This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2020.3042192, IEEE Transactions on Pattern Analysis and Machine Intelligence

Auto-encoding and Distilling Scene Graphs for Image Captioning

Xu Yang, Hanwang Zhang, and Jianfei Cai

Abstract—We propose Scene Graph Auto-Encoder (SGAE) that incorporates the language inductive bias into the encoder-decoder image captioning framework for more human-like captions. Intuitively, we humans use the inductive bias to compose collocations and contextual inferences in discourse. For example, when we see the relation "a person on a bike", it is natural to replace "on" with "ride" and infer "a person riding a bike on a road" even when the "road" is not evident. Therefore, exploiting such bias as a language prior is expected to help the conventional encoder-decoder models reason as we humans and generate more descriptive captions. Specifically, we use the scene graph — a directed graph ($\mathcal{G}$) where an object node is connected by adjective nodes and relationship nodes — to represent the complex structural layout of both image ($\mathcal{I}$) and sentence ($\mathcal{S}$). In the language domain, we use SGAE to learn a dictionary set ($\mathcal{D}$) that helps reconstruct sentences in the $\mathcal{S} \rightarrow \mathcal{G}_\mathcal{S} \rightarrow \mathcal{D} \rightarrow \mathcal{S}$ auto-encoding pipeline, where $\mathcal{D}$ encodes the desired language prior and the decoder learns to caption from such a prior; in the vision-language domain, we share $\mathcal{D}$ in the $\mathcal{I} \rightarrow \mathcal{G}_\mathcal{I} \rightarrow \mathcal{D} \rightarrow \mathcal{S}$ pipeline and distill the knowledge of the language decoder of the auto-encoder to that of the encoder-decoder based image captioner to transfer the language inductive bias. In this way, the shared $\mathcal{D}$ provides hidden embeddings about descriptive collocations to the encoder-decoder and the distillation strategy teaches the encoder-decoder to transform these embeddings to human-like captions as the auto-encoder. Thanks to the scene graph representation, the shared dictionary set, and the Knowledge Distillation strategy, the inductive bias is transferred across domains in principle. We validate the effectiveness of SGAE on the challenging MS-COCO image captioning benchmark, where our SGAE-based single-model achieves a new state-of-the-art 129.6 CIDEr-D on the Karpathy split, and a competitive 126.6 CIDEr-D (c40) on the official server, which is even comparable to other ensemble models. Furthermore, we validate the transferability of SGAE on two more challenging settings: transferring inductive bias from other language corpora and unpaired image captioning. Once again, the results of both settings confirm the superiority of SGAE.

Index Terms—Image Captioning, Scene Graph, Transfer Learning, Memory Network, Knowledge Distillation.

1 Introduction

Modern image captioning models employ an end-to-end encoder-decoder framework [1], [2], [3], [4], [5], i.e., the encoder encodes an image into vector representations and then the decoder decodes them into a language sequence. Since its invention is inspired from neural machine translation [6], this framework has experienced several significant upgrades such as the top-bottom [7] and the bottom-up [3] visual attentions for dynamic encoding, and the reinforced mechanism for sequence decoding [8], [9]. However, a ubiquitous problem has never been substantially resolved: when we feed an unseen image scene into the framework, we usually get a simple and trivial caption about the salient objects such as "a dog on the floor", which is no better than just a list of detected objects [2]. This situation is particularly unfavorable in front of the booming "mid-level" vision techniques nowadays, where we can already detect and segment almost everything in an image [10], [11], [12].

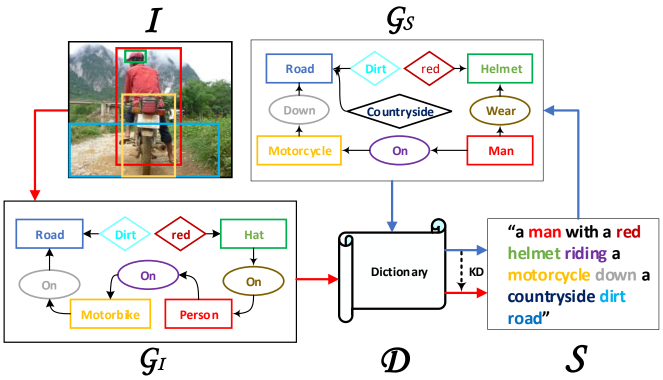


Fig. 1. Illustration of incorporating the auto-encoding of scene graphs (blue arrows) into the conventional encoder-decoder framework for image captioning (red arrows), where the language inductive bias is encoded in the trainable shared dictionary $\mathcal{D}$. In $\mathcal{D} \rightarrow \mathcal{S}$, the dark dot line from the blue arrow to the red arrow denotes the Knowledge Distillation strategy, which teaches the encoder-decoder to reason as the auto-encoder. Word colors correspond to nodes in image and sentence scene graphs.

Humans are good at composing sentences about a visual scene. Unsurprisingly, cognitive evidence [13] shows that the visually grounded language generation is not end-to-end and largely attributed to "high-level" symbolic reasoning, that is, once we abstract the scene into symbols, the generation will be almost *disentangled* from the visual perception. For example, as shown in Fig. 1, from the scene abstraction "helmet-on-human" and "dirt-road", we can say "a man with a helmet in countryside" by using commonsense knowledge like "a dirt road appears in

- *Xu Yang is with the Multimedia and Interactive Computing Lab, School of Computer Science and Engineering, Nanyang Technological University, Singapore.*
E-mail: s170018@e.ntu.edu.sg
- *Hanwang Zhang is currently an Assistant Professor at Nanyang Technological University, Singapore.*
E-mail: hanwangzhang@ntu.edu.sg
- *Jianfei Cai is currently a Professor and serving as the Head for the Data Science & AI Department at Faculty of IT, Monash University, Australia.*
E-mail: Jianfei.Cai@monash.edu

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2020.3042192, IEEE Transactions on Pattern Analysis and Machine Intelligence

countryside". In fact, such collocations and contextual inference in human language can be considered as the *inductive bias* that is apprehended by us from everyday practice, which makes us perform better than machines in high-level reasoning [14], [15]. However, the direct exploitation of the inductive bias, e.g., early template/rule-based caption models [16], [17], is well-known to be ineffective compared to the encoder-decoder ones, due to the large gap between visual perception and language composition.

In this paper, we propose to incorporate the inductive bias of language generation into the encoder-decoder framework for image captioning, benefiting from the complementary strengths of both symbolic reasoning and end-to-end multi-modal feature mapping. In particular, we use scene graphs [18], [19] to bridge the gap between the two worlds. A scene graph ($\mathcal{G}$) is a unified representation that connects 1) the objects (or entities), 2) their attributes, and 3) their relationships in an image ($\mathcal{I}$) or a sentence ($\mathcal{S}$) by directed edges. Due to the recent advances in spatial Graph Convolutional Networks (GCNs) [20], [21], [22], we can embed the graph structure into the vector representations, which can be seamlessly integrated into the encoder-decoder. Our key insight is that the vector representations are expected to transfer the inductive bias from the pure language domain to the vision-language domain.

To encode the language inductive bias, we propose the Scene Graph Auto-Encoder (SGAE), which is a sentence self-reconstruction network in the $\mathcal{S} \rightarrow \mathcal{G}_{\mathcal{S}} \rightarrow \mathcal{D} \rightarrow \mathcal{S}$ pipeline (blue arrows in Fig. 1), where $\mathcal{D}$ is a trainable dictionary set for the re-encoding purpose of the node features, the $\mathcal{S} \rightarrow \mathcal{G}_{\mathcal{S}}$ module is a fixed off-the-shelf scene graph language parser [23], $\mathcal{G}_{\mathcal{S}} \rightarrow \mathcal{D}$ is an Attentional GCN (AGCN) [24] to selectively encode the scene graph into embeddings, and $\mathcal{D} \rightarrow \mathcal{S}$ is a trainable RNN-based language decoder [3]. Note that $\mathcal{D}$ consists of descriptive and imaginary collocations — the language inductive bias — we extract from training SGAE and $\mathcal{D} \rightarrow \mathcal{S}$ learns to compose these collocations into human-like captions. By sharing such $\mathcal{D}$ containing ample language inductive bias in the encoder-decoder pipeline $\mathcal{I} \rightarrow \mathcal{G}_{\mathcal{I}} \rightarrow \mathcal{D} \rightarrow \mathcal{S}$ (red arrows in Fig. 1) and teaching this encoder-decoder to reason as the auto-encoder by Knowledge Distillation (KD) (vertical dark dot line $\mathcal{D} \rightarrow \mathcal{S}$ in Fig. 1, also see Section 5.2), the language inductive bias is transferred to guide the end-to-end image captioning. In particular, the $\mathcal{I} \rightarrow \mathcal{G}_{\mathcal{I}}$ module is a visual scene graph detector [25] and we introduce a multi-modal AGCN (MAGCN) for the $\mathcal{G}_{\mathcal{I}} \rightarrow \mathcal{D}$ module in the captioning pipeline, to complement necessary visual cues that are missing due to the imperfect visual detection. Interestingly, $\mathcal{D}$ can be considered as a working memory [26] that helps to re-key the encoded nodes from $\mathcal{I}$ or $\mathcal{S}$ to more generic representations with smaller domain gaps (see Section 4.3).

We implement the proposed SGAE-based captioning model by using the recently released visual encoder [27] and language decoder [3] with an RL-based training strategy [8]. Extensive experiments on MS-COCO [28] validate the superiority of using SGAE in image captioning. Particularly, in terms of the popular CIDEr-D metric [29], we achieve an absolute 9-point improvement over a strong baseline, which is a reimplemented version of Up-Down [3]. Our *single-model* reaches a new state-of-the-art score, achieving 129.6 on the Karpathy split and a competitive 126.6 on the official test server, which is even comparable to many ensemble models. Furthermore, we test the transferability of our SGAE in two more challenging settings: transferring inductive bias from other language corpora and unpaired image captioning. Particularly, when another Web language corpus is used to train the auto-encoder, we observe consistent improvements in CIDEr-D than the captioner without this corpus. For unpaired image captioning, our SGAE obtains a 10.6-point improvement in CIDEr-D than the baseline.

In summary, the work makes the following technical contributions:

- A novel Scene Graph Auto-Encoder (SGAE) for learning the graph representation of the language inductive bias.
- A multi-modal Attentional Graph Convolutional Network for modulating scene graphs into visual representations.
- An SGAE-based encoder-decoder image captioner with a shared dictionary and a Knowledge Distillation objective, both guiding the language decoding.
- Various captioning tasks and settings to confirm the superiority of our SGAE.

This paper is an extension of our preliminary work [30] with significant modifications. Specifically,

- We partitioned the original one dictionary $\mathcal{D}$ in [30] into three more fine-grained dictionaries $\mathcal{D}_O, \mathcal{D}_A, \mathcal{D}_R$, each of which preserves one kind of specific inductive bias: object, attribute, and relation knowledge.
- We exploited the Knowledge Distillation strategy to further enhance the inductive bias transfer from the pure language domain to the vision-language domain.
- We replaced the original GCN for encoding sentence scene graphs with a more advanced Attentional Graph Convolutional Network (AGCN) and replaced the original Multi-modal GCN for encoding image scene graphs with MAGCN to selectively incorporate different semantic contents to achieve stronger representation power.
- We validated the effectiveness of the new SGAE with these three refinements by extensive experiments and the results demonstrate the effectiveness of each refinement.
- We also added two more challenging tasks: unpaired image captioning and transfer inductive bias from a wild language corpus. The experiment results validate the effectiveness and the potential of our SGAE in both two settings.

2 Related Work

2.1 Image Captioning

There is a long history for researchers to develop automatic image captioning methods. The preliminary image captioners are usually templates/statistic based models [31], [32], [33] that they first form the templates according to the language statistic and then fill the words based on the recognized visual patterns to complete the captions. Though the language inductive bias is incorporated into these templates, the performances of these captioners are limited since the processes for composing templates and for filling words are not jointly trained.

