Look Twice as Much as You Say: Scene Graph Contrastive Learning for Self-Supervised Image Caption Generation

Chunhui Zhang
Brandeis University
Waltham, Massachusetts, USA
chunhuizhang@brandeis.edu

Chao Huang
University of Hong Kong
Hong Kong, China
chaohuang75@gmail.com

Youhuan Li
Hunan University
Changsha, Hunan, China
liyouhuan@hnu.edu.cn

Xiangliang Zhang
University of Notre Dame
South Bend, Indiana, USA
xzhang33@nd.edu

Yanfang Ye
University of Notre Dame
South Bend, Indiana, USA
yye7@nd.edu

Chuxu Zhang*
Brandeis University
Waltham, Massachusetts, USA
chuxuzhang@brandeis.edu

ABSTRACT

Images are commonly used for various information and knowledge applications, such as advertising and recommendation. Automating image caption generation will significantly improve image accessibility. This cross-modal task, which takes image as input and text as output, however, is difficult for learning. Though prior methods achieve good performance for image caption generation, they rely on either supervised learning which requires sufficient labeled data or unsupervised learning which needs external dataset as language pivot. In this paper, we propose SGCL, a novel **S**cene **G**raph **C**ontrastive **L**earning model for self-supervised image caption generation. SGCL adopts the pre-training and fine-tuning pipeline. Specifically, we first apply scene graph generation and objection detection method to encode scene graph and visual information in the image as feature representation. Later, a decoder network based on graph attention network and recurrent neural network is further designed to generate sequential text as caption. To enable contrastive learning in SGCL, we design scene graph augmentations as contrastive views of images and train the model effectively without ground-truth labels through contrastive learning. Additionally, we introduce the pre-trained word embedding and the context projector to enrich the text representation in the decoder network, which benefits model pre-training. Once the pre-training phase is finished, we further fine-tune the model for the image caption generation task with limited labeled data. Extensive experiments on benchmark dataset demonstrate that SGCL outperforms state-of-the-art models (both supervised and unsupervised).

CCS CONCEPTS

• **Computing methodologies → Neural networks**; • **Information systems → Web applications**; **Social networks**.

KEYWORDS

Graph contrastive learning, image caption generation

ACM Reference Format:
Chunhui Zhang, Chao Huang, Youhuan Li, Xiangliang Zhang, Yanfang Ye, and Chuxu Zhang. 2022. Look Twice as Much as You Say: Scene Graph Contrastive Learning for Self-Supervised Image Caption Generation. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3511808.3557382

1 INTRODUCTION

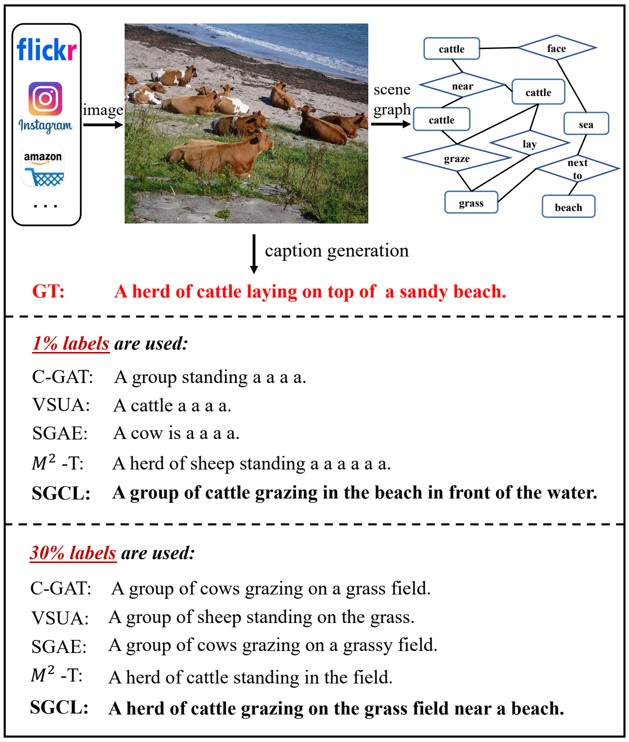
Image, as a common form to present information, is useful for various data science applications such as advertising, recommendation, and searching. Caption summarizes the information inside an image with a piece of text. Thus, automating accurate image caption generation can significantly improve image accessibility and help users to grasp the information quickly [30, 34]. However, this non-trivial task, which spans both image and text for cross-modal representation learning, is challenging to be solved. In recent years, image caption generation models have achieved substantial developments. Some of them reach or surpass humans' performance in a few evaluation metrics and generate natural language-like text [23, 50]. These methods are mainly trained under labels' supervision and require a large amount of exactly labeled image caption data.

Recently, contrastive learning has been proposed to mitigate label-hungry issue for vision, language, and graph [6, 15, 31, 46]. In general, it aims to pre-train a model in self-supervised manner and further fine-tune the pre-trained model on downstream task with limited labels. A common idea is to maximize the similarity between representations from different augmented views of the same input data in the latent space [46]. In the pre-training phase, the unsupervised pretext task leverages the model backbone (e.g., neural network) to learn discriminate representations. Later in the fine-tuning phase, the pre-trained backbone is further trained on a small fraction of labeled data for downstream task (e.g., ImageNet classification) and may outperform some supervised models [36].

Considering the success of contrastive learning, a natural idea is to leverage it to image caption generation for alleviating the dependence on a large amount of well-collected image-description pairs. However, there are two special characteristics of image caption generation that make the implementation of this idea challenging:

[*] Corresponding author.

CIKM '22, October 17–21, 2022, Atlanta, GA, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557382



GT: A herd of cattle laying on top of a sandy beach.

1% labels are used:

C-GAT: A group standing a a a a.
VSUA: A cattle a a a a.
SGAE: A cow is a a a a.
M^2 -T: A herd of sheep standing a a a a a a.
SGCL: A group of cattle grazing in the beach in front of the water.

30% labels are used:

C-GAT: A group of cows grazing on a grass field.
VSUA: A group of sheep standing on the grass.
SGAE: A group of cows grazing on a grassy field.
M^2 -T: A herd of cattle standing in the field.
SGCL: A herd of cattle grazing on the grass field near a beach.

Figure 1: Example of image caption generation results of the proposed model SGCL and baseline models with limited labeled data (GT: ground-truth).

(1) **Cross-modal data** - Unlike previous studies working on single-modal data (e.g., image, text, or graph), image caption generation is a cross-modal task on the intersection of image and text; (2) **Complex task** - Image caption generation is a complex task that has to generate new content rather than simple classification or prediction task studied in previous work. For the first characteristic, conventional image caption generation models require text ground-truth labels to supervise model for transferring the representation from image to language in latent space. Without ground-truth labels' supervision in model training, it is still doubtful whether contrastive methods can make learned representation from given visual scene input be effective to generate an acceptable describing sentence. For the second characteristic, image captioning model has to decode and generate sentence. When training a language decoder, teacher forcing uses ground-truth sentence as input for the decoder of the language model. It relies on neural machine translation (NMT) and guides language decoder model to converge smoothly and generate sentence similar to ground-truth text. Thus, it is essential to design language decoders that can be trained without the supervision of NMT and ground-truth text for the non-traditional self-supervised image caption generation problem.

