Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Visual commonsense reasoning with directional visual connections*

Yahong HAN†‡[1,2], Aming WU[1], Linchao ZHU[3], Yi YANG[3]

[1]*College of Intelligence and Computing, Tianjin University, Tianjin 300350, China*

[2]*Tianjin Key Lab of Machine Learning, Tianjin University, Tianjin 300350, China*

[3]*School of Computer Science, University of Technology Sydney, Sydney 2007, Australia*

†E-mail: yahong@tju.edu.cn

Received Dec. 25, 2020; Revision accepted Feb. 6, 2021; Crosschecked Apr. 22, 2021

Abstract: To boost research into cognition-level visual understanding, i.e., making an accurate inference based on a thorough understanding of visual details, visual commonsense reasoning (VCR) has been proposed. Compared with traditional visual question answering which requires models to select correct answers, VCR requires models to select not only the correct answers, but also the correct rationales. Recent research into human cognition has indicated that brain function or cognition can be considered as a global and dynamic integration of local neuron connectivity, which is helpful in solving specific cognition tasks. Inspired by this idea, we propose a directional connective network to achieve VCR by dynamically reorganizing the visual neuron connectivity that is contextualized using the meaning of questions and answers and leveraging the directional information to enhance the reasoning ability. Specifically, we first develop a GraphVLAD module to capture visual neuron connectivity to fully model visual content correlations. Then, a contextualization process is proposed to fuse sentence representations with visual neuron representations. Finally, based on the output of contextualized connectivity, we propose directional connectivity to infer answers and rationales, which includes a ReasonVLAD module. Experimental results on the VCR dataset and visualization analysis demonstrate the effectiveness of our method.

Key words: Visual commonsense reasoning; Directional connective network; Visual neuron connectivity; Contextualized connectivity; Directional connectivity

https://doi.org/10.1631/FITEE.2000722 **CLC number:** TP181

1 Introduction

Recent advances in visual understanding focus mainly on the recognition-level perception of visual content, e.g., object detection (Girshick, 2015; Liu et al., 2016) and segmentation (Badrinarayanan et al., 2017; Chen LC et al., 2018), or the recognition-level grounding of visual concepts, e.g., image captioning (Xu et al., 2015; Lu et al., 2017) and visual question answering (Ben-younes et al., 2017; Anderson et al., 2018). For complete visual understanding, a model must move forward from perception to reasoning, which includes cognitive inferences based on the associated visual contents and related commonsense knowledge. To promote the development of cognition-level reasoning based on complete visual understanding, the task of visual commonsense reasoning (VCR) (Zellers et al., 2019) was proposed along with a well-devised new dataset. Compared with traditional visual question answering, in VCR, given an image, a model is required to not only answer a question about the thorough understanding of the correlated details of the visual content, but also provide a rationale, e.g., contextualized with related

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 61876130 and 61932009)

ORCID: Yahong HAN, https://orcid.org/0000-0003-2768-1398

visual details and background knowledge, to justify why the answer is correct. To solve cognition-level reasoning based on recognition-level understanding, a recognition-to-cognition network (R2C) (Zellers et al., 2019) was first proposed to conduct VCR step by step, i.e., grounding the meaning of natural language with respect to the referred objects, contextualizing the meaning of an answer with respect to the question and related global objects, and finally, reasoning over the shared representation to obtain an answer. However, because R2C focuses on associating the representations of sentences and visual contents and does not design a proper reasoning scheme, it does not effectively address VCR.

Recent studies (Park and Friston, 2013; Bola and Sabel, 2015) on brain networks have suggested that brain function or cognition can be described as a global and dynamic integration of local neuronal connectivity. Particularly, local integration captures short-range connections, which is helpful for collecting much fine-grained information, whereas global integration focuses on long-range connections, which is beneficial for subserving higher-order cognition. Moreover, such global and local integration is context-sensitive with respect to a specific cognition task. Inspired by this idea, in this study, we propose a directional connective network (DCN) for VCR. As shown in Fig. 1, the main process of DCN is to dynamically reorganize (integrate) visual neuron connectivity, which is contextualized by the meaning of questions and answers in the current reasoning task. Based on the reorganized connectivities, directional information is considered to further improve reasoning ability.

Specifically, our network consists mainly of three modules, i.e., visual neuron connectivity, contextualized connectivity, and directional connectivity for reasoning. Taking an image that includes multiple object-bounding boxes as the input, we first devise a module of conditional GraphVLAD to represent the image's visual neuron connectivity, which includes multiple centers to dynamically capture question-related visual content. The visual neuron connectivity serves as the base function for global and local integration in the reasoning process. Meanwhile, as a key step for associating visual and linguistic information, a long short-term memory (LSTM) network (Hochreiter and Schmidhuber, 1997) is used to extract the sequential information in sentences. Then, we fuse the sentence representations with those of the visual neurons, which creates contextualization.

For contextualized connectivity, we employ a graph convolutional neural network (GCN) to sufficiently integrate local and global connectivity. For example, in Fig. 1, connections between "He" and "Person4," "Person4" and "Person3," and "Person3" and "table" could all be incorporated in the contextualized connectivity, which deepens the association of visual and linguistic contents. However, contextualized connectivity lacks direction information, which is an important clue for cognitive reasoning (Feltovich et al., 1997). Taking the answer sentence in Fig. 1 as an example, there exists a directional connection from "Person4" to "Person3" via the predicate "tell," and from "Person1" to "sandwich" via

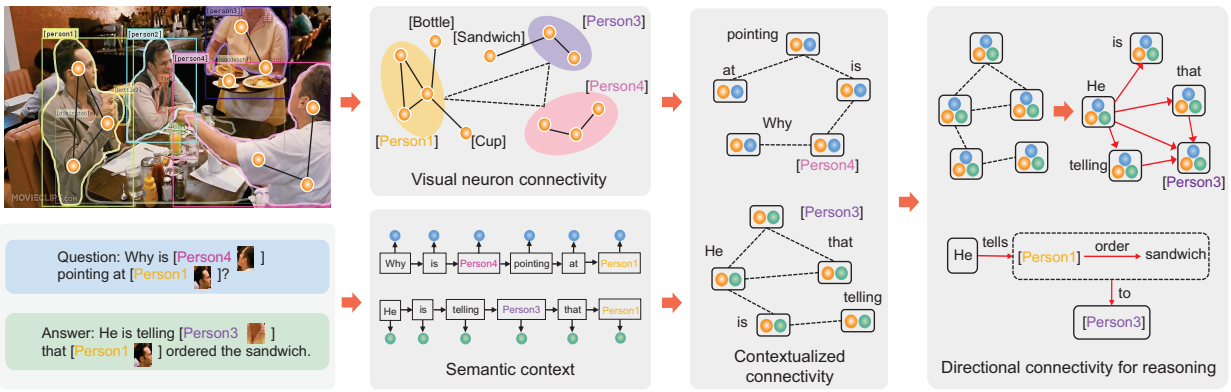

Fig. 1 Overview of our DCN method. The yellow, blue, and green circles indicate visual elements, questions, and answer representations, respectively. Our method includes mainly visual neuron connectivity, contextualized connectivity, and directional connectivity for reasoning. For semantic context, two LSTM units are used to extract sentence representations. References to color refer to the online version of this figure

the predicate "order." To this end, we explored the design of a directional connectivity module including a ReasonVLAD module to improve reasoning performance. Particularly, this module learns the semantic direction of input features, which is used to compute a directional adjacency matrix and then helps reason correct answers.