Compared with these early works, the modern captioning models have achieved striking advances due to the progress of the deep learning techniques in both computer vision, e.g., more accurate visual pattern recognition systems [27], [34], and natural language processing fields, e.g., encoder-decoder pipeline [35] and attention mechanism [6]. Inspired by them, researchers have developed attention based encoder-decoder image captioners. Particularly, S&T [36] builds an encoder-decoder captioner with end-to-end training and SA&T [7] exploits attention mechanism to generate words by adaptively focusing on different regions of an image at different time steps. After that, more advanced visual attention mechanisms have been proposed for generating better captions. For example, KWTL [37] can decide whether to attend to the image for generating vision-related words or just function words, bottom-up attention [3] selects regions only from saliency regions which are more likely to contain objects, and CAVP [4] and LBPF [38] stack more attention sub-networks for multiple-step reasoning. Interestingly, researchers have also designed semantic attention mechanisms [39], [40] to directly incorporate semantic labels recognized from the image into the captioners. For example, some methods exploit object [2], attribute [41], and relationship [42] knowledge into their captioning models. Also, CNM [5] collocates a series of modules to represent such semantic knowledge for captioning. Compared with all these approaches, our method uses the scene graph as the bridge to integrate object, attribute, and relationship knowledge to discover more meaningful semantic contexts from the preserved language inductive bias for more descriptive captions.

In addition to the progress of the network architectures in image captioning, some other methods are proposed to address the "exposure bias" problem [8], [9] for ameliorating the optimization. Specifically, when we train an image captioner, the words are generated conditioned on the ground-truth words, while during inference, the words are generated conditioned on the sampled words. Thus, there is a mismatch between the training and the inference, which causes performance degradation. Scheduled sampling strategy [43] is proposed to relieve the problem, which generates new words also conditioned on the sampled words during training. Furthermore, Reinforcement Learning based rewards [8], [44] are used to train the captioners by measuring the similarity between the sampled sentences and the ground-truth sentences. Considering these two techniques can alleviate the "exposure bias" problem and improve the performances, we also adopt them in this work.

Another line of image captioning research is on a more challenging task: Unpaired Image Captioning [45], which only has an image set and a caption corpus during training but without any image-caption pairs. A few preliminary works, including Pivoting Language [45], adversarial training [46], [47] and multi-modal embedding alignment [48], have been proposed to tackle this problem. In this work, we also extend our SGAE to such unpaired scenarios to validate its transferability between the vision and language domains.

2.2 Scene Graphs

Visual scene graphs [49], [50] contain the structured semantic contents of an image, which typically include the knowledge of the present objects, their attributes, and their pairwise relationships. They are beneficial for many vision tasks like VQA [51], [52], image generation [18], [53], and visual grounding [54]. By observing the great potential of scene graphs, a variety of approaches have been proposed to generate scene graphs from the images [24], [25], [55], [56], [57], [58], [59]. These methods usually first exploit some advanced object detectors, e.g., Faster RCNN [27], YOLO [60], or Mask RCNN [61], to detect the discrete objects and then recognize the pairwise object relationships so as to connect the discrete object nodes to construct a graph.

Besides extracting scene graphs from visual scenes, scene graphs can also be extracted from textual data, e.g., applying a network to parse image captions for scene graph generation [19], or designing certain rules to transfer the object and relationship classification results [62] or the dependency parser tree of a text [23] into the resultant scene graph. In this research, we exploit Motif [25] and Spice [23] to extract scene graphs from images and captions, respectively.

2.3 Graph Convolutional Networks

Graph Convolutional Network (GCN) generalizes CNN from regular grid data, e.g., image, speech, or text to any irregular graph data, e.g., social networks, citation networks, or knowledge graphs. There are two types of GCNs: spectral GCN [63], [64], [65] and spatial GCN [66], [67], [68]. Considering spectral GCN is usually used for the undirected graph and our scene graphs are directed, we apply spatial GCN in our research to transform a scene graph into a series of continuous embeddings, which can be seamlessly integrated into the deep network. Specifically, we deploy an Attentional GCN [24] to selectively incorporate the attribute and relationship nodes into an embedding to avoid redundancy brought by the noisy scene graphs. We also follow the suggestions in [22] to design our GCN to achieve stronger representation power.

2.4 Memory Networks

Recently, many works try to augment a working memory into a network for preserving a dynamic knowledge base to facilitate the subsequent inference, e.g., for preserving training samples to achieve one-shot learning [26], for preserving the context of a document to solve language modeling [69], question answering [70], [71], and reading comprehension [72], for preserving visual knowledge to deal with visual question answering [73], or even for preserving a series of reasoning decisions to address long term machine reasoning [74]. Among these methods, differentiable attention mechanisms [70], [71] are usually applied to preserve (extract) useful knowledge in (from) the memory for the tasks at hand. The extraction of the knowledge is achieved by query-key attention computation where the query is some kind of context knowledge (e.g., the questions in question answering task) and the key comes from the memory. Inspired by these methods, we also implement a memory architecture to preserve language inductive bias, which is represented by the sentence scene graph embedding during the training of the auto-encoder while being extracted during image captioning by using the visual scene graph embedding as the query.

2.5 Knowledge Distillation

Knowledge Distillation (KD) abstracts the response to input of a teacher learner as the knowledge and distils such knowledge to another student learner for achieving knowledge transfer [75]. Since KD does not require that the teacher and student learners own the same architecture or even the same task [76], it can be applied to various machine learning applications. For example, it can achieve model compression by teaching a small network to learn from a huge network trained by a large-scale dataset [77], [78]. Lifelong Learning [79], [80] also exploits KD to preserve the knowledge of the previous datasets to avoid catastrophic forgetting. KD has also been exploited for protecting modern learning systems from the attack of adversarial examples [81] by reducing the amplitude of network gradients. In this research, we treat the language scene

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2020.3042192, IEEE Transactions on Pattern Analysis and Machine Intelligence

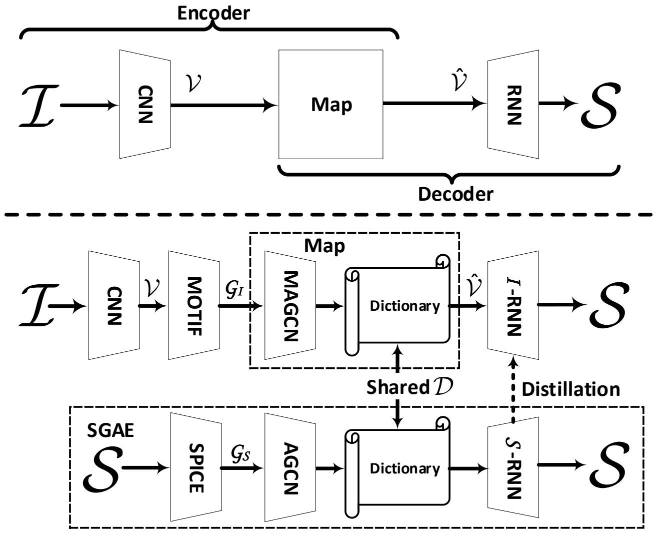


Fig. 2. Top: the conventional encoder-decoder. Bottom: our proposed encoder-decoder, where the novel SGAE embeds the language inductive bias in the dictionary set, which is shared to the image captioner, and in the decoder $\mathcal{S}$-RNN of SGAE, whose knowledge is distilled into the decoder $\mathcal{I}$-RNN of the image captioner.

graph auto-encoder and the image captioning encoder-decoder as the teacher and the student, respectively, by which we hope the encoder-decoder learns to compose captions as the auto-encoder. By distilling knowledge, the inductive bias is better transferred between the pure-language and the vision-language domains.

3 Encoder-Decoder Revisited

As illustrated in Fig. 2, given an image $\mathcal{I}$, the target of image captioning is to generate a natural language sentence $\mathcal{S} = \{s_1, s_2, ..., s_T\}$ describing the image. A state-of-the-art encoder-decoder image captioner can be formulated as:

$$\begin{aligned} \textbf{Encoder:} &\quad \mathcal{V} \leftarrow \mathcal{I}, \\ \textbf{Map:} &\quad \hat{\mathcal{V}} \leftarrow \mathcal{V}, \\ \textbf{Decoder:} &\quad \mathcal{S} \leftarrow \hat{\mathcal{V}}. \end{aligned} \tag{1}$$

Usually, the encoder is a Convolutional Neural Network (CNN) [27], [34] that extracts the image feature $\mathcal{V}$; the map is the widely used attention mechanism [3], [7] that re-encodes the visual features into more informative $\hat{\mathcal{V}}$ which is dynamic to language generation; the decoder is an RNN-based language decoder for the sequence prediction of $\mathcal{S}$. Given a ground-truth caption $\mathcal{S}^*$ for $\mathcal{I}$, we can train this encoder-decoder model by minimizing cross-entropy loss:

$$L_{XE} = -\log P(\mathcal{S}^*), \tag{2}$$

or by maximizing a reinforcement learning (RL) based reward [8] as:

$$R_{RL} = \mathbb{E}_{\mathcal{S}^s \sim P(\mathcal{S})}(r(\mathcal{S}^s; \mathcal{S}^*)), \tag{3}$$

where r is a sentence-level metric for the sampled sentence $\mathcal{S}^s$ and the ground-truth $\mathcal{S}^*$, e.g., the CIDEr-D [29] metric.

This encoder-decoder framework is the core pillar underpinning almost all the state-of-the-art image captioners since S&T [36]. However, it is widely shown brittle to dataset bias [2], [82]. We propose to exploit the language inductive bias, which is beneficial, to confront the dataset bias for more human-like image captioning. As shown in Fig. 2, the proposed framework is formulated as:

$$\begin{aligned} \textbf{Encoder:} &\quad \mathcal{V} \leftarrow \mathcal{I}, \\ \textbf{Map:} &\quad \hat{\mathcal{V}} \leftarrow R(\mathcal{V}, \mathcal{G}_{\mathcal{I}}; \mathcal{D}),\ \mathcal{G}_{\mathcal{I}} \leftarrow \mathcal{V}, \\ \textbf{Decoder:} &\quad \mathcal{S} \leftarrow \hat{\mathcal{V}}. \end{aligned} \tag{4}$$

We focus on modifying the Map module by introducing the scene graph $\mathcal{G}_{\mathcal{I}}$ into a re-encoder R parameterized by a shared dictionary set $\mathcal{D}$. In particular, we first propose a Scene Graph Auto-Encoder (SGAE) to learn the dictionary set $\mathcal{D}$ and the decoder $\mathcal{S}$-RNN which both embed the language inductive bias from sentence to sentence self-reconstruction (see Section 4) with the help of scene graphs. Then, we equip the encoder-decoder with the proposed SGAE to be our overall image captioner (see Section 5). Specifically, we use a novel Multi-modal Attentional Graph Convolutional Network (MAGCN) (see Section 5.1) to re-encode the image features by using $\mathcal{D}$ and a Knowledge Distillation strategy to train the encoder-decoder pipeline (see Section 5.2), narrowing the gap between vision and language.

4 Auto-Encoding Scene Graphs

In this section, we will introduce how to learn $\mathcal{D}$ through self-reconstructing sentence $\mathcal{S}$. As shown in Fig. 2, the process of reconstructing $\mathcal{S}$ is also an encoder-decoder pipeline. Thus, by slightly abusing the notations, we can formulate SGAE as:

$$\begin{aligned} \textbf{Encoder:} &\quad \mathcal{X} \leftarrow \mathcal{G}_{\mathcal{S}} \leftarrow \mathcal{S}, \\ \textbf{Map:} &\quad \hat{\mathcal{X}} \leftarrow R(\mathcal{X}; \mathcal{D}), \\ \textbf{Decoder:} &\quad \mathcal{S} \leftarrow \hat{\mathcal{X}}. \end{aligned} \tag{5}$$

Next, we detail every component in Eq. (5).

4.1 Scene Graphs

We introduce how to implement the step $\mathcal{G}_{\mathcal{S}} \leftarrow \mathcal{S}$, i.e., from sentence to scene graph. Formally, a scene graph is a tuple $\mathcal{G}_{\mathcal{S}} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ and $\mathcal{E}$ are the sets of nodes and edges, respectively. There are three kinds of nodes in $\mathcal{N}$: object node o, attribute node a, and relationship node r. We denote o_i as the i-th object, r_{ij} as the relationship between o_i and o_j, and $a_{i,l}$ as the l-th attribute of o_i. For each node in $\mathcal{N}$, it is represented by a trainable d-dimensional label embedding vector of the corresponding semantic label, i.e., $\boldsymbol{e}_o$, $\boldsymbol{e}_a$, and $\boldsymbol{e}_r$ for object, attribute, and relationship label embedding, respectively. In our implementation, d is set to $1,000$. The edges in $\mathcal{E}$ are formulated as follows:

- if an object o_i owns an attribute $a_{i,l}$, assigning a directed edge from $a_{i,l}$ to o_i;
- if there is one relationship triplet $< o_i - r_{ij} - o_j >$ appearing, assigning two directed edges from o_i to r_{ij} and from r_{ij} to o_j, respectively.