Our Contributions. To address the above challenges, in this paper, we propose a novel model named SGCL for **S**cene **G**raph **C**ontrastive **L**earning to generate image caption in self-supervised manner. SGCL aims to incorporate objects and relations in the image into caption generation model for better performance. Specifically, we leverage scene graph to abstract objects and relations information in the image, and apply object detection model to encode the image representation. The encoded representation is further decoded to sequential text (caption) through graph attention network and recurrent neural network. The model follows pre-training and fine-tuning pipeline. In the pre-training phase, we design scene graph augmentations as the contrastive views of images and train the model effectively without teacher forcing or language ground-truth labels. Besides, we introduce GloVe pre-trained word embedding layer [29] to enrich the text representation in the decoder, and design a context projector to summarize the sentence representation and enlarge the queue of negative keys for contrastive learning. Additionally, big dropout rate is adopted in the output sentence layer to benefit model training. After pre-training, the model is further fine-tuned to the image caption generation task using a small amount of labeled data (fine-tuning phase). We conduct extensive experiments on the benchmark dataset to show the superior performance of SGCL over state-of-the-art models (both supervised learning and unsupervised learning based). For example, Figure 1 shows better image caption generation results of SGCL with the comparison to some representative baseline methods. The improvement is significant when labeled data is limited (1%). To the best of our knowledge, this is the first work to study self-supervised image caption generation, which is novel and significant.

2 RELATED WORK

This work is closely related to two research lines: image caption generation and graph contrastive learning.

2.1 Image Caption Generation

Many works have been proposed for image caption generation. The pipeline usually includes two steps: in the first step, the object detector encodes image into feature representation. Then in the second step, the caption generation model uses feature as input and decodes it to output sentence. In the early period, Show-Tell model [40] encodes the image into vector representation using CNN and generates sentence description using RNN, respectively. Recently, the bottom-up attention mechanism has been widely used to generate a caption for the image, such as BUTD, C-GAT [2, 27]. The series of bottom-up methods allow the model to predict an abundant and plentiful set of detected objects, including both items and contextual regions and enable model to learn better feature representation of image for caption generation. Meanwhile, the graph is used to close the gap between image representation and language representation because of its explicit modeling of object entities, visual relationships, and attributes [44]. The scene graph works as input for graph neural network, and then is exploited at the decoding stage to generate caption sentence. Also, the concept of scene graph is extended on caption. CGVRG [33] leverages a semantic relationship graph on the image to improve caption generation. Different from previous studies that require a large amount of labeled data, our proposed SGCL applies self-supervised learning to alleviate the heavy demand of image-text labeled pairs.

2.2 Graph Contrastive Learning

Self-supervised contrastive learning has been one of the most competitive methods for representation learning in image [4, 15], text [10, 43], and graph [31, 38, 41, 46, 47, 49]. In graph contrastive learning, one basic idea is to minimize the distance of the different views' representation vectors from the same sample and maximize that value from the different samples [31]. For example, DGI [38] maximizes mutual information between patch representations and corresponding high-level summaries of graphs. GCC [31] constructs subgraph samples by random walk as contrastive samples of graphs. Besides, some research focus on designing effective data augmentation such as node (edge) deletion (addition) to construct contrastive pairs for self-supervised learning on graph data such as social networks and biochemical molecules [46]. JOAO [45] proposes an automatic and flexible method of selecting data augmentations to learn better representation for graphs. SelfLinKG [24] applies graph contrastive method to link concepts in massive heterogeneous graphs. A recent work finds that contrastive encodings from first-order neighbors from graph-specific perspective can achieve SOTA performance [14]. In this work, we generalize graph contrastive learning on scene graph of image instead of being limited in social or biochemical graph.

3 PRELIMINARY

Before presenting SGCL, we first introduce preliminary techniques related to our model design.

3.1 Object Detection

Given an image, the object detection task is to locate and label all objects on it. Faster R-CNN [32] is a widely used model for object detection. The procedure of Faster R-CNN can be divided into two steps: for an input image, firstly, the region proposal network generates regions of interest that scroll over the image's feature map to detect potential objects and box up objects with bounding boxes. Secondly, the object detection model extracts feature vectors for proposal regions with pooling operation. The objective function of Faster R-CNN is defined as follows:

$$L(p_i, t_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*), \quad (1)$$

where p_i is the predicted probability of i-th object anchor to be an object, p_i^* is binary ground-truth label to indicate whether the object anchor is positive. Besides, t_i is a vector which includes four coordinate values of the predicted bounding box of an object and t_i^* is the ground-truth bounding box of the object. L_{cls} is classification loss to justify whether the anchor is an object or not while L_{reg} is regression loss of bounding box of an object and it is activated for positive anchors. The extracted feature of an object is used as attribute of a node in the scene graph. In our model, we will apply a pre-trained Faster R-CNN to detect objects in each image, and then store feature maps of objects for later use.

3.2 Scene Graph Generation

Scene graph includes a graph structure for explicitly annotating an image with its items and existing relations between items [18]. We consider scene graph $\mathcal{G}$ of an image with two types of nodes: objects and relations. Object nodes interact with each other through relation nodes (see Figure 2). All edges in scene graph are between object node and relation node. Additionally, object and relation nodes are associated with object and relation labels, respectively. Formally, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ denote the sets of node and edge, respectively. Many studies have been proposed to generate scene graph. In this study, we will apply the Iterative Message Passing model (IMP) [42] to generate scene graph. Specifically, IMP has two sets of features as input: image feature $\mathcal{X}$ and object proposal B. It aims to generate optimal object classification x_i^{cls}, object proposals x_i^{bbox}, and relationship $x_{i\to j}$ between object i and object j. In formal, the goal is:

$$(x_i^{cls}, x_i^{bbox}, x_{i\to j})^* = \arg\max_{\mathbf{x}} \Pr(\{x_i^{cls}, x_i^{bbox}, x_{i\to j}\}_{i,j\in V} | \mathcal{X}, B), \quad (2)$$

where V is the node set of scene graph, $\Pr(\cdot)$ is formulated as:

$$\Pr(\{x_i^{cls}, x_i^{bbox}, x_{i\to j}\} | \mathcal{X}, B) = \prod_{i\in V} \prod_{j\neq i} \Pr(x_i^{cls}, x_i^{bbox}, x_{i\to j} | \mathcal{X}, B). \quad (3)$$

IMP first uses Faster R-CNN to obtain initial object proposals x_i^{bbox} and visual features of objects x_i^{cls} from $\mathcal{X}$. Then, the first-layer node GRU and edge GRU take the object proposals' visual features and the union-box features as the node and edge features, respectively. Next, the message pooling module processes and fuses both object features and relationship features. The final GRU unit predicts the scene graph's components: object categories and relationship types.

3.3 Graph Attention Network

Graph neural networks (GNNs) [13, 17, 21, 35, 39, 48] are powerful tools for modeling graph data such as social network, biological molecules, and recommendation systems. Specifically, given a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ and $\mathcal{E}$ denote node set and edge set, respectively. GNN is a learnable mapping function $f : v \rightarrow \mathcal{R}^d$ that maps each node $v \in \mathcal{V}$ into a d-dimensional latent representation space. Representations of nodes are updated by using messages from neighboring nodes which include all nodes connected by the current node's edges. In particular, Graph Attention Network (GAT) [39] implicitly defines neighboring nodes' weight factors by a self-attention mechanism over node features. To be more specific, let α_{ij} denote the importance of node j on node i. First, the intermediate weight coefficients w_{ij} is computed as follows:

$$w_{ij} = \mathrm{f}(\mathbf{W}\mathbf{e}_i \parallel \mathbf{W}\mathbf{e}_j), \quad (4)$$

where $\mathbf{e}$ is node embedding, $\mathbf{W}$ is parameter matrix, $\parallel$ is concatenation operator, and f is a single-layer feed forward neural network. Then, the *softmax* function is utilized to normalize w_{ij} across all neighboring nodes:

$$\alpha_{ij} = \frac{\exp(\text{Leaky\_ReLU}(w_{ij}))}{\sum_{k\in\mathcal{N}(i)} \exp(\text{Leaky\_ReLU}(w_{ik}))}, \quad (5)$$

where $\mathcal{N}(i)$ is the neighbor set of node i. With attention coefficient α_{ij}, the node feature is updated by aggregating neighboring nodes:

$$\mathbf{e}_i = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{j\in\mathcal{N}(i)} \alpha_{ij}^k \mathbf{W}^k \mathbf{e}_j\right), \quad (6)$$

where $\sigma(\cdot)$ is an activation function, $\parallel$ denotes the concatenation of multi-head attentions, K is the number of attention heads. In this work, we will employ GAT as one key component in SGCL for modeling scene graph information of image.