The contributions of this paper are summarized as follows:

1. To establish VCR, we propose a directional connective network (DCN) to dynamically reorganize the visual neuron connectivity that is contextualized by the meaning of questions and answers.

2. To fully model correlations of visual content, we develop a GraphVLAD module to capture visual neuron connectivity. Particularly, based on the image-question information, this module can dynamically capture question-related visual information, which is helpful for reasoning the correct answer.

3. To improve the reasoning accuracy, we propose a module of directional connectivity to infer answers or rationales. Particularly, this module includes a ReasonVLAD module, which helps enhance the information association of different models and improve the reasoning ability. Experimental results and extensive visual analysis demonstrate the effectiveness of our method.

2 Related work

2.1 Visual question answering

Given an image and a corresponding question, visual question answering (VQA) (Antol et al., 2015; Bansal et al., 2020; Chen L et al., 2020; Le et al., 2020) aims to select the correct answer from a set of proposed answers. To address VQA, models should require a sufficient understanding of visual and linguistic content. Recently, many effective methods have been proposed in the VQA task, including those based on attention (Kim et al., 2018; Malinowski et al., 2018), multi-modal fusion (Gao et al., 2018), and visual reasoning (Narasimhan et al., 2018; Cadene et al., 2019; Pan, 2019, 2020). Most methods focus on locating the related visual content that corresponds to the given question. However, they lack commonsense reasoning. To promote research into commonsense reasoning, a new VCR task was proposed by Zellers et al. (2019). Given a query-image pair, this task requires models to choose correct answers and rationales justifying why the answer is true. The challenges include mainly a thorough understanding of vision and language as well as a method to accurately infer correct responses (answers or rationales). In this study, we propose a DCN model for VCR, which is proved to be effective in the experiment.

2.2 Graph convolutional neural network

GCN (Kipf and Welling, 2016; Monti et al., 2017; Veličković et al., 2018; Zhou et al., 2018) generalizes the convolutional neural network to graph-structured data. By encoding both the structure of the graph surrounding a node and the feature of the node, a GCN can learn representation for every node effectively. Based on the benefit of capturing relations between nodes, many methods have been proposed to employ GCN for reasoning (Narasimhan et al., 2018; Schwartz et al., 2019). Particularly, Narasimhan et al. (2018) used GCN to infer answers. However, they constructed only an undirected graph for reasoning, neglecting the directional information between nodes. The directional information is often considered an important factor for inference (Feltovich et al., 1997). In this study, we propose directional connectivity to infer answers, which has been shown to be effective for VCR (Wu et al., 2019; Zellers et al., 2019).

3 Directional connective network

Fig. 2 shows the framework of a DCN model, which consists of visual neuron connectivity, contextualized connectivity, and directional connectivity for reasoning. Particularly, we can see that GraphVLAD and ReasonVLAD modules exist in our method. They are used to extract question-related visual content and help reason correct answers, respectively. In the following, we will introduce the details of our method.

3.1 Visual neuron connectivity

As a key to our method, the goal of visual neuron connectivity (Fig. 3a) is to obtain a global representation of an image, which is helpful for a thorough understanding of visual content. It includes mainly

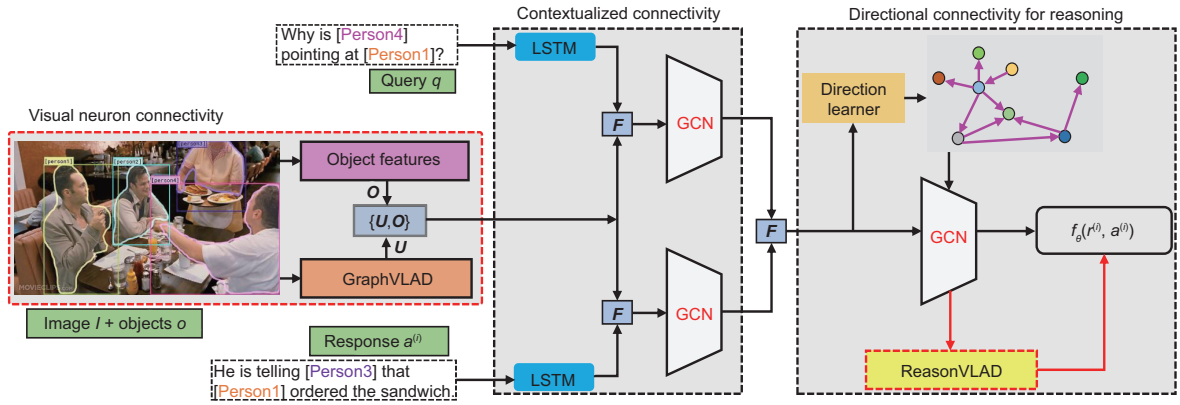

Fig. 2 Framework of the DCN method, including mainly visual neuron connectivity, contextualized connectivity, and directional connectivity for reasoning. $\{U, O\}$: the set including the output U of GraphVLAD and object features O; f_θ: the prediction function for responses (answers or rationales); F: a fusion operation

visual element connectivity and the computation of conditional centers and GraphVLAD.

3.1.1 Visual element connectivity

We first use a pre-trained network, e.g., ResNet (He et al., 2016), to obtain the feature map $\boldsymbol{X} \in \mathbb{R}^{w\times h\times m}$ of an image, where w, h, and m separately indicate the width, height, and number of channels. Here, we take each element of the feature map as a visual element. We take the output $\boldsymbol{Y} \in \mathbb{R}^n$ of the LSTM unit (Hochreiter and Schmidhuber, 1997) at the last time step as the query representation (question or question with a correct answer).

In general, there exists a certain relation between objects of an image (Chen YP et al., 2019). As shown in the left part of Fig. 3a, relations (solid and dotted lines) exist not only between elements (yellow circles) in the same object region, but also among various objects (Person1, Person3, Person4, and background). Obviously, capturing these relations is helpful for a thorough understanding of the entire scene. In this study, we employ GCN to capture these relations. Specifically, we seek to construct an undirected graph $G_g = (V, \xi, \boldsymbol{A})$, where ξ is the set of graph edges to learn and $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ $(N = wh)$ is the corresponding adjacency matrix. Each node $\nu \in V$ corresponds to one element of the feature map, and the size of V is set to N. We first reshape $\boldsymbol{X}$ to $\tilde{\boldsymbol{X}} \in \mathbb{R}^{N\times m}$. Then, we define an adjacency matrix for an undirected graph as $\boldsymbol{A} = \mathrm{softmax_r}(\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^{\mathrm{T}}) + \boldsymbol{I}_d$, where $\boldsymbol{I}_d$ indicates the identity matrix and $\mathrm{softmax_r}$ indicates that the softmax operation is in the row direction.