Fig. 3 shows one example of $\mathcal{G}_{\mathcal{S}}$, which contains 7 nodes in $\mathcal{N}$ and 6 directed edges in $\mathcal{E}$.

We use the scene graph parser provided by Spice [23] to get scene graphs $\mathcal{G}_{\mathcal{S}}$ from sentences, where a syntactic dependency tree is built [83] and then a rule-based method [62] is applied for transforming the tree to a scene graph.

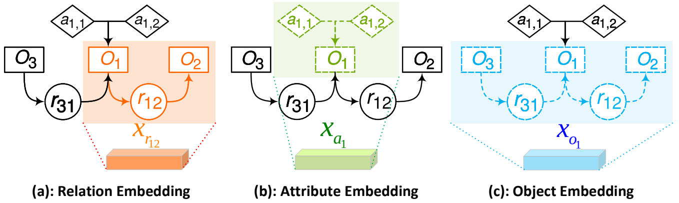


Fig. 3. Attentional Graph Convolutional Network. In particular, it is a spatial convolutional network, where the colored neighborhood is “convolved” for the resultant embedding. The dashed lines in (b) and (c) mean these knowledge flows are modulated by the soft attention weights.

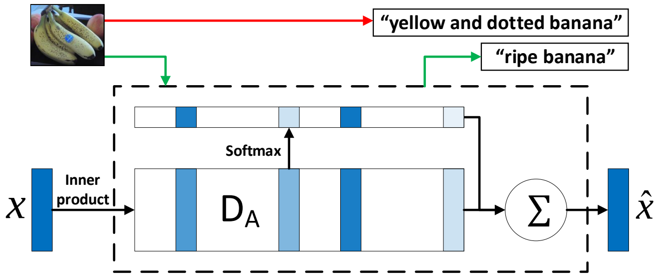


Fig. 4. The visualization of the re-encoder function R. The black dashed block shows the operation of this function. The top part demonstrates how “imagination” is achieved by re-encoding: the green line shows the generated phrase by re-encoding, while the red line shows the one without re-encoding. For convenience, we only visualize the attribute dictionary $\mathcal{D}_A$ here.

4.2 Attentional Graph Convolutional Network

We present the implementation for the step $\mathcal{X} \leftarrow \mathcal{G}_\mathcal{S}$ in Eq. (5), i.e., how to transform the original node embeddings $\boldsymbol{e}_o$, $\boldsymbol{e}_a$, and $\boldsymbol{e}_r$ into a new set of context-aware embeddings $\mathcal{X}$. Formally, $\mathcal{X}$ contains three kinds of d-dimensional embeddings: relationship embedding $\boldsymbol{x}_{r_{ij}}$ for relationship node r_{ij}, object embedding $\boldsymbol{x}_{o_i}$ for object node o_i, and attribute embedding $\boldsymbol{x}_{a_i}$ for object node o_i. In our implementation, d is set to $1,000$. We use four *spatial graph convolutions*: g_r, g_a, g_s, and g_o for generating the above mentioned three kinds of embeddings. In our implementation, all these four functions have the same structure with independent parameters: a vector concatenation input to two fully-connected layers with a ReLU in between (FC-ReLU-FC). We use two-layer perceptrons instead of one-layer perceptron (which is used in our preliminary work [30]) to achieve stronger representation power [22].

Relationship Embedding $\mathbf{x}_{r_{ij}}$: Given one relationship triplet $< o_i - r_{ij} - o_j >$ in $\mathcal{G}_\mathcal{S}$, we incorporate the context of a relationship triplet into $\mathbf{x}_{r_{ij}}$:

$$\boldsymbol{x}_{r_{ij}} = g_r([\boldsymbol{e}_{o_i}, \boldsymbol{e}_{r_{ij}}, \boldsymbol{e}_{o_j}]), \tag{6}$$

where $[\cdot,\cdot]$ means the concatenation operation. Fig. 3 (a) shows such an example.

Attribute Embedding $\boldsymbol{x}_{a_i}$: Given one object node o_i with all its attributes $a_{i,1:Na_i}$ in $\mathcal{G}_\mathcal{S}$, where Na_i is the number of attributes this object o_i has, then we use an Attentional GCN (AGCN) [24] to incorporate the context of this object and all its attributes into $\boldsymbol{x}_{a_i}$:

$$\boldsymbol{x}_{a_i} = \sum_{l=1}^{Na_i} p_l^a g_a([\boldsymbol{e}_{o_i}, \boldsymbol{e}_{a_{i,l}}]), \tag{7}$$

where p_l^a is the adaptive soft attention weight for the l-th attribute:

$$\begin{aligned} q_l^a &= \boldsymbol{w}_a^T \tanh(\boldsymbol{W}_a[\boldsymbol{e}_{o_i}, \boldsymbol{e}_{a_{i,l}}]), \\ \boldsymbol{p^a} &= \{p_1^a, p_2^a, ..., p_{Na_i}^a\} = \text{Softmax}(\{q_1^a, q_2^a, ..., q_{Na_i}^a\}), \end{aligned} \tag{8}$$

where $\boldsymbol{w}_a$ is a trainable vector and $\boldsymbol{W}_a$ is a trainable matrix. Compared with our preliminary work [30], we apply this AGCN because not all the attributes are equally important to the final caption and thus we use the context of o_i with its attribute $a_{i,l}$ to decide whether this attribute is important or not. By this AGCN the knowledge flow in the graph can be modulated for generating more accurate captions. Fig. 3 (b) shows such an attribute embedding.

Object Embedding $\boldsymbol{x}_{o_i}$: In $\mathcal{G}_\mathcal{S}$, o_i can act as “a subject” or “an object” in relationships, which means o_i will play distinguishable roles denoted by different edge directions. Then, different functions should be used to incorporate such knowledge, i.e., the functions g_s and g_o. To avoid the ambiguous meaning of the same “predicate” in a different context, knowledge of the whole relationship triplets where o_i appears should be incorporated into $\boldsymbol{x}_{o_i}$. One simple example for ambiguity is that, in <hand-with-cup>, the predicate “with” may mean “hold”, while in <head-with-hat>, “with” may mean “wear”. In addition, similar to the attribute embedding, AGCN is also applied here to modulate the knowledge flow. Therefore, $\boldsymbol{x}_{o_i}$ is calculated as:

$$\begin{aligned} \boldsymbol{x}_{o_i} = & \sum_{o_j \in obj(o_i)} p_j^o \, g_o([\boldsymbol{e}_{o_i}, \boldsymbol{e}_{o_j}, \boldsymbol{e}_{r_{ij}}]) \\ & + \sum_{o_k \in sbj(o_i)} p_k^s \, g_s([\boldsymbol{e}_{o_k}, \boldsymbol{e}_{o_i}, \boldsymbol{e}_{r_{ki}}]), \end{aligned} \tag{9}$$

where $o_j \in obj(o_i)$ and $o_k \in sbj(o_i)$ indicate (o_i, o_j) and (o_k, o_i) form subject-object pairs, respectively, e.g., $obj(o_1) = \{o_2\}$ in Fig. 3 (c); and p_j^o, p_k^s are soft attention weights computed in similar ways with the same trainable vector $\boldsymbol{w}_o$ and matrix $\boldsymbol{W}_o$, e.g., p_j^o is:

$$\begin{aligned} q_j^o &= \boldsymbol{w}_o^T \tanh(\boldsymbol{W}_o[\boldsymbol{e}_{o_i}, \boldsymbol{e}_{o_j}, \boldsymbol{e}_{r_{ij}}]), \\ \boldsymbol{p}^o &= \{p_1^o, p_2^o, ..., p_{N_{obj_i}}^o\} = \text{softmax}(\{q_1^o, q_2^o, ..., q_{N_{obj_i}}^o\}), \end{aligned} \tag{10}$$

where N_{obj_i} is the size of the set $obj(o_i)$. Fig. 3 (c) shows this object embedding.

4.3 Dictionary

Now we introduce how to learn the dictionary set $\mathcal{D}$ and how to use it to re-encode $\hat{\mathcal{X}} \leftarrow R(\mathcal{X}; \mathcal{D})$ in Eq. (5). Our key idea is inspired by using the working memory to preserve a global knowledge base for run-time inference, which is widely used in textual QA [70], VQA [69], and one-shot classification [26]. Our $\mathcal{D}$ aims to embed language inductive bias in language composition. Therefore, we propose to place the dictionary learning into the sentence self-reconstruction framework. Compared with our preliminary work [30] where we only use one dictionary to preserve all the knowledge, here we set $\mathcal{D} = \{\mathcal{D}_R, \mathcal{D}_A, \mathcal{D}_O\}$ to preserve knowledge about relationship embeddings (Eq.(6)), attribute embeddings (Eq.(7)), and object embeddings (Eq.(9)), respectively. For convenience, we remove the subscript of $\mathcal{D}$. Formally, we denote a dictionary as a $d \times K$ matrix $\mathcal{D} = \{\boldsymbol{d}_1, \boldsymbol{d}_2, ..., \boldsymbol{d}_K\}$, where K is the total number of the entries and is set to $5,000$ for each dictionary in implementation. Given an embedding vector $\boldsymbol{x} \in \mathcal{X}$, the re-encoder function R can be formulated as:

$$\hat{\boldsymbol{x}} = R(\boldsymbol{x}; \mathcal{D}) = \mathcal{D}\boldsymbol{\alpha} = \sum_{k=1}^{K} \alpha_k \boldsymbol{d}_k, \quad (11)$$

where $\boldsymbol{\alpha} = \text{softmax}(\mathcal{D}^T \boldsymbol{x})$ can be viewed as the "query-key" operation in memory network [70], [72], as in Fig. 4. Since these operations are all differentiable, $\mathcal{D}$ can be learned during the end-to-end training.

Intuitively, after training, this $\mathcal{D}$ preserves the "rules" which we humans usually use to compose descriptive sentences. When we insert $\mathcal{D}$ into the image captioner pipeline, given the noisy and simple image scene graphs, the captioner will exploit the preserved "rules" in $\mathcal{D}$ to transform them into embeddings for more descriptive captions. For example, in Fig. 6, such "rules" can summarize "a street with many cars" as "busy" in (a); add one more descriptive phrase "lush green" to decorate the trees in (b); achieve certain commonsense reasoning that infers "a ripe banana" from "a yellow and dotted banana" in (c); and use one more suitable action "perch" for the bird to replace the trivial relation "on" in (d).

4.4 Unpaired Image Captioning

To further validate the transferability between the pure language domain and the vision-language domain, we exploit our SGAE to address unpaired image captioning task [45]. In this setting, we only have an image set $\mathcal{I}$ and a language corpus $\mathcal{S}$, while we do not have any paired image-caption samples for training the image captioner. To deal with this task, we first learn an auto-encoder by reconstructing sentences $\mathcal{S}$ from the sentence scene graphs $\mathcal{G}_{\mathcal{S}}$ of these sentences. Importantly, in this unpaired setting, we only train SGAE. During testing, once we extract image scene graphs $\mathcal{G}_{\mathcal{I}}$ from the images (see Section 5.1), we directly input them into the trained auto-encoder to generate the captions. More formally, we can write the process as:

$$\begin{aligned} \textbf{Training:} \quad & \mathcal{S} \leftarrow \hat{\mathcal{X}} \leftarrow \mathcal{D} \leftarrow \mathcal{X} \leftarrow \mathcal{G}_{\mathcal{S}} \leftarrow \mathcal{S}, \\ \textbf{Testing:} \quad & \mathcal{S} \leftarrow \hat{\mathcal{X}} \leftarrow \mathcal{D} \leftarrow \mathcal{X} \leftarrow \mathcal{G}_{\mathcal{I}} \leftarrow \mathcal{I}. \end{aligned} \quad (12)$$

The results and the comparisons with the state-of-the-art models in Section 6.4.2 validate the superiority of our SGAE in unpaired image captioning as well.