4 THE PROPOSED MODEL

In this section, we present details of the proposed SGCL (Figure 2). First, we present the architecture of SGCL. Then, we introduce scene graph augmentation strategies for contrastive learning. Finally, we describe additional designs to refine the model.

4.1 Scene Graph Contrastive Learning

We develop contrastive learning-based method to pre-train the model by the following three steps: graph and feature generation, caption generation, contrastive learning.

4.1.1 *Graph and Feature Generation.*

At first, we propose to generate both scene graph and object feature of an image as the model input. Specifically, we apply an object detection model Faster-RCNN [32] and a scene graph generation model Iterative Message Passing [42] to generate object features $\mathcal{X}$ and scene graph $\mathcal{G}$, respectively: (1) Faster-RCNN first predicts object proposals, and then extracts small feature maps for object proposals. Finally, a MLP layer is used to classify each feature map for each object proposal; (2) Iterative Message Passing not only detects objects for the input image like Faster-RCNN but also predicts the relationship between detected objects by edge GRUs. It outputs a scene graph for the image, where nodes are detected objects and edges are predicted relationships of objects. Given $\mathcal{G}$ and $\mathcal{X}$, we combine them as the input pair $(\mathcal{G}, \mathcal{X})$ of an image. Additionally, each input pair is processed with randomly data augmentation (including both graph augmentation on $\mathcal{G}$ and feature augmentation on $\mathcal{X}$) to generate contrastive pairs $(\mathcal{G}_q, \mathcal{X}_q)$ and $(\mathcal{G}_k, \mathcal{X}_k)$ as the input for two branches (query branch and key branch) of the following contrastive learning step. The detail of data augmentation is described in Section 4.2.

4.1.2 *Caption Generation.*

Next, based on the generated data pairs in the first step, we design a graph neural network and recurrent neural network-based decoder network for caption generation. Specifically, the query data $(\mathcal{G}_q, \mathcal{X}_q)$ (or key data $(\mathcal{G}_k, \mathcal{X}_k)$) is processed by a query decoder network $f_q(\cdot)$ (or a key decoder network $f_k(\cdot)$). As shown in Figure 2, the two branches have the same neural network architecture. The decoder network takes augmented data, e.g., $(\mathcal{G}_q, \mathcal{X}_q)$, as input and outputs sentence logits:

$$\mathbf{S}_q = f_q(\mathcal{G}_q, \mathcal{X}_q), \tag{7}$$

where f_q consists of two LSTMs [16] and a graph attention network [39]. The first LSTM is designed to compute attention over the image, which works as a top-down attention layer $\mathrm{LSTM}_1(\cdot)$ and the second LSTM works as a language model $\mathrm{LSTM}_2(\cdot)$ to predict tokens of image caption. Specifically, the first LSTM layer is formulated as:

$$\mathbf{h}_1^t = \mathrm{LSTM}_1([\mathbf{h}_2^{t-1}; \mathbf{w}^{t-1}; \bar{\mathcal{X}}; \bar{\mathcal{G}}], \mathbf{h}_1^{t-1}), \tag{8}$$

where $\mathbf{h}_2^{t-1}$ is the language model's (the second LSTM's) output state at the last $t{-}1_{th}$ step, $\mathbf{w}^{t-1}$ is the pre-trained GloVe embedding of word predicted in the previous step (see Section 4.3.1 for detail), $\bar{\mathcal{X}}$ is the average object feature vector over $\mathcal{X}$ and $\bar{\mathcal{G}}$ is the average graph feature vector over $\mathcal{G}$. Then, $\mathrm{LSTM}_1(\cdot)$'s output state $\mathbf{h}_1^t$ and object feature $\mathcal{X}$ are processed by a hierarchical attention strategy:

$$\mathbf{x}^t = \mathrm{Att}_{\mathcal{X}}(\mathcal{X}, \mathbf{h}_1^t), \tag{9}$$

where $\mathrm{Att}_{\mathcal{X}}$ is a MLP layer as the first part of hierarchical attention, $\mathbf{x}^t$ is concatenated with $\mathbf{h}_1^t$ to form *condition context*. Furthermore, we take *condition context* as an additional input of graph attention network (GAT):

$$\mathbf{g}^t = \mathrm{GAT}([\mathcal{G}, [\mathbf{h}_1^t; \mathbf{x}^t]), \tag{10}$$

where $\mathbf{g}^t$ is the generated conditional graph embedding. Different from traditional GAT, we consider both neighbors' features and external condition context $\mathbf{h}_1^t$ which is the output state of $\mathrm{LSTM}_1(\cdot)$ to update node embedding in the scene graph. Afterwards, the hidden state $\mathbf{h}_1^t$ is used again to derive an attention weighted embedding $\mathcal{G}^t$ over the graph embedding $\mathbf{g}^t$:

$$\mathcal{G}^t = \mathrm{Att}_{\mathcal{G}}(\mathbf{g}^t, [\mathbf{h}_1^t; \mathbf{x}^t]), \tag{11}$$

where $\mathrm{Att}_{\mathcal{G}}$ is another MLP layer as the second part of hierarchical attention. Next, we take $\mathcal{G}^t$, $\mathbf{h}_1^t$, as well as $\mathbf{x}^t$ as the input of language model and formulate the second LSTM as follows:

$$\mathbf{h}_2^t = \mathrm{LSTM}_2([\mathbf{h}_1^t; \mathbf{x}^t; \mathcal{G}^t], \mathbf{h}_2^{t-1}). \tag{12}$$

The output hidden state $\mathbf{h}_2^t$ is used to generate word logit at step t:

$$\mathbf{S}_q^t = \mathrm{MLP}\left\{\mathrm{Dropout}(\mathbf{h}_2^t)\right\}, \tag{13}$$

where the output layer is a combination of Dropout operation and a MLP layer. The dropout rate is set to a large value to strengthen scene graph augmentation (see Section 4.3.3 for detail). With the generated sentence logits $\mathbf{S}_q$, we further design a context projector to obtain final embedding vector of query data sample (or key data sample) as follows:

$$\mathbf{e}_q = \varphi(\mathbf{S}_q), \tag{14}$$

where context projector φ consists of a mean pooling layer over the whole sentence followed by 2 layers of fully connected network (FC). Section 4.3.2 will provide more details.

4.1.3 *Contrastive Learning.*

Finally, we leverage the generated embedding vector $\mathbf{e}$ of each data sample to formulate the contrastive learning objective. We follow the general procedure in MoCo [15]. Specifically, given a query sample q, a positive sample k_+ and a set of negative samples $\{k_1, k_2, ..., k_K\}$ (key dictionary queue) for q, the similarity between query and key is measured as dot product of two embeddings: $sim(q, k) = \mathbf{e}_q{\cdot}\mathbf{e}_k$. Then, the InfoNCE loss is used to model similarity between each query-key pair:

$$\mathcal{L}_{InfoNCE} = -\log \frac{\exp(sim(\mathbf{q}, \mathbf{k}_+)/\tau)}{\sum_{i=0}^{K} \exp(sim(\mathbf{q}, \mathbf{k}_i)/\tau)}, \tag{15}$$

where τ is a temperature hyper-parameter, K is the number of negative samples which is the length of the dictionary queue (Note: $\mathbf{k}_0 = \mathbf{k}_+$). Let Θ_q and Θ_k denote the set of neural network parameters in key branch and query branch, respectively. The Adam optimizer [20] is used to update the query network parameters Θ_q while the key network parameters Θ_k are updated as follows:

$$\Theta_k \leftarrow m\Theta_k + (1 - m)\Theta_q, \tag{16}$$

where m is the momentum hyper-parameter. By this strategy, the most outdated samples are gradually replaced by the new samples.

