Information Sciences 623 (2023) 812–831

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Label-attention transformer with geometrically coherent objects for image captioning

Shikha Dubey [a], Farrukh Olimov [b], Muhammad Aasim Rafique [a], Joonmo Kim [c], Moongu Jeon [a,*]

[a] Gwangju Institute of Science and Technology (GIST), School of Electrical Engineering and Computer Science, Gwangju, South Korea
[b] Threat Intelligence Team, Monitorapp, Seoul, South Korea
[c] Dankook University, Department of Computer Engineering, Jukjeon, South Korea

ARTICLE INFO

Article history:
Received 17 April 2022
Received in revised form 4 December 2022
Accepted 7 December 2022
Available online 13 December 2022

Keywords:
Image captioning
Transformers
Self-attention
Label-attention
Geometrically coherent proposals
Memory-augmented-attention

ABSTRACT

Encoder-decoder-based image captioning techniques are generally utilized to describe meaningful information present in an image. In this work, we investigate two unexplored ideas for image captioning using the transformer: 1) an object-focused label attention module (LAM), and 2) a geometrically coherent proposal (GCP) module that focuses on the scale and position of objects to benefit the transformer model by attaining better image perception. These modules demonstrate the enforcement of objects' relevance in the surrounding environment. Furthermore, they explore the effectiveness of learning an explicit association between vision and language constructs. LAM and GCP tolerate the variation in objects' class and its association with labels in multi-label classification. The proposed framework, label-attention transformer with geometrically coherent objects (LATGeO), acquires proposals of geometrically coherent objects using a deep neural network (DNN) and generates captions by investigating their relationships using LAM. The module LAM associates the extracted objects classes to the available dictionary using self-attention layers. Object coherence is acquired in the GCP module using the localized ratio of the proposals' geometrical features. In this study, experimentation results are performed on MSCOCO dataset. The evaluation of LATGeO on MSCOCO advocates that objects' relevance in surroundings and their visual features binding with geometrically localized ratios and associated labels generate improved and meaningful captions.

1. Introduction

Image captioning is one of the prevalent challenges in scene understanding and commonly used techniques leverage solutions in computer vision (CV) and natural language processing (NLP). It manifests the inherent challenges of spatial, temporal, and sequential data modalities. Another obtrusive challenge in image captioning is the transformation of a spatial modality to a sequential modality that establishes transcriptions of a scene. A widely adopted solution presented in literature is to use an encoder-decoder architecture where an encoder extracts features, and a decoder transcribes the captions. For instance, in the deep learning frameworks, convolution neural networks (CNNs) and recurrent neural networks (RNNs) are adopted for encoding and decoding, respectively. CNNs are employed for extracting spatial visual features from the given

* Corresponding author.
E-mail addresses: shikha.d@gm.gist.ac.kr (S. Dubey), olimov.farrukh@gm.gist.ac.kr (F. Olimov), aasimrafique@gist.ac.kr (M.A. Rafique), q888@dankook.ac.kr (J. Kim), mgjeon@gist.ac.kr (M. Jeon).

https://doi.org/10.1016/j.ins.2022.12.018
0020-0255/

image and RNNs are frequently used with attention layers to preserve and capture the distant association in sequential data [43,48]. An inherent bound in a language is the length of sentences which does not allow the use of an attention layer with feed-forward networks and thus requires sequential modeling. Despite of state-of-the-art (SOTA) algorithms showing promising results, they do not learn relationships among objects and surroundings and are intolerant to variation in class to label associations in multi-label classification tasks. In this study, a "class" represents the detected object's class, and the "label" represents the word present in the dictionary.

This study proposes a novel architecture for image captioning that uses concrete features of objects and their surroundings, the localized ratio of objects by utilizing their geometrical properties, and employs a label-attention module (LAM) for objects compliance following the language rules. In particular, we assimilate high-level cognition by generating proposals from images, utilizing the available geometrical formations of proposals, and learning their relationship with surroundings and labels. The proposals are common identifiable vision interpretations that are detected objects in images. The learning of the proposed system is inspired by the recent advancements in encoder-decoder neural networks with self-attention layers called transformers [32]. Variants of transformers are in active use in image captioning and are discussed in detail in Section 2. Briefly, this study proposes a novel architecture that uses a label-attention transformer with geometrically coherent objects (LATGeO).

LATGeO uses geometrically coherent object proposals and label-attention to learn the relationship among objects. Here, the object detector is used for extracting object proposals. The main idea of extracting objects from images is to provide the proposed architecture with fine-grained information about the content of the image along with the entire image, while the geometrical properties identify the association among objects. The geometrically coherent properties are encapsulated for better learning of the relative positions and size of objects; for example, to learn the relative size of objects, van→car, from "a young boy standing in front of a van" to "a young boy standing in front of a police car" and to learn the relative positions of objects, parked next→leaning against, and background "flooded street", from "a bike parked next to a metal rack in a" to "a bicycle leaning against fence in a flooded street" (see Fig. 1). A similar study is proposed in [14], but our proposed architecture considers the ratio of objects' dimensions. In a transformer composition, multiple encoders are stacked, and the output of an encoder is passed to the next encoder in the stack. Usually, the output of the last encoder in a stack is passed to the first decoder in similarly stacked decoders. However, a recent study [5] discusses a composition of an encoder stack fully connected to a decoder stack. The fully connected composition inspires LATGeO because it explores multi-level geometrical and visual representations of objects in an image. Recently, ConVit [7] has shown promising results by improving the performance of the model in image classification tasks which is an explicit result of utilizing inductive bias in a transformer-based model. Similarly, this study also explores both hard- and soft-inductive biases' impact on the image captioning task.

Contributions of this study are summed up as follows:

1. An extrinsic and novel technique, label-attention module (LAM), emphasizes grammatical constructs, particularly nouns/pronouns, resulting in the improvement of image captioning. LAM bridges the gap between visual and language domains. It benefits the soft-inductive bias of a transformer to attend the visual information in images using language constructs.
2. Geometrically coherent proposals (GCP) module, focuses on the structure of detected objects utilizing localized ratios of their geometrical features.
3. A LATGeO framework that integrates LAM and GCP in a transformer for the image captioning task.[1] LATGeO relates object proposal embeddings with less significant surroundings to discover objects and background coherence, which supports the GCP module.
4. Extensive experimentation that includes evaluation on the MSCOCO dataset, a comprehensive ablation study, and visualization of the impact of LAM and GCP on LATGeO.

This paper is organized as follows: Section 2 details the recent developments in image captioning research, and Section 3 explains the proposed architecture LATGeO. Section 4 details extensive experiments performed to support the proposed methodology with ablation studies, and Section 5 briefs the conclusion of this work.

2. Related works

Machine vision algorithms evolved over the past two decades and extend to resolve challenging problems in scene understanding like image captioning problems. Image captioning fuses the progress in cutting-edge approaches from CV and NLP. The progress in image captioning techniques is categorized into four sections in our review work: template-based, deep neural network-based, attention-based, and transformer-based.

2.1. Template-based image captioning

Conventional algorithms are based on two approaches: 1) in earlier strategies like in [21], image retrieval techniques use a few collections of keywords as templates from image-caption data pairs and generate captions for the retrieved images

[1] The code is publicly available on https://github.com/shikha-gist/Image-Captioning/.

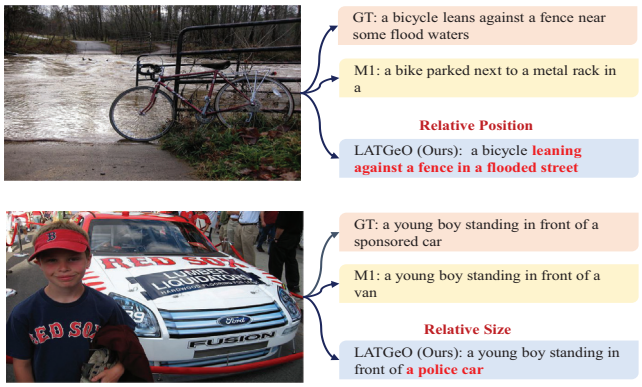

Fig. 1. Examples of images with captions show the utility of background and geometrically localized ratio information. The text in red shows the significant improvements in the captions (LATGeO captions) compared to the baseline transformer model (M1). GT: ground truth. Ml: Model without relative localization, size, and background information. Boxes filled in yellow and blue contain generated captions.

utilizing annotated captions, and 2) later approaches like in [20] practice bottom-up algorithms that infer sentence parts like nouns, verbs, and adjectives from images and apply pre-defined caption templates to generate the image descriptions. Recently, Lu et al. [25] have adopted specific image regions explicitly bound to slot locations of template captions. These template slots are packed with visual features of the extracted objects. Template-based captioning methods require human-crafted templates, which limits the generalization of these techniques. Conventional image captioning methods treat this task as a combination of CV and NLP, though process them in independent pipelines. The pipeline often is not scaleable and becomes intractable with a huge amount of data available today. Whereas, deep neural networks (DNNs) glue the pipeline in an atomic end-to-end system and niche big data for its learning.

2.2. Deep neural network-based image captioning

With the advent of artificial intelligence (AI), DNNs have influenced all branches of AI with no exception to image captioning [33,44,19,50,6]. Early works of image captioning employing CNN and RNN or long short-term memory (LSTM) in an encoder-decoder composition created a remarkable impact. In [33], image captioning is treated as a conventional machine translation problem by transforming an image into N-dimensional vector representation and feeding it as an input to the RNN decoder. Vinyals et al. [33] apply a deep CNN network to encode vision features from the entire image and utilize RNN to generate captions by maximizing the likelihood of a target caption. A constraint of this procedure is that it is challenging to represent all objects and their attributes in an image as a single feature vector. Consequently, scene graphs and object detection techniques [41,44,50] are incorporated for image captioning to address this constraint. Likewise, the attributes information is additionally presented to the RNN input to learn the relationship among objects [44,50]. Furthermore, the authors of [37] propose a recall network, an encoder-decoder architecture. This network is constructed with CNN and two LSTMs. CNN is for extracting visual features, and LSTMs are for the transmission of sequential information and integrating visual features. However, DNNs are still inefficient to associate the visual features with the language model, due to the data modalities of domain's constructs. Our model, LATGeO explores these relationships between the modalities of different domains by utilizing the proposed attention modules in the transformer network.

Model optimization plays a vital role in training DNN. Recently, [47,28] have suggested improvements in the optimization technique for training. [47] attempts to boost the training using actor-critic reinforcement learning (RL) to optimize non-differentiable quality metrics. Our proposed framework also incorporates a similar boosting training technique to improve the model's final performance. Moreover, the policy gradient technique [6] for RL also exhibited improved performance. Furthermore, Ren et al. [28] introduce a policy network (comprising of CNN and RNN), and a value network (consisting of CNN, RNN, and multi-layer perceptron (MLP)) to generate captions.

2.3. Attention-based image captioning

The attention layers added to the DNN particularly to obtain the superior results of sequential learning tasks and revive recent image captioning developments [29,24,34,42,4]. The early development of attention-based techniques introduces a spatial attention model using image feature maps to generate image captions, which is further extended to a channel-wise attention module in [4]. Next, Xu et al. [38] propose an attention-gated LSTM model where the output gate incorporates visual attention forwarded to the cell state of LSTM. Lu et al. [24] introduce an adaptive attention model on visual sentinel by deciding which region of an image should be attended to for extracting meaningful image features to generate sequential caption words. To learn the multi-level dependencies in objects, [11] proposes a multi-stage image captioning model consisting of one convolutional encoder and multiple stacked attention-based decoders to generate fine captions. In our pro-

posed technique, multi-level dependencies are learned by a single transformer. A series of recent works address the relational reasoning among regions using various compositions of activation layers in RNN and CNN [17,34,2,48,36,10]. Huang et al. [16] propose the refinement network to correlate semantic information with attributes to improve image captions. Whereas [45] proposes a transformation matrix to map visual features to context features. Our proposed framework learns such a relationship using geometrical information of the extracted objects without additional attributes similar to [14].