4.5 Training of SGAE

We deployed the top-down attention structure [3] as the $\mathcal{S}$-RNN for reconstructing $\mathcal{S}$, which contains two LSTM layers and one attention module in between. This structure has proven to be a powerful one and most state-of-the-art captioning models use it as the backbone, e.g., GCN-LSTM [42], RFNet [84], and LBPF [38].

Here we first show the prototypical architecture of our $\mathcal{S}$-RNN and then specify the values of the variables in this network. Specifically, at time step t, given the last word s_{t-1} and the embedding set $\mathcal{X}$, $\mathcal{S}$-RNN calculates the word probability vector $\boldsymbol{z}_t$ as follows:

$$\begin{aligned} \textbf{Input:} \quad & \boldsymbol{i}_t = [\text{Embed}(s_{t-1}), \boldsymbol{h}_2^{t-1}] \\ \textbf{LSTM1:} \quad & \boldsymbol{h}_1^t = \text{LSTM}_1(\boldsymbol{i}_t; \boldsymbol{h}_1^{t-1}), \\ \textbf{Attention:} \quad & \hat{\boldsymbol{x}}^t = \text{ATT}(\mathcal{X}, \boldsymbol{h}_1^t), \\ \textbf{LSTM2:} \quad & \boldsymbol{h}_2^t = \text{LSTM}_2([\boldsymbol{h}_1^t, \hat{\boldsymbol{x}}^t]; \boldsymbol{h}_2^{t-1}), \\ \textbf{Output:} \quad & \boldsymbol{z}_t = \text{Softmax}(\text{FC}(\boldsymbol{h}_2^t)), \end{aligned} \quad (13)$$

where $[\cdot]$ means the concatenation operation; Embed$(\cdot)$ is a learnable word embedding layer; LSTM1$(\cdot)$ and LSTM2$(\cdot)$ are two different LSTM layers whose hidden unit sizes are both set to 1,000; and the attention sub-network ATT is:

$$\begin{aligned} \textbf{Input:} \quad & \mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}, \boldsymbol{h}_1^t, \\ \textbf{Attention Weights:} \quad & b_n^t = \boldsymbol{\omega}_b^T \tanh(\boldsymbol{W}_v \boldsymbol{x}_n + \boldsymbol{W}_h \boldsymbol{h}_1^t), \\ & \boldsymbol{\beta}^t = \{\beta_1^t, \beta_2^t, ..., \beta_N^t\} = \text{softmax}(\boldsymbol{b}^t), \\ \textbf{Output:} \quad & \hat{\boldsymbol{x}}^t = \sum_{n=1}^{N} \beta_n^t \boldsymbol{x}_n, , \end{aligned} \quad (14)$$

where $\boldsymbol{W}_v$, $\boldsymbol{W}_h$ are trainable matrices and $\boldsymbol{\omega}_b$ is a trainable vector.

Given a sentence $\mathcal{S}$, we first extracted its scene graph and then used AGCN operations (Eq. (6), (7), (9)) to compute relationship, attribute, and object embeddings and then grouped them as three embedding sets $\mathcal{X}_R, \mathcal{X}_A, \mathcal{X}_O$. For example, $\mathcal{X}_R$ contains all the relationship embeddings $\boldsymbol{x}_{r_{ij}}$ computed by Eq. (6). These embedding sets were input into three dictionaries $\mathcal{D}_R, \mathcal{D}_A, \mathcal{D}_O$ to get the re-encoded embedding sets $\hat{\mathcal{X}}_R, \hat{\mathcal{X}}_A, \hat{\mathcal{X}}_O$ by Eq. (11) (e.g., $\hat{\mathcal{X}}_A = R(\mathcal{X}_A; \mathcal{D}_A)$), respectively. Then we grouped $\hat{\mathcal{X}}_R, \hat{\mathcal{X}}_A, \hat{\mathcal{X}}_O$ together as the embedding set $\mathcal{X}$ in Eq. (14) to calculate the attended vector $\hat{\boldsymbol{x}}^t$. Then we used Eq. (13) to calculate the word probability vector $\boldsymbol{z}_t$ and trained the network by cross-entropy loss (Eq. (2)) where the ground-truth probability vector $\boldsymbol{z}_t^*$ is an one-hot vector corresponding to the ground-truth word s_t^*.

The whole SGAE (AGCN, $\mathcal{S}$-RNN, and the dictionary set) were all randomly initialized. When we trained this SGAE, in the first 10 epochs, we removed the dictionary set in the pipeline to train a rudiment AGCN and $\mathcal{S}$-RNN. In the next 30 epochs, we inserted the dictionary set into the pipeline to jointly train the dictionary, AGCN, and $\mathcal{S}$-RNN. Removing the dictionary set at the beginning was to avoid learning trivial values since the graph embeddings ($\mathcal{X}_R, \mathcal{X}_A, \mathcal{X}_O$) had not learnt meaningful inductive bias at that moment. The learning rate was initialized to $5e^{-4}$ for all parameters and decayed by 0.8 for every 5 epochs. The batch size was set to 100 and Adam optimizer [85] was used. Note that the training of SGAE is unsupervised, that is, SGAE offers the potential never-ending learning from large-scale unsupervised inductive bias for $\mathcal{D}$. Some preliminary studies are reported in Section 6.2.2. Note that these training strategies were used in both paired and unpaired image captioning.

5 SGAE-based Encoder-Decoder

In this section, we introduce the overall model: SGAE-based Encoder-Decoder image captioner as in Fig. 2 and Eq. (4).

5.1 Multi-modal AGCN

The original image features extracted by CNN are not ready for the dictionary re-encoding (Eq. (11)) due to the large gap between vision and language. To this end, we propose a Multi-modal Attentional Graph Convolution Network (MAGCN) to map the visual features $\mathcal{V}$ into a set of scene graph-modulated features $\mathcal{V}'$.

Here, the scene graph $\mathcal{G}_{\mathcal{I}}$ is extracted by an image scene graph parser that contains an object proposal detector, an attribute classifier, and a relationship classifier. In our implementation, we use Faster-RCNN as the object detector [27], MOTIFS relationship detector [25] as the relationship classifier, and we use our attribute classifier: an FC-ReLU-FC-Sigmoid network head for multi-label

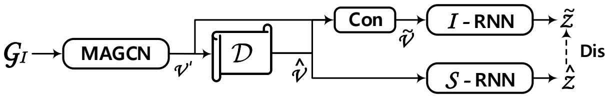


Fig. 5. The sketch of Knowledge Distillation, where **Con** denotes the concatenation operation.

classification. The image-parsed $\mathcal{G}_{\mathcal{I}}$ is different from the sentence-parsed $\mathcal{G}_{\mathcal{S}}$ in the sense that the object node o_i here contains knowledge from two domains: the semantic domain (the predicted category of this object) and the visual domain (the RoI feature extracted from the pre-trained Faster RCNN). Thus we have two different representations: the label embedding $\boldsymbol{e}_{o_i}$ and the RoI feature $\boldsymbol{v}_{o_i}$ to represent o_i. Since the modality differences are important in a vision-language task like image captioning, we hope to get a fused feature which contains the knowledge of such modality differences. Thus, following [86], we fuse $\boldsymbol{e}_{o_i}$ with $\boldsymbol{v}_{o_i}$ into a new node feature $\boldsymbol{u}_{o_i}$:

$$\boldsymbol{u}_{o_i} = \text{ReLU}(\boldsymbol{W}_1\boldsymbol{e}_{o_i} + \boldsymbol{W}_2\boldsymbol{v}_{o_i}) - (\boldsymbol{W}_1\boldsymbol{e}_{o_i} - \boldsymbol{W}_2\boldsymbol{v}_{o_i})^2, \quad (15)$$

where $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are the trainable matrices, here the $(\cdot)^2$ is the elementwise square operation. In this formula, the first term is a general operation to fuse the features of two different modalities, while the second part calculates their differences. The rest attribute and relationship node representations $\boldsymbol{u}_a$ and $\boldsymbol{u}_r$ are the learnable label embeddings of the corresponding attribute and relationship labels which are predicted by our attribute and relationship classifiers, respectively. The differences between the two scene graphs generated from $\mathcal{I}$ and $\mathcal{S}$ are visualized in Fig. 1, where $\mathcal{G}_{\mathcal{I}}$ is usually much simpler and noisier than $\mathcal{G}_{\mathcal{S}}$, e.g., there are so many relationships “on” in $\mathcal{G}_{\mathcal{I}}$.

Similar to AGCN used in Section 4.2, MAGCN also has an ensemble of four functions f_r, f_a, f_s and f_o, each of which has the same structure: FC-ReLU-FC with independent parameters. The computations of the relationship, attribute and object embeddings are similar to Eq. (6), Eq. (7), and Eq. (9) with the inputs $\{\boldsymbol{u}_o, \boldsymbol{u}_a, \boldsymbol{u}_r\}$. By replacing the functions $g_{\#}$ by $f_{\#}$ and the inputs $\boldsymbol{e}_{\#}$ by $\boldsymbol{u}_{\#}$, we can calculate the multi-modal graph embeddings of $\mathcal{G}_{\mathcal{I}}$. Similarly, as in Section 4.5, the embeddings are grouped into three sets: $\mathcal{V}_R^{'}, \mathcal{V}_A^{'}, \mathcal{V}_O^{'}$ and these three sets are further grouped into an embedding set $\mathcal{V}^{'}$. Then we adopt Eq. (11) to re-encode each representation $\boldsymbol{v}'$ in $\mathcal{V}'$ with respect to the corresponding dictionary (e.g., using $\mathcal{D}_R$ to re-encode the embeddings in $\mathcal{V}_R^{'}$) to get $\hat{\boldsymbol{v}}$ and group these re-encoded representations as $\hat{\mathcal{V}}$:

$$\hat{\mathcal{V}} = R(\mathcal{V}'; \mathcal{D}). \quad (16)$$

5.2 Knowledge Distillation

We deploy another top-down attention structure [3] $\mathcal{I}$-RNN for generating captions and the network structure is the same as Eq. (13) and (14) but with different inputs and parameters. Specifically, in $\mathcal{I}$-RNN, the input feature set $\mathcal{X}$ in Eq. (14) is set to $\widetilde{\mathcal{V}}$, which is the concatenation of $\mathcal{V}'$ and $\hat{\mathcal{V}}$:

$$\widetilde{\mathcal{V}} = \{\widetilde{\boldsymbol{v}}_1, ..., \widetilde{\boldsymbol{v}}_N\} = \{[\boldsymbol{v}_1', \hat{\boldsymbol{v}}_1], ..., [\boldsymbol{v}_N', \hat{\boldsymbol{v}}_N]\}, \quad (17)$$

where $[\cdot,\cdot]$ means the concatenation operation and N is the number of the representations in $\mathcal{V}'$ (or $\hat{\mathcal{V}}$). In this way, $\mathcal{V}'$ and $\hat{\mathcal{V}}$ respectively provide visual clues from the image and high-level semantic contexts from language inductive bias to the decoder for correct and human-like captions.

Compared with $\mathcal{S}$-RNN (Section 4.5) where the input embedding $\boldsymbol{x}$ is a d-dimensional representation, the input embedding of $\mathcal{I}$-RNN $\widetilde{\boldsymbol{v}}$ is a $2d$-dimensional representation. Therefore, the sizes of the LSTM layers and the attention modules in $\mathcal{I}$-RNN are different from $\mathcal{S}$-RNN. As a result, we can not directly use the parameters of $\mathcal{S}$-RNN to initialize $\mathcal{I}$-RNN, which weakens the transferability of the inductive bias. Fortunately, Knowledge Distillation [75] provides another perspective about knowledge which is the mapping from input vectors to output vectors of a trained system. Inspired by this, we can treat the inductive bias as the mapping from input to output of the trained $\mathcal{S}$-RNN and distill such inductive bias to $\mathcal{I}$-RNN for more descriptive captions.