Once the contrastive learning-based pre-training phase is finished, we further fine-tune the pre-trained model to image captioning task with a small number of labeled image-sentence pairs.

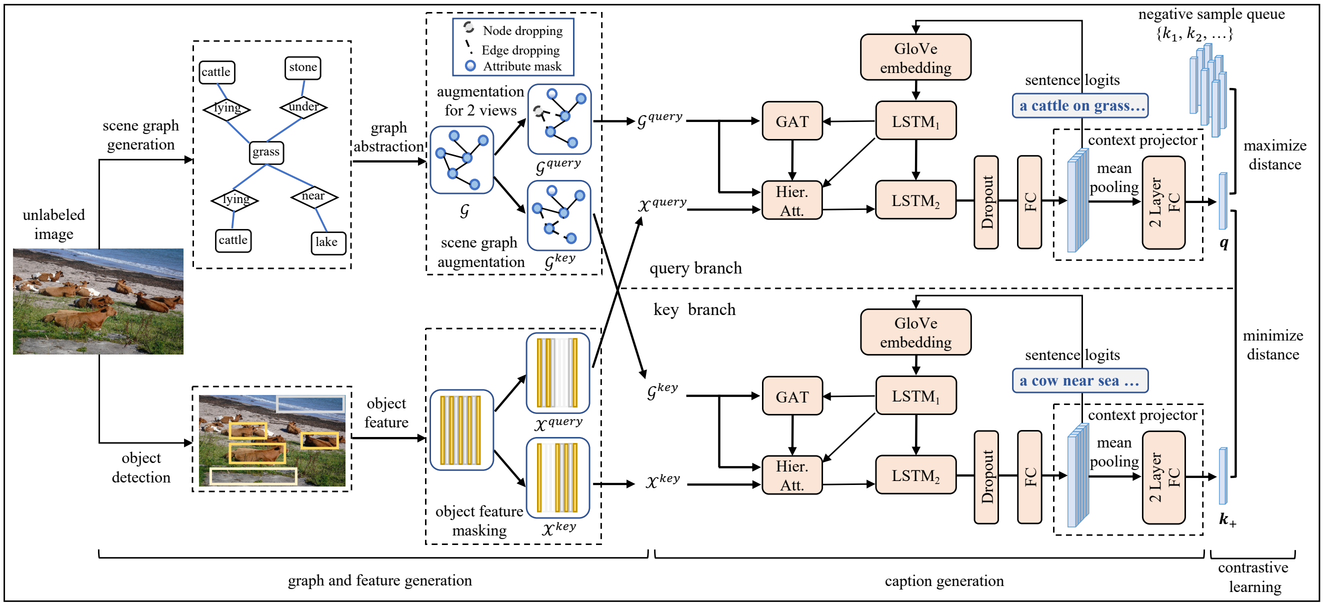


Figure 2: The overall framework of SGCL. First, SGCL generates scene graph and object features from an image. Then, SGCL develops two decoder networks (query branch and key branch) based on graph attention network (GAT) and recurrent neural network (LSTM) for generating embeddings of query sample and key sample. Finally, SGCL applies contrastive objective between query-key pair to optimize the model.

4.2 Scene Graph Augmentation

Scene graph contrastive learning requires sufficient data samples to fully pre-train the model. Inspired by the recent success of graph data augmentation [46], we propose to generate scene graph augmentations. Specifically, given the scene graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we consider following strategies for graph augmentation.

- **Node dropping.** It will randomly drop some nodes and its connected edges in $\mathcal{G}$. Our basic assumption is that dropping some nodes will not largely affect the semantics of the original image corresponding to the given scene graph.
- **Edge dropping.** It will perturb the connectivity of $\mathcal{G}$ by randomly removing a certain percentage of edges. Similar to node dropping, we assume that when some edges are discarded, the scene graph loses part of its relational representation while the semantics of the corresponding image is not overly altered.
- **Attribute masking.** It masks off some attributes of edge or node in $\mathcal{G}$. The basic assumption is that the partial masking of some node or edge attributes does not significantly impact the prediction of the model. The remaining unmasked areas and the contextual information provided by the surrounding nodes and edges can recover the attributes of the masked nodes or edges, and assist in learning the semantics of the corresponding image.
- **Object feature masking.** Besides data augmentation on the scene graph, we also consider data augmentation on the features of detected objects of the image. Particularly, we randomly mask some portion (e.g., 20%) of object features as augmentation. We assume that remaining unmasked regions of image features can provide enough representation in learning semantics of image.

The above data augmentation strategies are jointly used to generate sufficient data samples for contrastive learning in SGCL.

4.3 Additional Designs

Based on overall contrastive learning framework, we also introduce additional designs to further refine the model.

4.3.1 Pretrained Word Embedding Freezing. Existing image caption models often use teacher forcing to supervise the sentence generation. However, in SGCL, we are supposed to avoid teacher forcing and not use any labeled samples to train the model. During SGCL self-supervised training, we only use the previous step's output word rather than a ground-truth word as the current step's input, like a normal auto-regression model's inference period. Without teacher forcing, we design an alternative way by using the pre-trained word embedding to enrich language representation for self-supervised training. Specifically, we use GloVe word embedding [29] as the input of decoder network (LSTM module in Figure 2) in SGCL. Additionally, we freeze the word embedding layer during model pre-training phase. In model fine-tuning phase, we defrost the word embedding layer and optimize it with few labeled data.

4.3.2 Context Projector. In standard image caption generation model, the output sentence logits is a 3D tensor with size of `[batch_size, sentence_length, word_map_len]`. However, the size of that sentence logits is too big to store negative keys in the dictionary queue. In other words, the large word map length (totally 9490 words in this model) requires large storage of sentence logits in contrastive learning with a large number of negative samples, making SGCL infeasible when only limited memory space is available. To address this issue, we design a context projector to summarize output sentence embedding and reduce storage demand. Specifically, given the sentence logits $\mathbf{S}$, the projector is formulated as follows:

$$\mathbf{S}_p = \text{MLP}\left\{\text{MeanPool}(\mathbf{S})\right\}, \tag{17}$$

where MeanPool is mean pooling operation over `sentence_length` dimension and MLP is a two-layer FC with ReLU activation. According to Eq. 17, the projector first uses average pooling to summarize sentence embedding, and then applies neural network to generate compact embedding with context information. Besides, the projector saves memory and makes the storage of a large queue of negative samples feasible.

4.3.3 Big Dropout Rate. The dropout operation casually introduces the noise into the sentence representation when positive pairs are from the same scene graph. A very recent study [10] has discussed the role of dropout in contrastive sentence embedding. Thus, we are motivated to introduce a big dropout rate (e.g., 50%) in the output layer of SGCL (before the context projector layer) in order to learn better sentence embedding.

5 EXPERIMENTS

In this section, we conduct extensive experiments on the benchmark dataset to evaluate the model performance by comparison with state-of-the-art baseline models. We first describe experimental settings, and then discuss the performance comparison of different methods and effectiveness of each component/design in the proposed model. At last, several real case studies are provided.