To utilize the individual object features for more reliable context learning, object detection and attention module are combined in [1,22]. These studies encode the visual features of extracted objects and transfer them to the attention recurrent network for generating image captions. Likewise [1] has additional information on object attributes to refine the predicted captions. [22] utilizes a convolutional graphical model to represent structured information in the form of detected objects and their relationships. This information is passed through a hierarchical attention-based module for caption generation at each time step. [39] studies Hierarchical-Attention by using a generative adversarial network (GAN) based model. The authors of [8] also propose an attention module on LSTM generative model. Recent algorithms [43,48] propose the visual relationship attention on extracted objects region and investigate the visual relationship among them for generating captions. However, the algorithm [43] employs a Graph Convolutional Networks on detected objects and a LSTM network to generate captions based on the attention module. Several studies [13,40,30,35,9] have proposed attention modules for visual semantics to achieve better association among visual and contextual/textual information. CaptionNet [40] initializes memory to achieving better visual semantics in a LSTM network, similarly, Sammani et al. [30] propose selective memory attention in an LSTM-based auto-encoder-decoder network. Furthermore, image-level semantics are fused with textual context using a context-gating technique proposed by [35]. For model optimization, Rennie et al. [29] propose self-critical learning, which optimizes models based on evaluation metrics such as CIDEr-D, resulting in significant performance improvements over the methods that use cross-entropy objectives only. The authors in [46] propose a Context-Aware Visual Policy network (CAVP) for RL-based algorithms to attend complex visual compositions every time step.

These attention mechanisms have exhibited promising results on the image captioning task, however, lack in learning the relationship among background and objects. Beyond that, these methods lack in learning long-range dependencies among objects and language constructs.

2.4. Transformer-based image captioning

Because transformers are SOTA in NLP [32], image captioning adopted transformers for caption transcription. [49] recently explored the transformer by learning an attention module using contextualized embedding for individual regions and examined visual relationships with the help of spatial object regions. [15] proposed additional attention on top of the multi-head attention of the transformer for image captioning. A recent study [14] transfers encoded visual features of the extracted objects through the transformer architecture to learn the relative appearance among objects. Next, the extracted encoded information of objects' bounding boxes along with the visual information is passed through a multi-head self-attention mechanism to learn the object dependencies on each other. In comparison, [5] learns such dependencies among the regions of interest (objects spatial regions) with the help of memory-augmented attention. Moreover, the authors in [18] have exploited an object detector, Faster R-CNN, to extract multi-view features of objects before passing them through the encoder of the transformer. Faster R-CNNs are employed at the multi-level of the network to extract deep visual features. Although our proposed method, LATGeO, is motivated by these recent works of self-attention mechanisms [14,5], it is diverse in many aspects. LATGeO learns the relationship among objects as well as a relationship with a background. Furthermore, LATGeO proposes to combine different geometrically coherent features using localized ratios which is distinctive from the method proposed in [14]. Beyond meshed transformer [5], LATGeO utilizes the LAM in the decoder part of the multi-head attention module. In addition, we have experimented LATGeO with several object proposal methods like DETR [3], a newly introduced transformer-based object detection method. Our proposed technique learns relationships among objects and their context without requiring any additional information such as attributes and semantics. Details of our proposed approach are presented in the following sections.

3. Method

This work proposes a label-attention transformer that uses geometrically coherent objects for image captioning. The complete framework is depicted in Fig. 2. LATGeO generates meaningful transcriptions by exploring tangible objects' multi-level fine-grained features representation of an image. In the LATGeO framework, first, we extract objects from an image called object proposals inside the "Object Detector" block. Second, these proposals are assigned to labels from the known classes, "Objects' Classes", after passing these classes through the LAM, "Label-Attention Module" block. Next, an effective geometrical relationship of the object proposals is computed inside the "Geometrically Coherent Proposals" module. Finally, a multi-level representation of the object proposals and the less significant features are passed as input to the final learnable block of LATGeO, a fully connected encoder-decoder transformer. In the end, the decoder generates an image caption. The details of each component of the framework are explained in subsequent subsections.

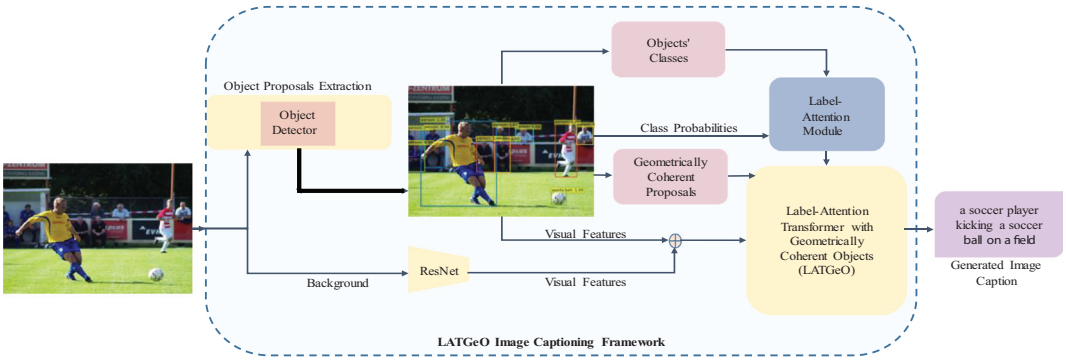

Fig. 2. The proposed architecture for image captioning, LATGeO. The "Object Detector" block extracts the object proposals with their classes, probabilities, features, and bounding boxes. "Geometrically Coherent Proposals" block or GCP block generates the objects' coherence utilizing objects geometrical properties. "Label-Attention Module" (LAM) associates objects classes with the dictionary labels. At last, all this information along with background features are passed through the "LATGeO" transformer block to generate image captions.

3.1. Object proposals and background

Proposals are essential components of the strategy presented in this study, therefore, we deliberate numerous choices of a proposal with distinctive features. An essential proposal is translated into invariant and covariant features, which helps to define a semantic relationship among proposals. It is empirically decided to use object detection in images to generate meaningful proposals. The detected objects, termed as object proposals, generate 2048-dimensional visual features. Furthermore, each object proposals' class probabilities, classes, visual features, and geometrical features are exploited for their use in image captioning. The classes and their probabilities generated from object proposals are fed to the LAM, and visual and geometrical features are fed to the transformer module in LATGeO. This methodology has generated proposals data by utilizing the SOTA DNN object detectors, i.e., Faster R-CNN [27], and DETR [3].

The first DNN network, Faster R-CNN is a two-stage object detection model; a base CNN model (ResNet in our study) for features extraction in the first stage, and a region proposal network (RPN) to generate the bounding boxes using the intersection over union (IoU) method in the second stage. The second DNN network for proposal generation employed in this study is DETR, which uses a transformer architecture for multiple objects detection.

Faster R-CNN is specifically fine-tuned for image captioning, making it suitable for further use in our detailed experiments. Performance comparison of our proposed architecture using both object detectors, DETR and Faster R-CNN, is provided in the discussion Section 4.4.1 to assert the choice later. In addition to the proposals, visibly less-significant features are fed to LATGeO, which renders the relationships between object proposals and background. A pre-trained ResNet [12] model is used to extract the background features then, these features undergo LATGeO block.

3.2. Geometrically coherent proposals

Image captions are hard to generate with proposals' features and labels only; therefore, this study adopts a natural coherence among proposals. In the GCP block, the coherent relationship is computed using the geometrical properties of the bounding boxes of the object proposals. Any pair of object proposals (a, b) can be fed to the GCP block, where $a = (x_a, y_a, w_a, h_a)$, and $b = (x_b, y_b, w_b, h_b)$ are coordinates of the bounding boxes. In this block, the relative geometrical coherence $\xi(a, b)$ of these proposals is calculated as follows:

$$\xi(a,b) = \left(log\left(\frac{x_a}{x_b}\right), log\left(\frac{y_a}{y_b}\right), log\left(\frac{w_a}{w_b}\right), log\left(\frac{h_a}{h_b}\right)\right), \tag{1}$$

where $(x_a, y_a), (x_b, y_b)$ are center coordinates, (w_a, h_a) and (w_b, h_b) are widths and heights of objects a and b, respectively. Semantically, Eq. (1) gives a simple ratio of the scale dimensions (Ratio-Comparison) of two object proposals, different from (L1-Comparison) of object proposals used in [14]. Moreover, a coherent relationship of each object proposal with the background is calculated using Eq. (1), where the input image dimension is considered as the bounding box of the background. For further processing, the weights η for the attention mechanism utilizing these geometrical features are calculated as below:

$$\eta_G^{ab} = ReLU([Emb(\xi)w_G, M_G]), \tag{2}$$

where $Emb(\cdot)$ represents the embedding of objects' relative geometrical features. The embedding function, $Emb(\cdot)$ projects the proposals' relative geometrical coherence vector $\xi(a, b)$ into a high-dimensional embedding using a sinusoidal function, similar to the study [14]. Weight w_G is a learned d_{model}-dimensional vector that projects embedding vectors down to a scalar, where G represents geometrical components. The term M_G is the trainable memory slots of size M, concatenated with the

projected scalars. Then, a non-linear activation function $ReLU(\cdot)$ is applied. The weights η_G^{ab} computed in Eq. (2) pass through the encoder layer (see Eq. (22) below) and the geometrical features are propagated through the LATGeO block.

3.3. Label-attention module

It is challenging to transcribe the features extracted from images into meaningful captions because the dictionary has more transcriptions (words) than their equivalent detected proposals. Therefore, this study considers meaningful labels from language models to reduce the combinations of transcription. This is achieved by passing the extracted features through a LAM as illustrated in Fig. 3(a). LAM learns the association between labels and detected proposals, and attends meaningful related classes of proposals. LAM generates embeddings of label-attention, which are fed to the LATGeO module. The detailed working of LAM is as follows:

First, LAM associates all object proposals' classes with the available dictionary D and generates label embeddings using the following equation:

$$L_O = \left[emb\left(Idx\left(w^j : C_O^i == D(w^j)\right)\right)\right],\ i = 1, 2, \ldots, o_n + 1, \tag{3}$$

where C_O^i represents a class of the i^{th} object proposal, $D(w^j)$ represents the j^{th} word present in dictionary D, and $Idx(\cdot)$ maps an object proposal's class to the corresponding word in the dictionary and returns the corresponding index of the j^{th} word as a label. The function $emb(\cdot)$ converts the label of the i^{th} object proposal into a learnable high-dimensional embedding of order d_{model}. Further, o_n is the number of object proposals, and plus one denotes the background. In Eq. (3), $L_O \in \mathscr{R}^{(o_n+1)\times d_{model}}$ represents the label embeddings, where i^{th} row is a d_{model}-dimensional embedded label of the i^{th} object proposal. Afterward, LAM adjusts the significance of the associated L_O, such that proposals with higher probabilities give preference to their corresponding L_O, and computes weightage matrix W_L. This matrix is computed using element-wise multiplication of L_O with the class probabilities of object proposals as given in the following equations:

$$R_O = \left[\left(Pr\left(C_O^i\right)\right)_{\times d_{model}}\right],\ i = 1, 2, \ldots, o_n + 1, \tag{4}$$

$$W_L = L_O * R_O. \tag{5}$$

In Eq. (4), $Pr\left(C_O^i\right)$ represents the probability of the i^{th} object proposal's class, and $R_O \in \mathscr{R}^{(o_n+1)\times d_{model}}$ defined as ranks of the detected classes. The i^{th} row of R_O is a d_{model}-dimensional vector with d_{model} times repeated value of $Pr\left(C_O^i\right)$. Next, we pass L_O, and W_L to the multi-head attention module, $Multi-Head(\cdot)$ of the vanilla transformer [32], where $Multi-Head(\cdot)$ consists of h numbers of identical attention heads as given below:

$$L_{Att} = \sigma(Multi-Head(Q, K, V)), \tag{6}$$

$$Multi-Head(Q, K, V) = [Head_1, Head_2, ..Head_h]W^0, \tag{7}$$

$$Head_i = Att(Q_i, K_i, V_i), i = 1, 2, \ldots, h. \tag{8}$$

Eq. (6) gives the embedded label-attention L_{Att}, by taking sigmoid, $\sigma(\cdot)$ of $Multi-Head(\cdot)$ attention's output. Eq. (7) concatenates the output of all the attention heads, where $W^0 \in \mathscr{R}^{d_{model}\times d_{model}}$ is the learned projection matrix for multi-head attention. Each head calculates the self-attention $Att(\cdot)$ simultaneously, given in Eq. (8), using different projection matrices W^Q, W^K, and W^V for queries Q, keys K, and values V, respectively. In the case of L_{Att}, matrices Q, K, and V are calculated as follows:

$$Q_i = L_O\ W_i^Q, \tag{9}$$

$$K_i = W_L\ W_i^K, \tag{10}$$

$$V_i = L_O\ W_i^V. \tag{11}$$

In Eqs. (9), (10), (11), matrices Q_i, K_i, and V_i are Q, K, and V for the i^{th} head, respectively. Similarly, $W_i^Q \in \mathscr{R}^{d_{model}\times d_Q}, W_i^K \in \mathscr{R}^{d_{model}\times d_K}$, and $W_i^V \in \mathscr{R}^{d_{model}\times d_V}$ are projection matrices for the i^{th} head, $Head_i$, and $d_Q = d_K = d_V = d_{model}/h$. Self-attention, $Att(\cdot)$, for each head is calculated as follows:

$$Att(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_K}}\right)V, \tag{12}$$