$$\begin{cases} \boldsymbol{M} = \boldsymbol{A}\tilde{\boldsymbol{X}}, \\ \tilde{\boldsymbol{M}} = \tanh(\boldsymbol{w}_f^c * \boldsymbol{M} + \boldsymbol{b}_f^c) \odot \sigma(\boldsymbol{w}_g^c * \boldsymbol{M} + \boldsymbol{b}_g^c), \end{cases} \tag{1}$$

where $\boldsymbol{w}_f^c \in \mathbb{R}^{1\times m\times n}$, $\boldsymbol{w}_g^c \in \mathbb{R}^{1\times m\times n}$, $\boldsymbol{b}_f^c \in \mathbb{R}^n$, and $\boldsymbol{b}_g^c \in \mathbb{R}^n$ are the trainable parameters, "$*$" represents the convolutional operation, and "$\odot$" indicates an element-wise product. Each row of the matrix $\boldsymbol{M}$ represents a feature vector of a node, which is a weighted sum of the neighboring node features of the current node. $\tilde{\boldsymbol{M}} \in \mathbb{R}^{N\times n}$ indicates the output of the GCN.

3.1.2 Computation of conditional centers

Because $\tilde{\boldsymbol{M}}$ captures only relations between visual elements and cannot fully understand the image, we consider using NetVLAD (Jégou et al., 2010; Arandjelović et al., 2018) to further enhance the representation of an image. By learning multiple centers, i.e., visual words, NetVLAD can use these centers to describe a scene (Arandjelović et al., 2018). However, these centers are learned based on the overall dataset and reflect the attributes of the dataset. In other words, these centers are independent of the current input data; they ignore the characteristic of the input data and reduce the accuracy of the representation. Here, we consider making an affine transformation for the initial centers and using these transformed centers to represent the content of an image.

Specifically, we first define the initial centers $\boldsymbol{C} = \{\boldsymbol{c}_i \in \mathbb{R}^n, i = 1, 2, ..., K\}$. Based on the current

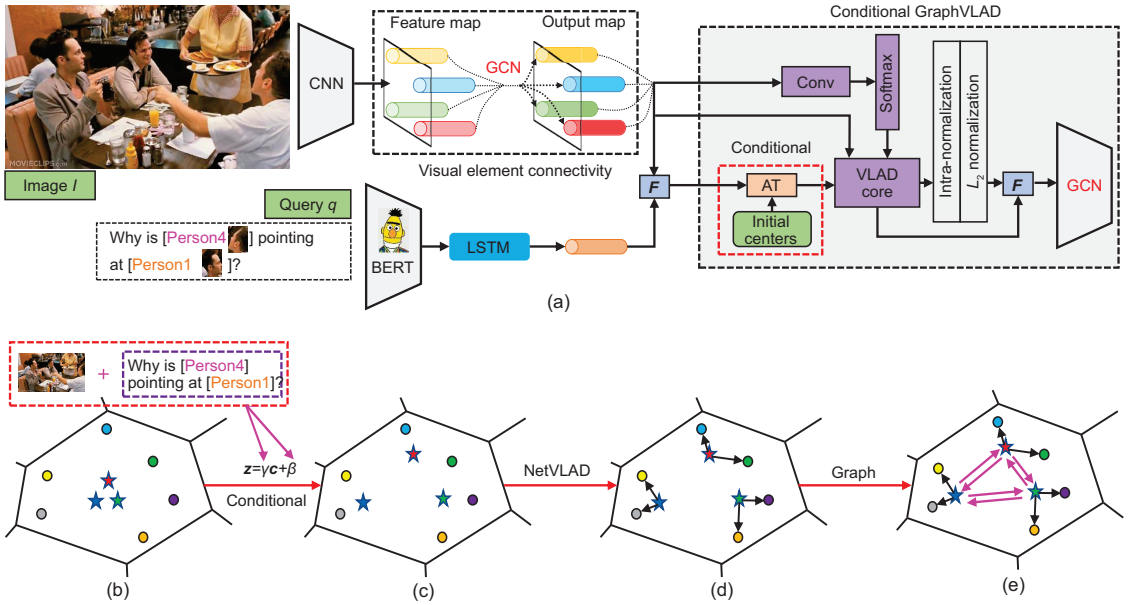

Fig. 3 Process of visual neuron connectivity (a), initial state of NetVLAD (b), conditional centers after an affine transformation (c), and results of NetVLAD (d) and GraphVLAD (e). We use the fusion of image and question to compute the parameters γ and β. AT: affine transformation. References to color refer to the online version of this figure

input query-image pairs, we make the affine transformation (Perez et al., 2017) for the initial centers:

$$\begin{cases} \gamma = f(\langle \tilde{\boldsymbol{M}}, \tilde{\boldsymbol{Y}} \rangle), \\ \beta = h(\langle \tilde{\boldsymbol{M}}, \tilde{\boldsymbol{Y}} \rangle), \\ \boldsymbol{z}_i = \gamma \boldsymbol{c}_i + \beta, \end{cases} \tag{2}$$

where $\langle a, b \rangle$ represents the concatenation of a and b. By stacking $\boldsymbol{Y}$, we obtain $\tilde{\boldsymbol{Y}} \in \mathbb{R}^{N \times n}$. We separately use a two-layer convolutional network to define f and h. $\boldsymbol{z}_i \in \mathbb{R}^n$ indicates the i^{th} generated conditional center. Here, we take the concatenated result of the representations of both input images and their corresponding queries as the input of f and h to compute parameters γ and β. Because parameters γ and β are learned based on the input query-image pairs, these two parameters reflect the character of the current input data. Equipped with the affine transformation, the initial centers are made to move toward the input features, which improves the accuracy of the residual operation (Fig. 3d) of NetVLAD. As shown in Figs. 3b and 3c, after affine transformation, the centers move toward the features (color circles). Finally, we use $\boldsymbol{Z} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_K\}$ to indicate the new conditional centers.

3.1.3 Computation of GraphVLAD

Next, we use $\boldsymbol{Z}$ and $\tilde{\boldsymbol{M}}$ to perform the NetVLAD operation:

$$\boldsymbol{D}_j = \sum_{i=1}^{N} \frac{\mathrm{e}^{\boldsymbol{w}_j^{\mathrm{T}} \tilde{\boldsymbol{M}}_i + \boldsymbol{b}_j}}{\sum_{j'} \mathrm{e}^{\boldsymbol{w}_{j'}^{\mathrm{T}} \tilde{\boldsymbol{M}}_i + \boldsymbol{b}_{j'}}} (\tilde{\boldsymbol{M}}_i - \boldsymbol{z}_j), \tag{3}$$

where $\{\boldsymbol{w}_j\}$ and $\{\boldsymbol{b}_j\}$ are sets of trainable parameters for each center $\boldsymbol{z}_j$ $(j = 1, 2, ..., K)$. Finally, we use $\boldsymbol{D} \in \mathbb{R}^{K \times n}$ to indicate the output of NetVLAD.

