Latent Memory-Augmented Graph Transformer for Visual Storytelling

Mengshi Qi*
School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China

Jie Qin
College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China

Di Huang
State Key Laboratory of Software Development Environment, Beihang University, Beijing, China

Zhiqiang Shen
CMU, Pittsburgh, USA
MBZUAI, Abu Dhabi, UAE

Yi Yang
ReLER Lab
University of Technology Sydney
Sydney, Australia

Jiebo Luo
Department of Computer Science
University of Rochester
NY, USA

ABSTRACT

Visual storytelling aims to automatically generate a human-like short story given an image stream. Most existing works utilize either scene-level or object-level representations, neglecting the interaction among objects in each image and the sequential dependency between consecutive images. In this paper, we present a novel Latent Memory-augmented Graph Transformer (**LMGT**), a Transformer based framework for visual story generation. LMGT directly inherits the merits from the Transformer, which is further enhanced with two carefully designed components, i.e., a graph encoding module and a latent memory unit. Specifically, the graph encoding module exploits the semantic relationships among image regions and attentively aggregates critical visual features based on the parsed scene graphs. Furthermore, to better preserve inter-sentence coherence and topic consistency, we introduce an augmented latent memory unit that learns and records highly summarized latent information as the story line from the image stream and the sentence history. Experimental results on three widely-used datasets demonstrate the superior performance of LMGT over the state-of-the-art methods.

CCS CONCEPTS

• **Computing methodologies** → **Natural language generation**; *Scene understanding*.

KEYWORDS

Visual Storytelling, Transformer, Scene Graph, Memory Network

ACM Reference Format:
Mengshi Qi, Jie Qin, Di Huang, Zhiqiang Shen, Yi Yang, and Jiebo Luo. 2021. Latent Memory-Augmented Graph Transformer for Visual Storytelling. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3474085.3475236

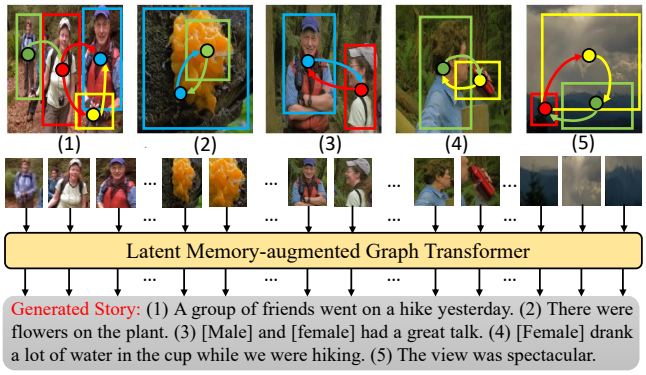

Figure 1: Illustration of the task of visual storytelling. Given an image stream with its corresponding scene graphs shown in the first row, our proposed LMGT can encode image regions integrated with critical semantic relationships into feature embeddings, and then decode them to a human-like, coherent, and informative story.

1 INTRODUCTION

With the popularity of social networks, a tremendous number of users routinely share a series of photos, along with their related comments/stories on social media platforms such as *Instagram* and *Flickr*. Consequently, a new task of visual storytelling [1, 24, 27, 34, 37, 43, 62, 64, 72], which aims at automatically generating a narrative story for an image stream (as shown in Figure 1), has recently attracted increasing attention in the multimedia community.

Given a stream of images, humans are capable of composing a suitable story line and then generating a sequence of sentences. A good automatic storyteller should imitate humans to completely understand the visual contents and their relationships (*e.g.*, objects, actions, scenes, and human-object interactions), and explicitly exploit the sequential and contextual dependencies along the image sequence. As such, most existing approaches for (single) image captioning [9, 14, 16, 22, 25, 32, 53, 56, 66, 75] cannot be directly adopted to handle this task since they neglect the sequential dependencies over the image stream when composing a story with multiple isolated sentences rather than a holistic story line. In addition, the

*Corresponding author: qidash@gmail.com. This work is partly supported by the Innovation Research Group Project of NSFC under Grant 61921003.

MM '21, October 20–24, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475236

inter-image temporal gap and visual change in an image stream are often far greater than the inter-frame variations in a video, making video captioning methods [4, 13, 21, 29, 41, 49, 50, 54, 73, 76–78] perform unsatisfactorily for visual storytelling either.

Recently, a number of approaches have been specifically proposed to address this challenging task [19, 20, 23, 27, 33, 43, 64, 67, 68, 70, 72]. Most of them employ either scene- or object-level image representations based on Recurrent Neural Networks (RNNs), *e.g.*, Long Short-Term Memory (LSTM) [17] and Gated Recurrent Units (GRU) [8]. However, the majority of them neglect the semantic interactions among objects in each image. Moreover, RNNs based methods usually suffer from the insufficient memory to store a long sequence with complex temporal dependencies [9, 32, 46, 47, 51, 59], thus failing to generate a coherent, human-like long narrative. As an alternative, the Transformer [59] has been recently proposed with potentially better performance than RNNs in sequence modeling and achieved remarkable successes in a variety of tasks, especially in language understanding. However, how to effectively employ the Transformer for visual storytelling has not yet been explored.

Since the capability of long-term sequential modeling is crucial for story generation, in this paper, we aim to unleash the potential of the Transformer to handle our challenging task. However, it is non-trivial to account for all the following specific requirements of the task in addition to long-term modeling: 1) the need to understand visual contents and their implicit semantic relationships in each image, and 2) the need to maintain inter-sentence coherence and topic consistency throughout the whole story. To meet these requirements in the context of the Transformer and inspired by the human way of telling a story, we propose a novel *Latent Memory-augmented Graph Transformer (**LMGT**)* for visual storytelling. LMGT shares the basic architecture with the Transformer, but enhances it with two carefully designed components, *i.e.*, a graph encoding module and a latent memory unit. Specifically, the graph encoding module encodes visual embeddings simultaneously integrated with structured semantic relationships among various image regions by constructing a scene graph. In addition, the latent memory unit is introduced to capture latent contextual clues as the story line, as well as record the previous history of images and generated sentences to preserve topic consistency and inter-sentence coherence, wherein the memory state can be updated based on the current input and previous memory. Finally, we integrate the feature embeddings learned from both the graph encoding module and the latent memory unit, and then decode them into a human-like, coherent and informative story for the given image stream.

The main advantages of LMGT are three-fold. First, LMGT exploits the power of the Transformer for visual storytelling, whose delicately designed self-attention mechanisms can substantially improve long-term sequence modeling. Second, we enhance the vanilla Transformer by incorporating the graph encoding module, which enables the Transformer to handle more complex structures and capture important semantic relationships by virtue of a scene graph. Last but not least, the proposed latent memory unit leverages highly summarized latent information to propagate history for future sentence generation, significantly improving the basic self-attention and maintaining inter-sentence coherence and topic consistency. Note that our proposed components can be easily plugged into the vanilla Transformer, while maintaining its unique advantages of parallel computation and flexible modularity.

In summary, the main contributions of this work include:

1) We introduce a novel Latent Memory-guided Graph Transformer for visual storytelling by simultaneously capturing the visual relationships in each image and the sequential dependencies across the image stream.

2) We design a graph encoding module to exploit the interaction between image regions in each image and further enhance the feature embeddings with the detected semantic relationships through scene graph parsing.