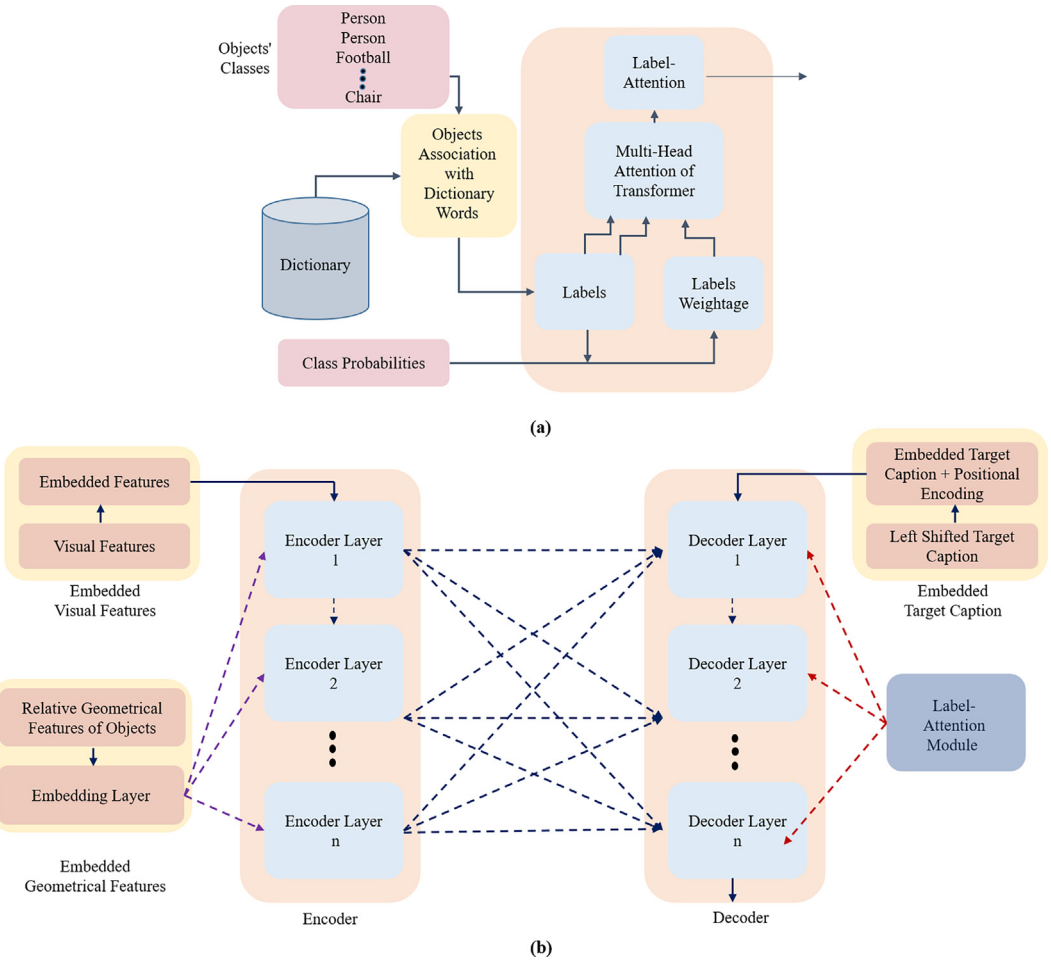

Fig. 3. Configurations of (a) LAM: associates objects' classes with dictionary words, and (b) LATGeO: fully-connects encoder and decoder layers, where the encoder layer encapsulates visual and geometrical features and decoder layer encapsulates LAM and target captions.

where d_K is a scaling factor in $Att(\cdot)$ module. The $softmax(\cdot)$ function assigns higher probability to the (Q, K) pair which are more correlated. Later, L_{Att} is used to enforce the output of each encoder layer $(E_{Out}(n))$ by utilizing element-wise multiplication as follows:

$$\widehat{I} = E_{Out}(n) * L_{Att}, \;\; n = 1, \ldots, L, \tag{13}$$

where L is the number of encoder layers, and $\widehat{I}$ is used as inputs to sigmoid gating (see Eq. (26) below) of the decoder layer in LATGeO, described in Section 3.4.3.

3.4. LATGeO

Detailed configuration of the LATGeO block is presented in Fig. 3(b) and Fig. 4. This block of transformer takes embedded visual features of proposals and background, LAM features, and GCP features as input and generates image-caption as output. The encoder inputs are the embeddings of proposals, background, and GCP and the decoder inputs are the LAM embedding, and encoder output. The details are provided in the following sections.

3.4.1. Embedded visual features

Features of object proposals and background are concatenated together (shown in Fig. 2) and then, these concatenated features X, projected down to a d_{model}-dimensional vector as an embedding using trainable matrix W (see Eq. (14) and Eq. (15) below). This process generates embeddings of the visual features (shown in Fig. 3(b)).

$$\boldsymbol{X} = \left[\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_{o_n}, \boldsymbol{x}_{o_{n+1}}\right], \boldsymbol{x_i} \in \mathscr{R}^{2048 \times 1}, \tag{14}$$

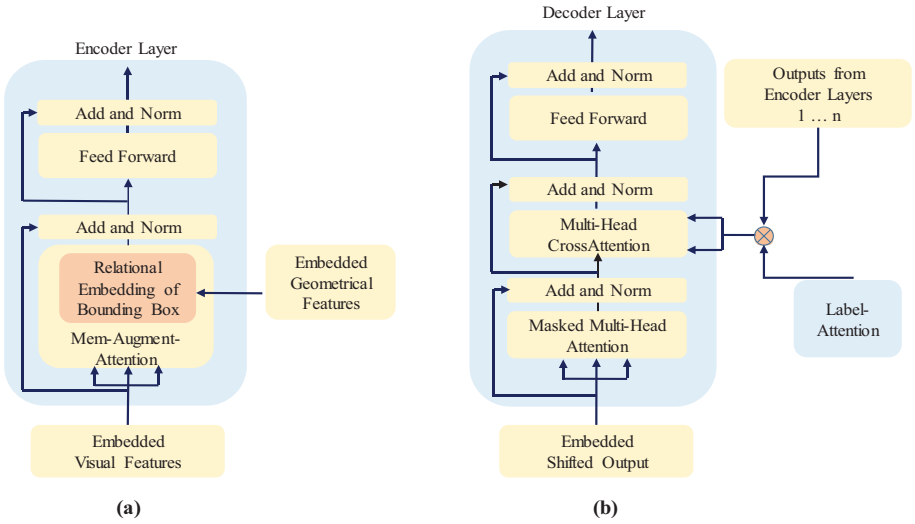

Fig. 4. (a) A detailed structure of the encoder layer. (b) A detailed structure of the decoder layer.

$$V_f = X^T\ W. \tag{15}$$

In Eq. (14) and Eq. (15), $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{o_n}$ represent n number of objects features, $\boldsymbol{x}_{o_{n+1}}$ represents background features, V_f represents embedded visual features, and weights $W \in \mathscr{R}^{2048 \times d_{model}}$.

3.4.2. Encoder layer

LATGeO encoder is composed of L identical encoder layers. Each layer is composed of two components: a multi-head memory-augmented-attention $MA_{Att}(\cdot)$ and a position-wise feed-forward network. These two components also have residual connections among themselves [5]. The detailed structure of the LATGeO encoder layer is shown in Fig. 4(a). The GCP embeddings, along with the embedded visual features, are utilized in the $MA_{Att}(\cdot)$ mechanism as follows:

$$MA_{Att}(Q_E, K_E, V_E, \eta_G) = Multi - Head_E(Q_E, K_E, V_E, \eta_G), \tag{16}$$

$$Multi - Head_E(Q_E, K_E, V_E, \eta_G) = \left[Head_{E_1}, Head_{E_2}, ..Head_{E_h}\right] W_1^0, \tag{17}$$

where η_G represents GCP embeddings already defined in Eq. (2). The terms Q_E, K_E, and V_E represent Q, K, and V, respectively, for the encoder (E) layer. The module $Multi - Head_E(\cdot)$ represents the multi-head attention process in the encoder layer, and W_1^0 is the learned projection matrix for $Multi - Head_E(\cdot)$.

$$Q_{E_i} = W_{q_i} V_f, \tag{18}$$

$$K_{E_i} = \left[W_{k_i} V_f, M_{k_i}\right], \tag{19}$$

$$V_{E_i} = \left[W_{v_i} V_f, M_{v_i}\right]. \tag{20}$$

In Eqs. (18), (19), (20), matrices Q_{E_i}, K_{E_i}, and V_{E_i} are Q, K, and V for the i^{th} encoder head, respectively. Similarly, W_{q_i}, W_{k_i}, and W_{v_i} are learnable projection matrices for the i^{th} encoder head, $Head_{E_i}$ along with different trainable memory slots M_{k_i}, and M_{v_i} of size M, respectively, like in [5]. Each head $Head_{E_i}$ is simultaneously calculated as below:

$$Head_{E_i}(Q_{E_i}, K_{E_i}, V_{E_i}, \eta_G) = \eta\ V_{E_i}, i = 1, 2, \ldots, h, \tag{21}$$

where h is the number of heads. The term η^{ab} is calculated as follows [5]:

$$\eta^{ab} = \frac{\eta_G^{ab} \exp(\eta_V^{ab})}{\sum_{l=1}^{(o_n+1+M)} \eta_G^{al} \exp(\eta_V^{al})}, \tag{22}$$

$$\eta_V = \frac{Q_E K_E^T}{\sqrt{d_k}}. \tag{23}$$

In Eq. (22), η_V represents attention weights of visual features (calculated using Eq. (23)), and η^{ab} represents the fused attention weights for the $MA_{Att}(\cdot)$ mechanism. The weights η^{ab} are calculated by incorporating geometric attention weights η_G and visual attention weights η_V. Moreover, all the encoder layers, consisting of h number of heads, are stacked so that the l^{th} layer takes input from the previous layer $(l-1)^{th}$. The output from $MA_{Att}(\cdot)$ is passed through the next component, a position-wise feed-forward network, to generate embedded non-singular affine transformations. Thus, each encoder layer generates $E_{Out}(n)$ (employed in Eq. (13)) after encapsulating $MA_{Att}(\cdot)$ and embedded transformations. Note that both embedded features passed through residual connection and a layer norm operation.

3.4.3. Decoder layer

LATGeO decoder is composed of L identical decoder layers. Each layer is composed of three components: masked multi-head attention, a multi-head cross-attention, and a position-wise feed-forward network. These components also have residual connections among themselves [5]. The detailed structure of the LATGeO decoder layer is shown in Fig. 4(b). A target caption C^N of length N is a sequence of words in a particular order. The masked multi-head attention module complies with the positional order only by taking input sequence $C^{<t}$ of length less than t for predicting the t^{th} word. The positional order of words is achieved by employing the sinusoidal positional encoding method [32]. The masked multi-head attention generates embedded sequence vectors Y. In this component, C^{t-1} induces Q, whereas K and V are derived from previous words $C^{<t}$ of the target caption. The other components of the decoder layer take inputs $\widehat{I}$ and Y to predict the next word. The attention mechanism in the decoder layer $M_{Att}(\cdot)$ is computed from the gated cross-attentions similar to the one used in the study [5], as follows:

$$M_{Att}\left(\widehat{I}, Y\right) = \sum_{i=1}^{L} \alpha_i * CrossAtt\left(\widehat{I}^i, Y\right), \tag{24}$$

$$CrossAtt\left(\widehat{I}^i, Y\right) = Att\left(W_q Y, W_k \widehat{I}^i, W_v \widehat{I}^i\right), \tag{25}$$

$$\alpha_i = \sigma\left(W_i\left[Y, CrossAtt\left(\widehat{I}^i, Y\right)\right] + b_i\right). \tag{26}$$

In Eq. (24), $M_{Att}\left(\widehat{I}, Y\right)$ is a weighted sum over encoder-decoder cross-attention $CrossAtt(\cdot)$, which is employed in all the decoder layers to attend the outputs from each encoder layer. In $CrossAtt(\cdot)$, $\hat{I}^i$ represents the combined input from the i^{th} encoder layer output and LAM. In Eq. (25), the module $CrossAtt(\cdot)$ computes cross associations between the encoder and decoder by utilizing $Att(\cdot)$ module, where Q is derived from the decoder, and K and V are derived from the encoder layer. A sigmoid gating technique in Eq. (26) is applied to the outputs of each encoder layer before passing them as inputs to the decoder layer. Eq. (26) computes matrix α_i, which yields encoders affinity with the sequence vector Y. This matrix provides relevance measures between $CrossAtt(\cdot)$ and Y. The function $\sigma(\cdot)$ represents the sigmoid activation function, vector b_i is a trainable bias, and W_i represents a weight matrix. The multi-head cross-attention for the decoder layer is calculated by concatenating all attention heads, where each attention head of the decoder layer consists of $M_{Att}\left(\widehat{I}, Y\right)$. At last, similar to [5], the next word C^t is generated using the softmax function after encapsulating the output from the multi-head cross-attention with a position-wise feed-forward network, residual connection, and a layer norm.