In addition, as shown in Fig. 3d, NetVLAD captures only relations between elements and centers. Because NetVLAD is computed based on visual elements where relations exist, we consider that there should exist certain relations between outputs. Here, we employ GCN to capture these relations. Specifically, we first concatenate the NetVLAD output and conditional centers, i.e., $\tilde{\boldsymbol{Z}} = \langle \boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_K \rangle$, $\tilde{\boldsymbol{Z}} \in \mathbb{R}^{K \times n}$, and $\boldsymbol{H} = \langle \boldsymbol{D}, \tilde{\boldsymbol{Z}} \rangle$. Then, we define an adjacency matrix for an undirected graph as $\boldsymbol{B} = \mathrm{softmax}_{\mathrm{r}}(\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}) + \boldsymbol{I}_d$. The following processes are the same as Eq. (1). Finally, we use $\boldsymbol{U} \in \mathbb{R}^{K \times n}$ to indicate the output of GraphVLAD. By this operation, we obtain the global image information, which complements local object features $\boldsymbol{O} \in \mathbb{R}^{L \times n}$

(L indicates the number of objects) extracted by a pre-trained network and the GCN. Finally, the set $\boldsymbol{S}=\{\boldsymbol{U},\boldsymbol{O}\}$ is taken as the global representation of an image.

3.2 Contextualized connectivity

The goal of contextualized connectivity is to not only capture the relevance between linguistic features and the global representation $\boldsymbol{S}$, but also extract deep semantic existing in sentences according to visual information. Specifically, LSTM is employed to obtain the representations $\tilde{\boldsymbol{Q}}\in\mathbb{R}^{P\times n}$ and $\tilde{\boldsymbol{A}}\in\mathbb{R}^{J\times n}$ of the query and response, respectively, where P and J indicate the lengths of the query and response, respectively. Next, we introduce the processing of the query. An attention operation is first used to obtain the relevance between the query and the global representation:

$$\begin{cases}\boldsymbol{F}_{qu}=\mathrm{softmax_r}(\tilde{\boldsymbol{Q}}\boldsymbol{U}^{\mathrm{T}}),\\ \boldsymbol{F}_{qo}=\mathrm{softmax_r}(\tilde{\boldsymbol{Q}}\boldsymbol{O}^{\mathrm{T}}),\\ \boldsymbol{Q}_U=\boldsymbol{F}_{qu}\boldsymbol{U},\\ \boldsymbol{Q}_O=\boldsymbol{F}_{qo}\boldsymbol{O}.\end{cases}\tag{4}$$

Then, we take the concatenation of $\boldsymbol{Q}_U$, $\boldsymbol{Q}_O$, and $\tilde{\boldsymbol{Q}}$ as $\boldsymbol{Q}_F\in\mathbb{R}^{P\times(3n)}$, where $\boldsymbol{U}$ and $\boldsymbol{O}$ are the output of the GraphVLAD. Here, we obtain only sequential features, rather than the structural information (Xu et al., 2018) which is helpful for a better understanding of the sentence semantics. Meanwhile, LSTM is limited by long-term information dilution (Vaswani et al., 2017), which weakens the capacity of the sentence representation. Here we consider using GCN to extract structural information. Specifically, we define an adjacency matrix for an undirected graph as $\boldsymbol{Q}=\mathrm{softmax_r}(\boldsymbol{Q}_F\boldsymbol{Q}_F^{\mathrm{T}})+\boldsymbol{I}_d$. The following processes are the same as Eq. (1). Finally, we use $\boldsymbol{Q}_g\in\mathbb{R}^{P\times n}$ to indicate the output of this network. The processing of responses is the same as that of queries. The representation of response generated by a GCN is defined as $\boldsymbol{A}_g\in\mathbb{R}^{J\times n}$.

3.3 Directional connectivity with ReasonVLAD

3.3.1 Directional connectivity

Directional information is an important clue for cognitive reasoning, and using directional information can improve the reasoning accuracy (Feltovich et al., 1997). Here, we propose a semantic direction-based GCN for reasoning. Specifically, we first use $\tilde{\boldsymbol{A}}$ to obtain the attention representation $\boldsymbol{Q}_a\in\mathbb{R}^{J\times n}$ of $\boldsymbol{Q}_g$. The processes are the same as Eq. (4). Then, $\boldsymbol{Q}_a$ and $\boldsymbol{A}_g$ are concatenated as $\boldsymbol{E}_{qa}\in\mathbb{R}^{J\times(2n)}$. Next, based on $\boldsymbol{E}_{qa}$, we first try to learn the direction information:

$$\begin{cases}\boldsymbol{D}_{qa}=\phi(\boldsymbol{E}_{qa}),\\ \boldsymbol{G}_t=\boldsymbol{D}_{qa}\boldsymbol{D}_{qa}^{\mathrm{T}},\\ \boldsymbol{D}_t=\mathrm{sign}(\boldsymbol{G}_t),\\ \boldsymbol{V}_e=\mathrm{softmax_r}(\mathrm{abs}(\boldsymbol{G}_t)),\end{cases}\tag{5}$$

where $\mathrm{abs}(\cdot)$ is the operation of absolute value. Here, ϕ is defined as a directional function, which is a one-layer convolutional network without activation. Also, to learn the direction, we do not use rectified linear unit (ReLU) activation in the last layer of the network ϕ. By using the sign function, we obtain the direction $\boldsymbol{D}_t$, where "-1" and "1" separately indicate the negative and positive correlations. Next, based on the output $\boldsymbol{D}_t$ of the sign function, we compute the adjacency matrix.

$$\begin{cases}\boldsymbol{H}=\boldsymbol{D}_t\odot\boldsymbol{V}_e+\boldsymbol{I}_d,\\ \boldsymbol{M}_t=\boldsymbol{H}\boldsymbol{E}_{qa},\\ \boldsymbol{R}_t=\tanh(\boldsymbol{w}_f^r*\boldsymbol{M}_t+\boldsymbol{b}_f^r)\odot\sigma(\boldsymbol{w}_g^r*\boldsymbol{M}_t+\boldsymbol{b}_g^r),\end{cases}\tag{6}$$

where $\boldsymbol{H}$ indicates the adjacency matrix, and $\boldsymbol{w}_f^r\in\mathbb{R}^{1\times(2n)\times n}$, $\boldsymbol{w}_g^r\in\mathbb{R}^{1\times(2n)\times n}$, $\boldsymbol{b}_f^r\in\mathbb{R}^n$, and $\boldsymbol{b}_g^r\in\mathbb{R}^n$ indicate the trainable parameters. Finally, we take $\boldsymbol{R}_t\in\mathbb{R}^{J\times n}$ as the GCN output. By this operation, our model not only learns the direction information between nodes, but also leverages the GCN computation information, resulting in accurate inference. In the experiment, compared with an undirected GCN, our method improves performance significantly.

