Learning the Dynamics of Visual Relational Reasoning via Reinforced Path Routing

Chenchen Jing,[1] Yunde Jia,[1] Yuwei Wu,[1]* Chuanhao Li,[1] Qi Wu[2]

[1]Beijing Laboratory of Intelligent Information Technology,
School of Computer Science, Beijing Institute of Technology, China
[2]Australian Centre for Robotic Vision, University of Adelaide, Australia
{chenchen.jing,jiayunde,wuyuwei,lichuanhao}@bit.edu.cn, qi.wu01@adelaide.edu.au

Abstract

Reasoning is a dynamic process. In cognitive theories, the dynamics of reasoning refers to reasoning states over time after successive state transitions. Modeling the cognitive dynamics is of utmost importance to simulate human reasoning capability. In this paper, we propose to learn the reasoning dynamics of visual relational reasoning by casting it as a path routing task. We present a reinforced path routing method that represents an input image via a structured visual graph and introduces a reinforcement learning based model to explore paths (sequences of nodes) over the graph based on an input sentence to infer reasoning results. By exploring such paths, the proposed method represents reasoning states clearly and characterizes state transitions explicitly to fully model the reasoning dynamics for accurate and transparent visual relational reasoning. Extensive experiments on referring expression comprehension and visual question answering demonstrate the effectiveness of our method.

Introduction

Cognitive theories reveal that reasoning is a dynamic process where the state of the intelligent system is always changing over time in its state space (Engelfriet and Treur 1994; Port and Van Gelder 1995). The dynamics of reasoning is described by sequences of reasoning states over time after successive reasoning steps (Jonker and Treur 2002). Formally, a reasoning state is an intermediate representation of a reasoning process. A transition from one reasoning state to another reasoning state formalizes one reasoning step. Modeling the reasoning dynamics is critical to simulate the reasoning capability of humans (Jonker and Treur 2003).

In this paper, we propose to learn the reasoning dynamics of visual relational reasoning by casting it as a path routing task. Visual relational reasoning involves the sequential processing of visual information such as relationships and objects in images. Figure 1 shows the path routing for visual relational reasoning in the context of visual question answering (VQA). In path routing, we require a reasoning model to explore paths, *i.e.*, sequences of nodes, over a visual graph, whose nodes denote objects in an image and edges denote relationships among the objects. For example, to answer the question "What is the person to the left of the glasses holding?" in Figure 1 (a), the model should progressively attend to the glasses, the person, and the kite on the graph based on their relationships. Different from existing modular methods (Andreas et al. 2016; Johnson et al. 2017) and holistic methods (Guo, Xu, and Tao 2019; Hu et al. 2019) that focus on reasoning structure modeling or contextual representations learning, we focus on reasoning dynamics learning in path routing shown in Figure 1 (d). By exploring such paths, the reasoning states are represented clearly and the reasoning state transitions are characterized explicitly to enable the modeling of the reasoning dynamics.

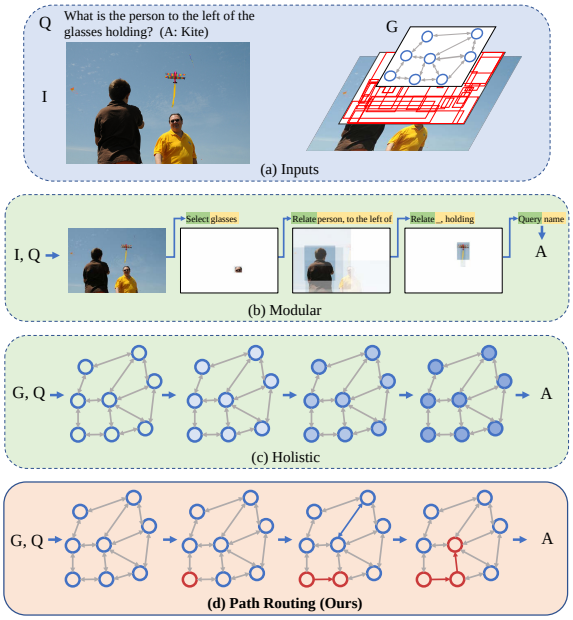

Figure 1: Illustrations of visual relational reasoning in the context of visual question answering. (a) An input question, an input image, and a graph representing the relational layout of the image. We do not show all edges in the graph for simplicity. (b) The reasoning process of typical modular methods. (c) The reasoning process of typical graph-based holistic methods. (d) The path routing on the graph, where red circles and red solid arrows comprise the explored path.