3.4.4. Encoder decoder connection

The connections from the encoder to decoder layers used in LATGeO are shown in Fig. 3(b), which depicts that our proposed model has a fully connected encoder-decoder. The results from Eq. (26) provide gated input for each layer of the decoder. In this study, various compositions of the encoder-decoder connections are explored, such as single-connection, skipped-connections, and residual-connections compositions, and they are described in A.2. The detailed structure of LATGeO encoder and decoder layers is shown in Fig. 4. A benefit of using the selected composition of the encoder-decoder in LATGeO is to attend the output of all encoder layers and reevaluate if the stack of encoders misses a valuable relationship. The effectiveness of the selected composition is further discussed in Section 4.3.

3.5. Training objective functions

LATGeO is trained using masked cross-entropy objective function (XE) [14] and label-smoothing function [31]. Afterward, it is tuned with RL using beam search on the generated sequence of words [1]. The XE objective function, $L(\phi)$, is the sum of the negative log-likelihood of the correctly predicted words at each step given as follows:

$$L(\phi) = -\sum_{n=1}^{N} log\left(p_\phi\left(Sw_n^{LS} | I, Sw_1^{LS}, \ldots, Sw_{n-1}^{LS}\right)\right), \tag{27}$$

where ϕ is the learning parameters, Sw_n represents a one-hot vector for the n^{th} word in a ground truth sentence, N is the length of the image-caption, I is an input image, and LS represents label-smoothing with smoothing parameter ϵ [31]. The architecture optimized on the best validation score of CIDEr-D metrics is obtained after supervised learning. The reward function $r(\cdot)$ of RL based on the CIDEr-D score of a generated caption is utilized to calculate the final policy gradient. This gradient [5] computes the reward for each step and it is calculated as follows:

$$\nabla_\phi L(\phi) = -\frac{1}{k}\sum_{j=1}^{k}\Big(\Big(r\Big(S^j\Big) - \beta\Big)\nabla_\phi log\Big(p_\phi\Big(S^j\Big)\Big)\Big). \tag{28}$$

In Eq. (28), S^j is the j^{th} sentence in the beam, k is the beam size, and p_ϕ represents a policy: an "action" of predicting the next word. The policy gradient employs a baseline β as a mean of the rewards, which differs from the rewards based on the greedy decoding used in the earlier methods [29,1], given as below:

$$\beta = \left(\sum_{j=1}^{k} r\Big(S^j\Big)\right)/k. \tag{29}$$

The proposed architecture, LATGeO, is trained using captions of a specific length, N, represented as a vector. This vector representation is generated using an embedding layer of dimensions of d_{model} and then fed into a decoder layer. Elements' order in the sequence is represented using positional encoding, which is added to the decoder input. Positional encoding can be seen as a vector representation of numbers in the range $[1, N]$ projected to d_{model} dimensional decoder. The model takes the output of previously generated words as input to generate the next word during the prediction phase.

4. Implementation and evaluation

4.1. Dataset

In this study, we use the MSCOCO dataset [23]. The dataset consists of 123,287 labeled images and is split using the standard Karpathy technique [19] into 113,287 images in the train set, and 5,000 each in the validation set and the test set. Each image in the dataset has 5 different target captions. For online testing, the split of the dataset is different in such a way that there are 82,783 images in training, 40,504 images in the validation, and 40,775 images in the test sets. It is important to note that target captions of images for online testing are not available publicly. During testing, the target captions are converted into lower-case, and each caption is restricted to a length of $N = 22$ words. Following the work of [5], we place "START", and "END" keywords at the beginning and in the end, respectively, for each caption. For LAM, a dictionary is made of words that occurred more than five times in the whole corpus, resulting in a vocabulary size of 10,021 distinct words. Less frequent words are substituted with the "UNK" keyword.

4.2. Experimental design and implementation

During training and testing, the proposals are generated utilizing Faster R-CNN [27] with a base ResNet−101 [12]. Then the features (classes, probabilities, and bounding boxes) of the extracted proposals along with background features are passed through the GCP and LAM blocks as shown in Fig. 2. Finally, utilizing all this information the captions are generated with the help of the LATGeO transformer. During training, we also pass the target captions to the decoder layer. Faster R-CNN is fine-tuned on the Visual Genome dataset [1,5,19], which contains 1600 object classes. In addition to the objects classes, this dataset provides annotations for objects attributes (like colors, sizes, etc.) and their relationships (like below, under, on, in, etc.). However, this study only uses annotations of objects classes; for future experimentation the other available annotations will be incorporated into the LAM of the proposed framework. Objects with class probabilities greater than 0.7 are selected as proposals, then a maximum of 50 object proposals per image are selected including the background. Similarly, a 2048-dimensional features vector for less significant details is extracted for each image using ResNet−50 [12]. Words are embedded using linear projection of one-hot vector representations of 512-dimensions, which is the same as the input dimensions of our proposed transformer model. Moreover, sinusoidal encoding is used for the positional encoding of words in a target caption [5]. In our study, smoothing parameter ϵ is set to 0.1. In LATGeO, input and output dimensions d_{model} of encoder-decoder architecture are set to 512. The number of heads h is set to 8, and memory size M is set to 40. The number of stacked encoders and decoders layers L is set to 3 (further details in A.3).

LATGeO is trained on an Nvidia 1080Ti machine with RAM 16 GB, using masked cross-entropy objective function, XE (see Eq. (27) above). We employ Adam optimizer with a learning rate scheduling strategy [32] and performed 10,000 warmup iterations. After supervised learning, in RL, the reward function (see Eq. (28) above) is used with patience of 5 based on the CIDEr-D score of the validation set. The reward is achieved by decoding sentences using beam search with a beam-size of $k = 5$ and a learning rate of $5 * 10^{-6}$. All experiments are performed with a batch size of 50. Moreover, the regularization is constructed using an early-stopping technique based on the CIDEr-D score.

4.3. Evaluation

4.3.1. Evaluation metrics

We have evaluated the proposed architecture's performance using regularly used evaluation metrics [5], i.e., BLEU-1, BLEU-4, METEOR, ROUGE-L, SPICE, and CIDEr-D. The quantitative results on the MSCOCO 2014 test set from the Karapathy split and the MSCOCO server evaluation test set are given in Table 1, Table 2, and Table 3.

4.3.2. Evaluation on MSCOCO Karapathy split

LATGeO is compared with the recent best single-model algorithms as well as with recent ensemble-model algorithms. The proposed model trained using XE objective function outperforms all SOTA single-model [44,37,50,25,48,4,22,2,13,1,38,11,45,36,34,16,14,5] and ensemble-model algorithms [33,29,24], as shown in Table 1. Furthermore, it improves scores for all evaluation metrics compared to the transformer-based algorithms such as MeshTrans [5] and ObjRel-Trans [14], i.e., 2.7% and 3.2% improvement on CIDEr-D scores, respectively. For a fair comparison, we have trained MeshTrans [5] (our base model) with a similar preprocessing and hyper-parameters to our training of LATGeO.

Furthermore, LATGeO, when trained with RL, boosts the performance and produces the highest BLEU-1, METEOR, SPICE, and CIDEr-D scores, which are presented in Table 2. LATGeO outperforms DNN-based [28,6,37,47,44,41], attention-based [39,29,1,22,11,36,10,9,17,46,40,43,30,35,42], and transformer-based algorithms [14,18,5,15,49] for most of the evaluation metrics. As shown, LATGeO outperforms the MeshTrans (our base model) [5] for all evaluation metrics with a 2.5% CIDEr-D score improvement and shows the superiority of our model over MeshTrans. It also outperforms with a 3.4% CIDEr-D score improvement compared to another transformer-based algorithm [14], which includes geometrical features different from ours, and with 1.9% CIDEr-D score improvement compared to [15]. Additionally, LATGeO outperforms the transformer-based algorithm [49] for all metrics (except ROUGE-L) with a 2.5% CIDEr-D score improvement. LATGeO shows 0.8% CIDEr-D score improvement compared to [18]. Here, we have compared the SV model of the work [18] with LATGeO. Although [41] gives better BLEU-4 and ROUGE-L scores than LATGeO, they use an ensemble technique to present their evaluation metrics. Similarly, [49] gives a better ROUGE-L score than LATGeO, because in their study they have additionally utilized spatial relationship and visual region attention module. LATGeO single-model, however, shows better evaluation results than [41,49] on other metrics, including CIDEr-D, where we achieve 2.6% and 2.5% improvement, respectively.

4.3.3. Evaluation on online COCO server

Table 3 illustrates the online performance of our proposed architecture, LATGeO, on the COCO test server. We have utilized single-model LATGeO for the online evaluation. For a fair comparison, we have summarized the comparison of our model only with the top-performing single-models from the server leader-board [37,47,44,13,8,45,38,34,22,11,10,46,40,35,43,42]. Moreover, as per our knowledge, our proposed framework, LATGeO, is the first to report a transformer-based

Table 1
LATGeO evaluation using BLEU-1, BLEU-4, METEOR, ROUGE-L, and CIDEr-D scores on Karpathy's split MSCOCO test dataset (all values are in percentage %). The symbol ⊕ denotes an ensemble model, and the rest are single models, Bold figures depict the best results. The symbol * represents values after training the model using the authors' code and employing the same data preprocessing routine as ours. The studies marked with # utilize ResNet-152 based visual features.

	Model	BLEU-1	BLEU-4	METEOR	ROUGE-L	CIDEr-D
DNN	NICv2⊕ [33]	-	32.1	25.7	-	99.8
	MSM # [44]	73.0	32.5	25.1	-	98.6
	Recall [37]	73.4	32.2	25.9	-	101.6
	LSTM_p + ATT_s [50]	73.8	32.7	26.1	54.1	101.8
Template	NBT [25]	75.5	34.7	27.1	-	107.2
Attention	Fine-Grain [48]	71.2	26.5	24.7	-	88.2
	SCA-CNN [4]	71.9	31.1	25.0	53.1	95.2
	Obj-R + Rel-A [22]	73.2	32.8	25.6	53.4	96.5
	Bawg-LSTM+mean [2]	71.9	30.2	25.3	-	99.8
	VD-SAN [13]	73.4	32.2	25.4	-	99.9
	Up-Down[1]	74.5	33.4	26.1	54.4	105.4
	SCST (Att2in) ⊕ [29]	-	32.8	26.7	55.1	106.5
	AttM[38]	75.7	33.7	26.3	55.1	106.8
	Adaptive ⊕ [24]	74.2	33.2	26.6	-	108.5
	Stack-Cap (C2F)[11]	76.2	35.2	26.5	-	109.1
	ALT-ALTM [45]	75.1	35.5	27.4	55.9	110.7
	GateCap_A[36]	75.9	35.5	27.4	56.3	110.8
	ARL[34]	75.9	35.8	**27.8**	56.4	111.3
	att-ref[16]	76.4	36.1	27.6	56.4	114.5
Transformer	Up-Down + ObjRel-Trans[14]	75.6	33.5	27.6	56.0	112.6
	MeshTrans*[5]	75.7	35.4	27.8	56.4	113.1
	LATGeO (Ours)	**76.5**	**36.4**	**27.8**	**56.7**	**115.8**

Table 2
LATGeO results after training with RL (CIDEr-D optimization) (all values are in percentage %). The symbol ⊕ denotes an ensemble model, and the rest are single models, Bold figures depict the best results. The symbol * represents values after training the model using the authors' code and employing the same data pre-processing routine as ours. The studies marked with # utilize ResNet-101 based visual features.