3.3.2 ReasonVLAD

After obtaining the output of the directional connectivity module, we design a ReasonVLAD module to enhance the information association of different models and improve the reasoning ability. Specifically, after contextualized connectivity, we obtain the question-related representation $\boldsymbol{Q}_g\in\mathbb{R}^{P\times n}$. Based on the response representation $\tilde{\boldsymbol{A}}\in\mathbb{R}^{J\times n}$, we employ an attention mechanism to select related information from $\boldsymbol{Q}_g$. The processes are shown as

follows:

$$\begin{cases} \boldsymbol{F}_{qa} = \mathrm{softmax_r}(\tilde{\boldsymbol{A}}\boldsymbol{Q}_g^{\mathrm{T}}), \\ \boldsymbol{A}_Q = \boldsymbol{F}_{qa}\boldsymbol{Q}_g. \end{cases} \tag{7}$$

Taking the sum of $\boldsymbol{A}_Q$ and $\boldsymbol{R}_t$ as the input, our model learns multiple centers to sufficiently represent the semantic information existing in the fused representations. Specifically, we first define the initial centers $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_i \in \mathbb{R}^n, i = 1, 2, ..., K\}$. Next, the processes of ReasonVLAD are shown as follows:

$$\boldsymbol{\Phi}_j = \sum_{i=1}^{J} \frac{\mathrm{e}^{\boldsymbol{w}_j^{\mathrm{T}}\tilde{\boldsymbol{S}}_i + \boldsymbol{b}_j}}{\sum_{j'} \mathrm{e}^{\boldsymbol{w}_{j'}^{\mathrm{T}}\tilde{\boldsymbol{S}}_i + \boldsymbol{b}_{j'}}} (\tilde{\boldsymbol{S}}_i - \boldsymbol{\theta}_j), \tag{8}$$

where $\tilde{\boldsymbol{S}} \in \mathbb{R}^{J\times n}$ indicates the sum of $\boldsymbol{A}_Q$ and $\boldsymbol{R}_t$. Finally, we use $\boldsymbol{\Phi} \in \mathbb{R}^{J\times n}$ to represent the output of NetVLAD. The advantage of ReasonVLAD is mainly that, with the help of the learned centers, this module can sufficiently capture the fused information, which reduces the loss of related information and then improves reasoning ability.

3.4 Prediction layer and loss function

After obtaining the ReasonVLAD output, we use $\boldsymbol{R}_t$ and employ the attention mechanism to select related information from $\boldsymbol{\Phi}$. Then we concatenate $\boldsymbol{R}_t$ and the attention result across the channel dimension. Next, we compute a global vector representation via a max-pooling operation across the node dimension. This operation helps determine a permutation-invariant output and focuses on the impact of the graph structure (Norcliffe-Brown et al., 2018). Finally, we compute classification logits through a two-layer multilayer perceptron (MLP) with ReLU activation.

Given a query-image pair, the VCR task gives four response choices. In this study, we train our model using a multi-class cross-entropy loss between the set of responses and the labels, i.e., $l(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\sum_{i=1}^{4} y_i \log \hat{y}_i$, where $\boldsymbol{y}$ denotes the ground truth and $\hat{\boldsymbol{y}}$ is the prediction result.

4 Experiments

In this section, we evaluate our method on the VCR dataset. This dataset contains 2.9×10^5 pairs of questions, answers, and rationales, and more than 1.1×10^5 unique movie scenes. This task considers three modes: $Q \to A$ (given a question, select the correct answer), $QA \to R$ (given a question and the correct answer, select the correct rationale), and $Q \to AR$ (given a question, select the correct answer, then the correct rationale). For the $Q \to AR$ mode, if either the wrong answer or the wrong rationale is obtained, no points will be received.

Implementation details are as follows: We use ResNet50 (He et al., 2016) to extract image and object features. BERT (Devlin et al., 2019) is used as the word embedding. The feature map is $\boldsymbol{X} \in \mathbb{R}^{12\times 24\times 512}$. The size of the hidden state of LSTM is set to 512. For Eq. (1), we use a one-layer GCN, and 32 centers are used to compute GraphVLAD. For Eq. (2), we separately use a two-layer network to define f and h. Their parameters are all set to $1 \times 1024 \times 512$ and $1 \times 512 \times 512$. Next, we use a one-layer GCN to capture relations between centers. The parameter settings of the GCN are the same as those of Eq. (1). For contextualized connectivity, we separately use a two-layer GCN to process queries and responses. Their parameter settings are the same as those of Eq. (1). For Eq. (5), a one-layer GCN is used for reasoning. In addition, the parameters of the network ϕ are set to $1 \times 1024 \times 512$. For ReasonVLAD (Eq. (8)), we use 28 centers. During training, we use the Adam optimizer with a learning rate of 2×10^{-3}.

4.1 Performance of our method

We evaluate our method on the three modes of the VCR task. The results based on the validation set are shown in Table 1. We can see that some of state-of-the-art VQA methods, e.g., MUTAN (Ben-younes et al., 2017) and BottomUpTopDown (Anderson et al., 2018), do not perform well on this task. These methods usually focus on associating the visual representations with the corresponding linguistic representations, which lacks the ability to infer and results in unsatisfactory performance. Meanwhile, compared with the baseline method, on the three VCR task modes, our method is 3.8%, 3.5%, and 4.8% better than R2C, respectively. This shows that our method is effective. We can also see that BERT-based pretraining methods (Li LH et al., 2019; Lu et al., 2019; Su et al., 2019; Li G et al., 2020) obtain outstanding reasoning performance. These methods usually employ a multi-layer transformer (Vaswani et al., 2017) as the backbone, and extend it to take both visual and linguistic embedded

features as input. By pretraining based on massive-scale visual-linguistic datasets, they can extract powerful generic representations and improve the performance of visual-linguistic tasks significantly.

In Fig. 4, we show qualitative results of our method. First, we can see that compared with commonly used VQA datasets (Antol et al., 2015; Goyal et al., 2017), the questions and responses of the VCR task are more complex. Therefore, it is difficult to obtain the correct answers and rationales by directly employing recognition of visual contents. The first row corresponds to the correct result. Given an image and a corresponding complex question, our method selects the right answer and the corresponding rationale. This further demonstrates the effectiveness of our method. The second and third rows correspond to failed cases. In the case of the second row, our method selects the correct answer and a wrong rationale, and in the case of the third row, our method does not select the correct answer or rationale. This indicates that when questions and responses involve more commonsense knowledge, it is difficult for the model to select the correct answer and correct corresponding rationale.