5.1 Experimental Settings

5.1.1 Dataset. In this work, we employ the most widely used benchmark dataset **MSCOCO** [5] for the following experiments. The dataset is collected from web by searching images of 80 object categories with diverse scene types on the Flickr website. Specifically, the images were gathered by searching for pairs of 80 object categories and various scene types. The goal of the MSCOCO image collection process was to gather images containing multiple objects in their natural context. Given the visual complexity of most images in the dataset, they pose challenge for image caption generation. In total, the dataset includes 122,585 images. Each image has corresponding 5 human-annotated captions as ground-truth. In addition, we apply *karpathy split* [19] (a widely employed split method in image caption generation) to split the training set (112,585 images), validation set (5,000 images), and test set (5,000 images).

5.1.2 Evaluation Metrics and Baseline Methods. To evaluate the performances of our model and baseline methods, we adopt five widely used metrics: BLEU-k [28] (k = 1, 2, 3, 4), CIDEr [37], METEOR [8], ROUGE-L [22], and SPICE [1]. Besides, we compare our model with nine recent baselines including both supervised models and unsupervised models. Supervised baseline models are introduced as following:

- **Dlc Transformer** [26] improves transformer-based image caption generation model by adopting dual way self-attention as attention modules.
- **ASGC** [3] uses user intention to generate more detailed and fine-grained captions.
- M^2 **Transformer** [7] uses a mesh-like decoder to exploit memorized low-level and high-level representations at the same time to generate more accurate descriptions.
- **SGAE** [44] stores inductive bias from human language as background knowledge to improve the amount of information in the generated captions.
- **VSUA** [12] designs attention model to generate descriptions for input images. Specifically, the training object is maximizing a variational lower bound.
- **C-GAT** [27] proposes a conditional graph attention network to process scene graph generated from image then boost the caption generation performance.
- **Sub-GC** [50] decomposes scene graph into several sub-graphs which can capture richer semantic information to benefit caption generation results.
- **BUTD** [2] applies Bottom-Up and Top-Down Attention to gradually get fine-grained representations and comprehensively locate all object regions.
- **Add-Att** [25] proposes Visual Sentinel, which catches the most informative objects to generate image caption efficiently.

Differently, unsupervised models are outlined as following:

- **UIC** [9] collects external Shutterstock dataset which uses image-language pairs to augment cross-modal representation.
- **I2t+nmt** [11] trains image Chinese caption generation model with external labeled image Chinese caption dataset, then trains a neural machine translation model with external Chinese-English dataset as a pivoting module. This pivoting module translates the Chinese captions to English captions.

For all baseline methods, we follow the default settings provided by the authors' open source codes of the original papers.

5.1.3 Reproducibility Settings. We train SGCL on unlabeled dataset for 80 epochs, and then fine-tune the model with different label fractions (1%, 5%, 10%, 15%, 20%, 25% and 30% labels) for 50 epochs. For data augmentation over object features, we set the random mask rate as 20%. For data augmentation over scene graph, the rate of node drop is 15%, the rate of edge drop is 15%, and the rate of attribute masking is 20%. In contrastive learning, the parameter τ of loss function $\mathcal{L}_{InfoNCE}$ is set to 0.07. The momentum value equals 0.99. The size of negative samples queue is 131072. The embedding size is set to 512. We implement SGCL by PyTorch. The mini-batch size is set to 512 and learning rate is set to 0.07 with decay factor = 0.9. We choose Adam [20] as the optimizer.

5.2 Performance Comparison

The performances of all models with different label percentages (from 1% to 30%) are shown in Figure 3. According to this figure, we can obtain several findings: (1) SGCL outperforms all baseline methods for all metrics and all used label percentages, demonstrating the superiority of our model for image caption generation; (2) The improvement of SGCL over supervised learning models (M^2 Transformer, SGAE, VSUA, C-GAT, Sub-GC, BUTD, and Add-Att) and unsupervised learning models (UIC and i2t-nmt) ranges from 4.6% to 102% and 2.8% to 538.9%, respectively. The improvement is more significant when limited labels are available (1%). It demonstrates the capability of contrastive learning in learning rich text representation from massive unlabeled images and the effectiveness of self-supervised pre-training in eliminating label-hungry issues for image caption generation; (3) unsupervised learning baseline (UIC) achieve better performance than supervised learning baselines when label percentage is small (1%) while it is much worse than those methods when label percentage increases. It indicates

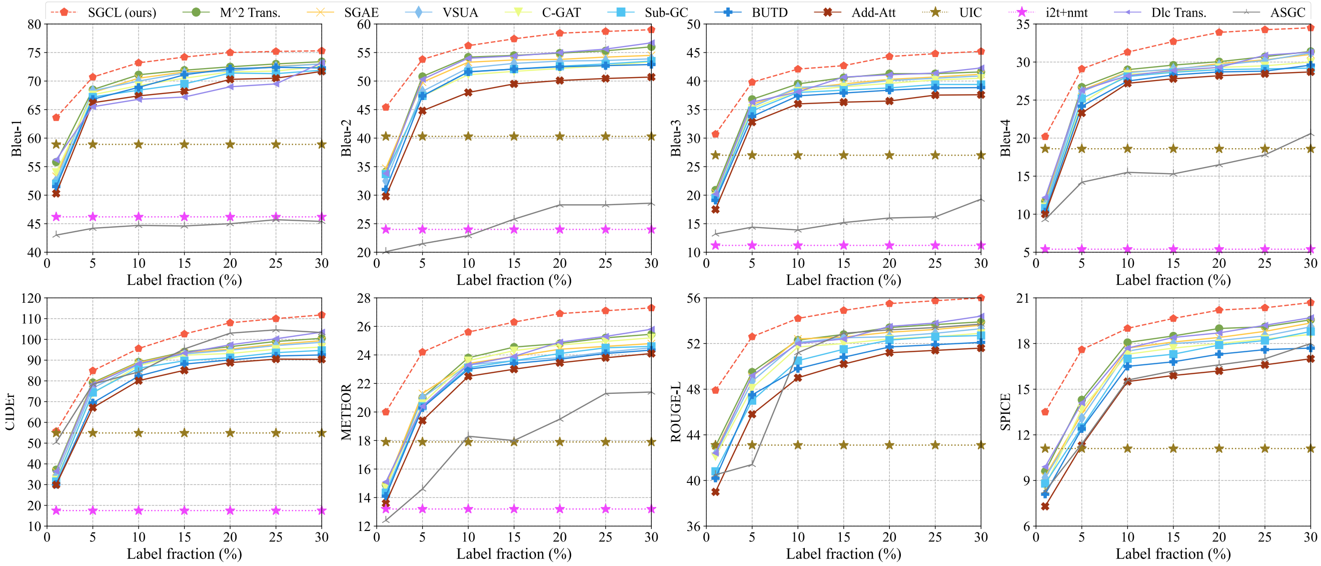


Figure 3: Performances of all models with limited labels (Note that ROUGE-L and SPICE of i2t+nmt are not shown due to missing values in the original work).

that existing supervised learning models heavily rely on labeled information of image caption and increasing labeled data can significantly improve their performances.

5.3 Ablation Study

Our model contains several essential components/designs such as graph augmentation and context projector. In order to analyze the contributions of different components/designs, we conduct several ablation studies by considering each of them independently. In the following result tables, we denote BLEU-*k*, CIDEr, METEOR, ROUGE-L, SPICE as B-*k*, C., M., R.-L, S. for short, respectively. Note that we only show model results with 1% and 30% training labels due to space limitations.