Specifically, when we apply Knowledge Distillation to train $\mathcal{I}$-RNN, as shown in Fig. 5, we first input the re-encoded representation set $\hat{\mathcal{V}}$ of an image into the trained $\mathcal{S}$-RNN, i.e., setting $\mathcal{X}$ to $\hat{\mathcal{V}}$ in Eq. (14) and then calculating Eq. (13) to get a series of word probabilities $\hat{\mathcal{Z}} = \{\hat{\boldsymbol{z}}_1, ..., \hat{\boldsymbol{z}}_T\}$ with the temperature set to 5 in the softmax layer, where $\hat{\boldsymbol{z}}_t$ is the output of the softmax layer at time step t. Meantime, we input the concatenation feature set $\widetilde{\mathcal{V}}$ of the same image into $\mathcal{I}$-RNN, i.e., setting $\mathcal{X}$ to $\widetilde{\mathcal{V}}$ in Eq. (14) to get a series of word probabilities $\widetilde{\mathcal{Z}} = \{\widetilde{\boldsymbol{z}}_1, ..., \widetilde{\boldsymbol{z}}_T\}$ with the same temperature. Then, we calculate the KL divergence between $\hat{\mathcal{Z}}$ and $\widetilde{\mathcal{Z}}$ as the Knowledge Distillation objective:

$$L_{KD} = \text{KL}(\hat{\mathcal{Z}}||\widetilde{\mathcal{Z}}) = \sum_{t=1}^{T} \text{KL}(\hat{\boldsymbol{z}}_t||\widetilde{\boldsymbol{z}}_t), \quad (18)$$

which teaches $\mathcal{I}$-RNN to reason as $\mathcal{S}$-RNN.

5.3 Training and Inference

Following the common practice in deep-learning feature transfer [87], [88], we used the SGAE pre-trained $\mathcal{D}$ in auto-encoder as the initialization for the $\mathcal{D}$ in our overall encoder-decoder for image captioning (Eq.(4)). In particular, we intentionally used a very small learning rate to fine-tune $\mathcal{D}$ to impose the sharing purpose. The overall training loss is hybrid: we first used $L_{XE} + 25L_{KD}$ for 20 epochs and then used $-R_{RL} + 5L_{KD}$ for another 80 epochs; where L_{XE} and R_{RL} are cross-entropy loss (Eq. (2)) and RL-based reward (Eq. (3)), respectively. We set the importance weight of L_{KD} to 25 in the first 20 epochs since the magnitudes of the gradients of L_{KD} scale as $1/5^2$ (temperature is set to 5) [75] and then empirically set this importance weight to 5 in the other 80 epochs. The learning rate was initialized to $5e^{-5}$ for $\mathcal{D}$ and $5e^{-4}$ for the other parameters. All these learning rates were decayed by 0.8 for every 5 epochs. The batch size was set to 100 and Adam optimizer [85] was used. For inference in language generation, we adopted the beam search strategy [8] with a beam size of 5.

6 Experiments

6.1 Datasets and Metrics

MS-COCO [28]. There are two standard splits of MS-COCO: the official online test split and the 3rd-party Karpathy split [89]. The first split has $82,783/40,504/40,775$ train/val/test images and the second split has $113,287/5,000/5,000$ train/val/test images. Each of image has about 5 human labelled captions for training. For captions, we used the following steps to pre-process them: we first tokenized the texts on white space; then we change all the words to lowercase; we also deleted the words which appear less than 5 times; at last, we trimmed each caption to a maximum of 16 words. This results in a vocabulary of $10,369$ words.

Visual Genome [50] (VG). This dataset has abundant scene graph annotations, e.g., object categories, object attributes, and pairwise relationships, which can be exploited to respectively train the object proposal detector, the attribute classifier, and the relationship classifier, as our image scene graph parser (see Section 5.1). Since the object, attribute, and relationship annotations are very noisy in VG dataset, we filtered them by keeping the objects, attributes, and relationships which appear more than $2,000$ times in the training set. After filtering, the remained 305 objects, 103 attributes, and 64 relationships were used to train our object detector, attribute classifier and relationship classifier, respectively.

SentiCap [90]. Compared with MS-COCO and VG which only contain factual descriptions (e.g., "a girl" or "a street") of the images, this dataset has a quite different text distribution: it provides 4,892 positive (e.g., "a beautiful girl") and 3,977 negative (e.g., "an ugly street") sentiment captions.

We used a similar way as in MS-COCO to pre-process the captions in VG and SentiCap. It is noteworthy that we only exploited these additional text descriptions in the ablation studies in Section 6.2.2, which were not used for the training in the other experiments.

Metrics. We employed five standard automatic evaluation metrics to measure the similarities between the generated captions and the ground-truth captions: CIDEr-D [29], BLEU [91], METEOR [92], ROUGE [93] and SPICE [23].

6.2 Ablation Studies

We conducted extensive ablation studies for architecture (Section 6.2.1), language corpus (Section 6.2.2), and sentence reconstruction quality (Section 6.2.3). For simplicity, we use **SGAE-KD** to denote our SGAE-based encoder-decoder captioning model trained by the Knowledge Distillation objective L_{KD} (Eq. (18)) to distinguish from **SGAE**, where L_{KD} was not used.

6.2.1 Architecture

Comparing Methods. To quantify the importance of the proposed AGCN, MAGCN, the dictionary set $\mathcal{D}$, and the Knowledge Distillation objective, we gradually incorporated these techniques into a strong baseline and compared their performances:

Base: We followed the pipeline given in Eq. (1) and the network in Eq. (13) without using AGCN, MAGCN, $\mathcal{D}$, and L_{KD}. This baseline is the benchmark for other ablation baselines. Importantly, this is also the reimplemented version of Up-Down model [3].

MAGCN: We added MAGCN to compute the multi-modal embedding set $\mathcal{V}'$ for validating the importance of MAGCN. We also compared MAGCN with the one used in our preliminary work [30]: **MGCN**, where attentional gates were not incorporated into the GCN. Noteworthy, this is one kind of the reimplemented version of GCN-LSTM model [42] with a much smaller batch size (100) compared with the original one (1024).

$\mathcal{D}$ **w/o AGCN**: We learned $\mathcal{D}$ by using Eq. (5), while AGCN was not used and only the word embeddings of $\mathcal{S}$ were input to the decoder. Also, MAGCN was not used in Eq. (4). This baseline is designed for validating the importance of AGCN and MAGCN.

TABLE 1
The performances of various methods on MS-COCO Karpathy split. The metrics: B@N, M, R, C and S denote BLEU@N, METEOR, ROUGE-L, CIDEr-D and SPICE. The best results are marked in boldface.

Models	B@1	B@4	M	R	C	S
SCST [8]	–	34.2	26.7	55.7	114.0	–
LSTM-A [41]	78.6	35.5	27.3	56.8	118.3	20.8
StackCap [44]	78.6	36.1	27.4	–	120.4	–
Up-Down [3]	79.8	36.3	27.7	56.9	120.1	21.4
CAVP [4]	–	38.6	28.3	58.5	126.3	21.6
RFNet [84]	80.4	37.9	28.3	58.3	125.7	21.7
LBPF [38]	80.5	38.3	28.5	58.4	127.6	22.0
CNM [5]	80.8	38.7	28.4	58.7	127.4	21.8
GCN-LSTM [42]	80.5	38.2	28.5	58.3	127.6	22.0
Base	79.9	36.8	27.7	57.0	120.6	20.9
MGCN	80.2	37.2	27.9	57.5	123.4	21.2
MAGCN	80.3	37.4	28.0	57.6	123.9	21.3
$\mathcal{D}$ w/o AGCN	80.2	37.3	27.8	58.0	124.2	21.4
$\mathcal{D}\#1$+AGCN	80.5	37.5	28.0	58.2	125.9	21.5
$\mathcal{D}\#3$+GCN	80.5	37.5	28.1	58.3	126.0	21.4
$\mathcal{D}\#3$+AGCN	80.6	37.8	28.1	58.5	126.4	21.7
SGAE-pre [30]	80.8	38.4	28.4	58.6	127.8	22.1
SGAE	80.9	38.5	28.5	**58.8**	128.4	22.2
SGAE-KD	**81.0**	**38.8**	**28.8**	**58.8**	**129.6**	**22.4**

$\mathcal{D}\#3$**+AGCN:** Compared to Base, $\mathcal{D} = \{\mathcal{D}_R, \mathcal{D}_A, \mathcal{D}_O\}$ was learned by using AGCN in Eq. (5). And MAGCN was not used in Eq. (4). This baseline is designed for validating the importance of the shared $\mathcal{D}$. We also compared it with another two baselines: $\mathcal{D}\#3$**+GCN** where attentional gates were not used and $\mathcal{D}\#1$**+AGCN** where only one dictionary $\mathcal{D}$ was used.

SGAE: When we used all the AGCN, MAGCN, and the dictionary set $\mathcal{D}$, the baseline is called SGAE. We also compare it with the one in our preliminary version **SGAE-pre** [30], which used GCN, MGCN, and only one dictionary.

SGAE-KD: When we applied L_{KD} (Eq. (18)) to train $\mathcal{I}$-RNN based on the baseline SGAE, we got our integral model.

Results. The bottom of Table 1 shows the performances of the ablation baselines on MS-COCO Karpathy split. Compared with Base and our preliminary version SGAE-pre, our SGAE-KD can boost the CIDEr-D by absolute 9 and 1.8, respectively. By comparing MAGCN, $\mathcal{D}$ w/o AGCN, and $\mathcal{D}\#1$+AGCN with Base, we can find that all the performances are improved, which demonstrate that the proposed MAGCN, AGCN, and $\mathcal{D}$ are all indispensable for improving the performances. In addition, by refining the graph convolution operations (MAGCN vs. MGCN and AGCN vs. GCN) and partitioning the dictionary to more fine-grained ones ($\mathcal{D}\#3$+AGCN vs. $\mathcal{D}\#1$+AGCN), we also achieve certain improvements. These observations validate the importance of the refinements of the components in SGAE. Furthermore, by applying the Knowledge Distillation strategy to teach $\mathcal{I}$-RNN to reason as $\mathcal{S}$-RNN, the performance of the whole system is also improved, which suggests that the transferability of the language inductive bias is further enhanced.

Qualitative Examples. Fig. 6 shows 6 examples of the generated captions using different models. We can see that compared with captions generated by Up-Down (Base) [3], the descriptions of GCN-LSTM (MAGCN) [42] usually contain more about object attributes and pairwise relationships. For captions generated by SGAE [30], they are more complex and descriptive. For example, in Fig. 6 (a), the word "busy" is used to describe the heavy traffic; in (b) the scene "forest" can be deduced from "trees"; and in (c), the attribute "ripe" is inferred from the color of the banana. It can also be observed that our SGAE-KD generates more descriptive captions which contain more interesting details than SGAE, e.g., the action "perch" which is more exclusive for the animal bird is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2020.3042192, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 2
The performances of using different language corpora.

Models	B@1	B@4	M	R	C	S
Base	79.9	36.8	27.7	57.0	120.6	20.9
VG	80.4	38.1	28.3	58.4	125.1	21.7
COCO	81.0	38.8	28.8	58.8	129.6	22.4
Senti#Pos	80.1	37.0	27.9	57.2	122.8	21.2
Senti#Neg	80.2	37.2	28.0	57.4	123.1	21.2
VG+COCO	**81.3**	**38.9**	**29.1**	**58.9**	**130.3**	**22.5**

generated in (d); the word "pile" more accurately describes the state of the rocks in (e); and the word "spray" instead of "stand" is used to describe the action between the hydrant and the water in (f).

6.2.2 *Language Corpus*

Comparing Methods. To test the performances of transferring the inductive bias from other language corpora, we also learned $\mathcal{D}$ from the text corpora of VG [50] and SentiCap [90]. Among them, VG has a similar language corpus as COCO while SentiCap's corpus is very dissimilar. By learning $\mathcal{D}$ from them, we can get a more comprehensive understanding of the effects brought by the pre-trained $\mathcal{D}$. After learning $\mathcal{D}$, we inserted it in Eq. (16) to get $\hat{\mathcal{V}}$ for captioning. The results are demonstrated in Table 2, where **VG/COCO/VG+COCO** learn $\mathcal{D}$ by texts of VG/COCO/VG+COCO and **Senti#Pos/Senti#Neg** use positive and negative captions from SentiCap, respectively.