Table 1 Performance of our DCN model on the VCR dataset

Model	Accuracy (%)		
	$Q \to A$	$QA \to R$	$Q \to AR$
VisualBERT	70.8	73.2	52.2
ViLBERT	72.4	74.5	54.0
Unicoder-VL	72.6	74.5	54.5
VL-BERT	73.8	74.4	55.2
Revisited VQA	39.4	34.0	13.5
BottomUpTopDown	42.8	25.1	10.7
MLB	45.5	36.1	17.0
MUTAN	44.4	32.0	14.6
R2C (baseline)	63.8	67.2	43.1
DCN	**67.6**	**70.7**	**47.9**

To further demonstrate the effectiveness of our method, we show more qualitative results in Fig. 5. We can see that our method selects the correct answers and corresponding rationales. Particularly, taking the first image as an example, we can see that the given question is complex. Our method selects the correct answers and corresponding rationales, with resulting scores of 98% and 99%, respectively. For the third example, our method selects

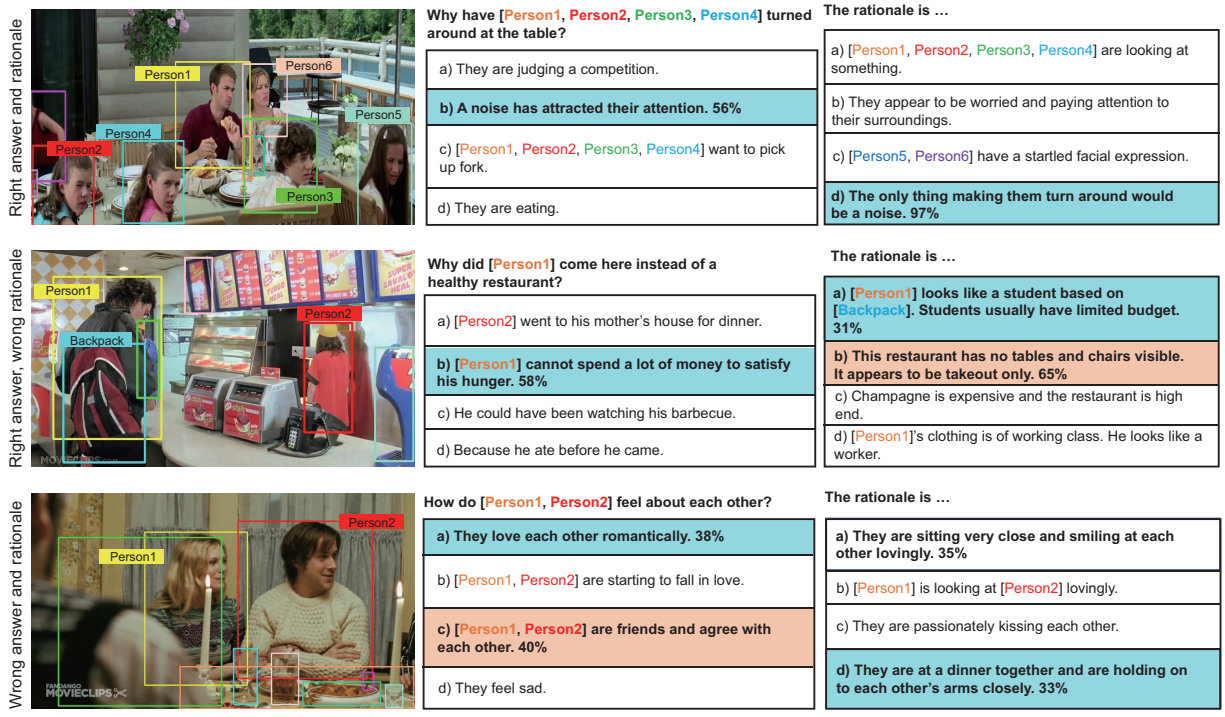

Fig. 4 Qualitative examples from the DCN. Correct choices are highlighted in blue, and incorrect inferences are in orange. The number after each option indicates the score given by our model. The first row is a successful case. The second and last rows correspond to two failed cases. References to color refer to the online version of this figure

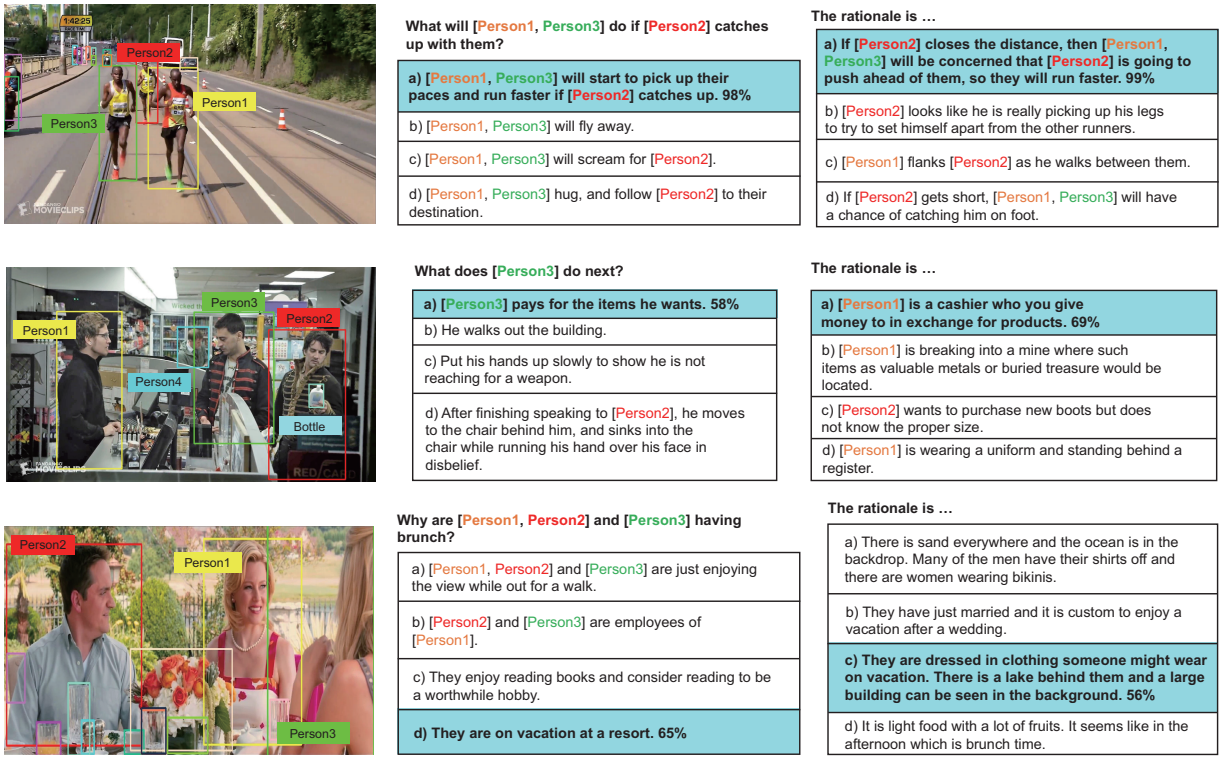

Fig. 5 More qualitative examples from the DCN. Correct choices are highlighted in blue. We can see that our method selected the correct answers and rationales. References to color refer to the online version of this figure

the correct answer and the corresponding rationale with scores of 65% and 56%. This shows that our method is helpful in promoting a thorough understanding of visual content and improving reasoning ability, which further demonstrates the effectiveness of our method.

4.2 Ablation analysis

In this subsection, we perform an ablation analysis of our proposed method.

4.2.1 GraphVLAD

The number of centers is an important hyper-parameter for GraphVLAD. If few centers are used, visual content will not be sufficiently represented. Conversely, if more centers are used, the number of parameters and computational costs will increase. In $Q \rightarrow A$, $QA \rightarrow R$, and $Q \rightarrow AR$ modes, the performance scores of 16 and 48 centers separately are 66.8%, 69.2%, 46.6% and 67.0%, 69.6%, 46.8%. For our method, the performance of 32 centers is the best.