5.3.1 Scene Graph Augmentation. As one of the key procedures in scene graph contrastive learning, graph augmentation plays important role in affecting final performance. Here we study the impact of scene graph augmentation. Specifically, we conduct experiments to evaluate the model performance with different augmentation strategies. We consider three graph augmentations - node dropping (N), edge dropping (E), node attribute masking (A), one object feature augmentation - object feature masking (O), and their combination to train the self-supervised model and report their performances in Table 1, where the best results are highlighted in bold. According to this table, it is easy to find that the combination of four augmentations works better than a single augmentation method or the combination of a graph augmentation and object feature augmentation. The model without any data augmentation has the worst performance. It demonstrates that self-supervised image caption generation requires various data augmentations to generate sufficient contrastive pairs for model pre-training. Besides, we can see that node attribute masking contributes most to graph augmentation as the model with this augmentation strategy achieve better performance than other two graph augmentations for most metrics.

Table 1: Performances of different model variants with various graph augmentation strategies (Note: N - node dropping, E - edge dropping, A - node attribute masking, O - object feature masking).

Label	N	E	A	O	B-1	B-2	B-3	B-4	C.	M.	R.-L	S.
1%					61.8	43.8	28.9	18.2	47.7	18.5	46.3	11.9
				✓	62.5	44.6	30.0	19.1	49.2	19.9	47.0	13.1
	✓			✓	62.5	44.5	29.0	18.5	52.9	19.1	47.3	13.2
		✓		✓	63.1	44.3	28.8	18.6	52.2	19.3	47.2	13.0
			✓	✓	63.0	45.1	29.9	19.3	53.3	19.6	47.5	13.3
	✓	✓	✓	✓	**63.6**	**45.4**	**30.7**	**20.2**	**55.0**	**20.0**	**47.9**	**13.5**
5%					69.4	51.7	36.6	26.2	75.9	22.2	49.4	16.3
				✓	70.3	53.0	38.6	27.9	79.4	23.9	51.9	17.3
	✓			✓	62.5	52.8	38.9	28.5	81.4	19.1	51.9	17.2
		✓		✓	63.1	53.5	38.7	28.6	82.2	19.3	52.0	17.1
			✓	✓	70.3	53.3	39.2	28.1	82.3	24.1	52.2	17.4
	✓	✓	✓	✓	**70.7**	**53.8**	**39.8**	**29.1**	**84.9**	**24.2**	**52.6**	**17.6**
10%					71.0	51.9	37.9	28.2	85.5	23.1	50.7	17.2
				✓	71.3	53.1	38.2	32.1	86.7	24.5	52.0	18.2
	✓			✓	72.0	54.4	39.2	33.7	89.8	24.7	53.2	18.5
		✓		✓	71.9	54.3	38.8	33.9	87.5	24.5	53.0	18.4
			✓	✓	72.5	55.4	41.6	34.1	92.1	24.7	53.5	18.7
	✓	✓	✓	✓	**73.2**	**56.2**	**42.1**	**31.3**	**94.6**	**25.6**	**54.2**	**19.0**
20%					73.3	55.7	41.9	31.9	102.7	24.8	53.3	18.4
				✓	74.1	57.7	43.1	31.7	105.7	25.7	54.9	19.4
	✓			✓	74.3	57.5	43.5	33.3	106.6	26.1	55.1	19.6
		✓		✓	74.2	58.0	43.6	33.4	105.4	26.3	55.3	19.7
			✓	✓	74.4	58.1	44.0	33.7	107.1	26.5	55.5	19.9
	✓	✓	✓	✓	**75.0**	**58.4**	**44.3**	**33.9**	**108.0**	**26.9**	**55.5**	**20.2**
30%					73.6	56.4	42.8	32.7	107.1	25.3	53.7	19.0
				✓	74.5	58.4	44.0	32.4	110.2	26.1	55.2	19.2
	✓			✓	74.6	58.3	44.4	34.1	110.6	26.5	55.6	19.9
		✓		✓	74.5	58.5	44.7	34.3	110.7	26.6	55.5	20.3
			✓	✓	74.8	58.4	44.8	34.6	110.9	26.8	55.9	20.5
	✓	✓	✓	✓	**75.3**	**59.0**	**45.2**	**34.5**	**111.8**	**27.3**	**56.0**	**20.7**

5.3.2 Pre-trained Word Embedding Freezing. In the sentence embedding layer of SGCL, we use GloVe pre-trained word embedding as the input, and then freeze word embedding for self-supervised

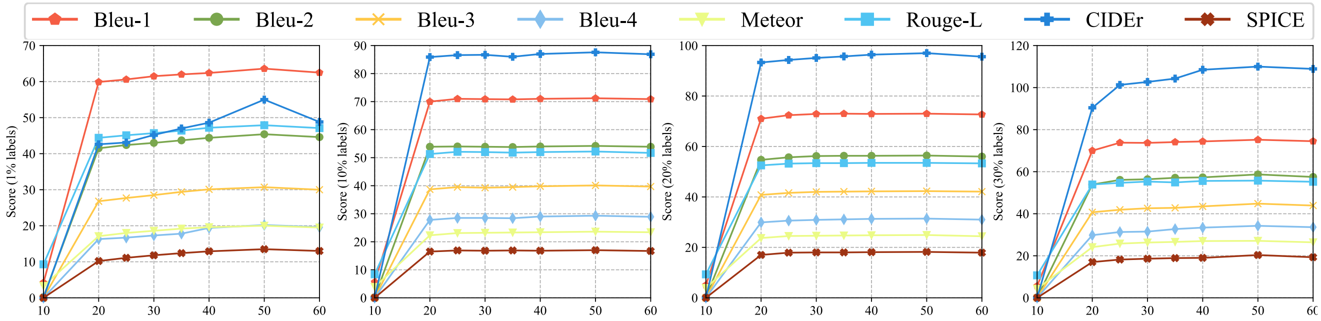
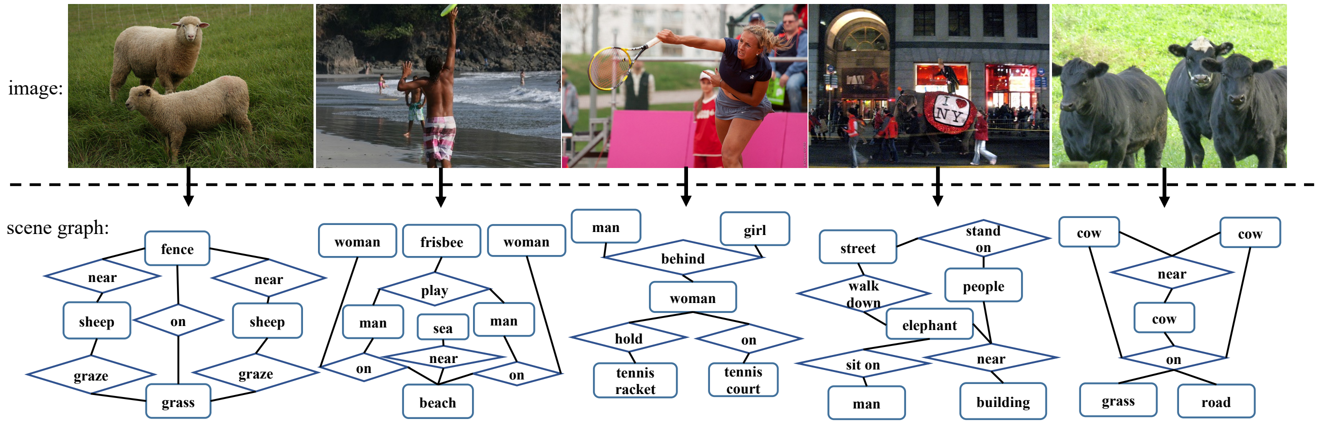


Figure 4: Impact of dropout rate at the output layer on model performance.



