, so in future work, we will try to reduce the size of the model to improve efficiency. In the meantime, we will try more effective feature enhancement methods for further improvement, especially for improving fusion methods for multi-level granularity features.

Acknowledgement: This research was supported by the Yunnan Provincial Key Laboratory of Software Engineering, the Kunming Provincial Key Laboratory of Data Science and Intelligent Computing.

Funding Statement: This research was funded by (i) Natural Science Foundation China (NSFC) under Grant Nos. 61402397, 61263043, 61562093 and 61663046; (ii) Open Foundation of Key Laboratory in Software Engineering of Yunnan Province: No. 2020SE304. (iii) Practical Innovation Project of Yunnan University, Project Nos. 2021z34, 2021y128 and 2021y129.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: D. Zhang, Y. Ma; data collection: Y. Ma; analysis and interpretation of results: D. Zhang, Y. Ma, H. Wang; draft manuscript preparation: H. Wang, A. Ren, J. Liang. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The authors confirm that the data supporting the findings of this study are available within the article.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. K. Punia, M. Kumar, T. Stephan, G. G. Deverajan and R. Patan, "Performance analysis of machine learning algorithms for big data classification: Ml and AI-based algorithms for big data analysis," *International Journal of E-Health and Medical Communications (IJEHMC)*, vol. 12, no. 4, pp. 60–75, 2021.

[2] S. M. Nagarajan, G. G. Deverajan, U. Kumaran, M. Thirunavukkarasan, M. D. Alshehri *et al.,* "Secure data transmission in internet of medical things using RES-256 algorithm," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 12, pp. 8876–8884, 2022.

[3] D. Gurari, Y. Zhao, M. Zhang and N. Bhattacharya, "Captioning images taken by people who are blind," in *Proc. of European Conf. on Computer Vision*, Glasgow, UK, pp. 417–434, 2020.

[4] H. Ahsan, N. Bhalla, D. Bhatt and K. Shah, "Multi-modal image captioning for the visually impaired," *arXiv Preprint arXiv:2105.08106*, 2021.

[5] W. Li, Z. Qu, H. Song, P. Wang and B. Xue, "The traffic scene understanding and prediction based on image captioning," *IEEE Access*, vol. 9, pp. 1420–1427, 2020.

[6] M. Stefanini, M. Cornia, L. Baraldi, S. Cascianelli, G. Fiameni *et al.,* "From show to tell: A survey on deep learning-based image captioning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 539–559, 2022.

[7] J. Wang, M. Zhu, D. Sun, B. Wang, W. Gao *et al.,* "MCF3D: Multi-stage complementary fusion for multi-sensor 3D object detection," *IEEE Access*, vol. 7, pp. 90801–90814, 2019.

[8] N. Razmjooy, F. R. Sheykhahmad and N. Ghadimi, "A hybrid neural network–world cup optimization algorithm for melanoma detection," *Open Medicine*, vol. 13, no. 1, pp. 9–16, 2018.

[9] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson *et al.,* "Bottom-up and top-down attention for image captioning and visual question answering," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, pp. 6077–6086, 2018.

[10] M. Suhail, A. Mittal, B. Siddiquie, C. Broaddus, J. Eledath *et al.,* "Energy-based learning for scene graph generation," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, pp. 13936–13945, 2021.

[11] D. Rao, S. Huang, Z. Jiang, G. G. Deverajan and R. Patan, "A dual deep neural network with phrase structure and attention mechanism for sentiment analysis," *Neural Computing and Applications*, vol. 33, no. 3, pp. 11297–11308, 2021.

[12] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning and A. Y. Ng, "Grounded compositional semantics for finding and describing images with sentences," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 207–218, 2014.

[13] A. Karpathy and F. F. Li, "Deep visual-semantic alignments for generating image descriptions," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, pp. 3128–3137, 2015.

[14] O. Vinyals, A. Toshev, S. Bengio and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, pp. 3156–3164, 2015.

[15] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville *et al.,* "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. of the 32nd Int. Conf. on Machine Learning*, Lille, France, pp. 2048–2057, 2015.

[16] Y. Qin, J. Du, Y. Zhang and H. Lu, "Look back and predict forward in image captioning," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, pp. 8367–8375, 2019.

[17] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[18] Z. J. Zha, D. Liu, H. Zhang, Y. Zhang and F. Wu, "Context-aware visual policy network for fine-grained image captioning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 2, pp. 710–722, 2019.

[19] L. Huang, W. Wang, Y. Xia and J. Chen, "Adaptively aligned image captioning via adaptive attention time," in *Proc. of the 33rd Int. Conf. on Neural Information Processing Systems*, Red Hook, NY, USA, pp. 8942–8951, 2019.

[20] S. D. Khan and S. Ba, "Scale and density invariant head detection deep model for crowd counting in pedestrian crowds," *The Visual Computer*, vol. 37, no. 8, pp. 2127–2137, 2021.

[21] L. Huang, W. Wang, J. Chen and X. Y. Wei, "Attention on attention for image captioning," in *Proc. of the IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, Seoul, Korea (South), pp. 4634–4643, 2019.

[22] Y. Pan, T. Yao, Y. Li and T. Mei, "X-Linear attention networks for image captioning," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, pp. 10971–10980, 2020.