Next, we analyze the effect of conditional centers and the GCN for GraphVLAD. Here, we keep other components of our method unchanged. In the experiment, we find that employing conditional operation improves the performance by 0.9%, 0.7%, and 1.0%. This shows that the conditional operation is helpful in capturing visual contents sufficiently, which improves performance. Meanwhile, we find that employing GCN in GraphVLAD improves performance by 0.8%, 1.0%, and 0.9%. This shows that employing graph neural networks to integrate centers is helpful for a thorough understanding of visual contents. These analyses further demonstrate the effectiveness of the proposed GraphVLAD module.

In Fig. 6, we show two t-SNE (van der Maaten and Hinton, 2008) examples of conditional centers. The queries of Figs. 6a and 6b are "Who does the dog belong to?" and "What will happen after the person pushes the lifeboat over the edge of the ship?" We can see that the positions of centers are related to the complexity of visual contents and queries. When an image contains rich content and its corresponding query is complex, e.g., Fig. 6b, to capture rich visual information to answer the query, these centers

will learn to spread farther apart from each other. Meanwhile, when the image content and its corresponding query contain relatively little information, e.g., Fig. 6a, to focus on visual information that is related to the query, these centers will adaptively become more concentrated. In this way, we can obtain an effective visual representation, which is helpful for the following contextualization and reasoning.

In Fig. 7, we show more visualization results. We can see that when the given image contains rich content and the corresponding query is complex, the centers will spread out to capture much more

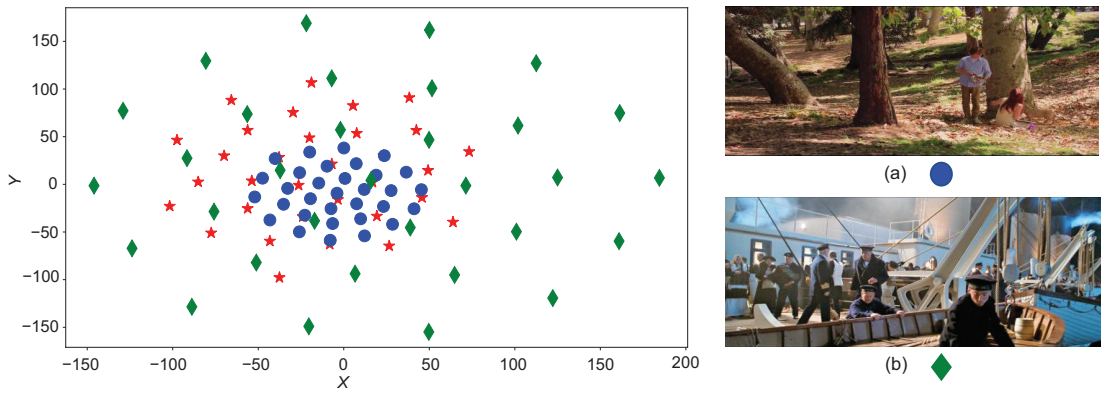

Fig. 6 t-SNE plot of conditional centers. Red pentagrams: initial centers; blue circles and green rhombuses: two different conditional centers. (a) and (b) are used to compute the blue and green centers, respectively. References to color refer to the online version of this figure

Fig. 7 t-SNE plot of conditional centers: (a) example 1; (b) example 2. Red pentagrams: initial centers; blue circles: different conditional centers. References to color refer to the online version of this figure

semantic information, whereas when the content is less rich and query is simpler, the centers will become more concentrated. This further shows that with the help of the conditional operation, our method can dynamically capture query-related visual information that is helpful for reasoning the correct answers and rationales.

4.2.2 Contextualized connectivity

In this study, we separately employ a GCN to capture the semantics of queries and responses. To prove that this operation is effective, we compare it with a common operation, i.e., using a GCN to process the concatenation of vision, query, and response. In $Q \rightarrow A$, $QA \rightarrow R$, and $Q \rightarrow AR$ modes, compared with our method, the performance of the common operation is 1.1%, 0.9%, and 1.2% worse, showing the effectiveness of our method.

4.2.3 Directional connectivity with ReasonVLAD

In this study, we design directional connectivity for reasoning. Here, we perform an ablation analysis of this module. When we remove the directional operation and keep other components of our model unchanged, the performance separately degrades by 0.9%, 1.1%, and 1.3%. This demonstrates that direction information is important for accurate reasoning. Capturing direction information can improve performance. Next, we perform an ablation analysis of the ReasonVLAD module. Employing ReasonVLAD is beneficial for sufficiently capturing information of the fused representation. Here, we analyze the effect of employing different numbers of centers. When the number of centers is set to 24 and 32, the performance degrades. This shows that employing more centers increases the number of parameters, which results in overfitting and affects performance; employing few centers does not capture the fused information effectively. For our method, the performance of employing 28 centers is the best.

5 Conclusions

In this paper, we have proposed a directional connective network for visual commonsense reasoning. Specifically, this network consists of three graph-based modules, i.e., visual neuron connectivity, contextualized connectivity, and directional connectivity including ReasonVLAD. Visual neuron connectivity promotes a thorough understanding of visual contents, contextualized connectivity captures the relevance between linguistic features and global representations, and directional connectivity enhances reasoning ability based on learned direction information. We have proposed a ReasonVLAD module to sufficiently capture the fused information. Experimental results based on the VCR dataset and visualization analysis demonstrated the effectiveness of our method.

Contributors

Yahong HAN designed the research. Aming WU conducted the experiments and drafted the manuscript. Linchao ZHU helped organize the manuscript. Yi YANG revised the paper.

Compliance with ethics guidelines

Yahong HAN, Aming WU, Linchao ZHU, and Yi YANG declare that they have no conflict of interest.

References

Anderson P, He XD, Buehler C, et al., 2018. Bottom-up and top-down attention for image captioning and visual question answering. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.6077-6086. https://doi.org/10.1109/CVPR.2018.00636

Antol S, Agrawal A, Lu JS, et al., 2015. VQA: visual question answering. Proc IEEE Int Conf on Computer Vision, p.2425-2433. https://doi.org/10.1109/ICCV.2015.279

Arandjelović R, Gronat P, Torii A, et al., 2018. NetVLAD: CNN architecture for weakly supervised place recognition. *IEEE Trans Patt Anal Mach Intell*, 40(6):1437-1451. https://doi.org/10.1109/TPAMI.2017.2711011

Badrinarayanan V, Kendall A, Cipolla R, 2017. SegNet: a deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans Patt Anal Mach Intell*, 39(12):2481-2495. https://doi.org/10.1109/TPAMI.2016.2644615

Bansal A, Zhang YT, Chellappa R, 2020. Visual question answering on image sets. European Conf on Computer Vision, p.51-67. https://doi.org/10.1007/978-3-030-58589-1_4

Ben-younes H, Cadene R, Cord M, et al., 2017. MUTAN: multimodal tucker fusion for visual question answering. Proc IEEE Int Conf on Computer Vision, p.2631-2639. https://doi.org/10.1109/ICCV.2017.285