3) We propose a latent memory unit to learn the highly summarized latent information as story lines, and store the sequential history to make the generated sentences more coherent and the overall topic more consistent.

4) Extensive experiments on three public benchmarks show that the proposed LMGT achieves the state-of-the-art performance, while qualitatively generating more human-like coherent stories than the competitors.

2 RELATED WORK

Transformer and Visual Captioning. The Transformer [59] has achieved significant successes in recent years, especially in natural language processing (NLP), such as machine translation [59], text generation [12], pre-trained language modeling [10, 12, 71], multimodal representation [6, 57, 58] and documents summarization [38]. The Transformer has strong abilities to draw global dependencies between input and output and takes the advantage of large-batch parallel training. Recently, a few attempts [9, 16, 22, 31, 32, 53, 77] have leveraged the Transformer for image and video captioning. For instance, Herdade *et al.* [16] captured the geometric weights between entities in the image as the attention of the Transformer for generating captions. Huang *et al.* [22] enhanced the self-attention by scaling the weights of the final attended information considering image contextual information. Inspired by the advantages of the Transformer, we enhance the vanilla Transformer with graph encoding and latent memory recording in a more effective way to address visual story generation, which are significantly different from [9, 31]. To the best of our knowledge, we are the first to propose such a Transformer-based model to address this specific task, which may provide a reference in this line of research.

Visual Storytelling. The task was first introduced in [24] with a specific dataset, *i.e.*, the *Visual StoryTelling Dataset (VIST)*. To tackle the challenging task, on the one hand, prior efforts [27, 43, 64, 72] attempted to directly adapt the methods in visual captioning (*e.g.*, RNNs based sequence-to-sequence models with attention mechanisms) to story generation. For example, Kim *et al.* [27] introduced two levels of hierarchical RNNs with attention mechanisms in terms of global encoding level and local image level to address multi-image cued story generation. On the other hand, several recent works [1, 18, 19, 33, 34, 37, 62, 67, 70] have devoted to incorporating semantic knowledge to improve the quality of the generated story. For instance, Li *et al.* [33] inferred semantic concepts and captured cross-modal rules for visual storytelling, and Hsu *et al.* [19] distilled a wealthy of words from an external knowledge graph to generate more interesting stories. In addition, another line of

approaches [20, 23, 68] have demonstrated the effectiveness of adopting reinforcement learning into the model with newly proposed rewards, *e.g.*, Wang *et al.* [68] designed an adversarial reward learning approach by implicitly imitating human demonstrations. However, the story generated from most of the previous methods are not yet satisfied, due to the limited memory capacity of RNNs based architectures and neglecting the wealthy interaction among image regions. Differently, we propose a novel Transformer-based framework in this paper, and we equip the model with two specifically designed components to better fulfill this task, by simultaneously capturing structural semantic relationships, latent story lines, and long-term sequential dependencies.

3 PRELIMINARY

3.1 Transformer and Self-Attention

Recently, the vanilla Transformer model [59] and its variants have obtained significant achievements in various domains, especially in NLP. The key to its success is largely attributed to the proposed self-attention mechanism. This mechanism learns a query matrix Q, a key matrix K of dimension d_k and the corresponding value matrix V. Formally, it can be formulated as follows:

$$\text{Att}(Q, K, V) = \text{Softmax}(\frac{QK^T}{\sqrt{d_k}})V. \tag{1}$$

Furthermore, the Transformer employs the multi-head attention to combine h scaled dot-product attentions running in parallel, enabling the model to jointly attend to important information from various embeddings at different positions. Given the input queries, keys, and values which can be mapped onto h subspaces, the h-head self-attention can be calculated as follows:

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(H_1, \cdots, H_h)W^O, \\ \text{where} \quad H_i &= \text{Att}(QW_i^Q, KW_i^K, VW_i^V), \end{aligned} \tag{2}$$

where W_i^Q, W_i^K and $W_i^V \in \mathbb{R}^{\frac{d}{h} \times d}$ denote the independent head projection matrix, respectively, and $W^O = \{W_i^O \in \mathbb{R}^{\frac{d}{h} \times d}, i \in \{1, \cdots, h\}\}$ refers to a fully connected layer to concatenate the output from the h heads.

Generally, there are L layers in the encoder and decoder of the Transformer model. In the l-th layer, the multi-head self-attention module takes the hidden states of the previous layer, *i.e.*, H^{l-1}, as input and obtains attention-aware output, which is then fed to a feed-forward layer. Besides, residual connection [15] and layer normalization [3] are also leveraged in each layer.

4 PROPOSED APPROACH

4.1 Overview

Problem Formulation. We define the input image stream as $\mathcal{I} = \{I_1, \cdots, I_N\}$, where N indicates the total number of images in the stream. Our goal is to generate a short story y composed of N sentences, *i.e.*, $y = \{y_1, \cdots, y_N\}$, where each sentence y_n consists of T words, *i.e.*, $y_n = \{w_1, \cdots, w_T\}$. To capture semantic relationships among different image regions, we construct a set of scene graphs $\mathcal{G}$ for the given images based on a pre-trained scene graph parser, *i.e.*, $\mathcal{G} = \{G_1, \cdots, G_N\}$, where G_n denotes the scene graph of the n-th image in the stream.

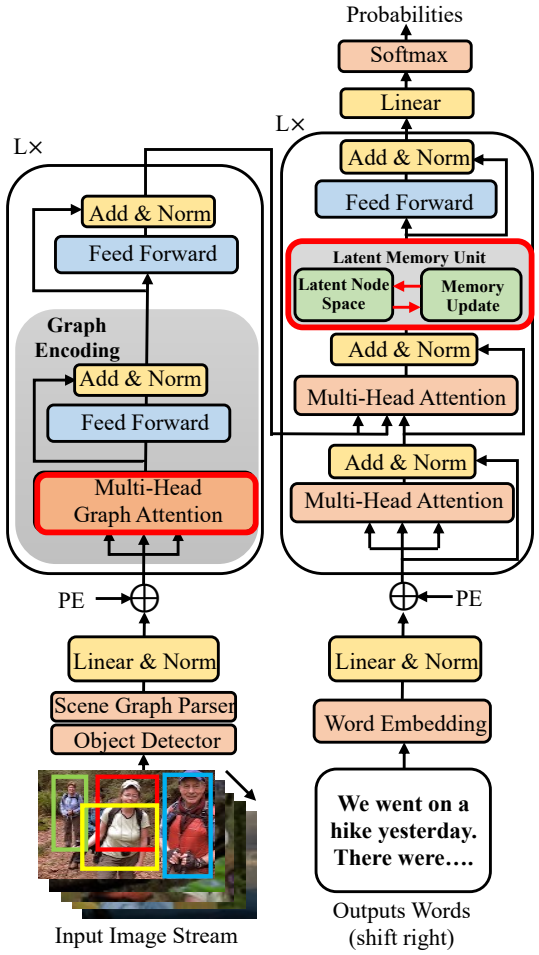

Figure 2: Overview of the proposed *Latent Memory-augmented Graph Transformer*, which mainly consists of two carefully designed components marked with the red borders: a *graph encoding module* to obtain implicit semantic relational embeddings of input image regions based on scene graphs, and a *latent memory unit* to help the Transformer record the important contextual and historic information as latent memory. "PE" denotes Positional Encoding.