*corresponding author

To achieve this goal, we present a reinforced path routing method, which uses a reinforcement learning (RL) based reasoning model to learn multi-step paths, for accurate and transparent visual relational reasoning. The proposed method regards the reasoning model as an agent that navigates on a structured visual graph based on an input sentence to infer the reasoning result. At each time step, the agent uses a navigation policy to decide which node should be selected to extend the current path. The navigation policy, which contains a language attention and a history attention, enables the agent to exploit language cues, history states, and history decisions in current decision-making. The agent will be rewarded if it finds a path based on which the correct result is inferred.

We further introduce a pre-training strategy that warm-starts the agent via supervised learning (SL). The strategy encourages the agent to navigate over the graph from one node to another node in a soft manner. In pre-training, we progressively sharpen the probability distribution of the navigation policy, to encourage the agent to gradually learn to focus on only one node. A coverage loss is used to enforce the model to attend to different nodes at different steps. The pre-training strategy is thus capable of preventing the model from degradation in the RL stage. Extensive experiments on two visual relational reasoning tasks, referring expression comprehension (Hu et al. 2016; Qiao, Deng, and Wu 2020), visual question answering (Antol et al. 2015; Wu et al. 2017), demonstrate our method achieves accurate reasoning with a transparent reasoning process.

The contributions of this paper are two-fold:

1. We are the first to learn the dynamics of visual relational reasoning by casting it as a path routing task to achieve accurate and transparent reasoning.
2. We present a reinforced path routing method that can learn multi-step paths without any additional annotation. The explored paths can make the reasoning processes of our method more human-understandable.

Related Work

Visual Relational Reasoning

Existing visual relational reasoning methods for referring expression comprehension and visual question answering can be divided into two categories: modular and holistic. Modular methods (Andreas et al. 2016; Johnson et al. 2017; Hu et al. 2018; Shi, Zhang, and Li 2019; Hong et al. 2019; Liu et al. 2019a; Chen et al. 2021) focus on reasoning structure modeling and assembles various neural modules for different input sentence-image pairs. They show good compositionality and interpretability in various tasks but usually have high model complexity and inferior performance on tasks over real-world images. Holistic methods (Hudson and Manning 2018; Perez et al. 2018; Hu et al. 2019; Wang et al. 2019; Yang, Li, and Yu 2019a; Jing et al. 2020a; Liu et al. 2020; Liao et al. 2020; Jing et al. 2020b; Deng et al. 2021) focus on contextual representation learning and use a single model for different inputs. They usually stack attention mechanisms or graph convolution operations to learn informative representations and perform reasoning in the latent space. By contrast, our method focuses on reasoning dynamics learning by casting the reasoning task as path routing. We explicitly characterize the reasoning states and the state transitions to learn the reasoning dynamics for accurate reasoning with a transparent reasoning process.

The NSM (Hudson and Manning 2019a), the XMN (Shi, Zhang, and Li 2019), and the SGMN (Yang, Li, and Yu 2020) also build a visual graph and perform reasoning by traversing the graph. Nonetheless, in each step of reasoning, they weighted sum all nodes in the graph to represent the current state. Therefore, which node contributes most to the next step of reasoning is not clear, and the reasoning process is hard to understand. Our method uses only one attended node to represent the state of each step and combines the state and historical states for the next step of reasoning. Only the attended nodes are involved in reasoning. Thus the path formed by the attended nodes serves as an explanation for the reasoning process.

Reinforcement learning for REC and VQA

Reinforcement learning has been widely applied to vision-and-language tasks such as REC and VQA. Nonetheless, most of them use the RL as a technique to estimate gradients of non-differentiable components to guarantee the models can be optimized in an end-to-end manner. For example, neural modular networks (Hu et al. 2017; Johnson et al. 2017; Mascharka et al. 2018) use the REINFORCE algorithm (Williams 1992) to estimates gradients of the layout generator. Tang *et al.* (2019) use the REINFORCE to explore tree structure to model visual context for VQA. Wu, Xu, and Yang (2017) use the RL to learn to move and reshape a bounding box for one-stage REC. Different from these methods, we model the reasoning process as a Markov decision process and use the RL to achieve sequential reasoning for both REC and VQA.