Quantitative Results. By comparing Base with the models using additional text corpora in Table 2, we find that using additional texts can boost the performances even when a dissimilar corpus like SentiCap is used (Senti#Pos vs. Base or Senti#Neg vs. Base). We can also see that the more similar the language corpus is, the higher the scores can be achieved. For example, COCO is obviously better than the other single language corpus model (COCO vs. VG) and the similar corpus VG outperforms the dissimilar ones (VG vs. Senti#Pos or Senti#Neg). Such observations suggest that $\mathcal{D}$ can memorize common inductive bias from the additional unmatched language corpus or specific inductive bias from a matched language corpus for generating better captions. Also, when we compose two similar corpora VG and COCO to learn $\mathcal{D}$, a consistent improvement can be achieved compared with the model with only one corpus (COCO+VG vs COCO).

Human Evaluation. The purpose for introducing human evaluation is to avoid the performance bias towards COCO since we measure the quantitative performance based on ground-truth captions of COCO, while our dictionary $\mathcal{D}$ could come from a different corpus with a very different style from COCO. In particular, we conducted human evaluation with 50 annotators who are fluent in English and unaware of our work. We divided 50 annotators into 5 groups and asked each group to respectively compare the following pairwise methods: Base vs. COCO, Base vs. VG, Base vs. Senti, COCO vs. VG, COCO vs. Senti. We randomly sampled 100 images from the COCO test set and assigned these same images to each group with the captions generated by the corresponding two comparing methods. For Senti captions, one half of them were generated by Senti#Pos and another half were generated by Senti#Neg. For each group with 10 annotators, it was further divided into two sub-groups that every 5 annotators were given the same 50 images for comparing.

TABLE 3
Results of human evaluation, where each row compares two methods. In each column, "1st"/"2nd" lists the percentage of votes which prefer the fist/second method in that row and "Similar" denotes the votes for "two captions are similar".

Models	1st	2nd	Similar
COCO vs. Base	69%	12%	19%
VG vs. Base	65%	14%	21%
Senti vs. Base	73%	11%	16%
COCO vs. VG	36%	29%	35%
COCO vs. Senti	34%	38%	28%

TABLE 4
The performances of the sentence reconstruction using different scene graphs.

Models	B@1	B@4	M	R	C	S
$\hat{\mathcal{X}}$	90.3	53.8	34.3	66.5	153.2	30.6
$\mathcal{X}$	**93.9**	**65.2**	**38.5**	**71.8**	**177.0**	**34.3**
SGAE-KD	81.0	38.8	28.8	58.8	129.6	22.4

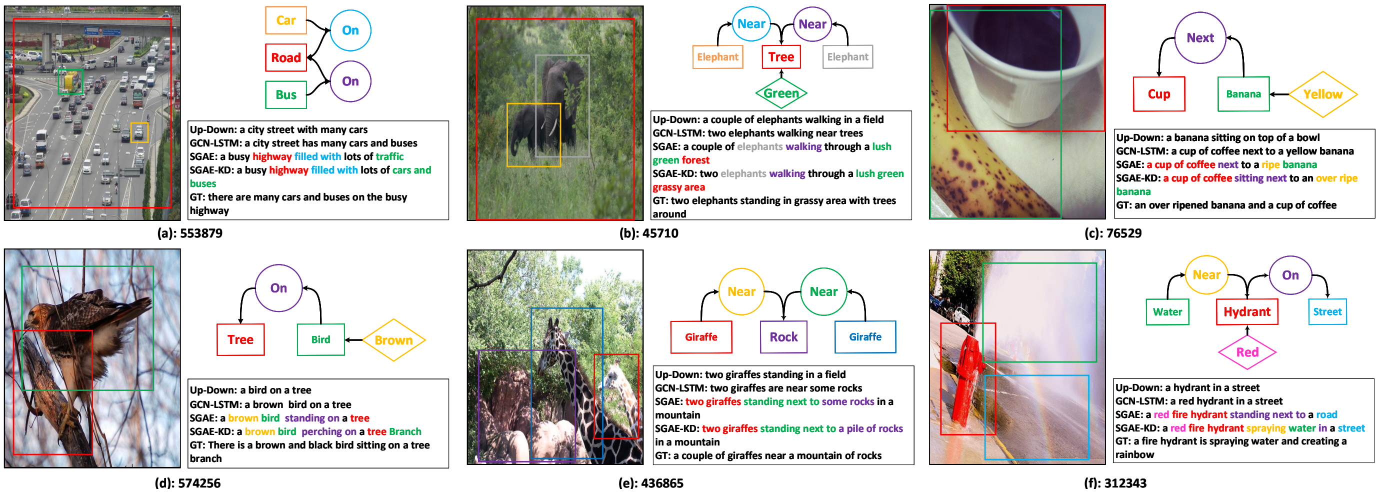
During the comparisons, each annotator was asked that which one of the two captions is more descriptive, e.g., containing or inferring more interesting details ("a lush forest" vs. "a forest"), using more exclusive words ("a seagull" vs. "a bird"), using more summary phrases ("busy street" vs. "many people in a street"). We set three options for the annotators to vote: "the first caption is better", "the second caption is better", and "two captions are similar". In this process, the annotators did not know where these captions are from. After collecting the votes, we calculated the ratio and reported them in Table 3.

Analyses of Human Evaluation. 1) For the comparisons of COCO/VG/Senti vs. Base, we can see that the models using $\mathcal{D}$ generate more descriptive captions, getting 69%/65%/73% votes, respectively. 2) The comparison of COCO vs. VG suggests using COCO is better, although the gap is not as large as that in the case of COCO vs. Base. 3) More interestingly, the comparison of COCO vs. Senti suggests that the generated captions by using Senti are a little more descriptive than using COCO. This might be because the language inductive bias from the sentiment captions evokes the annotators' emotions to vote more for Senti, despite the sentiment words do not return high quantitative scores in Table 2.

Qualitative Examples. Fig. 7 shows 4 examples in human evaluation, where the captions are generated by four methods. Generally, compared with captions generated by Base, the captions of VG, COCO, and Senti are more descriptive. Specifically, the captions generated by different language corpora reflect the inductive bias of such corpora: COCO and VG use some factual language to describe the image while Senti prefers to convey different sentiments. For example, in Fig. 7 (a), COCO and VG describe there is "a fire hydrant" and its colour, while Senti tags this "fire hydrant" with the adjective "ugly", which reflects the negative sentiment.

6.2.3 *Sentence Reconstruction*

Comparing Methods. We investigated how well the sentences are reconstructed in training SGAE (Eq. (5)) with or without the re-encoder $R(\mathcal{X};\mathcal{D})$, that is, we denote $\hat{\mathcal{X}}$ as the pipeline using $\mathcal{D}$ and $\mathcal{X}$ as the pipeline directly reconstructing sentences from their scene graph embeddings. Such results are given in Table 4.



Up-Down: a city street with many cars
GCN-LSTM: a city street has many cars and buses
SGAE: a busy highway filled with lots of traffic
SGAE-KD: a busy highway filled with lots of cars and buses
GT: there are many cars and buses on the busy highway

(a): 553879

Up-Down: a couple of elephants walking in a field
GCN-LSTM: two elephants walking near trees
SGAE: a couple of elephants walking through a lush green forest
SGAE-KD: two elephants walking through a lush green grassy area
GT: two elephants standing in grassy area with trees around

(b): 45710

Up-Down: a banana sitting on top of a bowl
GCN-LSTM: a cup of coffee next to a yellow banana
SGAE: a cup of coffee next to a ripe banana
SGAE-KD: a cup of coffee sitting next to an over ripe banana
GT: an over ripened banana and a cup of coffee

(c): 76529

Up-Down: a bird on a tree
GCN-LSTM: a brown bird on a tree
SGAE: a brown bird standing on a tree
SGAE-KD: a brown bird perching on a tree Branch
GT: There is a brown and black bird sitting on a tree branch

(d): 574256

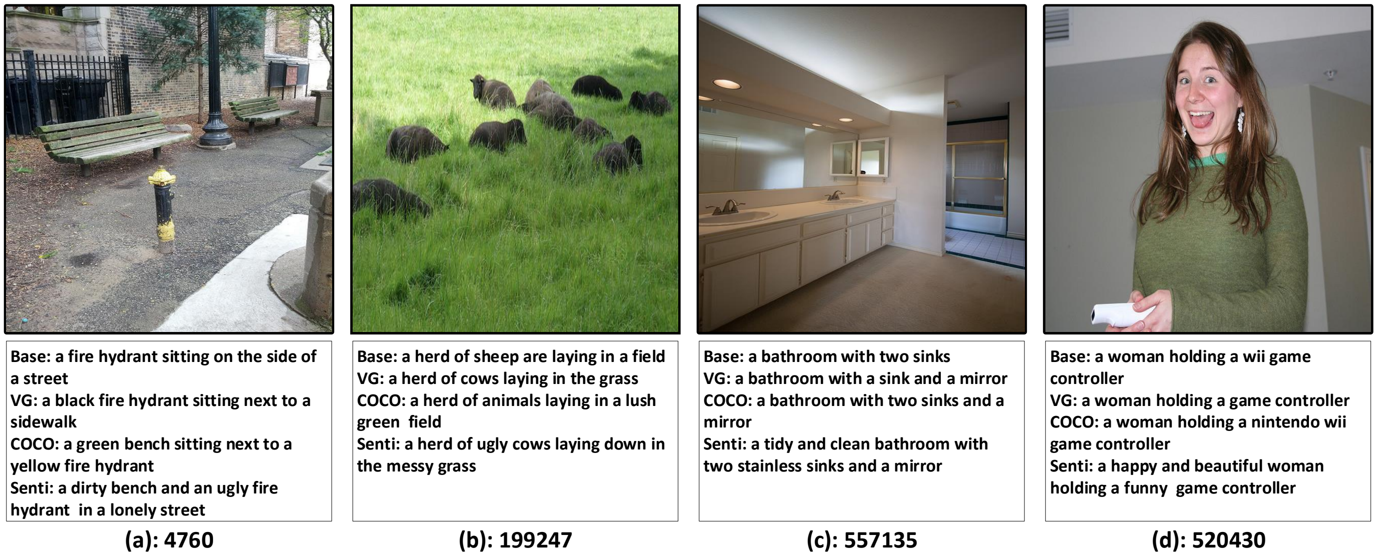
Up-Down: two giraffes standing in a field
GCN-LSTM: two giraffes are near some rocks
SGAE: two giraffes standing next to some rocks in a mountain
SGAE-KD: two giraffes standing next to a pile of rocks in a mountain
GT: a couple of giraffes near a mountain of rocks

(e): 436865

Up-Down: a hydrant in a street
GCN-LSTM: a red hydrant in a street
SGAE: a red fire hydrant standing next to a road
SGAE-KD: a red fire hydrant spraying water in a street
GT: a fire hydrant is spraying water and creating a rainbow

(f): 312343

Fig. 6. Qualitative examples of different methods. The comparing methods are Base, the reimplemented version of Up-Down [3], MAGCN, the reimplemented version of GCN-LSTM [42], SGAE, the upgraded version of our preliminary work [30], and our SGAE-KD. For each figure, the image scene graph is pruned to avoid clutter. The id refers to the image id in MS-COCO. Word colors correspond to nodes in the detected scene graphs.



Base: a fire hydrant sitting on the side of a street
VG: a black fire hydrant sitting next to a sidewalk
COCO: a green bench sitting next to a yellow fire hydrant
Senti: a dirty bench and an ugly fire hydrant in a lonely street

(a): 4760

Base: a herd of sheep are laying in a field
VG: a herd of cows laying in the grass
COCO: a herd of animals laying in a lush green field
Senti: a herd of ugly cows laying down in the messy grass

(b): 199247

Base: a bathroom with two sinks
VG: a bathroom with a sink and a mirror
COCO: a bathroom with two sinks and a mirror
Senti: a tidy and clean bathroom with two stainless sinks and a mirror

(c): 557135

Base: a woman holding a wii game controller
VG: a woman holding a game controller
COCO: a woman holding a nintendo wii game controller
Senti: a happy and beautiful woman holding a funny game controller

(d): 520430

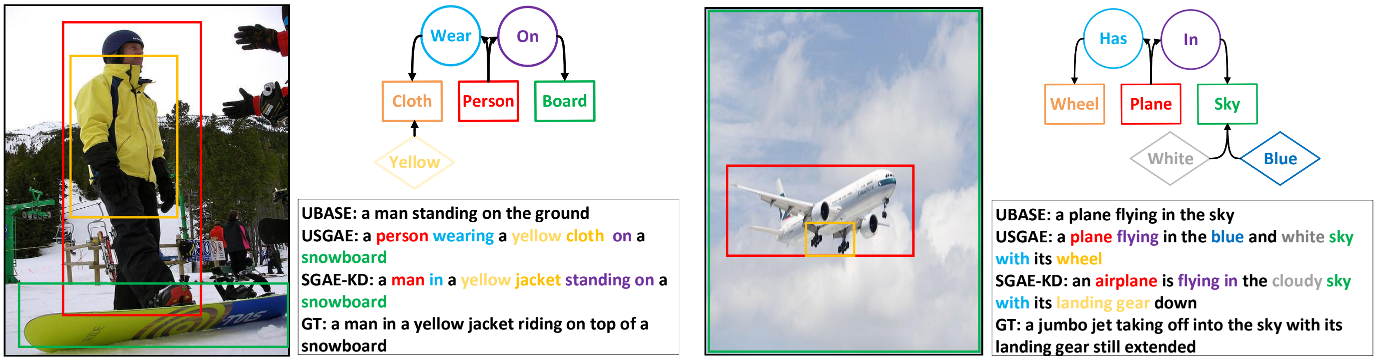
Fig. 7. Captions generated by using different language corpora. The id refers to the image id in MS-COCO.