As illustrated in Figure 2, the overall framework of our proposed method follows a typical Transformer-based encoder-decoder architecture. The encoder and decoder are both composed of several stacks of multi-head attention layers and feed-forward layers. Moreover, we introduce a graph encoding module in the encoder, to encode the input image regions with the corresponding semantic relationships based on the parsed scene graph, which transfers the structural semantic knowledge from images to text descriptions. In the meantime, a latent memory unit is incorporated into the decoder, in order to capture and record latent historical information as the story line to maintain inter-sentence coherence and topic consistency. Finally, the story decoder generates a coherent and informative human-like story.

4.2 Graph Encoding Module

In order to effectively encode the constructed scene graph into hidden states and infer the semantic corretion among image regions, we propose a graph encoding module (GEM). GEM consists of a graph attention layer, a feed-forward layer, as well as residual connection and layer normalization. In this module, we input the node representations extracted from the object detector, and then enrich the new relation-aware node embeddings by aggregating the neighborhood information from the scene graph. Most importantly, the module can learn to assign high attention weights to the critical semantic relations in each image, which is essential to compose an informative story.

Scene Graph Construction. Scene graphs [35, 45, 48, 69, 74] can encode structured knowledge from visual images, which generally contain objects and semantic relationships between objects, such as *"flower-on-plant"* in Figure 1 (2). In our work, for the n-th image, we first utilize a pre-traied Faster R-CNN [52] as the object detector and obtain a set of $\mathcal{K}$ objects, denoted by $\mathcal{V}_n = \{v_{n,1}, \cdots, v_{n,\mathcal{K}}\}$. Then, we adopt a scene graph parser [74] pre-trained on the Visual Genome dataset [30] to detect the relationships between objects and construct the scene graph $G_n = \{\mathcal{V}_n, \mathcal{E}_n\}$, where $\mathcal{V}_n$ and $\mathcal{E}_n$ denote the set of nodes (*i.e.*, objects) and edges (*i.e.*, relationships), respectively. An edge $e_{n,(i,j)}$ represents the directed connection from node $v_{n,i}$ to $v_{n,j}$. Additionally, we indicate the incoming and outgoing neighbor sets of the node $v_{n,i}$ as $\mathcal{N}_{n,i}^{in}$ and $\mathcal{N}_{n,i}^{out}$, respectively. Initially, we employ an embedding layer to convert the feature of each node $v_{n,i}$ (*i.e.*, the visual feature of an object) and edge $e_{n,(i,j)}$ (*i.e.*, the word embedding of a relationship label) in the graph to the dense vectors $\hat{v}_{n,i}$ and $\hat{e}_{n,(i,j)}$ of the same dimension, respectively.

Owing to the information propagated through the directed edges in the scene graph, our graph encoding module can learn two kinds of representations for each node: 1) the incoming representation that can aggregate the features from the incoming edges and the corresponding incoming neighbor nodes; 2) the outgoing representation which can be obtained from the outgoing edges and the corresponding outgoing neighbor nodes. We denote $\overrightarrow{H}_{n,i}^{l}$ and $\overleftarrow{H}_{n,i}^{l}$ as the incoming and outgoing representations of node $v_{n,i}$ at the l-th layer, respectively. The initial input embedding of each node can be set as follows:

$$\overrightarrow{H}_{n,i}^{l} = \overleftarrow{H}_{n,i}^{l} = W_v \cdot \hat{v}_{n,i} + b_v, \tag{3}$$

where W_v and b_v denote the learnable weight and bias matrices, respectively.

Then, we adopt the graph attention [61] as the aggregation operator to capture the global semantic information and exploit the critical relationships in the scene graph. The joint representation of each node is a result of aggregating its own embedding and the embeddings of connected edges and neighbor nodes, which can be formulated as follows:

$$\widetilde{v}_{n,(i,j)}^{l} = W_e \cdot (\overrightarrow{H}_{n,i}^{l} \| \hat{e}_{n,(i,j)}^{l} \| \overleftarrow{H}_{n,j}^{l}) + b_e, \tag{4}$$

where W_e and b_e are the learnable weight and bias matrices, respectively, and $\cdot\|\cdot$ refers to the concatenation operator. Furthermore, we compute the multi-head graph attention of all incoming and outgoing relationships of each node:

$$\begin{aligned} \overrightarrow{g}_{n,i}^{l} &= \|_{o=1}^{h} \Big(\sum_{j \in \mathcal{N}_i^{out}} \alpha_{n,(i,j)}^{x} \widetilde{v}_{n,(i,j)}^{l} W_v^o \Big) \cdot W^O, \\ \alpha_{n,(i,j)}^{o} &= \frac{\exp\Big(\frac{\overrightarrow{H}_{n,i}^{l-1} W_q^o \cdot (\widetilde{v}_{n,(i,j)}^{t} W_k^o)^\top}{\sqrt{d_k}}\Big)}{\sum_{j \in \mathcal{N}_i^{out}} \exp\Big(\frac{\overrightarrow{H}_{n,i}^{l-1} W_q^o \cdot (\widetilde{v}_{n,(i,j)}^{t} W_k^o)^\top}{\sqrt{d_k}}\Big)}, \end{aligned} \tag{5}$$

where $\overrightarrow{g}_{n,i}^{l}$ denotes the new incoming representation for node $v_{n,i}$ after the graph attention operation, and the new outgoing representation $\overleftarrow{g}_{n,i}^{l}$ can be obtained in the similar way.

After the graph attention sub-layer, we add a feed-forward layer, and leverage the residual connection and layer normalization to further enhance the representations as follows:

$$\begin{aligned} \overrightarrow{H}_{n,i}^{l} &= \text{LayerNorm}(\overrightarrow{g}_{n,i}^{l} + \overrightarrow{H}_{n,i}^{l-1}), \\ \overleftarrow{H}_{n,i}^{l} &= \text{LayerNorm}(\overleftarrow{g}_{n,i}^{l} + \overleftarrow{H}_{n,i}^{l-1}). \end{aligned} \tag{6}$$

Finally, the updated representation $H_{n,i}$ of the i-th node in image I_n can be achieved through concatenating the forward and backward embeddings with a linear transformation:

$$H_{n,i} = W_H \cdot (\overrightarrow{H}_{n,i}^{L} \| \overleftarrow{H}_{n,i}^{L}), \tag{7}$$

where W_H denotes the learnable weight matrix, and L refers to the number of layers in the graph encoding module.

4.3 Latent Memory Unit

Even though the vanilla Transformer shows the strong capability in long-term sequence modeling, it is still challenging to adapt it to our specific task due to its inability to preserve the core story line and history information that helps to ensure topic consistency and inter-sentence coherence. History information can provide rich semantic clues to generate subsequent more coherent sentences in a story. For example, humans can describe the current image as *"working in the office room"*, by imagination and inference based on the objects observed in the previous images, *e.g.*, *table*, *male*, and *computer*, which make up a story line.

To this end, we incorporate a latent memory unit into our LMGT. The memory unit has three main functions: 1) mapping the encoded graph feature into a latent space to obtain the highly summarized latent memory as the story line, 2) augmenting the hidden states with additional memory slots to store more semantic context, and 3) recurrently updating the memory by a gating function to record sequential history information.