	Model	BLEU-1	BLEU-4	METEOR	ROUGE-L	SPICE	CIDEr-D
DNN	RL-EmbeddedReward [28]	71.3	30.4	25.1	52.5	-	93.7
	RL-G-GAN[6]	-	29.9	24.8	52.7	19.9	102.0
	Recall [37]	75.8	33.1	24.7	-	103.7	
	Actor-Critic [47]	-	34.4	26.7	55.8	-	116.2
	MSM # [44]	78.6	35.5	27.3	56.8	-	118.3
	SGAE⊕ [41]	**81.0**	39.0	28.4	58.9	22.2	129.1
Attention	Hierarchical-Attention [39]	73.0	28.6	25.3	56.5	-	92.5
	SCST (Att2all) ⊕ [29]	-	35.4	27.1	56.6	-	117.5
	Up-Down[1]	79.8	36.3	27.7	56.9	21.4	120.1
	Obj-R + Rel-A[22]	79.2	36.3	27.6	56.8	21.4	120.2
	Stack-Cap (C2F)[11]	78.6	36.1	27.4	56.9	20.9	120.4
	GateCap_O[36]	79.3	37.3	27.9	57.7	-	124.0
	hLSTMat# [10]	79.9	37.5	28.5	58.2	22.3	125.6
	CFG [9]	80.5	38.3	28.2	58.3	21.6	125.4
	RFNet⊕ [17]	80.4	37.9	28.3	58.3	21.7	125.7
	Fine-Visual-policy[46]	-	38.6	28.3	58.5	21.6	126.3
	IFE+CapNet+FT [40]	80.4	38.5	28.8	58.3	22.5	127.6
	GCN-LSTM⊕ [43]	80.9	38.3	28.6	58.5	22.1	128.7
	ETN [30]	80.6	39.2	-	58.9	22.6	128.9
	SG+RWS+WR [35]	80.3	38.5	28.7	58.4	22.4	129.1
	SGAE -KD[42]	**81.0**	38.8	28.8	58.8	22.4	129.6
Transformer	ObjRel-Trans[14]	80.5	38.6	28.7	58.4	21.2	128.3
	VRAtt-Soft-Trans [49]	80.5	38.5	28.9	**61.8**	22.8	129.2
	MeshTrans*[5]	80.7	38.8	28.9	58.4	22.6	129.2
	AoANet[15]	80.2	38.9	**29.2**	58.8	22.1	129.8
	MT-sv[18] #	80.8	**39.8**	29.1	59.1	-	130.9
	LATGeO (Ours)	**81.0**	38.8	**29.2**	58.7	**22.9**	**131.7**

Table 3
Online evaluation of LATGeO on MSCOCO test server. The studies marked with *, and # utilize ResNet-152 and ResNet-101 based visual features, respectively. The results are sorted on the CIDEr-D scores. Only single model architectures are reported in this table (Oct, 2021).

	Model	BLEU-1		BLEU-2		BLEU-3		BLEU-4		METEOR		ROUGE-L		CIDEr-D	
		c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
DNN	Recall [37]	74.9	92.5	58.3	84.2	43.8	73.4	32.1	61.7	24.2	32.1	54.3	68.5	98.6	102.0
	Actor-Critic (single)[47]	77.8	92.9	61.2	85.5	45.9	74.5	33.7	62.5	26.4	34.4	55.4	69.1	110.2	112.1
	MSM* [44]	78.7	93.7	62.7	86.7	47.6	76.5	35.6	65.2	27.0	35.4	56.4	70.5	116.0	118.0
Attention	VD-SAN[13]	73.7	91.7	57.1	82.7	43.1	72.2	32.4	61.1	25.6	34.5	-	-	98.5	98.9
	SDCD[8]	74.8	-	52.5	-	36.5	-	23.5	-	23.5	-	50.5	-	104.1	-
	ALT-ALTM[45]	74.3	92.0	57.8	84.0	44.1	73.8	33.7	63.1	26.8	36.4	55.1	70.7	104.6	105.3
	AttM[38]	75.5	92.4	58.8	84.6	44.5	74.2	33.3	62.9	26.0	34.7	54.8	69.7	103.1	104.7
	ARL [34]	-	-	58.9	85.6	45.0	75.6	34.3	64.7	27.0	36.4	55.5	71.0	106.1	106.4
	Obj-R + Rel-A [22]	79.2	94.4	62.6	87.2	47.5	77.1	35.4	65.8	27.3	36.1	56.2	71.2	115.1	117.3
	Stack-Cap (C2F)[11]	77.8	93.2	61.6	86.1	46.8	76.0	34.9	64.6	27.0	35.6	56.2	70.6	114.8	118.3
	hLSTMat# [10]	79.4	94.4	63.5	88.0	48.7	78.4	36.8	67.4	28.2	37.0	57.7	72.2	120.5	122.0
	Fine-Visual-policy [46]	80.1	94.9	64.7	88.8	50.0	79.7	37.9	69.0	28.1	37.0	58.2	73.1	121.6	123.8
	IFE+CapNet+FT[40]	79.5	94.9	63.8	88.7	49.2	79.4	37.5	68.8	28.4	37.7	57.8	73.0	122.6	125.6
	SG+RWS+WR[35]	80.0	94.7	64.4	88.5	49.6	79.2	37.6	68.4	28.3	37.2	57.8	72.7	123.3	125.7
	GCN-LSTM[43]	-	-	**65.5**	89.3	**50.8**	80.3	**38.7**	69.7	28.5	37.6	**58.5**	73.4	125.3	126.5
	SGAE -KD[42]	-	-	-	-	50.1	79.9	38.2	69.3	28.7	37.9	58.4	**73.5**	124.5	126.6
Transformer	*LATGeO(Ours)*	**80.5**	**95.4**	64.8	**89.6**	50.0	**80.8**	37.9	**70.3**	**28.8**	**38.2**	58.1	73.2	**126.7**	**130.1**

single-model for online evaluation. Table 3 demonstrates that our model surpasses all the current SOTA methods on most of the evaluation metrics and achieves an improvement of 3.5% CIDEr-D score to the previous best single-model algorithm [42].

4.4. Discussion

4.4.1. DETR objects proposals

We have also experimented LATGeO with proposals generated using DETR [3] because of its several advantages over Faster R-CNN e.g., DETR does not use any computational expensive hand-crafted techniques like non-maximum suppression

(NMS). Object proposal is one of the essential parts of our proposed technique. In this study, ResNet-50 [12] is used as a base network for DETR. The RGB features of object proposals are passed through another ResNet-50 model to generate 2048-dimensional visual feature maps.

Table 4 presents a performance comparison of LATGeO using DETR and Faster R-CNN object detectors; when Faster R-CNN was trained on the MSCOCO detection dataset, it is represented as LATGeO-Faster R-CNN, and when it was fined-tuned with the Visual Genome dataset [19], it is represented as LATGeO-Faster R-CNN + G in Table 4. We further categorize these comparisons using two different objective functions: XE and RL. Faster R-CNN + G outperforms LATGeO with DETR because Faster R-CNN was fine-tuned on the Visual Genome dataset, which connects the visual domain to the language domain using 1600 object classes, whereas DETR has 91 object classes. In Table 4, when both object detectors, DETR and Faster R-CNN, have been trained on the MSCOCO detection dataset, LATGeO-DETR (XE and RL) models outperform LATGeO-Faster R-CNN (XE and RL) models. Therefore, we can conclude that fine-tuned DETR model on the Visual Genome dataset for image captioning may achieve better results than Faster R-CNN and will be investigated in future work. Moreover, Table 4 includes the performance evaluation on the decomposition of the SPICE metric to show the importance of the object proposals using objects and relationships among objects, including color, count, and size [14]. Using the SPICE metric decomposition, Faster R-CNN + G shows a better relationship among object proposals than LATGeO-DETR. This demonstrates the importance of using a better object proposals technique.

4.4.2. Ablation study: effectiveness of LATGeO modules

Table 5 illustrates the effectiveness of individual modules proposed in LATGeO. It asserts that modules OP, Bg, GCP, and LAM collectively produce the highest scores except for the BLEU-4 score. Moreover, Table 6 demonstrates the effectiveness of the GCP module along with the LAM by decomposing the SPICE metric into objects, attributes, relation, color, count, and size metrics. We have compared these metrics with a recent transformer-based model, MeshTrans [5], and other recent methods. As illustrated, LATGeo shows improvements in relation, attribute, count, and object metrics, compared to all other mentioned methods, though outperforms in all metrics compared to the MeshTrans model. Tables 5 and 6 show that GCP and LAM improve the overall performance of LATGeO and generate fine captions.

Table 4
LATGeO results using Faster R-CNN and DETR, where G stands for Visual Genome dataset.

Model	B-1	B-4	M	R	C	SPICE						
						All	Object	Att	Relation	Color	Count	Size
LATGeO-DETR (XE)	75.3	34.7	27.1	55.7	112.1	20.3	37.1	9.8	5.6	10.0	11.8	4.7
LATGeO-Faster R-CNN (XE)	70.6	30.2	–	52.3	94.1	–	–	–	–	–	–	–
LATGeO-Faster R-CNN + G (XE)	76.5	36.4	27.8	56.7	115.8	20.9	37.6	11.0	5.8	12.5	13.0	5.1
LATGeO-DETR (RL)	79.8	37.2	28.5	57.6	127.0	22.0	39.7	10.8	6.6	12.1	22.1	3.1
LATGeO-Faster R-CNN (RL)	76.0	33.1	–	54.5	110.5	–	–	–	–	–	–	–
LATGeO-Faster R-CNN + G (RL)	81.0	38.8	29.2	58.7	131.7	22.9	40.7	12.1	7.1	14.6	22.9	4.2

Table 5
A comparison of different modules proposed in LATGeO. Acronyms are defined as, OP: Object Proposals Module–Faster R-CNN, Bg: Background Features, GCP-L1: Geometrically Coherent Proposals: L1-Comparison, GCP-Ratio: Geometrically Coherent Proposals: Ratio-Comparison (Default), LAM: Label-Attention Module, GCP: Geometrically Coherent Proposals. Bold figures stand for the best performance in all.

Model	BLEU-1	BLEU-4	METEOR	ROUGE-L	CIDEr-D	SPICE
OP	80.5	38.6	28.7	58.4	129.2	22.5
OP + GCP-L1	80.3	38.4	28.9	58.6	129.9	22.7
OP + GCP-Ratio	80.5	38.5	**29.2**	58.4	130.4	22.7
OP + GCP + Bg	80.6	**38.9**	**29.2**	58.5	130.7	**22.9**
OP + GCP + Bg + LAM *(LATGeO) (Ours)*	**81.0**	38.8	**29.2**	**58.7**	**131.7**	**22.9**

Table 6
LATGeO evaluation with SPICE shows significant improvement in Relation, Attributes, Object, and Count metrics.