Analysis. As we can see, the performances of using direct scene graph features $\mathcal{X}$ are much better than those ($\hat{\mathcal{X}}$) imposed with $\mathcal{D}$ for re-encoding. This is reasonable since $\mathcal{D}$ regularizes the reconstruction and thus creates knowledge loss in this pure language domain. Interestingly, the gap between $\hat{\mathcal{X}}$ and SGAE-KD suggests that we should develop a more powerful image scene graph parser for improving the quality of $\mathcal{G}_{\mathcal{I}}$ in Eq. (4) and design a stronger re-encoder for extracting more preserved inductive bias when only low-quality visual scene graphs are available.

6.3 Comparisons with State-of-The-Arts

Comparing Methods. Though there are various captioning models developed in recent years, we only compare SGAE-KD with some encoder-decoder methods trained by cross-entropy loss (Eq. (2)) and RL-based reward (Eq. (3)) due to their superior performances. Specifically, we compare our methods with **SCST** [8], **StackCap** [44], **Up-Down** [3], **LSTM-A** [41], **GCN-LSTM** [42], **CAVP** [4], **RFNet** [84], **LBPF** [38], and **CNM** [5]. Among these methods, SCST and Up-Down are two strong baselines since almost all the following captioning models use the advanced self-critic reward provided by SCST and the visual features provided by Up-Down. Compared with SCST, StackCap proposes a more complex RL-based reward for learning captions with more details. LSTM-A, GCN-LSTM, CAVP, and CNM try to exploit the knowledge of visual scene graphs, where LSTM-A and GCN-LSTM respectively exploit attribute and relationship information, CAVP tries to recognize the pairwise relationships in the decoder, and CNM learns to collocate various types of modules to generate object, attribute, and relation words for composing the caption. The results of these methods are reported in Table 1, where the models are trained by cross-entropy loss (Eq. (2)) first and then by RL-based reward (Eq. (3)) and Table 5, where the models are only trained by cross-entropy loss (Eq. (2)).

Analysis. From Table 1 and 5, we can see that our single model outperforms the other image captioners. In Table 1, when both cross-entropy loss and RL-based reward are used, our single model



UBASE: a man standing on the ground
USGAE: a person wearing a yellow cloth on a snowboard
SGAE-KD: a man in a yellow jacket standing on a snowboard
GT: a man in a yellow jacket riding on top of a snowboard

(a): 247625

UBASE: a plane flying in the sky
USGAE: a plane flying in the blue and white sky with its wheel
SGAE-KD: an airplane is flying in the cloudy sky with its landing gear down
GT: a jumbo jet taking off into the sky with its landing gear still extended

(b): 262800

Fig. 8. Two examples of unpaired image captioning. The id refers to the image id in MS-COCO.

TABLE 5
The performances of various methods on MS-COCO Karpathy split trained by cross-entropy loss only.

Models	B@1	B@4	M	R	C	S
SCST [8]	–	30.0	25.9	53.4	99.4	–
LSTM-A [41]	73.4	32.6	25.4	54.0	100.2	18.6
StackCap [44]	76.2	35.2	26.5	–	109.1	–
Up-Down [3]	77.2	36.2	27.0	56.4	113.5	20.3
RFNet [84]	77.4	37.0	27.9	57.3	116.3	20.8
LBPF [38]	77.8	37.4	28.1	57.5	116.4	21.2
GCN-LSTM [42]	77.3	36.8	27.9	57.0	116.3	20.9
CNM [5]	77.6	37.1	27.9	57.3	116.6	20.8
Base	76.8	36.1	27.1	56.3	113.1	20.3
MAGCN	77.3	36.4	27.3	56.7	114.4	20.6
SGAE-pre [30]	77.6	36.9	27.7	57.2	116.7	20.9
SGAE	77.8	37.0	27.9	57.3	116.9	21.0
SGAE-KD	**78.2**	**37.3**	**28.1**	**57.4**	**117.1**	**21.3**

achieves a new state-of-the-art score among all the compared methods in terms of CIDEr-D, reaching 129.6. By transferring the inductive bias from the pure language domain to the vision-language domain, our method still has better performances even our decoder is not as sophisticated as CVAP or LBPF, even we do not use more advanced RL-based reward as StackCap, and even we do not mix up different visual features as the input as RFNet or CNM. Moreover, compared with GCN-LSTM which sets the batch size to 1,024, our smaller batch size (100) still leads to higher performances. Table 6 reports the performances of different methods test on the official server. Compared with the published captioning methods, our single model achieves the highest CIDEr-D c40 score, 126.6.

6.4 Unpaired Image Captioning

6.4.1 Comparing Methods

We designed the following ablation methods to validate the contributions of the scene graphs and the dictionary set $\mathcal{D}$ in unpaired image captioning. To distinguish the unpaired image captioners with our paired image captioners in Section 6.2.1, we add the capital letter “U” in the names of the following models.

UBase: When training, we only input the node embeddings of the sentence scene graph $\mathcal{G}_\mathcal{S}$ into the auto-encoder for reconstructing, where these embeddings were the word embeddings of the one-hot node labels. When testing, we only input the node embeddings of the image scene graph $\mathcal{G}_\mathcal{I}$ into the auto-encoder for captioning. Also, we did not use the dictionary set $\mathcal{D}$ in this baseline.

UMAGCN: Compared with UBase, we exploited the MAGCN for transforming both $\mathcal{G}_\mathcal{S}$ and $\mathcal{G}_\mathcal{I}$ to embeddings, which were input into the auto-encoder for captioning. Also, we did not use $\mathcal{D}$ in this baseline. This is an upgraded version of [47] without adversarial learning objective.

USGAE: We followed Eq. (12) to generate the captions. Compared with UMAGCN, the graph embeddings were re-encodered by R and then the re-mapped embeddings were used for captioning.

We also compare our USGAE with the following state-of-the-arts: **Pivoting** [45], **UIC** [46], and **SME** [48]. For fair comparisons, we only compare our USGAE with the models which are not trained by adversarial learning objectives since different methods have different adversarial learning strategies and it is beyond the scope of our research. Besides these models, we also list the results of SGAE-KD (Section 6.2.1) to show the gap between the unpaired case and the paired case.

6.4.2 Results and Analysis

Table 7 reports the performances of various methods of unpaired image captioning. We can observe that our USGAE achieves new state-of-the-art scores among all the unpaired image captioning methods. More importantly, our USGAE can complement other models by using our auto-encoder as initialization followed by fine-tuning with specific training strategies. By comparing UMAGCN with UBase and USGAE with UMAGCN, we can conclude that both the scene graph representations and the dictionary set play indispensable roles in transferring inductive bias from the pure language domain to the vision-language domain. Specifically, scene graph representations contain more comprehensive knowledge than single-node embeddings, which build a stronger bridge between two domains. The dictionary set $\mathcal{D}$ filters the noisy $\mathcal{G}_\mathcal{I}$ by the re-encoder, thus narrowing the gap between two domains. Furthermore, compared with the paired image captioning SGAE-KD, the upside potential for unpaired image captioning is huge.

Fig. 8 shows two examples about unpaired image captioning. We can observe that UBase only lists a few detected objects as the caption which does not contain any interesting details. In contrast, our USGAE can generate more descriptive captions, even comparable to SGAE-KD which is trained by paired supervisions.

TABLE 6
The performances of various methods on the online MS-COCO test server.

Model	B@3		B@4		M		R-L		C-D	
Metric	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
SCST [8]	47.0	75.9	35.2	64.5	27.0	35.5	56.3	70.7	114.7	116.0
LSTM-A [41]	47.6	76.5	35.6	65.2	27.0	35.4	56.4	70.5	116.0	118.0
StackCap [44]	46.8	76.0	34.9	64.6	27.0	35.6	56.2	70.6	114.8	118.3
Up-Down [3]	49.1	79.4	36.9	68.5	27.6	36.7	57.1	72.4	117.9	120.5
RFNet [84]	50.1	80.4	38.0	69.2	28.2	37.2	58.2	73.1	122.9	125.1
CAVP [94]	50.0	79.7	37.9	69.0	28.1	37.0	58.2	73.1	121.6	123.8
SGAE-pre [30]	**50.1**	79.6	37.8	68.7	28.1	37.0	58.2	73.1	122.7	125.5
CNM [5]	–	–	37.9	68.4	28.1	36.9	58.3	72.9	123.0	125.3
SGAE-KD	**50.1**	**79.9**	**38.2**	**69.3**	**28.7**	**37.9**	**58.4**	**73.5**	**124.5**	**126.6**

TABLE 7
The performances of unpaired image captioning.

Models	B@1	B@4	M	R	C	S
Pivoting [45]	46.2	5.4	13.2	–	17.7	–
UIC [46]	53.8	15.6	16.6	39.9	46.7	9.6
SME [48]	59.7	16.6	18.3	43.8	53.8	10.8
UBase	47.8	7.2	14.1	32.2	23.4	7.3
UMAGCN [47]	53.9	12.9	16.8	40.0	43.9	12.4
USGAE	**60.8**	**17.1**	**19.1**	**43.8**	**55.1**	**12.8**
SGAE-KD	81.0	38.8	28.8	58.8	129.6	22.4

7 Conclusions

We proposed to incorporate the language inductive bias — a prior for more human-like language generation — into the prevailing encoder-decoder framework for image captioning. In particular, we presented a novel unsupervised learning method: Scene Graph Auto-Encoder (SGAE) for embedding the inductive bias into a dictionary set, which can be shared as a re-encoder for language generation. Besides this, we also distilled the inductive bias into the language decoder of the image captioner to further improve the transferability. Thanks to these techniques, our image captioner generates more descriptive and human-like captions. We validated the SGAE-based framework by extensive ablations and comparisons with the state-of-the-art image captioners on MS-COCO captioning benchmark and the more challenging unpaired image captioning task. We believe that SGAE is a general solution for capturing the language inductive bias, which can also benefit other vision-language tasks.