Capturing the Latent Memory. To further summarize the latent memory as the story line from learned visual features, we propose an latent graph by mapping the original feature of each node (*i.e.*, $\{H_{n,i}\}$ in Eq. (7)) into a set of additional latent nodes, and then augmented those nodes with the previous memory. Specifically, we denote the introduced latent nodes as $Z = \{z_1, \cdots, z_m\}$ to represent the learned latent memory, where m indicates the number of latent nodes and $m << \mathcal{K}$. Formally, the mapping process from the original node to the latent one can be defined as:

$$z_j = \sum_{i=1}^{\mathcal{K}} \phi(H_{n,i}, \theta_j) W^\top H_{n,i}, \quad 1 \leqslant j \leqslant m, \tag{8}$$

where $\phi(H_{n,i}, \theta_j) = H_{n,i} \cdot \theta_j^\top$ refers to a projection function to map node feature $H_{n,i}$ to the latent node z_j, and θ_j indicates the learnable parameter of the j-th latent node.

Memory Construction. Specifically, in the proposed latent memory unit, the hidden states of keys and values are both augmented with the learned latent memory and extra memory slots to encode more semantic contextual clues, which are set as plain learnable vectors and can be updated step-by-step during training. Given the query set Q and the latent node set Z_t at step t, we formulate the memory-augmented attention operation as:

$$\begin{aligned} &\mathcal{S}(Q) = \text{Att}(W^Q Q, K, V), \\ &\text{where} \quad K = [W^K(Q \| y_{t-1} \| Z_t)], \quad V = [W^V(Q \| y_{t-1} \| Z_t)], \end{aligned} \tag{9}$$

where y_{t-1} denotes the embedding of the last output. It is worth noting that the proposed memory-augmented attention can also be adopted in a multi-head attention fashion, by repeating weight matrices (*i.e.*, W^Q, W^K and W^V) for h times, and finally concatenating the results of all the heads.

Memory Update. Furthermore, we propose a recurrent memory updating strategy to record sequential history information, as shown in the green box in Figure 3. In particular, when our model decodes the N-th image at the t-th time step, the memory unit can perform multi-head attention on the representations from the latent nodes Z_t and the memory embedding $\mathcal{M}_{t-1} \in \mathbb{R}^{T_m \times d}$ from the previous time step, where T_m denotes the length of recurrent memory state. We denote the input query matrix as $Q = \mathcal{M}_{t-1}$, and the key and value matrices as K and V during memory updating, respectively. We impose the residual connections onto the memory unit during the decoding process, which can be formulated as follows:

$$\widetilde{C}_t = \text{MLP}(\mathcal{S}(\mathcal{M}_{t-1}) + \mathcal{M}_{t-1}) + \mathcal{S}(\mathcal{M}_{t-1}) + \mathcal{M}_{t-1}, \tag{10}$$

where $\text{MLP}(\cdot)$ denotes the multi-layer perceptron. Furthermore, we introduce two types of gates to balance the inputs from $\mathcal{M}_{t-1}$ and y_{t-1}, *i.e.*, the forget gate G_t^f and the input gate G_t^i, which are respectively formulated as:

$$\begin{aligned} G_t^f &= \tanh(\mathcal{M}_{t-1}) \cdot W^f + Y_{t-1} \cdot V^f, \\ G_t^i &= \tanh(\mathcal{M}_{t-1}) \cdot W^i + Y_{t-1} \cdot V^i, \end{aligned} \tag{11}$$

where W and V refer to learnable weights in each gate. Both gates are used to control what information should be retained from the previous memory states, maintaining the inter-sentence coherence and topic consistency during the story generation. Thus, they are the most critical components in the memory unit. Note that the proposed memory unit employs the multi-head attention to encode the memory states and obtain the multiple memory slots rather than a single one as in LSTM [17] or GRU [7]. This can significantly boost the capability of modeling complex long-term structures. Then, the final output with the gating function is calculated as:

$$\mathcal{M}_t = \text{sigmoid}(G_t^f) \odot \mathcal{M}_{t-1} + \text{sigmoid}(G_t^i) \odot \tanh(\widetilde{C}_t), \tag{12}$$

where $\odot$ means the Hadamard product, and $\mathcal{M}_t$ is the updated memory of the latent memory unit at step t.

Propagation from Latent Memory. In order to generate the whole story considering the learned latent memory, we update

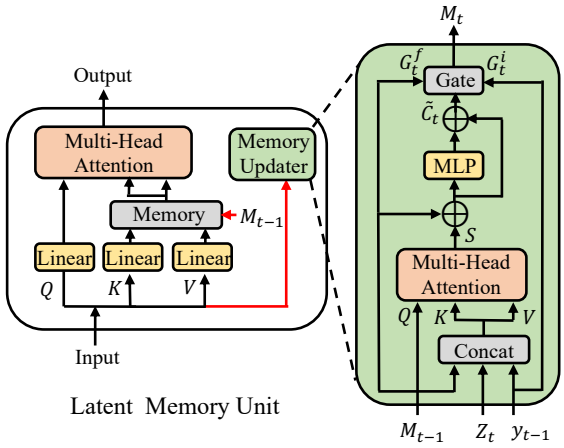

Figure 3: Illustration of memory construction and update in the proposed *latent memory unit*, where the embeddings of K and V can be concatenated with extra previous memory slots M_{t-1}, and the memory updater would update the current memory M_t.

the node features via propagating the information from the latent nodes with memory back to the original nodes:

$$\widetilde{H}_{n,i} = \text{Relu}(\sum_{j=1}^{d} \phi((H_{n,i} \| \mathcal{M}_{t-1}), \theta_j) \cdot z_j), \quad 1 \leqslant i \leqslant \mathcal{K}, \tag{13}$$

where the output $\widetilde{H}_{n,i}$ during the propagation can be leveraged as the final representation including the latent memory for feature $H_{n,i}$.

4.4 Story Decoder

The story decoder in our proposed LMGT takes the output of the encoder as input, and outputs each word of the sentence in a story. It shares a similar architecture with the encoder, containing L identical self-attention blocks and an augmented latent memory unit, followed by residual connection and layer normalization. When the decoder generates the t-th word in the n-th sentence, we denote $\hat{w}_t^n \in \mathbb{R}^{d' \times 1}$ as the embedding vector of the t-th word, and then the input embedding matrix at the t-th time step is:

$$W_{<t}^n = [\hat{w}_0^n; \cdots; \hat{w}_{t-1}^n], \quad W_{<t}^n \in \mathbb{R}^{d' \times t}, \tag{14}$$

where $\hat{w}_0^n$ denotes the feature vector of the start token of a sentence. In the (l+1)-th block, the input feature $H_{<t}^l \in \mathbb{R}^{d' \times t} = (h_1^l, \cdots, h_t^l)$ is fed to a multi-head self-attention sub-layer:

$$A_{:,t}^{l+1} = \text{MultiHead}(H_{:,t}^l, H_{<t}^l, H_{<t}^l), \tag{15}$$

where $H_{:,t}^l \in \mathbb{R}^{d' \times 1}$, $A_{:,t}^l \in \mathbb{R}^{d' \times 1}$. Note that we denote the input of the first block when $l = 0$ as $h_t^0 = \hat{w}_{t-1}^n$. Afterwards, the output of the self-attention sub-layer, *i.e.*, $A_{:,t}^{l+1}$, is aggregated with the feature embeddings of the image stream learned from the encoder and the memory states through the latent memory unit as described in Sec. 4.3. Finally, we can get the output after applying the feed-forward layer, and then fed it to the classifier to predict the next word based on the pre-defined vocabulary.