Model	SPICE						
	All	**Object**	**Attributes**	**Relation**	**Color**	**Count**	**Size**
Standard Transformer	21.1	38.6	9.6	6.3	9.2	17.5	2.0
ObjRel-Trans [14]	21.2	37.9	11.4	6.3	**15.5**	17.5	**6.4**
Up–Down [1]	21.4	39.1	10.0	6.5	11.4	18.4	3.2
hLSTMat [10]	22.3	40.3	11.2	6.4	15.2	14.4	3.7
MeshTrans [5]	22.6	40.0	11.6	6.9	12.9	20.4	3.5
LATGeO (Ours)	**22.9**	**40.7**	**12.1**	**7.1**	14.6	**22.9**	4.2

4.4.3. Qualitative analysis of LATGeO

We have shown image captions of selected images generated by the proposed framework for qualitative analysis in Fig. 5. This figure depicts the image captions generated by LATGeO and competitive techniques for parallel comparison. A row with qualitative analysis of the generated caption for each selected image is appended to highlight the improvements introduced in the captions presumably caused by proposed modules. For example, Fig. 5(a) depicts an improvement in the translation of image features to a precise collective noun using object proposals (OP) with LAM in the generated caption: "a man riding → a group of people riding". Similarly, Fig. 5(b) shows improvement in the generated caption up to a precise common noun by utilizing OP with label association: "two men → two people". Also, in Fig. 5(l), "a man and woman → a couple of people". Additional examples in Fig. 5(c–h) accentuate OP inclusion in LAM and its efficacy for LATGeO. Furthermore, Fig. 5(b) depicts the qualitative significance of GCP by adding an explicable adjective in the generated caption: "a kite → a large kite". The effect of adding background features in LATGeO is also evident in Fig. 5(a-c), (e-g), and (i-l). For instance, in Fig. 5(c), the generated caption is corrected with an accurate object in the background: "a refrigerator → a bathtub". Similarly, Fig. 5(g) shows improved captions from "metal rack in a → fence in a flooded street". Finally, examples about the influence of fine use of OP

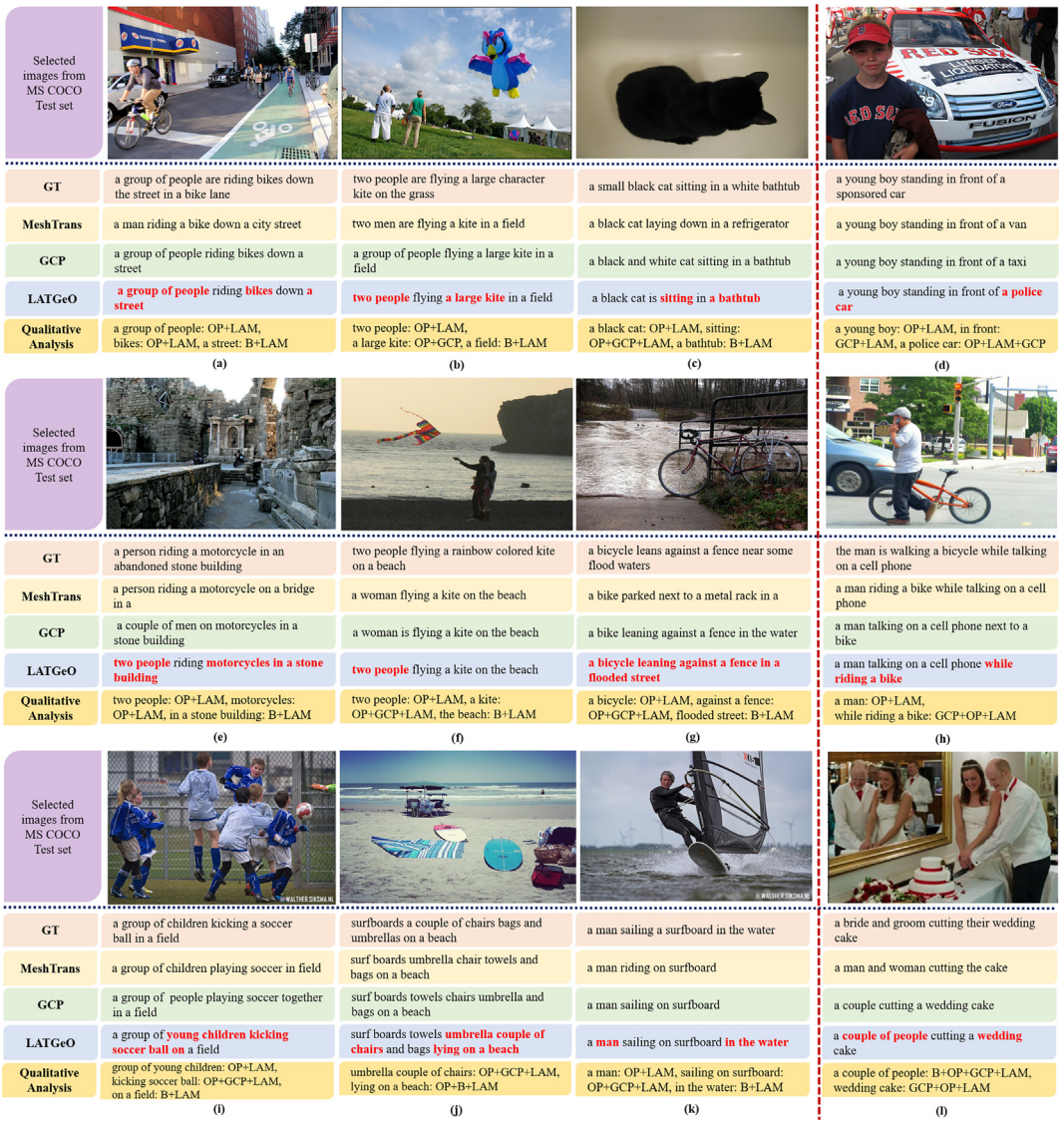

Fig. 5. Qualitative results of selected images from the MSCOCO test dataset. GT, MeshTrans, and GCP represent captions generated by ground truth, MeshTrans [5], and the GCP module of LATGeO, respectively. For brevity, OP denotes object proposals, and B denotes background. (a) -(c), (e) -(g), and (i) -(k), represent successful image captioning cases, and (d), (h), and (l) represent cases where LATGeO-generated captions show low accuracy. Last rows (Qualitative Analysis) brief the expected modules of the LATGeO which could be involved in generating accurate captions as well as show the improvement compared to the MeshTrans model. Highlighted words with red color in the LATGeO captions represent caption refinements compared to the MeshTrans model.

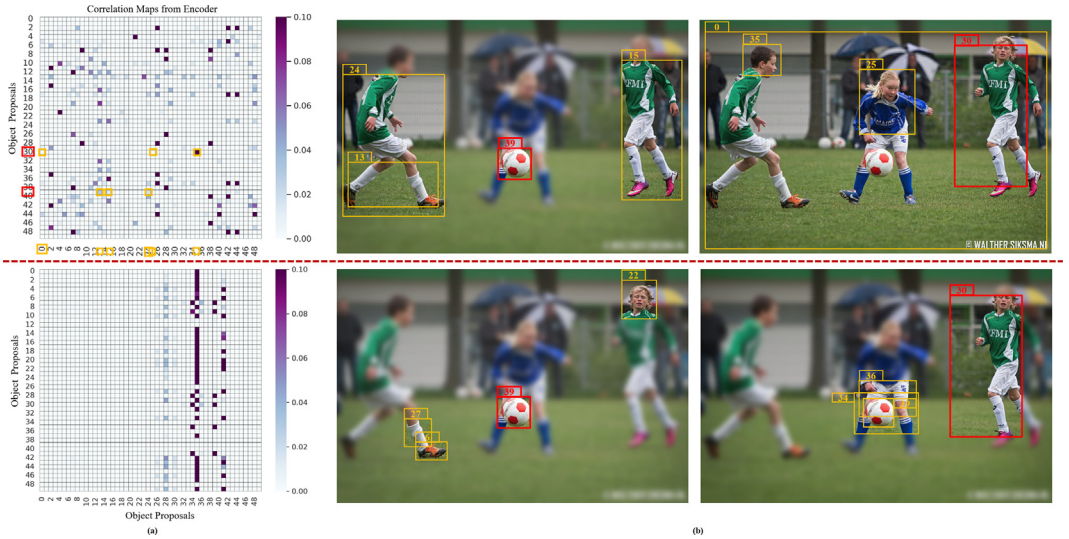

Fig. 6. Demonstration of correlation among object proposals after passing through LATGeO encoder. The top and bottom rows show results with and without GCP, respectively. (a) shows the correlation maps of object proposals. (b) shows the top-3 objects in the yellow bounding box giving higher correlated values with two randomly chosen object proposals 39 and 30 (left and right images, respectively). Correlated object proposals are highlighted on the original image with their id numbers on the top-left corner of the highlighted part.

with LAM and GCP are depicted in Figs. 5(d) and (l): "a sponsored car → a police car", and "cake → a wedding cake", respectively. Therefore, the qualitative results in Fig. 5 highlight the efficacy of background, OP, LAM, and GCP in LATGeO to generate semantically and syntactically concise image captions.

Furthermore, Fig. 6 demonstrates the effectiveness of the GCP module, where the correlation maps of object proposals are generated with the help of the encoder layer's attention module. Fig. 6(a) top image shows a correlation map constructed using a model trained with the GCP module, and the bottom image shows a correlation map generated without the GCP module. As demonstrated, correlations among objects achieved from a model with GCP have more diverse dependencies than the ones without GCP. Most object proposals are correlated with object numbers 35 and 41 (both belong to the "person" class) without GCP, whereas the GCP adds variation in the inclusion of proposals. Moreover, we have demonstrated the correlation of object proposals with the chosen proposals id numbers 30 and 39 marked in red bounding boxes and displayed their top-3 correlated object proposals with yellow bounding boxes (Fig. 6(b)). Proposal id number 39 ("football") in the first row is more correlated with proposal id numbers 24, 13, and 15, which depicts the correlation with persons present in the image showing an action of kicking a football, wearing football shoes, and football jerseys. Whereas, the model without the GCP module in the second row depicts the correlation of proposal id number 39 with proposals 27, 6, and 22, which mainly give attention to the person's face and shoes. Similarly, Fig. 6(b) also demonstrates the GCP module's effect on the proposal id number 30 ("boy or person"). This proposal shows a high correlation with other proposals like other players in the field along with the football field. Fig. 6 demonstrates that we achieve spatially distant object associations with the GCP module.

The significance of the LAM in the decoder is shown in Fig. 7, where we have demonstrated the association of extracted object proposals with particular words in the dictionary. This association is achieved from the decoder layer using a cross-attention module, which employs LAM and encoders' outputs. Such association process in LATGeO using the LAM module is zoomed-out and demonstrated inside a dotted orange box. The colored lines represent such associations generated from the high attention weights from the cross-attention module of the decoder. In the second row, the caption "young" shows more attention to the object proposals, containing faces of the players and their actions of playing and running. Similarly, the caption "children" is associated with the object proposals, containing all young boys and girls playing on the ground. Despite having many objects per image (max = 50), LATGeO can adequately generalize to map only a small number of objects per word. Furthermore, Fig. A.8 in A.1 depicts a detailed comparison of attention mapping between generated captions with and without LAM + GCP in LATGeO.

4.4.4. Future work

The proposed modules are tested with transformers for their seamless integration in end-to-end deep neural networks. However, the modules can be trained independently with the same dataset and different ground truth. The image captioning pipeline of conventional algorithms can include the output of LAM and GCP as additional data input for final caption generation. Moreover, the proposed techniques can be extended for the grounded situation recognition (GSR) [26] that describes activities and the roles associated with the activity in a given image. LAM module pays attention to the noun detected in the image and explores their relationship. GCP associates objects considering the geometric structure and scale. Apparently, the

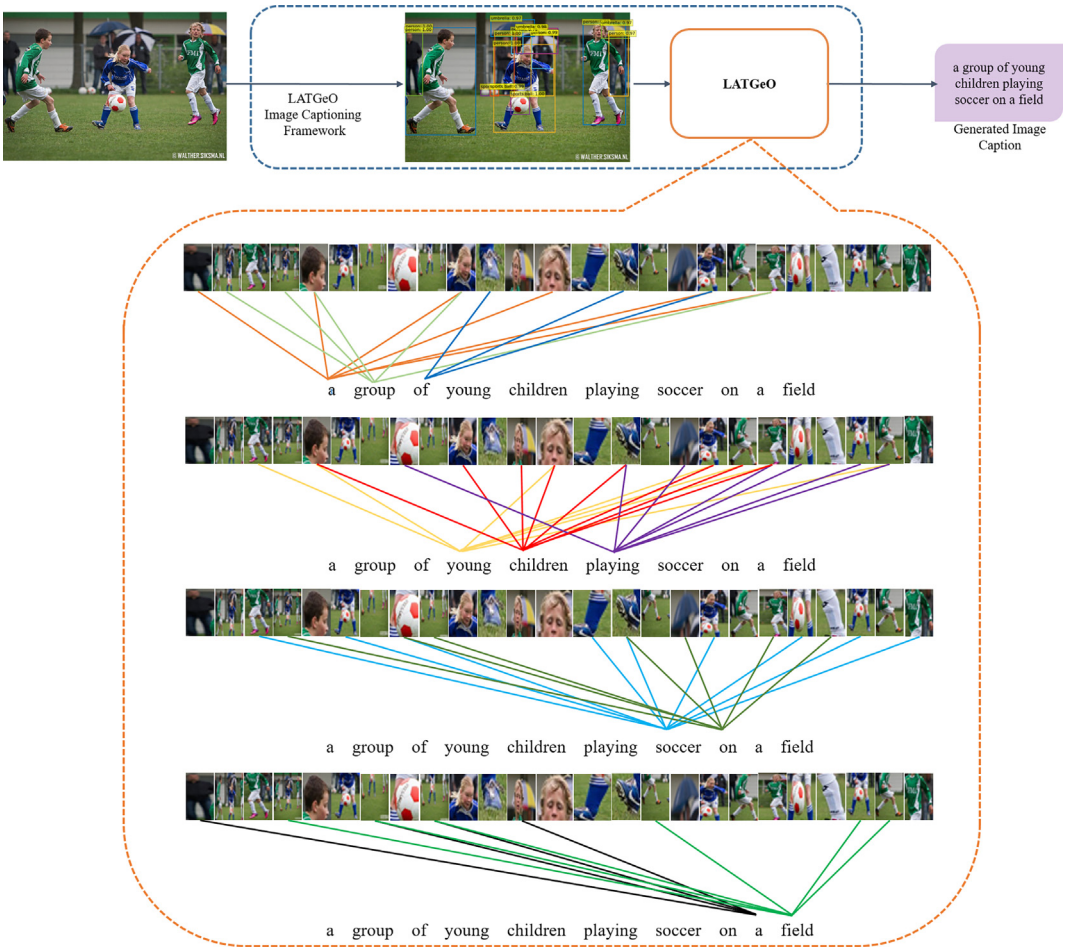

Fig. 7. Demonstration of extracted objects' association to words using the proposed LAM in the decoder of LATGeO.