References

[1] R. Luo, B. Price, S. Cohen, and G. Shakhnarovich, “Discriminability objective for training descriptive captions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6964–6974. 1

[2] J. Lu, J. Yang, D. Batra, and D. Parikh, “Neural baby talk,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7219–7228. 1, 3, 4

[3] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, “Bottom-up and top-down attention for image captioning and visual question answering,” in *CVPR*, vol. 3, no. 5, 2018, p. 6. 1, 2, 3, 4, 6, 7, 8, 10, 11, 12

[4] Z.-J. Zha, D. Liu, H. Zhang, Y. Zhang, and F. Wu, “Context-aware visual policy network for fine-grained image captioning,” *IEEE transactions on pattern analysis and machine intelligence*, 2019. 1, 3, 8, 10

[5] X. Yang, H. Zhang, and J. Cai, “Learning to collocate neural modules for image captioning,” in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 1, 3, 8, 10, 11, 12

[6] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *ICLR*, 2015. 1, 2

[7] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” in *International conference on machine learning*, 2015, pp. 2048–2057. 1, 2, 4

[8] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, “Self-critical sequence training for image captioning,” in *CVPR*, vol. 1, no. 2, 2017, p. 3. 1, 2, 3, 4, 7, 8, 10, 11, 12

[9] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, “Sequence level training with recurrent neural networks,” 2015. 1, 3

[10] R. Hu, P. Dollár, K. He, T. Darrell, and R. Girshick, “Learning to segment every thing,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1

[11] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, “Panoptic segmentation,” *arXiv preprint arXiv:1801.00868*, 2018. 1

[12] J. Redmon and A. Farhadi, “Yolo9000: Better, faster, stronger,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 6517–6525. 1

[13] D. Marr, “Vision: A computational investigation into the human representation and processing of visual information. mit press,” *Cambridge, Massachusetts*, 1982. 1

[14] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, “Building machines that learn and think like people,” *Behavioral and Brain Sciences*, vol. 40, 2017. 2

[15] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner *et al.*, “Relational inductive biases, deep learning, and graph networks,” *arXiv preprint arXiv:1806.01261*, 2018. 2

[16] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg, “Babytalk: Understanding and generating simple image descriptions,” in *CVPR*, 2011. 2

[17] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt *et al.*, “From captions to visual concepts and back,” in *CVPR*, 2015. 2

[18] J. Johnson, A. Gupta, and L. Fei-Fei, “Image generation from scene graphs,” *arXiv preprint*, 2018. 2, 3

[19] Y.-S. Wang, C. Liu, X. Zeng, and A. Yuille, “Scene graph parsing as dependency parsing,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, 2018, pp. 397–407. [Online]. Available: http://aclweb.org/anthology/N18-1037 2, 3

[20] D. Marcheggiani and I. Titov, “Encoding sentences with graph convolutional networks for semantic role labeling,” in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 1506–1515. 2

[21] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, “Gated graph sequence neural networks,” *arXiv preprint arXiv:1511.05493*, 2015. 2

[22] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, “How powerful are graph neural networks?” in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=ryGs6iA5Km 2, 3, 5

[23] P. Anderson, B. Fernando, M. Johnson, and S. Gould, “Spice: Semantic propositional image caption evaluation,” in *European Conference on Computer Vision*. Springer, 2016, pp. 382–398. 2, 3, 4, 8

[24] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, “Graph r-cnn for scene graph generation,” in *European Conference on Computer Vision*. Springer, 2018, pp. 690–706. 2, 3, 5

[25] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi, “Neural motifs: Scene graph parsing with global context,” in *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition, 2018, pp. 5831–5840. 2, 3, 6

[26] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," in *Advances in Neural Information Processing Systems*, 2016, pp. 3630–3638. 2, 3, 5

[27] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99. 2, 3, 4, 6

[28] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755. 2, 7

[29] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4566–4575. 2, 4, 8

[30] X. Yang, K. Tang, H. Zhang, and J. Cai, "Auto-encoding scene graphs for image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 685–10 694. 2, 5, 8, 10, 11, 12

[31] P. Kuznetsova, V. Ordonez, A. C. Berg, T. L. Berg, and Y. Choi, "Collective generation of natural image descriptions," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 2012, pp. 359–368. 2

[32] M. Mitchell, X. Han, J. Dodge, A. Mensch, A. Goyal, A. Berg, K. Yamaguchi, T. Berg, K. Stratos, and H. Daumé III, "Midge: Generating image descriptions from computer vision detections," in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2012, pp. 747–756. 2

[33] S. Li, G. Kulkarni, T. L. Berg, A. C. Berg, and Y. Choi, "Composing simple image descriptions using web-scale n-grams," in *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 2011, pp. 220–228. 2

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778. 2, 4

[35] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112. 2

[36] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *CVPR*, 2015. 2, 4

[37] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing when to look: Adaptive attention via a visual sentinel for image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 6, 2017, p. 2. 3

[38] Y. Qin, J. Du, Y. Zhang, and H. Lu, "Look back and predict forward in image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8367–8375. 3, 6, 8, 10, 11

[39] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4651–4659. 3

[40] Z. Gan, C. Gan, X. He, Y. Pu, K. Tran, J. Gao, L. Carin, and L. Deng, "Semantic compositional networks for visual captioning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5630–5639. 3

[41] T. Yao, Y. Pan, Y. Li, Z. Qiu, and T. Mei, "Boosting image captioning with attributes," in *IEEE International Conference on Computer Vision, ICCV*, 2017, pp. 22–29. 3, 8, 10, 11, 12

[42] T. Yao, Y. Pan, Y. Li, and T. Mei, "Exploring visual relationship for image captioning," in *Computer Vision–ECCV 2018*. Springer, 2018, pp. 711–727. 3, 6, 8, 10, 11

[43] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 1171–1179. 3

[44] J. Gu, J. Cai, G. Wang, and T. Chen, "Stack-captioning: Coarse-to-fine learning for image captioning," *AAAI*, 2017. 3, 8, 10, 11, 12

[45] J. Gu, S. Joty, J. Cai, and G. Wang, "Unpaired image captioning by language pivoting," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 503–519. 3, 6, 11, 12

[46] Y. Feng, L. Ma, W. Liu, and J. Luo, "Unsupervised image captioning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 4125–4134. 3, 11, 12

[47] J. Gu, S. Joty, J. Cai, H. Zhao, X. Yang, and G. Wang, "Unpaired image captioning via scene graph alignments," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 10 323–10 332. 3, 11, 12

[48] I. Laina, C. Rupprecht, and N. Navab, "Towards unsupervised image captioning with shared multimodal embeddings," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7414–7424. 3, 11, 12

[49] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei, "Image retrieval using scene graphs," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3668–3678. 3

[50] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International Journal of Computer Vision*, vol. 123, no. 1, pp. 32–73, 2017. 3, 8, 9

[51] J. Shi, H. Zhang, and J. Li, "Explainable and explicit visual reasoning over scene graphs," *arXiv preprint arXiv:1812.01855*, 2018. 3

[52] D. Teney, L. Liu, and A. van den Hengel, "Graph-structured representations for visual question answering," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 3233–3241. 3

[53] B. Zhao, L. Meng, W. Yin, and L. Sigal, "Image generation from layout," *arXiv preprint arXiv:1811.11389*, 2018. 3

[54] D. Liu, H. Zhang, Z.-J. Zha, and F. Wu, "Learning to assemble neural module tree networks for visual grounding," *arXiv preprint arXiv:1812.03299*, 2018. 3

[55] H. Zhang, Z. Kyaw, S.-F. Chang, and T.-S. Chua, "Visual translation embedding network for visual relation detection," in *CVPR*, vol. 1, no. 2, 2017, p. 5. 3

[56] X. Yang, H. Zhang, and J. Cai, "Shuffle-then-assemble: Learning object-agnostic visual relationship features," in *European Conference on Computer Vision*. Springer, 2018, pp. 38–54. 3

[57] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, "Scene graph generation by iterative message passing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2017. 3

[58] K. Tang, H. Zhang, B. Wu, W. Luo, and W. Liu, "Learning to compose dynamic tree structures for visual contexts," *arXiv preprint arXiv:1812.01880*, 2018. 3

[59] L. Chen, H. Zhang, J. Xiao, X. He, S. Pu, and S.-F. Chang, "Counterfactual critic multi-agent training for scene graph generation," *arXiv preprint arXiv:1812.02347*, 2018. 3

[60] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788. 3

[61] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969. 3

[62] S. Schuster, R. Krishna, A. Chang, L. Fei-Fei, and C. D. Manning, "Generating semantically precise scene graphs from textual descriptions for improved image retrieval," in *Proceedings of the fourth workshop on vision and language*, 2015, pp. 70–80. 3, 4

[63] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013. 3

[64] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in neural information processing systems*, 2016, pp. 3844–3852. 3

[65] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," *arXiv preprint arXiv:1506.05163*, 2015. 3

[66] A. Micheli, "Neural network for graphs: A contextual constructive approach," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 498–511, 2009. 3

[67] J. Atwood and D. Towsley, "Diffusion-convolutional neural networks," in *Advances in neural information processing systems*, 2016, pp. 1993–2001. 3

[68] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *International conference on machine learning*, 2016, pp. 2014–2023. 3

[69] C. Xiong, S. Merity, and R. Socher, "Dynamic memory networks for visual and textual question answering," in *International conference on machine learning*, 2016, pp. 2397–2406. 3, 5

[70] S. Sukhbaatar, J. Weston, R. Fergus *et al.*, "End-to-end memory networks," in *Advances in neural information processing systems*, 2015, pp. 2440–2448. 3, 5, 6

[71] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher, "Ask me anything: Dynamic memory

networks for natural language processing," in *International conference on machine learning*, 2016, pp. 1378–1387. 3

[72] A. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, and J. Weston, "Key-value memory networks for directly reading documents," *arXiv preprint arXiv:1606.03126*, 2016. 3, 6

[73] H. Xu and K. Saenko, "Ask, attend and answer: Exploring question-guided spatial attention for visual question answering," in *European Conference on Computer Vision*. Springer, 2016, pp. 451–466. 3

[74] D. A. Hudson and C. D. Manning, "Compositional attention networks for machine reasoning," *arXiv preprint arXiv:1803.03067*, 2018. 3

[75] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015. 3, 7

[76] D. Xu, W. Ouyang, X. Wang, and N. Sebe, "Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 675–684. 3

[77] N. Komodakis and S. Zagoruyko, "Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer," 2017. 3

[78] A. Polino, R. Pascanu, and D. Alistarh, "Model compression via distillation and quantization," *arXiv preprint arXiv:1802.05668*, 2018. 3

[79] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017. 3

[80] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 2001–2010. 3

[81] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2016, pp. 582–597. 3

[82] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick, "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017, pp. 1988–1997. 4

[83] D. Klein and C. D. Manning, "Accurate unlexicalized parsing," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2003, pp. 423–430. 4

[84] W. Jiang, L. Ma, Y.-G. Jiang, W. Liu, and T. Zhang, "Recurrent fusion network for image captioning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 499–515. 6, 8, 10, 11, 12

[85] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. 6, 7

[86] Y. Zhang, J. Hare, and A. Prügel-Bennett, "Learning to count objects in natural images for visual question answering," in *ICLR*, 2018. 7

[87] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018. 7

[88] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in neural information processing systems*, 2014, pp. 3320–3328. 7

[89] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3128–3137. 7

[90] A. Mathews, L. Xie, and X. He, "Senticap: Generating image descriptions with sentiments," *arXiv preprint arXiv:1510.01431*, 2015. 8, 9

[91] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 311–318. 8

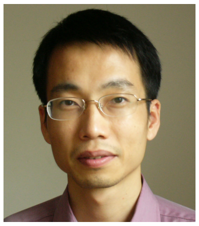
[92] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005, pp. 65–72. 8

[93] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," *Text Summarization Branches Out*, 2004. 8

[94] D. Liu, Z.-J. Zha, H. Zhang, Y. Zhang, and F. Wu, "Context-aware visual policy network for sequence-level image captioning," in *2018 ACM Multimedia Conference on Multimedia Conference*. ACM, 2018, pp. 1416–1424. 12

Xu Yang received the B.Eng. degree in Communication Engineering from Nanjing University of Posts and Telecommunications in 2013 and the M.Eng. degree in Information Processing from Southeast University in 2016. He is now a PhD candidate with Multimedia and Interactive Computing Lab, School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests mainly include computer vision, machine learning and Image Captioning.

Hanwang Zhang is currently an Assistant Professor at Nanyang Technological University, Singapore. He was a research scientist at the Department of Computer Science, Columbia University, USA. He has received the B.Eng (Hons.) degree in computer science from Zhejiang University, Hangzhou, China, in 2009, and the Ph.D. degree in computer science from the National University of Singapore in 2014. His research interest includes computer vision, multimedia, and social media. Dr. Zhang is the recipient of the Best Demo runner-up award in ACM MM 2012, the Best Student Paper award in ACM MM 2013, and the Best Paper Honorable Mention in ACM SIGIR 2016, and TOMM best paper award 2018. He is also the winner of Best Ph.D. Thesis Award of School of Computing, National University of Singapore, 2014.



Jianfei Cai (S'98-M'02-SM'07) received his PhD degree from the University of Missouri-Columbia. He is currently a Professor and serving as the Head for the Data Science & AI Department at Faculty of IT, Monash University, Australia. His major research interests include multimedia, computer vision and visual computing. He has published over 200 technical papers in international journals and conferences. He is currently an AE for IEEE TMM, and has served as an AE for IEEE TIP and TCSVT.