Method

The proposed method learns the dynamics of visual relational reasoning by constructing a graph to represent an input image and encouraging an agent to explore paths on the graph according to an input sentence, as shown in Figure 2. In this section, we first formally define visual relational reasoning as a path routing task and then illustrate the method.

Formulation

We focus on two visual relational reasoning tasks, referring expression comprehension (REC) and visual question answering (VQA). The REC task aims to localize an object described by a referring expression L in an image I represented by a set of objects $\mathcal{O} = \{o_i\}_{i=1}^{N}$, where N is the number of objects. The VQA task aims to provide an answer for a natural language question L about the image I. For simplicity, here we use the same notations L and I to represent the input sentence (*i.e.*, the referring expression/question) and the image, respectively.

To learn the dynamics of visual relational reasoning, we cast it as a path routing task on a visual graph $G = \{V, E\}$ representing the relational layout of the image I, where

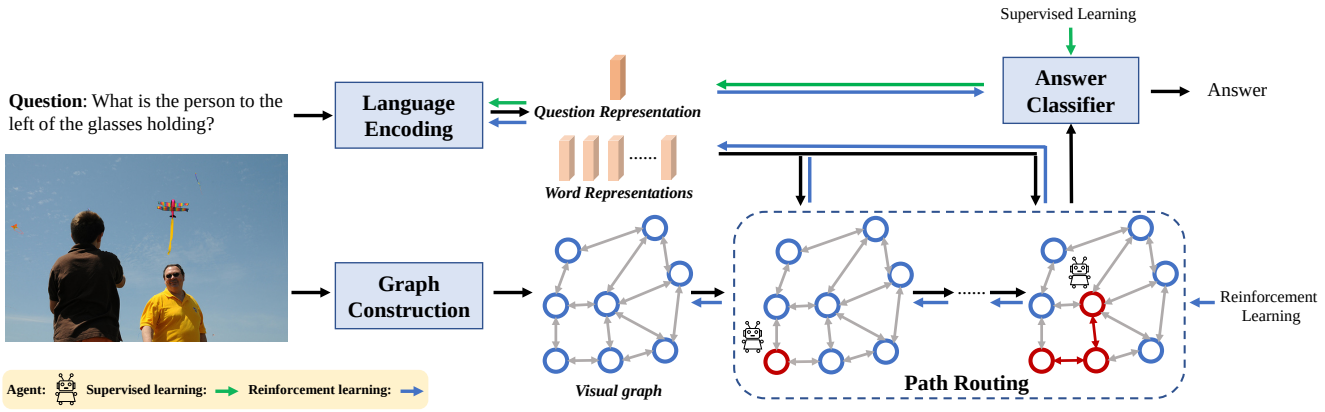

Figure 2: Overview of our method for visual question answering. Black arrows show the forward process. Blue arrows and green arrows show the backward processes of supervised learning and reinforcement learning, respectively. Our method constructs a visual graph from an input image and encodes an input question to obtain the representations of the question and words. An agent navigates on the graph according to the question. The learned path is combined with the question to predict the answer. A reinforcement learning loss and a supervised learning loss are used to train the agent in an end-to-end manner.

$V = \{v_i\}_{i=1}^N$ is a set of nodes corresponding to the objects in I. $E = \{e_{ij}\}_{i,j=1}^N$ denotes the relationships among objects. In path routing, a reasoning model learns to navigate on the graph G according to the sentence L to infer reasoning results. Here we formalize the reasoning dynamics in the context of path routing and define three concepts:

- **Reasoning state.** In path routing, we define the reasoning state s_t, the intermediate representation of a reasoning process, at each time step $t \in \{0, 1, ..., T\}$ by the current node $v_{u_t} \in V$. T denotes the number of time steps for path routing. $u_t \in \{1, 2, ..., N\}$ indicates the index of the node at the time step t.
- **State transition.** The reasoning model in path routing performs the transition from one reasoning state to another reasoning state. At each time step t, the model determines the next node $v_{u_{t+1}}$ based on the current node v_{u_t} to obtain the next state s_{t+1}.
- **Reasoning dynamics.** After the path routing, a path $p_T = \{v_{u_1}, v_{u_2}, ..., v_{u_T}\}$ is explored on the visual graph. Thus the reasoning dynamics $d = \{s_1, s_2, ..., s_T\}$, a time-indexed sequence of reasoning states, is determined.