proposed modules can help in improving the GSR performance. Moreover, attributes of the objects such as objects' colors, and sizes can be encapsulated as labels in the LAM to extract nouns present in the image along with their attributes. Another potential application of LATGeO is scene understanding in videos (e.g. anomaly detection and action recognition), where the two modules can add value by imposing vision constraints and biases.

5. Conclusions

This study demonstrates an image captioning technique named LATGeO, which explores the utility of object identity preservation along with surrounding information to generate meaningful captions of still images. LATGeO binds objects' features, surroundings, geometrical properties, and associated labels of semantically coherent objects using a transformer. The proposed architecture generates proposals using object detection algorithms and computes their geometrical coherence using a novel scale ratio-comparison technique that helps the transformer to attend spatially distant objects in a given image. The encoder and the associations of particular features of the objects are further reviewed, strengthened, and transcribed using labels of the detected objects by a decoder. The labels are a mapping of classes to the dictionary words using a proposed LAM. An extrinsic definition of proposal added to the decoder helped LATGeO to outperform SOTA algorithms on the MSCOCO test dataset. The proposed technique is trained with cross-entropy loss and fine-tuned with reward-based RL, which improves the results and scores better than many SOTA offline ensembles and shows outstanding performance in the online evaluation. This study also shows the effectiveness of introducing different forms of inductive biases (soft- and hard-inductive bias) in the transformer network. Future work can explore the utility of the proposed combination of biases in GSR, scene understanding, and anomaly detection using captions. In this study, we examined object detection for proposals, whereas other choices of proposals can be explored to explicitly guide the transformers. Similarly, object coherence and label generation can be further explored to improve the results.

CRediT authorship contribution statement

Shikha Dubey: Conceptualization, Methodology, Software, Validation, Formal analysis, Data curation, Writing - original draft, Visualization. **Farrukh Olimov:** Methodology, Data curation, Writing - original draft, Validation, Visualization. **Muhammad Aasim Rafique:** Visualization, Validation, Investigation, Data curation, Writing - review & editing. **Joonmo Kim:** Visualization, Validation, Writing - review & editing. **Moongu Jeon:** Resources, Supervision, Project administration, Funding acquisition, Writing - review & editing.

Data availability

Data associated with this research is available in publicly accessible repository. The associated dataset, MS COCO Dataset, can be found in the location: https://cocodataset.org. The code is publicly available on https://github.com/shikha-gist/Image-Captioning.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was financially supported partially by 1) the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2014–3-00077, AI National Strategy Project), 2) the Culture, Sports and Tourism R & D Program through the Korea Creative Content Agency (KOCCA) grant funded by the Ministry of Culture, Sports and Tourism in 2022 (Development of Intelligent Exhibition Commentary Platforms for Deaf-Korean Sign Language/Word Translation Systems, R2020060002), and 3) the GIST-MIT Research Collaboration grant funded by the GIST in 2022.

Appendix A. Additional experimentation

A.1. Attention mapping from the decoder layer

Fig. A.8 demonstrates the effectiveness of the proposed LAM and GCP with the help of attention mapping obtained from the decoder layer. The image on the left-side shows the cross-attention between detected objects and the words generated by the decoder module of the model with LAM and GCP. The image on the right-side shows the results of the same image

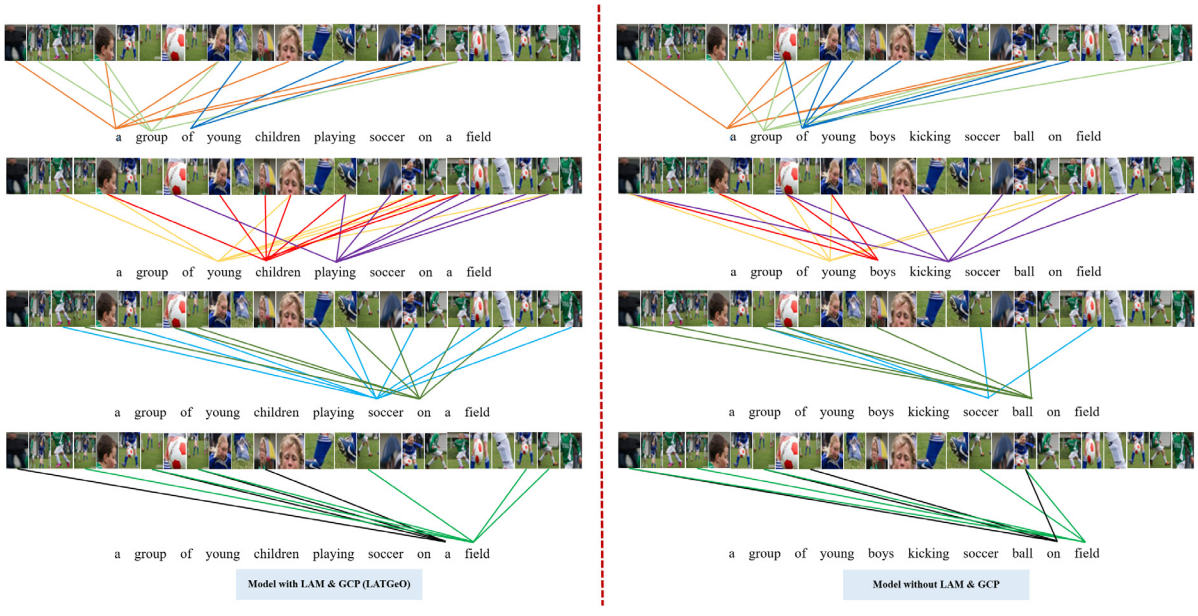

Fig. A.8. Comparison of cross attention mappings of models with (left) and without (right) GCP + LAM.

without the use of LAM and GCP. It can be observed in the first row of the left-side that strong attention between the phrase "group of" and the object proposals with a group of people in it when using the LAM and GCP modules. The same can not be observed on the right-side image. In the second row, the word "playing" has more significant attention connections with object proposals giving an impression of running, maneuvering, and handling the ball, which is visible with low frequency on the word "kicking" on the right-side. It is also considered as a benefit of using LAM. Similar connections can be observed in rows three and four with the words "soccer" and "field", respectively. It is considered that LAM adds a semantic value to the image to words translation.

A.2. Composition of encoder-decoder layers of proposed transformer

We have performed additional experiments to illustrate the effect of different types of connectivity between encoder-decoder layers. Fig. A.9 and Fig. A.10 demonstrate the details for connections in 3-layers and 6-layer architectures, respectively. Fig. A.9(a) shows the single-connection when one encoder output is passed as an input to the corresponding decoder layer. Fig. A.9(b) shows skip-connection when a randomly selected few encoder layers outputs are passed as input to the decoder layer after sigmoid gating. Fig. A.9(c) shows residual-connection [12] among encoder layers: when a residual connection is included among encoder layers along with a fully-connected transformer. Fig. A.9(d) shows residual-connection in encoder and decoder layers: a residual connection is included among encoder layers and decoder layers along with a fully-connected transformer. Fig. A.10(a) and (b) represent similar connections in 6-layers architecture.

Table A.7 demonstrates the comparative analysis of using different connectivity in LATGeO. This table demonstrates that our model with fully-connected encoder-decoder layers outperforms other mentioned connection techniques.

A.3. Number of encoder-decoder layers

Fig. A.9 and Fig. A.10 also demonstrate the proposed architecture with different numbers of encoder-decoder layers. Table A.7 shows the effect of using 3-layers and 6-layers in the proposed algorithm. This table concludes that using 3-layers of encoder-decoder and a fully-connected transformer shows the best results compared to other compositions.

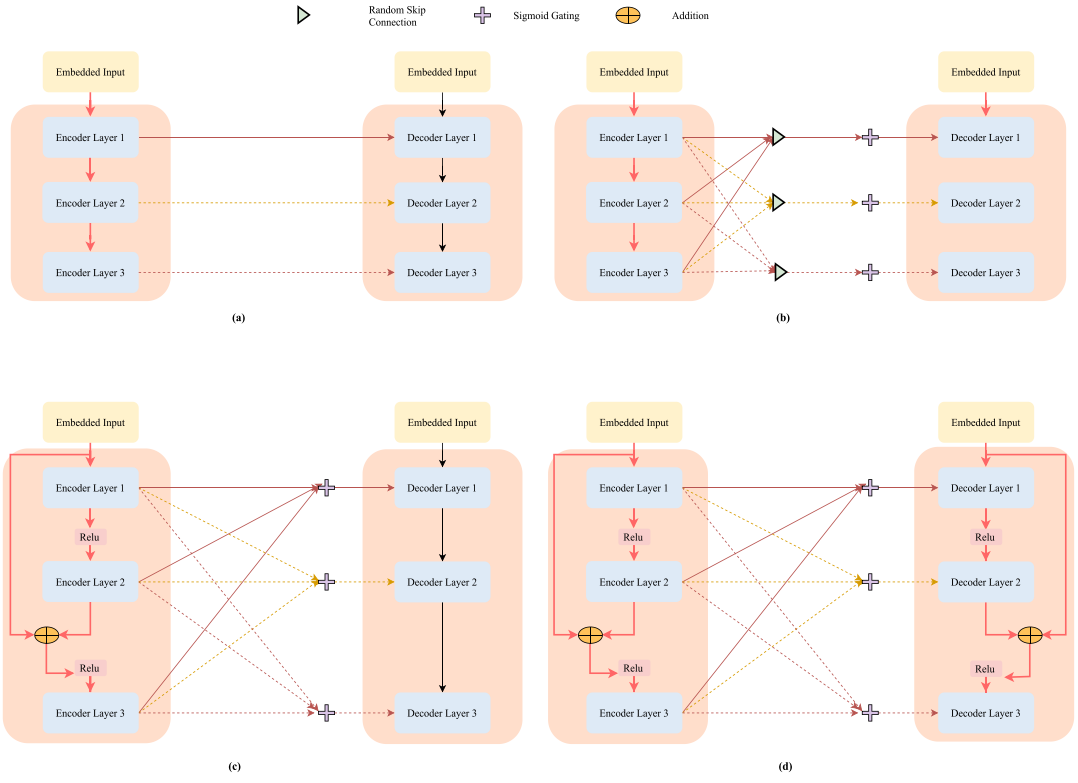

Fig. A.9. Compositions of connectivity between encoder and decoder of the transformer using 3-Layers. (a) Single-Connection. (b) Skip-Connection. (c) Residual-Connection among encoder layers with fully-connected layers. (d) Residual-connections in encoder and decoder layers with fully-connected layers.