4.5 Training Objective

Given a sequence of ground-truth sentences $y^*_{1:N}$ with words $w^n_{1:T}$, we train our model with the cross-entropy loss function as follows:

$$\mathcal{L}_{\mathrm{CE}}(\theta) = -\sum_{n=1}^{N}\sum_{t=1}^{T}\log(p_\theta(w^n_t|w^n_{1:t-1}, y^*_{1:N-1})), \tag{16}$$

where θ denotes the model's parameters, and N and T refer to the total number of sentences in a story and the number of words in a sentence, respectively. We adopt the beam search [40] to generate the sentence after decoding.

5 EXPERIMENTS

5.1 Datasets and Settings

Visual Storytelling Dataset (VIST) [24] totally contains 210,819 images and 50,200 stories collected from 10,117 *Flicker* albums, which are annotated with a number of event titles by Amazon Mechanical Turk (AMT). Each album includes five images and the corresponding story composed of five sentences. Following [24], we split all the samples into three sets, *i.e.*, 40,098 in the training set, 4,988 in the validation set, and 5,050 in the test set, respectively. In our experiments, we define an image album or image stream as a sequence of 5 images following [24].

Disney Dataset [43] and **NYC Dataset** [43]. Disney includes 7,717 blog posts and 60,545 images collected with the searching topic of "Disneyland". Accordingly, NYC consists of 11,863 blog posts and 78,467 images collected with "NYC" as the topic. Following [43], we take 70%/10%/20% of the whole dataset to form the training/validation /test set on both datasets, respectively.

Metrics. In our experiments, we adopt five widely-used automatic metrics in visual captioning, *i.e.*, BLEU [42], ROUGE-L [36], METEOR [11], CIDEr-D [60] and SPICE [2]. We adopt the public source codes released by Microsoft COCO Evaluation Server to calculate all the above-mentioned metrics [5].

Compared Methods. We compare our model with the following state-of-the-art methods: **seq2seq** [24], **BARNN** [39], **HAtt-Rank** [72], **HPSR** [64], **AREL** [68], **SRT** [65], **KLST** [70], **HSRL** [23], **VSCMR** [33], **SGVST** [67], **KLEV** [19] and **INet** [26]. Additionally, we choose two baseline approaches for image/video captioning, *i.e.*, **CNN-RNN** [63] and **HRNN** [28], and adapt them to our task by averaging the image features in the stream and concatenating all the sentences to become a short story[1].

5.2 Implementation Details

We implement our proposed model with PyTorch on two NVIDIA Tesla V100 GPUs. To parse the scene graph of each image, we adopt Faster-RCNN [52] with VGG-16 [55] or ResNet-152 [15] backbone as the object detector, and MOTIFS [74] as the relationship detector. Based on the confidence scores obtained by the two detectors, we select top-10 objects and top-20 relationships in each scene graph. We set the feature dimension of each image patch as 4,096, and then encode those features into a 512-dimensional embedding through a fully connected layer followed by the ReLU activation. For story generation, we build a specific vocabulary with 9,837 words, each of which appears at least three times in the training set, and employ the most frequent 150 object categories and 50 relationship labels from *Visual Genome* [30]. Each word can be initially embedded as a 512-dimensional vector by using GloVe [44], and sinusoidal positional encoding [59] is adopted to represent word positions in the sequence. Following the same hyper-parameter setting in [59], we set the input feature size as 512 for all self-attention layers and feed-forward layers in the Transformer, and our LMGT includes $h = 8$ heads in the multi-head self-attention layers. Both the encoder and decoder contain six identical self-attention layers (*i.e.*, $L = 6$), while the encoder has an additional graph encoding module and the decoder has an augmented latent memory unit. In the latent memory unit, we set the length of the recurrent memory state as 1 (*i.e.*, $T_m = 1$) and the dimension of extra memory vectors as 100 (*i.e.*, the number of latent nodes $d = 100$). During training, we adopt the Adam optimizer and set the learning rate as 0.0015. In total, our model is trained with 300 epochs, and the batch size is set to 100 image-story pairs. During decoding, we adopt beam search with the size of 3. Finally, we choose the best model when the METEOR score reaches the highest on the validation set, due to the advantage of METEOR over other metrics [60].

5.3 Quantitative Results

We report the quantitative performance of our proposed model compared with different baselines and state-of-the-art approaches across three benchmarks in Table 1. We can observe that our proposed LMGT with feature VGG16/512 or Res152/512 achieves superior performance over other baselines and state-of-the-art methods in terms of all metrics. In particular, our LMGT clearly exhibits better performance than RNNs based methods, *i.e.*, seq2seq, HRNN, BARNN and HAtt-Rank, by large margins. As we discussed previously, seq2seq and HRNN inherit the limitations from RNNs, having difficulty in capturing complex relationships and long-term dependency between different image regions and image streams. In contrast, our model not only inherits the benefits from the Transformer, but also extracts enriched feature embeddings by exploiting semantic relevance and inter-sentence coherence, thus bringing significant improvements. Notably, our proposed model even outperforms the models optimized with carefully designed reinforcement learning rewards, *i.e.*, AREL [68] and HSRL [23]. The above observations apparently confirms the validity of augmenting latent memory unit into the decoding process, because the highly summarized latent information can be reasonably modeled and recorded. Meanwhile, compared with the GCN based relation extraction approach (*i.e.*, SGVST [67]) and knowledge graph based methods (*i.e.*, KLST [70] and KLEV [19]), our proposed LMGT also achieves a great improvement. This can be attributed to the introduced graph encoding module which captures implicit semantic relationships and attentively aggregates the features from the most important regions based on the scene graph. In addition, we also evaluate several variants of our proposed model by removing either of the two proposed components or even both of them, the results of which can also be found in Table 1. Specifically, when removing both modules, the model is equivalent to the vanilla Transformer. From the results, we can conclude that by adding the proposed two

[1]In the experiments, the parameter settings of the above-mentioned methods are adopted from the corresponding papers. Since not all the papers report results on the validation sets, we only report the performance on the test sets of all the datasets.

Table 1: Performance comparison of our method with the state-of-the-art approaches on the VIST [24], Disney [43], and NYC [43] datasets, w.r.t. BLEU (B), ROUGE-L (R-L), CIDEr-D (C), METEOR (M), and SPICE (S). Here 'B-n' refers to BLEU score using up to n-grams, 'feature(IMG/TXT)' shows the captured image feature and the adopted dimension of word embedding. "†" and "‡" denote the methods utilized scene graph and reinforcement learning, respectively. 'GEM' and 'LMU' indicate our Graph Encoding Module and Latent Memory Unit, respectively. The best performance is highlighted in bold.