By exploring paths, the dynamics of visual relational reasoning is supposed to be fully modeled because reasoning states and state transitions are represented and characterized clearly. To this end, we present a reinforced path routing method that models path routing as a Markov Decision Process. The sentence L and the graph G comprise the external environment. An RL-based reasoning model serves as the agent, whose states and actions correspond to reasoning states and state transitions, respectively. In the following, we illustrate how we devise, reward, and optimize the reasoning model for visual relational reasoning.

Model

Feature Encoding For a visual graph $G = \{V, E\}$, we represent a node v_i via a local feature $\boldsymbol{l}_i \in \mathbb{R}^{d_l}$ encoding the appearance information, and a spatial feature $\boldsymbol{b}_i \in \mathbb{R}^{d_b}$ encoding its location and size. The two features are concatenated and projected into a common space $\mathbb{R}^d$ via a linear mapping as $\boldsymbol{v}_i = \boldsymbol{W}_v[\boldsymbol{l}_i; \boldsymbol{b}_i]$, where $\boldsymbol{W}_v \in \mathbb{R}^{d \times (d_l + d_b)}$, and $[\cdot; \cdot]$ denotes the concatenation operation of two vectors. We use a fully-connected graph to represent the image, which means there is an undirected edge e_{ij} between each pair of nodes v_i and v_j. We do not obtain the representation of e_{ij} as its representation is not used in path routing.

For a sentence L which contains a sequence of M words $\{w_k\}_{k=1}^M$, we use a Bi-LSTM (Schuster and Paliwal 1997) to encode the sequence and project it into the common space to obtain a sentence-level representation $\boldsymbol{L} \in \mathbb{R}^d$. The word representation $\boldsymbol{w}_k \in \mathbb{R}^d$ of a word w_k is obtained by concatenating corresponding forward and backward hidden vectors and projecting it into the common space. Note that we use bold letters to denote the representations of corresponding non-bold letters throughout this paper.

Policy Network In path routing, the agent is supposed to determine which node should be added to extend the current path. A navigation policy π_{nav} is introduced for the agent. At each time step t, the policy generates a probability distribution over all the nodes based on the current state s_t. The representation of the state is obtained via $\boldsymbol{s}_t = \boldsymbol{W}_s \boldsymbol{v}_{u_t}$, where $\boldsymbol{v}_{u_t}$ is the representation of the current node and the $\boldsymbol{W}_s \in \mathbb{R}^{d \times d}$ is a learnable matrix. For the initial state s_0, we calculate a mean node representation by averaging all node representations to represent it. We use an attention-based neural network to parameterize the policy network (as shown in Figure 3), which is introduced in the following.

Firstly, a *language attention* is introduced to enable the agent to focus on different parts of the sentence at different time steps. The agent uses the attention mechanism based

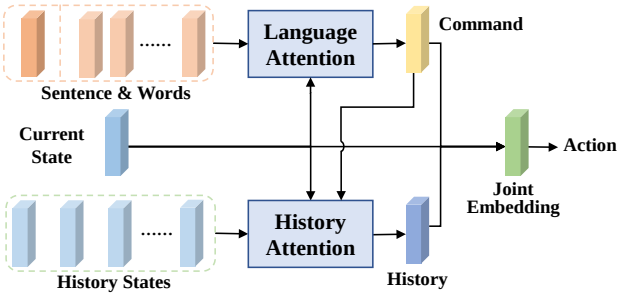

Figure 3: Architecture of the policy network. It uses a language attention to generate a textual command by focusing on important words, and a history attention to fuse the history nodes according to the command and the current state. The history embedding, the command, and the state are combined for action prediction.