Bola M, Sabel BA, 2015. Dynamic reorganization of brain functional networks during cognition. *NeuroImage*, 114:398-413. https://doi.org/10.1016/j.neuroimage.2015.03.057

Cadene R, Ben-younes H, Cord M, et al., 2019. MUREL: multimodal relational reasoning for visual question answering. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.1989-1998. https://doi.org/10.1109/CVPR.2019.00209

Chen L, Yan X, Xiao J, et al., 2020. Counterfactual samples synthesizing for robust visual question answering. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.10797-10806. https://doi.org/10.1109/CVPR42600.2020.01081

Chen LC, Papandreou G, Kokkinos I, et al., 2018. DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans Patt Anal Mach Intell*, 40(4):834-848. https://doi.org/10.1109/TPAMI.2017.2699184

Chen YP, Rohrbach M, Yan ZC, et al., 2019. Graph-based global reasoning networks. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.433-442. https://doi.org/10.1109/CVPR.2019.00052

Devlin J, Chang MW, Lee K, et al., 2019. BERT: pre-training of deep bidirectional transformers for language understanding. Proc Conf of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), p.4171-4186. https://doi.org/10.18653/v1/N19-1423

Feltovich PJ, Ford KM, Hoffman RR, 1997. Expertise in Context: Human and Machine. MIT Press, Cambridge, MA, USA, p.67-99.

Gao P, Li H, Li S, et al., 2018. Question-guided hybrid convolution for visual question answering. European Conf on Computer Vision, p.485-501. https://doi.org/10.1007/978-3-030-01246-5_29

Girshick R, 2015. Fast R-CNN. Proc IEEE Int Conf on Computer Vision, p.1440-1448. https://doi.org/10.1109/ICCV.2015.169

Goyal Y, Khot T, Summers-Stay D, et al., 2017. Making the V in VQA matter: elevating the role of image understanding in visual question answering. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.6325-6334.

He KM, Zhang XY, Ren SQ, et al., 2016. Deep residual learning for image recognition. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.770-778.

Hochreiter S, Schmidhuber J, 1997. Long short-term memory. *Neur Comput*, 9(8):1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Jégou H, Douze M, Schmid C, et al., 2010. Aggregating local descriptors into a compact image representation. Proc IEEE Computer Society Conf on Computer Vision and Pattern Recognition, p.3304-3311. https://doi.org/10.1109/CVPR.2010.5540039

Kim KM, Choi SH, Kim JH, et al., 2018. Multimodal dual attention memory for video story question answering. https://arxiv.org/abs/1809.07999

Kipf TN, Welling M, 2016. Semi-supervised classification with graph convolutional networks. https://arxiv.org/abs/1609.02907v4

Le TM, Le V, Venkatesh S, et al., 2020. Hierarchical conditional relation networks for video question answering. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.9969-9978. https://doi.org/10.1109/CVPR42600.2020.00999

Li G, Duan N, Fang YJ, et al., 2020. Unicoder-VL: a universal encoder for vision and language by cross-modal pre-training. Proc AAAI Conf on Artificial Intelligence, p.11336-11344. https://doi.org/10.1609/aaai.v34i07.6795

Li LH, Yatskar M, Yin D, et al., 2019. VisualBERT: a simple and performant baseline for vision and language. https://arxiv.org/abs/1908.03557

Liu W, Anguelov D, Erhan D, et al., 2016. SSD: single shot multibox detector. European Conf on Computer Vision, p.21-37. https://doi.org/10.1007/978-3-319-46448-0_2

Lu JS, Xiong CM, Parikh D, et al., 2017. Knowing when to look: adaptive attention via a visual sentinel for image captioning. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.3242-3250. https://doi.org/10.1109/CVPR.2017.345

Lu JS, Batra D, Parikh D, et al., 2019. ViLBERT: pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. https://arxiv.org/abs/1908.02265

Malinowski M, Doersch C, Santoro A, et al., 2018. Learning visual question answering by bootstrapping hard attention. European Conf on Computer Vision, p.3-20. https://doi.org/10.1007/978-3-030-01231-1_1

Monti F, Boscaini D, Masci J, et al., 2017. Geometric deep learning on graphs and manifolds using mixture model CNNs. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.5425-5434. https://doi.org/10.1109/CVPR.2017.576

Narasimhan M, Lazebnik S, Schwing AG, 2018. Out of the box: reasoning with graph convolution nets for factual visual question answering. Proc 32nd Int Conf on Neural Information Processing Systems, p.2659-2670.

Norcliffe-Brown W, Vafeias ES, Parisot S, 2018. Learning conditioned graph structures for interpretable visual question answering. https://arxiv.org/abs/1806.07243

Pan YH, 2019. On visual knowledge. *Front Inform Technol Electron Eng*, 20(8):1021-1025. https://doi.org/10.1631/FITEE.1910001

Pan YH, 2020. Miniaturized five fundamental issues about visual knowledge. *Front Inform Technol Electron Eng*, online. https://doi.org/10.1631/FITEE.2040000

Park HJ, Friston K, 2013. Structural and functional brain networks: from connections to cognition. *Science*, 342(6158):1238411. https://doi.org/10.1126/science.1238411

Perez E, Strub F, de Vries H, et al., 2017. FiLM: visual reasoning with a general conditioning layer. https://arxiv.org/abs/1709.07871v2

Schwartz I, Yu S, Hazan T, et al., 2019. Factor graph attention. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.2039-2048. https://doi.org/10.1109/CVPR.2019.00214

Su WJ, Zhu XZ, Cao Y, et al., 2019. VL-BERT: pre-training of generic visual-linguistic representations. https://arxiv.org/abs/1908.08530v1

van der Maaten L, Hinton G, 2008. Visualizing data using t-SNE. *J Mach Learn Res*, 9:2579-2605.

Vaswani A, Shazeer N, Parmar N, et al., 2017. Attention is all you need. Proc 31st Int Conf on Neural Information Processing Systems, p.6000-6010.

Veličković P, Cucurull G, Casanova A, et al., 2018. Graph attention networks. Proc Int Conf on Learning Representations.

Wu AM, Zhu LC, Han YH, et al., 2019. Connective cognition network for directional visual commonsense reasoning. Proc 33rd Conf on Neural Information Processing Systems, p.5669-5679.

Xu K, Ba JL, Kiros R, et al., 2015. Show, attend and tell: neural image caption generation with visual attention. Proc 32nd Int Conf on Machine Learning, p.2048-2057.

Xu K, Wu LF, Wang ZG, et al., 2018. Exploiting rich syntactic information for semantic parsing with graph-to-sequence model. Proc Conf on Empirical Methods in Natural Language Processing, p.918-924.

Zellers R, Bisk Y, Farhadi A, et al., 2019. From recognition to cognition: visual commonsense reasoning. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.6713-6724. https://doi.org/10.1109/CVPR.2019.00688

Zhou J, Cui GQ, Zhang ZY, et al., 2018. Graph neural networks: a review of methods and applications. https://arxiv.org/abs/1812.08434v3