Methods	Feature (IMG/TXT)	VIST [24]								Disney [43]	NYC [43]
		B-1	B-2	B-3	B-4	R-L	C	M	S	M	M
CNN-RNN [63]	VGG16/512	38.3	18.2	8.7	4.2	-	8.5	10.5	12.5	8.0	7.0
HRNN [28]	VGG16/512	34.9	16.0	7.7	3.7	-	6.5	10.0	10.2	7.7	6.1
seq2seq [24]	VGG16/512	36.5	16.5	7.5	3.5	-	6.8	10.3	9.9	7.6	7.4
BARNN [39]	VGG16/512	-	-	-	-	-	-	33.3	-	-	-
HAtt-Rank [72]	Res101/512	-	-	21.0	-	29.5	7.5	34.1	-	-	-
HPSR [64]	Res101/512	61.9	37.8	21.5	12.2	31.2	8.0	34.4	-	-	-
AREL‡ [68]	Res152/512	63.8	39.1	23.2	14.1	29.5	9.4	35.0	-	-	-
SRT [65]	VGG16/512	43.4	21.4	10.4	5.2	-	11.4	12.3	-	9.9	8.4
HSRL‡ [23]	Res152/512	-	-	-	12.3	30.8	10.7	35.2	-	-	-
VSCMR [33]	Res152/512	63.8	-	-	14.3	30.2	8.7	35.0	-	-	-
KLST† [70]	Res152/512	66.4	39.2	23.1	12.8	29.9	12.1	35.2	-	-	-
KLEV† [19]	Res101/512	45.1	-	-	5.6	24.1	9.6	29.6	-	-	-
SGVST† [67]	VGG16/512	65.1	40.1	23.8	14.7	29.9	9.8	35.8	-	-	-
INet [26]	Res152/512	64.4	40.1	23.9	14.7	29.7	10.0	35.6	-	-	-
LMGT (ours)	VGG16/512	66.9	40.5	24.2	15.1	31.9	12.5	36.3	22.7	10.5	9.1
LMGT (ours)	Res152/512	**67.5**	**41.6**	**25.0**	**16.7**	**32.8**	**12.9**	**37.2**	**23.1**	**11.6**	**9.5**
LMGT w/o GEM+LMU	VGG16/512	55.6	33.5	19.7	10.9	25.2	8.2	25.6	15.6	8.1	7.5
LMGT w/o GEM	VGG16/512	62.2	38.5	21.3	12.0	25.1	10.0	30.3	19.9	8.5	7.9
LMGT w/o LMU	VGG16/512	63.6	39.0	22.9	13.2	28.2	11.2	34.5	20.6	9.7	8.3

Table 2: Performance comparison of our method and several baseline approaches in terms of human evaluation on the VIST [24] dataset. The best performance is highlighted in bold.

Methods	Feature	Q1	Q2
CNN-RNN [63]	VGG16/512	8.01	6.97
HRNN [28]	VGG16/512	7.72	6.07
seq2seq [24]	VGG16/512	7.61	7.37
LMGT (ours)	VGG16/512	**9.22**	**8.35**
LMGT w/o GEM+LMU	VGG16/512	7.85	6.89
LMGT w/o GEM	VGG16/512	8.30	7.34
LMGT w/o LMU	VGG16/512	8.39	7.66

components, the performance of the vanilla Transformer can be significantly enhanced in terms of all metrics.

5.4 Human Evaluation

Since current automatic metrics may not be sufficient to comprehensively and accurately evaluate the effectiveness of our proposed model, we further conduct a human study to compare our approach with several baseline methods. In particular, we asked 30 volunteers (15 males and 15 females) to score from 1 to 10 (10 means the best performance) according to the following two criteria, after reading 50 randomly selected stories generated by our proposed model and other approaches on VIST [24]. **Question 1 (Q1)**: Whether the generated story is semantically coherent and relevant to the given image stream? **Question 2 (Q2)**: Whether the generated story is expressive and informative in story-style language? The quantitative results are listed in Table 2, where we can clearly observe that LMGT significantly outperforms other methods and achieves the highest scores in terms of both criteria. For example, our LMGT obtains 9.22 and 8.35 w.r.t. Q1 and Q2, which are higher than HRNN by more than 1.5 and 2.2, respectively. The low scores of CNN-RNN, HRNN, seq2seq demonstrate the limitations of RNNs based methods in modeling long-term sentences and complex semantic relationships, as well as the necessity to augmented a latent memory unit to remember the story line history. These subjective results once again demonstrate that our LMGT can effectively boost the quality of the generated story.

5.5 Qualitative Results

We visualize a few examples with the input image streams, the corresponding scene graphs, the ground-truth stories and the stories generated by LMGT as well as the baseline seq2seq [24] in Figure 4. As can be observed, the stories generated by our LMGT are more informative, human-like and closer to the ground-truth compared with the results of seq2seq. Meanwhile, LMGT successfully maintains the globally coherent story line, so that the generated sentence of each image is more consistent with the main topic as well as more coherently associated with the preceding/following sentences. As an example, we observe less details and several incoherence in the stories generated by seq2seq, whereas our LMGT identifies more related entities and the corresponding semantic relations in each sentence, *e.g.*, "drinks in the hand" and "fire on the field" in Figure 4 (a); "words/flowers on the beside" and "foods on the table"

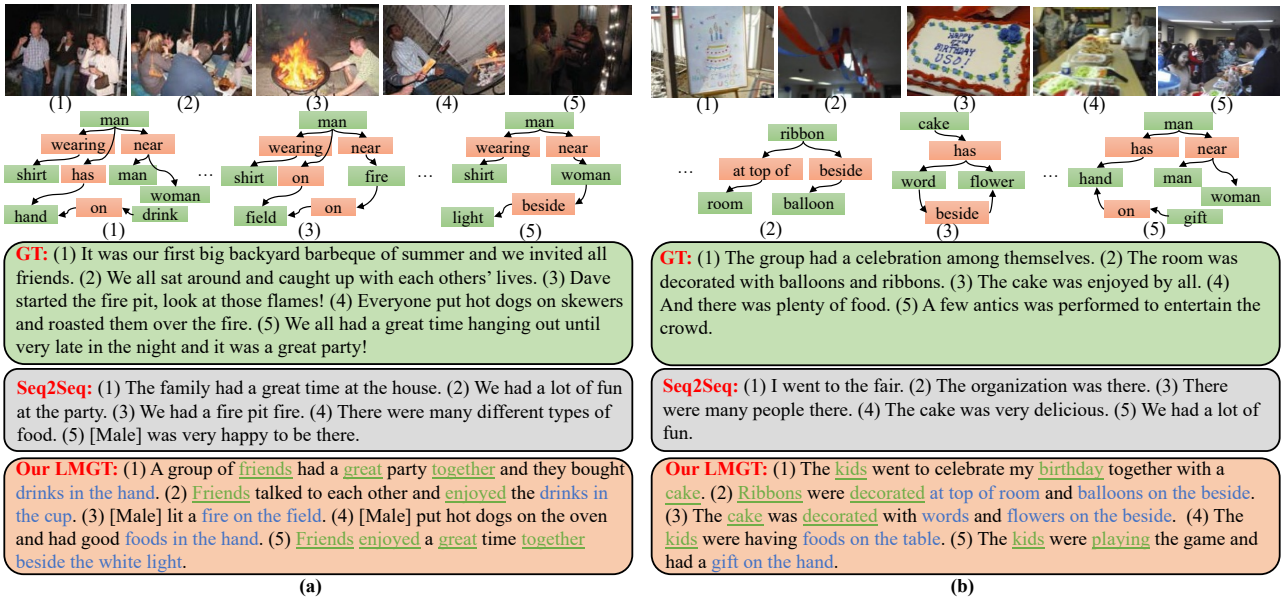

Figure 4: Qualitative examples with input image streams and scene graphs for visual storytelling by our proposed LMGT model compared with the ground truth (GT) and baseline *seq2seq* [24]. We denote the words in green and blue as the learned latent memory and captured semantic relations in the sentences, respectively.

in Figure 4 (b). These findings demonstrate the effectiveness of the introduced graph encoding module in LMGT, which detects the important relationships between image regions with the help of scene graphs to make the story composed of enriched objects and events, eventually leading to a complete and informative story. Moreover, our LMGT has the ability to capture the more important facts as the story line from the image stream to maintain the global coherence throughout the sentences, *e.g.*, "friends" and "together" in Figure 4 (a); "kids", "birthday" and "decorated" in Figure 4 (b), which are in accordance with their appearances and semantic symbols in image streams so that the story appears more consistent and smooth in the topic. We also notice that the seq2seq often misses the previous context when generating the subsequent sentences, while our generated stories are more inter-sentence coherent. This should be attributed to our proposed latent memory unit, which can propagate previous memory states to make the generated sentence more coherent with the previous ones.