[23] Y. N. Dauphin, A. Fan, M. Auli and D. Grangier, "Language modeling with gated convolutional networks," in *Proc. of the 34th Int. Conf. on Machine Learning*, Sydney, Australia, pp. 933–941, 2017.

[24] X. Yang, H. Zhang and J. Cai, "Learning to collocate neural modules for image captioning," in *Proc. of the IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, Los Alamitos, CA, USA, pp. 4249–4259, 2019.

[25] S. Yan, C. Shen, Z. Jin, J. Huang, R. Jiang *et al.,* "Pcpl: Predicate-correlation perception learning for unbiased scene graph generation," in *Proc. of the 28th ACM Int. Conf. on Multimedia*, Seattle, WA, USA, pp. 265–273, 2020.

[26] T. Yao, Y. Pan, Y. Li and T. Mei, "Exploring visual relationship for image captioning," in *Proc. of the European Conf. on Computer Vision (ECCV)*, Munich, Germany, pp. 684–699, 2018.

[27] X. Yang, K. Tang, H. Zhang and J. Cai, "Auto-encoding scene graphs for image captioning," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, pp. 10685–10694, 2019.

[28] X. Li and S. Jiang, "Know more say less: Image captioning based on scene graphs," *IEEE Transactions on Multimedia*, vol. 21, no. 8, pp. 2117–2130, 2019.

[29] L. Guo, J. Liu, J. Tang, J. Li, W. Luo *et al.,* "Aligning linguistic words and visual semantic units for image captioning," in *Proc. of the 27th ACM Int. Conf. on Multimedia*, New York, NY, USA, pp. 765–773, 2019.

[30] Y. Zhong, L. Wang, J. Chen, D. Yu and Y. Li, "Comprehensive image captioning via scene graph decomposition," in *Proc. of European Conf. on Computer Vision*, Glasgow, UK, pp. 211–229, 2020.

[31] K. Nguyen, S. Tripathi, B. Du, T. Guha and T. Q. Nguyen, "In defense of scene graphs for image captioning," in *Proc. of the IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, Montreal, QC, Canada, pp. 1407–1416, 2021.

[32] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv Preprint arXiv:1609.02907*, 2016.

[33] J. Bastings, I. Titov, W. Aziz, D. Marcheggiani and K. Sima'an, "Graph convolutional encoders for syntax-aware neural machine translation," *arXiv Preprint arXiv:1704.04675*, 2017.

[34] R. Zellers, M. Yatskar, S. Thomson and Y. Choi, "Neural motifs: Scene graph parsing with global context," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA, pp. 5831–5840, 2018.

[35] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona *et al.,* "Microsoft coco: Common objects in context," in *Proc. of European Conf. on Computer Vision*, Zurich, Switzerland, pp. 740–755, 2014.

[36] K. Papineni, S. Roukos, T. Ward and W. J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, Pennsylvania, USA, pp. 311–318, 2002.

[37] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proc. of the Acl Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, Ann Arbor, Michigan, USA, pp. 65–72, 2005.

[38] C. Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Proc. of the Text Summarization Branches Out*, Barcelona, Spain, pp. 74–81, 2004.

[39] R. Vedantam, C. Lawrence Zitnick and D. Parikh, "Cider: Consensus-based image description evaluation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, pp. 4566–4575, 2015.

[40] P. Anderson, B. Fernando, M. Johnson and S. Gould, "Spice: Semantic propositional image caption evaluation," in *Proc. of European Conf. on Computer Vision*, Amsterdam, Netherlands, pp. 382–398, 2016.

[41] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross and V. Goel, "Self-critical sequence training for image captioning," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp. 7008–7024, 2017.

[42] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata *et al.,* "Visual genome: Connecting language and vision using crowd sourced dense image annotations," *International Journal of Computer Vision*, vol. 123, no. 1, pp. 32–73, 2017.

[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv Preprint arXiv:1412.6980*, 2014.

[44] W. Wang, Z. Chen and H. Hu, "Hierarchical attention network for image captioning," in *Proc. of the AAAI Conf. on Artificial Intelligence*, Honolulu, Hawaii, USA, vol. 33, no. 1, pp. 8957–8964, 2019.

[45] J. Wang, W. Xu, Q. Wang and A. B. Chan, "Compare and reweight: Distinctive image captioning using similar images sets," in *Proc. of European Conf. on Computer Vision*, Glasgow, UK, pp. 370–386, 2020.

[46] Z. Wang, Z. Huang and Y. Luo, "Human consensus-oriented image captioning," in *Proc. of the Twenty-Ninth Int. Joint Conf. on Artificial Intelligence*, Yokohama Yokohama, Japan, pp. 659–665, 2021.

[47] W. Jiang, L. Ma, Y. G. Jiang, W. Liu and T. Zhang, "Recurrent fusion network for image captioning," in *Proc. of European Conf. on Computer Vision*, Munich,Germany, pp. 499–515, 2018.

[48] J. Ji, C. Xu, X. Zhang, B. Wang and X. Song, "Spatio-temporal memory attention for image captioning," *IEEE Transactions on Image Processing*, vol. 29, pp. 7615–7628, 2020.

[49] J. Ji, Z. Du and X. Zhang, "Divergent-convergent attention for image captioning," *Pattern Recognition*, vol. 115, pp. 107928, 2021.