GT:	Two sheep grazing on a lush green grass covered field.	A young man catching a frisbee on a sandy beach.	A woman taking a swing at a tennis ball.	A group of people walking down the street with an elephant.	A couple of cows stand on some grass.

1% labels **are used:**

M^2-T:	A sheep standing in a a a.	A man girl a a a a a.	A girl tennis a a tennis tennis	A elephant of in a a a.	A cow standing standing a a a.
SGAE:	A sheep of standing a a a.	A man is a a a a a.	A man girl a a a a a.	A man is a a a a a.	A sheep of in a a a a a.
VSUA:	A sheep of in a a a.	A man standing a a a a.	A girl girl a a a a a.	A people of a a a a a a.	A cow cow cow a a a a a.
C-GAT:	A group of a a a a.	A man is a a a a a.	A man girl a a a a a.	A street of a a a a.	A sheep of a a a.
SGCL:	**A couple of sheep standing in the grass in a field.**	**A group of people playing a frisbee standing by the sea.**	**A woman is holding a tennis racket on a tennis ball.**	**A group of people standing in a street with a building.**	**A herd of cows walking down a road in the grass.**

30% labels **are used:**

M^2-T:	A group of sheep standing in a fenced area.	Two men playing frisbee on a dirt field.	A woman is holding a tennis racket in her hand.	A man riding an elephant in front of a building.	A group of cows standing next to each other on a field.
SGAE:	A white sheep is standing in the grass.	A group of men playing a game of frisbee.	A man hitting a tennis ball on a tennis court.	A group of people standing next to an elephant.	A cow standing on top of a lush green field.
VSUA:	A couple of sheep standing next to each other.	A man holding a frisbee in his hand.	Two men playing frisbee on a dirt field.	An elephant standing in front of a building.	A group of cows are standing in the grass.
C-GAT:	A group of sheep grazing in a grassy field.	A man holding a frisbee in his hand.	A woman is playing tennis on the court.	A man riding on the back of an elephant.	A brown cow standing next to a brown cow.
SGCL:	**A couple of sheep standing on a lush green field near a fence.**	**A man is jumping in the air to catch a frisbee on a sea beach.**	**A woman is trying to hitting a tennis ball on a tennis court.**	**An elephant walking down a street with people in the background.**	**A herd of black cows standing next to each other on a lush green field.**

Figure 5: Case study: image caption generation results of SGCL and four selected baseline methods for different images with various contexts using 1% and 30% training labels (GT: ground-truth).

learning. Here we investigate the effect of loading pre-trained word embedding action and freezing word embedding action. Table 2 reports the model performance when adding different actions in sentence embedding layer. According to the result of this table, we can find that both actions improve image caption generation results. In other words, using pre-trained word embedding rather than using randomly initialized word embedding benefits model contrastive learning. Based on this step, freezing word embedding action further improves the model performance.

Table 2: Effectiveness of loading pre-trained word embedding (P) and freezing word embedding (F).

Label	P	F	B-1	B-2	B-3	B-4	C.	M.	R.-L	S.
1%			62.9	44.8	30.1	19.8	54.1	19.3	47.0	13.0
	✓		63.0	45.1	30.2	19.9	54.4	19.6	47.3	13.0
	✓	✓	**63.6**	**45.4**	**30.7**	**20.2**	**55.0**	**20.0**	**47.9**	**13.5**
5%			69.8	53.2	38.8	28.3	80.0	23.6	52.1	16.9
	✓		70.1	53.6	39.0	28.3	79.8	23.8	52.3	17.1
	✓	✓	**70.7**	**53.8**	**39.8**	**29.1**	**84.9**	**24.2**	**52.6**	**17.6**
10%			71.8	58.3	41.1	33.5	91.7	23.5	52.9	17.5
	✓		72.4	58.3	41.5	33.7	92.6	23.6	53.4	18.3
	✓	✓	**73.2**	**56.2**	**42.1**	**31.3**	**94.6**	**25.6**	**54.2**	**19.0**
20%			74.1	57.8	43.2	33.1	103.9	26.4	54.5	19.6
	✓		74.6	58.0	43.5	33.5	105.5	26.4	55.1	20.0
	✓	✓	**75.0**	**58.4**	**44.3**	**33.9**	**108.0**	**26.9**	**55.5**	**20.2**
30%			74.3	58.4	44.6	33.9	107.3	26.7	55.1	20.2
	✓		75.0	58.6	44.7	34.2	110.8	26.9	55.4	20.5
	✓	✓	**75.3**	**59.0**	**45.2**	**34.5**	**111.8**	**27.3**	**56.0**	**20.7**

5.3.3 Big Dropout Rate. In SGCL, dropout is adopted ahead of context projector in the output layer (Figure 2) to randomly remove a portion of encoded image features for generating output sentence logits. To study the impact of dropout operation, we evaluate model performance by varying dropout rate (from 10% to 60%), as shown in Figure 4. According to this figure, with the dropout rate gradually increasing, we can find that SGCL performs better. SGCL achieves the best performance when dropout rate is around 50%. However, when dropout rate is smaller than 20%, the model is not trained well and the small dropout rate leads to bad model performance. The result indicates big dropout rate has positive effect and brings good model performance, which is consistent with the conclusion in recent work for contrastive sentence embedding [10].

Table 3: Effectiveness of context projector at the output layer.

Label	Projector	B-1	B-2	B-3	B-4	C.	M.	R.-L	S.
1%		55.6	37.5	22.8	14.3	39.5	15.9	40.3	9.2
	✓	**63.6**	**45.4**	**30.7**	**20.2**	**55.0**	**20.0**	**47.9**	**13.5**
5%		68.3	50.1	35.6	25.0	75.1	22.2	49.7	15.3
	✓	**70.7**	**53.8**	**39.8**	**29.1**	**84.9**	**24.2**	**52.6**	**17.6**
10%		68.4	52.5	37.5	28.1	81.4	22.8	50.1	16.4
	✓	**73.2**	**56.2**	**42.1**	**31.3**	**94.6**	**25.6**	**54.2**	**19.0**
20%		69.1	53.7	39.6	29.4	98.4	24.1	50.3	17.2
	✓	**75.0**	**58.4**	**44.3**	**33.9**	**108.0**	**26.9**	**55.5**	**20.2**
30%		71.2	54.5	41.4	31.5	104.2	24.0	50.7	18.6
	✓	**75.3**	**59.0**	**45.2**	**34.5**	**111.8**	**27.3**	**56.0**	**20.7**

5.3.4 Context Projector. To generate output sentence embedding, we design a context projector consisting of a mean pooling layer and two-layer MLP in SGCL. To examine its importance, we remove context projector and only use the mean pooling operation to generate output sentence embedding. The comparison result is shown in Table 3. It is easy to see that the model with context projector has much better performance than that without context projector for all metrics. It demonstrates that the context projector has a significant impact on improving SGCL for image caption generation.

5.4 Case Study

In order to show direct comparison between the proposed model SGCL and some selective baseline models (i.e., M^2-Transformer, VSUA, SGAE, and C-GAT with relatively better performance), we further provide image caption generation results of these methods (with 1% and 30% training labels) for some real cases, as shown in Figure 5. To cover diverse cases, we select images that vary from each other in context. We have several findings from this figure: (1) when label percentage is small (1%), all baseline methods generate poor caption since these models are not trained well with limited labeled data; (2) given 30% labels, baseline methods can generate much better caption, showing that these methods require a sufficient amount of labeled data to train the model well and increasing labeled data can significantly improve their performances; (3) our model SGCL generates better caption (more similar to the ground-truth sentence) than all baseline methods for all cases. More importantly, it still can generate reasonable and meaningful caption when only limited labeled data (1%) are available, which demonstrates the strong capability of contrastive learning for self-supervised image caption generation.