5.6 Ablation Study

Effect of the Graph Encoding Module (GEM). We compare our full LMGT model to LMGT without GEM, and report the results in Tables 1 and 2. Note that 'LMGT w/o GEM' means the model replaces the graph attention layer by a self-attention layer. We can find from both tables that removing GEM brings an expected decrease in performance, revealing the proposed GEM is beneficial for injecting rich semantic relational knowledge into feature embeddings, and thus significantly boosts the performance.

Impact of the Latent Memory Unit (LMU). As shown in Tables 1 and 2, the objective and subjective performance of LMGT degrades a lot when removing LMU, suggesting LMU enhances the inter-sentence coherence for story generation, and the topic consistency can also be well preserved. Furthermore, as shown in Table 3, we evaluate different configurations for LMU. We can see that a default setting with sentence-level recurrence, two hidden layers, memory state length with $T_m = 1$, and 100 latent nodes achieves a good trade-off between effectiveness and efficiency.

Table 3: Ablation study of the proposed latent memory unit on VIST [24] with feature VGG16/512. "Re", "Len", "Layer", and "Node" refer to whether sentence-level recurrence is utilized, the length of the memory state, the number of hidden layers used, and the number of latent nodes, respectively.

Model	Re	Len	Layer	Node	B-4	M	C
LMGT	×	-	2	100	14.0	30.7	10.1
LMGT	✓	1	1	100	14.3	32.6	10.9
LMGT (default)	✓	1	2	100	15.1	36.3	12.5
LMGT	✓	1	5	100	15.3	36.5	12.6
LMGT	✓	2	2	100	14.9	35.8	12.0
LMGT	✓	5	2	100	14.6	35.3	11.5
LMGT	✓	1	2	50	14.2	33.7	11.2
LMGT	✓	1	2	150	14.5	35.9	12.1

6 CONCLUSION

In this paper, we proposed a novel graph Transformer, *i.e.*, LMGT, for visual storytelling. By virtue of the designed graph encoding module, important interactions and semantic relationships among visual objects can be encoded based on the parsed scene graph. Furthermore, the latent memory unit can endow the model the ability to record the sequential history and contextual relevance among sentences, thus maintaining a consistent story line. Extensive experimental results demonstrate that our proposed model outperforms the state-of-the-art methods as our generated stories are coherent, informative and more like human-written.

REFERENCES

[1] Vishal Anand, Raksha Ramesh, Ziyin Wang, Yijing Feng, Jiana Feng, Wenfeng Lyu, Tianle Zhu, Serena Yuan, and Ching-Yung Lin. 2020. Story Semantic Relationships from Multimodal Cognitions. In *MM*. ACM.

[2] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *ECCV*. Springer.

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[4] Elaheh Barati and Xuewen Chen. 2019. Critic-based Attention Network for Event-based Video Captioning. In *MM*. ACM.

[5] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. 2015. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325* (2015).

[6] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019. Uniter: Learning universal image-text representations. *arXiv preprint arXiv:1909.11740* (2019).

[7] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* (2014).

[8] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[9] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. 2020. Meshed-Memory Transformer for Image Captioning. In *CVPR*. IEEE/CVF.

[10] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. In *ACL*.

[11] Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*. 376–380.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[13] Lianli Gao, Zhao Guo, Hanwang Zhang, Xing Xu, and Heng Tao Shen. 2017. Video captioning with attention-based LSTM and semantic consistency. *IEEE Transactions on Multimedia* 19, 9 (2017), 2045–2055.

[14] Longteng Guo, Jing Liu, Jinhui Tang, Jiangwei Li, Wei Luo, and Hanqing Lu. 2019. Aligning linguistic words and visual semantic units for image captioning. In *MM*. ACM.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. IEEE.

[16] Simao Herdade, Armin Kappeler, Kofi Boakye, and Joao Soares. 2019. Image captioning: Transforming objects into words. In *NeurIPS*.

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

[18] Xudong Hong, Rakshith Shetty, Asad Sayeed, Khushboo Mehra, Vera Demberg, and Bernt Schiele. 2020. Diverse and Relevant Visual Storytelling with Scene Graph Embeddings. In *CNLL*.

[19] Chao-Chun Hsu, Zi-Yuan Chen, Chi-Yang Hsu, Chih-Chia Li, Tzu-Yuan Lin, Ting-Hao'Kenneth' Huang, and Lun-Wei Ku. 2020. Knowledge-Enriched Visual Storytelling. In *AAAI*.

[20] Junjie Hu, Yu Cheng, Zhe Gan, Jingjing Liu, Jianfeng Gao, and Graham Neubig. 2020. What Makes A Good Story? Designing Composite Rewards for Visual Storytelling.. In *AAAI*.

[21] Yaosi Hu, Zhenzhong Chen, Zheng-Jun Zha, and Feng Wu. 2019. Hierarchical global-local temporal modeling for video captioning. In *MM*. ACM.

[22] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. 2019. Attention on attention for image captioning. In *ICCV*. IEEE.

[23] Qiuyuan Huang, Zhe Gan, Asli Celikyilmaz, Dapeng Wu, Jianfeng Wang, and Xiaodong He. 2019. Hierarchically structured reinforcement learning for topically coherent visual story generation. In *AAAI*.

[24] Ting-Hao Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, et al. 2016. Visual storytelling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1233–1239.

[25] Jiayi Ji, Xiaoshuai Sun, Yiyi Zhou, Rongrong Ji, Fuhai Chen, Jianzhuang Liu, and Qi Tian. 2020. Attacking Image Captioning Towards Accuracy-Preserving Target Words Removal. In *MM*. ACM.

[26] Yunjae Jung, Dahun Kim, Sanghyun Woo, Kyungsu Kim, Sungjin Kim, and In So Kweon. 2020. Hide-and-Tell: Learning to Bridge Photo Streams for Visual Storytelling. In *AAAI*.

[27] Taehyeong Kim, Min-Oh Heo, Seonil Son, Kyoung-Wha Park, and Byoung-Tak Zhang. 2018. Glac net: Glocal attention cascading networks for multi-image cued story generation. In *ACL*.

[28] Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei. 2017. A hierarchical approach for generating descriptive image paragraphs. In *CVPR*. IEEE.

[29] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *ICCV*. IEEE.

[30] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. 2016. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. https://arxiv.org/abs/1602.07332

[31] Jie Lei, Liwei Wang, Yelong Shen, Dong Yu, Tamara L Berg, and Mohit Bansal. 2020. Mart: Memory-augmented recurrent transformer for coherent video paragraph captioning. In *ACL*.

[32] Guang Li, Linchao Zhu, Ping Liu, and Yi Yang. 2019. Entangled transformer for image captioning. In *ICCV*. IEEE.