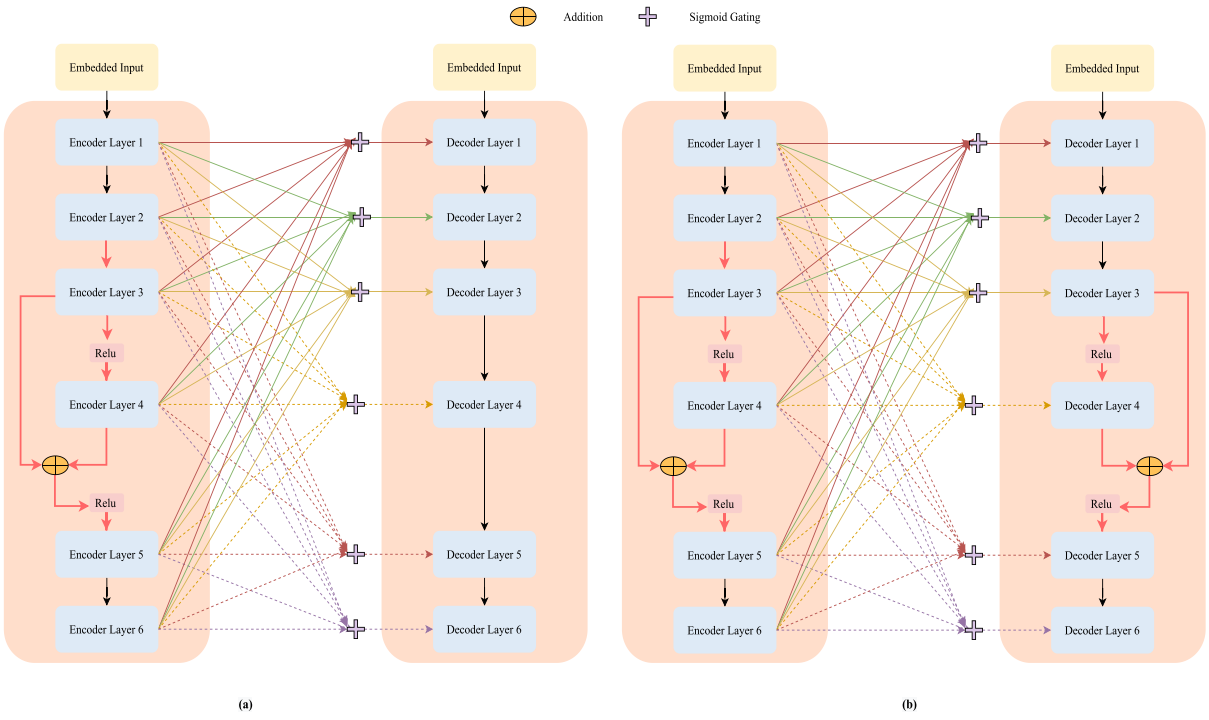

Fig. A.10. Compositions of connectivity between encoder and decoder of the transformer using 6-Layers. (a) Residual-Connection among encoder layers along with fully-connected layers. (b) Residual-connection in encoder and decoder layers with fully-connected layers.

Table A.7
LATGeO evaluation with various compositions of connectivity.

Model		BLEU-1	BLEU-4	METEOR	ROUGE-L	CIDEr-D	SPICE
Types of Layer-Connections	Layers						
Single-Connection	3	80.4	38.8	**29.2**	58.5	129.5	22.9
Skipped-Connection		80.0	38.3	29.1	58.3	128.8	**23.1**
Residual-Connection in Encoder		80.7	**39.0**	29.0	58.5	128.4	22.8
Residual-Connection in Encoder Decoder		80.2	38.8	**29.2**	58.5	129.3	**23.1**
Residual-Connection in Encoder	6	80.9	38.7	28.7	57.9	130.0	22.1
Residual-Connection in Encoder Decoder		80.6	38.4	29.0	58.2	130.6	22.5
Fully-Connected	6	80.6	38.1	29.1	58.1	129.2	22.8
Fully-Connected *LATGeO (Ours)*	3	**81.0**	38.8	**29.2**	**58.7**	**131.7**	22.9

References

[1] Peter Anderson, X. He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 6077–6086, 2018.
[2] Pengfei Cao, Zhongyi Yang, Liang Sun, Yanchun Liang, Mary Yang, Renchu Guan, Image captioning with bidirectional semantic attention-based guiding of long short-term memory, Neural Process. Lett. (2019) 1–17.
[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. ArXiv (2020), abs/2005.12872, 2020.
[4] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 6298–6306, 2017.
[5] Marcella Cornia, Matteo Stefanini, L. Baraldi, and R. Cucchiara. Meshed-memory transformer for image captioning. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 10575–10584, 2020.
[6] Bo Dai, S. Fidler, R. Urtasun, and Dahua Lin. Towards diverse and natural image descriptions via a conditional gan. IEEE International Conference on Computer Vision (ICCV), pages 2989–2998, 2017.
[7] Stéphane D'Ascoli, Hugo Touvron, Matthew L Leavitt, Ari S Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In Internation Conference on Machine Learning (ICML), 2021.
[8] Songtao Ding, Qu. Shiru, Yuling Xi, Shaohua Wan, Stimulus-driven and concept-driven analysis for image caption generation, Neurocomputing (2020).
[9] Junlong Feng, Jianping Zhao, Context-fused guidance for image captioning using sequence-level training, Comput. Intell. Neurosci. (2022).
[10] Lianli Gao, Xiangpeng Li, Jingkuan Song, Heng Tao Shen, Hierarchical lstms with adaptive attention for visual captioning, IEEE Trans. Pattern Anal. Mach. Intell. 42 (2020) 1112–1131.

[11] Jiuxiang Gu, Jianfei Cai, G. Wang, and Tsuhan Chen. Stack-captioning: Coarse-to-fine learning for image captioning. AAAI (2018), abs/1709.03376, 2018.
[12] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778, 2016.
[13] Xinwei He, Yang Yang, Baoguang Shi, Xiang Bai, Visual-densely semantic attention network for image caption generation. Neurocomputing, Vd-san, 2019.
[14] Simao Herdade, Armin Kappeler, Kofi Boakye, and Joao Soares. Image captioning: Transforming objects into words. In Advances in Neural Information Processing Systems, volume 32, 2019.
[15] Lun Huang, Wenmin Wang, J. Chen, and Xiao-Yong Wei. Attention on attention for image captioning. IEEE International Conference on Computer Vision (ICCV), pages 4633–4642, 2019.
[16] Yiqing Huang, Cong Li, Tianpeng Li, Weitao Wan, and Jiansheng Chen. Image captioning with attribute refinement. 2019 IEEE International Conference on Image Processing (ICIP), pages 1820–1824, 2019.
[17] Wenhao Jiang, Lin Ma, Yugang Jiang, W. Liu, and T. Zhang. Recurrent fusion network for image captioning. In European Conference on Computer Vision (ECCV), 2018.
[18] Yu. Jun, Li Jing, Yu Zhou, and Huang Qingming, Multimodal transformer with multi-view visual representation for image captioning, IEEE Transactions on Circuits and Systems for Video Technology, 2020.
[19] Andrej Karpathy, Fei-Fei Li, Deep visual-semantic alignments for generating image descriptions, IEEE Trans. Pattern Anal. Mach. Intell. 39 (2017) 664–676.
[20] Girish Kulkarni, Visruth Premraj, Vicente Ordonez, S. Dhar, Siming Li, Yejin Choi, A. Berg, and Tamara L. Berg. Babytalk: Understanding and generating simple image descriptions. IEEE Trans. Pattern Anal. Mach. Intell., 35:2891–2903, 2013.
[21] P. Kuznetsova, Vicente Ordonez, A. Berg, Tamara L. Berg, and Yejin Choi. Generalizing image captions for image-text parallel corpus. In ACL, 2013.
[22] Xiangyang Li, Shuqiang Jiang, Know more say less: Image captioning based on scene graphs, IEEE Trans. Multimedia 21 (2019) 2117–2130.
[23] Tsung-Yi Lin, M. Maire, Serge J. Belongie, James Hays, P. Perona, D. Ramanan, Piotr Dollár, and C.L. Zitnick. Microsoft coco: Common objects in context. In European Conference on Computer Vision (ECCV), 2014.
[24] Jiasen Lu, Caiming Xiong, Devi Parikh, and R. Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3242–3250, 2017.
[25] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Neural baby talk. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7219–7228, 2018.
[26] Sarah Pratt, Mark Yatskar, Luca Weihs, Ali Farhadi, and Aniruddha Kembhavi. Grounded situation recognition. In European Conference on Computer Vision (ECCV), 2020.
[27] Shaoqing Ren, Kaiming He, Ross B. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell., 39:1137–1149, 2015.
[28] Zhou Ren, Xiaoyu Wang, N. Zhang, Xutao Lv, and Li-Jia Li. Deep reinforcement learning-based image captioning with embedding reward. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1151–1159, 2017.
[29] Steven J. Rennie, E. Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1179–1195, 2017.
[30] Fawaz Sammani and Luke Melas-Kyriazi. Show, edit and tell: A framework for editing image captions. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
[31] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016), 2016.
[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in Neural Information Processing Systems, volume 30, 2017.
[33] A. Oriol Vinyals, Samy Bengio Toshev, D. Erhan, Show and tell: Lessons learned from the, mscoco image captioning challenge, IEEE Trans. Pattern Anal. Mach. Intell. 39 (652–663) (2015) 2017.
[34] Junbo Wang, Wei Wang, Liang Wang, Zhiyong Wang, D. Feng, and T. Tan. Learning visual relationship and context-aware attention for image captioning. Pattern Recognition (2020), 2020.
[35] Li Wang, Zechen Bai, Yonghua Zhang, Lu. Hongtao, Show, recall, and tell: Image captioning with recall mechanism, AAAI, 2020.
[36] Shiwei Wang, Long Lan, X. Zhang, and Zhigang Luo. Gatecap: Gated spatial and semantic attention model for image captioning. Multimedia Tools and Applications, 79:11531–11549, 2020.
[37] Wu. Lingxiang, Xu. Min, Jinqiao Wang, Stuart Perry, Recall what you see continually using gridlstm in image captioning, IEEE Trans. Multimedia (2018).
[38] C. Xu, Junzhong Ji, Meng long Zhang, and Xiaodan Zhang. Attention-gated lstm for image captioning. IEEE International Conference on Unmanned Systems and Artificial Intelligence (ICUSAI), pages 172–177, 2019.
[39] Shiyang Yan, Yuan Xie, Wu. Fangyu, Jeremy S. Smith, Lu. Wenjin, Bailing Zhang, Image captioning via hierarchical attention mechanism and policy gradient optimization, Signal Processing 167 (2020) 2020.
[40] Longyu Yang, Hanli Wang, Pengjie Tang, Qinyu Li, Captionnet: A tailor-made recurrent neural network for generating image descriptions, IEEE Trans. Multimedia (2021).
[41] X. Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 10677–10686, 2019.
[42] Xu. Yang, Hanwang Zhang, Jianfei Cai, Auto-encoding and distilling scene graphs for image captioning, IEEE Trans. Pattern Anal. Mach. Intell. 2020 (2020).
[43] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Exploring visual relationship for image captioning. In European Conference on Computer Vision (ECCV), 2018.
[44] Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, and Tao Mei. Boosting image captioning with attributes. IEEE International Conference on Computer Vision (ICCV), pages 4904–4912, 2017.
[45] Senmao Ye, Junwei Han, Nian Liu, Attentive linear transformation for image captioning, IEEE Trans. Image Process. (2018).
[46] Zhengjun Zha, Daqing Liu, Hanwang Zhang, Yongdong Zhang, Wu. Feng, Context-aware visual policy network for fine-grained image captioning, IEEE Trans. Pattern Anal. Mach. Intell. 2019 (2019).
[47] L. Zhang, Flood Sung, Feng Liu, T. Xiang, S. Gong, Yongxin Yang, and Timothy M. Hospedales. Actor-critic sequence training for image captioning. Neural Information Processing Systems (NIPS), 2017.
[48] Zongjian Zhang, Wu. Qiang, Yang Wang, F. Chen, High-quality image captioning with fine-grained and semantic-guided visual attention, IEEE Trans. Multimedia 21 (2019) 1681–1693.
[49] Zongjian Zhang, Wu. Qiang, Yang Wang, Fang Chen, Exploring region relationships implicitly: Image captioning with visual relationship attention, Image Vis. Comput. 109 (2021) 104146.
[50] Dexin Zhao, Zhi Chang, Shutao Guo, A multimodal fusion approach for image captioning, Neurocomputing 329 (2019) 476–485.