6 CONCLUSIONS

In this paper, we study the problem of self-supervised image caption generation, which has not been well investigated before. To solve the problem, we propose a novel model named SGCL - **S**cene **G**raph **C**ontrastive **L**earning. SGCL is pre-trained by a contrastive learning framework with the input of scene graph and object feature generated by scene graph generation and object detection methods. In SGCL, different views of each data sample are generated through graph perturbations and feature masking, and no text ground-truth label is required. Moreover, context projector and pre-trained word embedding are introduced to enrich text embedding and benefits model training. After self-supervised pre-training, the model is further fine-tuned to the image caption generation task with limited labeled data. Extensive experiments on the benchmark dataset demonstrate the superior performance of SGCL over state-of-the-art baseline methods. As an early work for self-supervised image caption generation, this paper can inspire many future works. For example, we will take benefit of the robust features obtained by contrast training to extend our framework to robust learning of low-quality images (e.g., defective images, low-pixel images) or noisy human-crafted annotations. In addition, we will also extend the application of the self-supervised image caption generation framework to detect dangerous elements on the web (e.g., racial hatred images on social media) with less well-labeled images.

ACKNOWLEDGMENTS

This work is partially supported by the NSF under grants IIS-2209814, IIS-2203262, IIS-2214376, IIS-2217239, OAC-2218762, CNS-2203261, CNS-2122631, CMMI-2146076, and the NIJ 2018-75-CX-0032. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any funding agencies.

REFERENCES

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *ECCV*. 382–398.

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*. 6077–6086.

[3] Shizhe Chen, Qin Jin, Peng Wang, and Qi Wu. 2020. Say as you wish: Fine-grained control of image caption generation with abstract scene graphs. In *CVPR*. 9962–9971.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*. 1597–1607.

[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] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2019. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *ICLR*.

[7] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. 2020. Meshed-Memory Transformer for Image Captioning. In *CVPR*. 10578–10587.

[8] Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *ACL Workshop*. 376–380.

[9] Yang Feng, Lin Ma, Wei Liu, and Jiebo Luo. 2019. Unsupervised image captioning. In *CVPR*. 4125–4134.

[10] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *EMNLP*. 6894–6910.

[11] Jiuxiang Gu, Shafiq Joty, Jianfei Cai, and Gang Wang. 2018. Unpaired image captioning by language pivoting. In *ECCV*. 503–519.

[12] 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 *ACM MM*. 765–773.

[13] Zhichun Guo, Wenhao Yu, Chuxu Zhang, Meng Jiang, and Nitesh V Chawla. 2020. GraSeq: graph and sequence fusion learning for molecular property prediction. In *CIKM*. 435–443.

[14] Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive multi-view representation learning on graphs. In *ICML*. 4116–4126.

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*. 9729–9738.

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

[17] Chao Huang, Huance Xu, Yong Xu, Peng Dai, Lianghao Xia, Mengyin Lu, Liefeng Bo, Hao Xing, Xiaoping Lai, and Yanfang Ye. 2021. Knowledge-aware coupled graph neural network for social recommendation. In *AAAI*. 4115–4122.

[18] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. 2015. Image retrieval using scene graphs. In *CVPR*. 3668–3678.

[19] Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *CVPR*. 3128–3137.

[20] Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

[21] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

[22] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *ACL Workshop*. 74–81.

[23] Xihui Liu, Hongsheng Li, Jing Shao, Dapeng Chen, and Xiaogang Wang. 2018. Show, tell and discriminate: Image captioning by self-retrieval with partially labeled data. In *ECCV*. 338–354.

[24] Xiao Liu, Li Mian, Yuxiao Dong, Fanjin Zhang, Jing Zhang, Jie Tang, Peng Zhang, Jibing Gong, and Kuansan Wang. 2021. OAG_know: Self-supervised Learning for Linking Knowledge Graphs. *TKDE* (2021).

[25] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. 2017. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *CVPR*. 375–383.

[26] Yunpeng Luo, Jiayi Ji, Xiaoshuai Sun, Liujuan Cao, Yongjian Wu, Feiyue Huang, Chia-Wen Lin, and Rongrong Ji. 2021. Dual-level collaborative transformer for image captioning. In *AAAI*. 2286–2293.

[27] Victor Siemen Janusz Milewski, Marie Francine Moens, and Iacer Calixto. 2020. Are Scene Graphs Good Enough to Improve Image Captioning?. In *IJCNLP*. 504–515.

[28] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*. 311–318.

[29] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *EMNLP*. 1532–1543.

[30] Xin Qian, Eunyee Koh, Fan Du, Sungchul Kim, Joel Chan, Ryan A Rossi, Sana Malik, and Tak Yeon Lee. 2021. Generating Accurate Caption Units for Figure Captioning. In *WWW*. 2792–2804.

[31] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. Gcc: Graph contrastive coding for graph neural network pre-training. In *KDD*. 1150–1160.

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS* (2015), 91–99.

[33] Zhan Shi, Xu Zhou, Xipeng Qiu, and Xiaodan Zhu. 2020. Improving Image Captioning with Better Use of Caption. In *ACL*. 7454–7464.

[34] Ruixiang Tang, Mengnan Du, Yuening Li, Zirui Liu, Na Zou, and Xia Hu. 2021. Mitigating Gender Bias in Captioning Systems. In *WWW*. 633–645.

[35] Yijun Tian, Chuxu Zhang, Zhichun Guo, Chao Huang, Ronald Metoyer, and Nitesh V Chawla. 2022. RecipeRec: A Heterogeneous Graph Learning Model for Recipe Recommendation. In *IJCAI*.

[36] Nenad Tomasev, Ioana Bica, Brian McWilliams, Lars Buesing, Razvan Pascanu, Charles Blundell, and Jovana Mitrovic. 2022. Pushing the limits of self-supervised ResNets: Can we outperform supervised learning without labels on ImageNet? *arXiv preprint arXiv:2201.05119* (2022).

[37] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *CVPR*. 4566–4575.

[38] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2018. Deep Graph Infomax. In *ICLR*.

[39] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR*.

[40] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *CVPR*. 3156–3164.

[41] Dongkuan Xu, Wei Cheng, Dongsheng Luo, Haifeng Chen, and Xiang Zhang. 2021. Infogcl: Information-aware graph contrastive learning. In *NeurIPS*. 30414–30425.

[42] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. 2017. Scene graph generation by iterative message passing. In *CVPR*. 5410–5419.

[43] Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. ConSERT: A Contrastive Framework for Self-Supervised Sentence Representation Transfer. In *ACL*. 5065–5075.

[44] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. 2019. Auto-encoding scene graphs for image captioning. In *CVPR*. 10685–10694.

[45] Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. 2021. Graph Contrastive Learning Automated. In *ICML*. 12121–12132.

[46] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. In *NeurIPS*. 5812–5823.

[47] Lu Yu, Shichao Pei, Lizhong Ding, Jun Zhou, Longfei Li, Chuxu Zhang, and Xiangliang Zhang. 2022. SAIL: Self-Augmented Graph Contrastive Learning. In *AAAI*. 8927–8935.

[48] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. 2019. Heterogeneous graph neural network. In *KDD*. 793–803.

[49] Jianan Zhao, Qianlong Wen, Shiyu Sun, Yanfang Ye, and Chuxu Zhang. 2021. Multi-view Self-supervised Heterogeneous Graph Embedding. In *ECML/PKDD*. 319–334.

[50] Yiwu Zhong, Liwei Wang, Jianshu Chen, Dong Yu, and Yin Li. 2020. Comprehensive image captioning via scene graph decomposition. In *ECCV*. 211–229.