[33] Jiacheng Li, Haizhou Shi, Siliang Tang, Fei Wu, and Yueting Zhuang. 2019. Informative Visual Storytelling with Cross-modal Rules. In *MM*. ACM.

[34] Jiacheng Li, Siliang Tang, Juncheng Li, Jun Xiao, Fei Wu, Shiliang Pu, and Yueting Zhuang. 2020. Topic Adaptation and Prototype Encoding for Few-Shot Visual Storytelling. In *MM*. ACM.

[35] Yikang Li, Wanli Ouyang, Bolei Zhou, Jianping Shi, Chao Zhang, and Xiaogang Wang. 2018. Factorizable net: an efficient subgraph-based framework for scene graph generation. In *ECCV*. Springer.

[36] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[37] Jen-Chun Lin, Wen-Li Wei, Yen-Yu Lin, Tyng-Luh Liu, and Hong-Yuan Mark Liao. 2020. Learning From Music to Visual Storytelling of Shots: A Deep Interactive Learning Mechanism. In *MM*. ACM.

[38] Yang Liu. 2019. Fine-tune BERT for extractive summarization. *arXiv preprint arXiv:1903.10318* (2019).

[39] Yu Liu, Jianlong Fu, Tao Mei, and Chang Wen Chen. 2017. Let your photos talk: Generating narrative paragraph for photo stream via bidirectional attention recurrent neural networks. In *AAAI*.

[40] Bruce T Lowerre. 1976. *The HARPY speech recognition system*. Technical Report. CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE.

[41] Yingwei Pan, Ting Yao, Houqiang Li, and Tao Mei. 2017. Video captioning with transferred semantic attributes. In *CVPR*. IEEE.

[42] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*.

[43] Cesc C Park and Gunhee Kim. 2015. Expressing an image stream with a sequence of natural sentences. In *NIPS*.

[44] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

[45] Mengshi Qi, Weijian Li, Zhengyuan Yang, Yunhong Wang, and Jiebo Luo. 2019. Attentive relational networks for mapping images to scene graphs. In *CVPR*. IEEE.

[46] Mengshi Qi, Jie Qin, Yi Yang, Yunhong Wang, and Jiebo Luo. 2021. Semantics-Aware Spatial-Temporal Binaries for Cross-Modal Video Retrieval. *IEEE Transactions on Image Processing* 30 (2021), 2989–3004.

[47] Mengshi Qi, Jie Qin, Xiantong Zhen, Di Huang, Yi Yang, and Jiebo Luo. 2020. Few-Shot Ensemble Learning for Video Classification with SlowFast Memory Networks. In *MM*. ACM.

[48] Mengshi Qi, Yunhong Wang, and Annan Li. 2017. Online cross-modal scene retrieval by binary representation and semantic graph. In *MM*. ACM.

[49] Mengshi Qi, Yunhong Wang, Annan Li, and Jiebo Luo. 2018. Sports video captioning by attentive motion representation based hierarchical recurrent neural networks. In *MMSports*. ACM.

[50] Mengshi Qi, Yunhong Wang, Annan Li, and Jiebo Luo. 2019. Sports video captioning via attentive motion representation and group relationship modeling. *IEEE Transactions on Circuits and Systems for Video Technology* 30, 8 (2019), 2617–2633.

[51] Mengshi Qi, Yunhong Wang, Jie Qin, Annan Li, Jiebo Luo, and Luc Van Gool. 2019. stagNet: an attentive semantic RNN for group activity and individual action recognition. *IEEE Transactions on Circuits and Systems for Video Technology* 30, 2 (2019), 549–565.

[52] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*.

[53] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*.

[54] Zhiqiang Shen, Jianguo Li, Zhou Su, Minjun Li, Yurong Chen, Yu-Gang Jiang, and Xiangyang Xue. 2017. Weakly supervised dense video captioning. In *CVPR*. IEEE.

[55] Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *ICLR*.

[56] Yuqing Song, Shizhe Chen, Yida Zhao, and Qin Jin. 2019. Unpaired cross-lingual image caption generation with self-supervised rewards. In *MM*. ACM.

[57] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. 2019. Videobert: A joint model for video and language representation learning. In *ICCV*. IEEE.

[58] Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*.

[59] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all

you need. In *NIPS*.
[60] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *CVPR*. IEEE.
[61] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *ICLR*.
[62] Paula Viana, Pedro Carvalho, Maria Teresa Andrade, Pieter P Jonker, Vasileios Papanikolaou, Inês N Teixeira, Luis Vilaça, José P Pinto, and Tiago Costa. 2020. Semantic Storytelling Automation: A Context-Aware and Metadata-Driven Approach. In *MM*. ACM.
[63] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *CVPR*. IEEE.
[64] Bairui Wang, Lin Ma, Wei Zhang, Wenhao Jiang, and Feng Zhang. 2019. Hierarchical photo-scene encoder for album storytelling. In *AAAI*.
[65] Jing Wang, Jianlong Fu, Jinhui Tang, Zechao Li, and Tao Mei. 2018. Show, reward and tell: Automatic generation of narrative paragraph from photo stream by adversarial training. In *AAAI*.
[66] Jing Wang, Jinhui Tang, and Jiebo Luo. 2020. Multimodal Attention with Image Text Spatial Relationship for OCR-Based Image Captioning. In *MM*. ACM.
[67] Ruize Wang, Zhongyu Wei, Piji Li, Qi Zhang, and Xuanjing Huang. 2020. Storytelling from an Image Stream Using Scene Graphs. In *AAAI*.
[68] Xin Wang, Wenhu Chen, Yuan-Fang Wang, and William Yang Wang. 2018. No metrics are perfect: Adversarial reward learning for visual storytelling. In *ACL*.
[69] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. 2017. Scene graph generation by iterative message passing. In *CVPR*. IEEE.
[70] Pengcheng Yang, Fuli Luo, Peng Chen, Lei Li, Zhiyi Yin, Xiaodong He, and Xu Sun. 2019. Knowledgeable Storyteller: A Commonsense-Driven Generative Model for Visual Storytelling.. In *IJCAI*.
[71] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*.
[72] Licheng Yu, Mohit Bansal, and Tamara L Berg. 2017. Hierarchically-attentive rnn for album summarization and storytelling. In *EMNLP*.
[73] Yitian Yuan, Lin Ma, Jingwen Wang, and Wenwu Zhu. 2020. Controllable Video Captioning with an Exemplar Sentence. In *MM*. ACM.
[74] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. 2018. Neural motifs: Scene graph parsing with global context. In *CVPR*. IEEE.
[75] Beichen Zhang, Liang Li, Li Su, Shuhui Wang, Jincan Deng, Zheng-Jun Zha, and Qingming Huang. 2020. Structural Semantic Adversarial Active Learning for Image Captioning. In *MM*. ACM.
[76] Shengyu Zhang, Ziqi Tan, Jin Yu, Zhou Zhao, Kun Kuang, Jie Liu, Jingren Zhou, Hongxia Yang, and Fei Wu. 2020. Poet: Product-oriented Video Captioner for E-commerce. In *MM*. ACM.
[77] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. 2018. End-to-end dense video captioning with masked transformer. In *CVPR*. IEEE.
[78] Yongqing Zhu and Shuqiang Jiang. 2019. Attention-based densely connected LSTM for video captioning. In *MM*. ACM.