on current state $\boldsymbol{s}_t$ and the sentence-level representation $\boldsymbol{L}$ to generate a command embedding:

$$\begin{aligned} \boldsymbol{c}_t &= \boldsymbol{W}_0 \sum_{k=1}^{M} \alpha_{t,k}^{L} \boldsymbol{w}_k, \\ \alpha_{t,k}^{L} &= \mathrm{Softmax}_k \left(\boldsymbol{W}_1 (\boldsymbol{w}_k \circ \boldsymbol{W}_2^t \sigma(\boldsymbol{W}_3 [\boldsymbol{L}; \boldsymbol{s}_t])) \right), \end{aligned} \tag{1}$$

where $\circ$ denotes the dot product operation of two vectors and σ denotes the RELU activation function. $\boldsymbol{w}_k$ is the representation of the k-th word in L. $\boldsymbol{W}_0 \in \mathbb{R}^{d \times d}$, $\boldsymbol{W}_1 \in \mathbb{R}^{d \times d}$, $\boldsymbol{W}_2^t \in \mathbb{R}^{d \times d}$ and $\boldsymbol{W}^3 \in \mathbb{R}^{d \times 2d}$ are learnable matrices. Specifically, $\boldsymbol{W}_2^t$ denotes a single learnable matrix for each iteration t. The obtained command embedding serves as an instruction for models to perform state transitions.

Then, based on the command and the current state, we fuse the history states via a *history attention* to generate the history embedding:

$$\begin{aligned} \boldsymbol{h}_t &= \sum_{i=1}^{t-1} \alpha_{t,i}^{H} \boldsymbol{W}_4 \boldsymbol{s}_i, \\ \alpha_{t,i}^{H} &= \mathrm{Softmax}_i \left(\boldsymbol{W}_5 \boldsymbol{s}_i \circ \boldsymbol{W}_6 [\boldsymbol{c}_t; \boldsymbol{s}_t] \right), \end{aligned} \tag{2}$$

where $\boldsymbol{W}_4 \in \mathbb{R}^{d \times d}$, $\boldsymbol{W}_5 \in \mathbb{R}^{d \times d}$ and $\boldsymbol{W}_6 \in \mathbb{R}^{d \times 2d}$ are learnable matrices.

Finally, the state representation, the command embedding, and the history embedding are fused as a joint embedding $\boldsymbol{g}_t = [\boldsymbol{c}_t; \boldsymbol{s}_t; \boldsymbol{h}_t]$ to obtain the next state. Concretely, the agent computes the probability of each node for being selected to extend the current path as

$$\alpha_{t,i}^{\mathrm{nav}} = \mathrm{Softmax}_i \left(\boldsymbol{W}_7 \boldsymbol{g}_t \circ (\boldsymbol{W}_8 [\boldsymbol{s}_t; \boldsymbol{v}_i] + \boldsymbol{W}_9 \boldsymbol{\eta}_{t,i}) \right), \tag{3}$$

where $\boldsymbol{W}_7 \in \mathbb{R}^{d \times 3d}$, $\boldsymbol{W}_8 \in \mathbb{R}^{d \times 2d}$ and $\boldsymbol{W}_9 \in \mathbb{R}^{d \times 1}$ are learnable matrices. The $\boldsymbol{v}_i$ is the representation of the i-th node. The vector $\boldsymbol{\eta}_t = \sum_{j=0}^{t-1} \boldsymbol{\alpha}_j^{\mathrm{nav}}$ is the accumulated probability distribution of previous time steps to enable the agent be aware of the history decisions. The action a_t is sampled from the distribution to obtain the next node $v_{u_{t+1}}$.

Output Module For each sentence, we use an existing language POS tagging method, the Spacy tool (Honnibal and Montani 2017), to obtain the part-of-speech (POS) tag of each word in it. Then we derive the number of steps T for path routing of the sentence by computing the number of nouns in it. The final state is thus s_T. For REC, we directly output the final node v_{u_T} as the predicted object for an input referring expression L. For VQA, we build a simple task-specific output module, an answer classifier. The answer classifier projects the inputs into a probability distribution over all possible answers as $\boldsymbol{\alpha}^{ans} = \boldsymbol{W}_{ans}[\boldsymbol{s}_T; \boldsymbol{L}]$, where $\boldsymbol{\alpha}^{ans}$ denotes the output probability distribution and $\boldsymbol{W}_{ans} \in \mathbb{R}^{N^a \times 2d}$ denotes a learnable matrix. N^a is the number of answers. The answer with the highest probability $\hat{a}$ is regarded as the predicted answer.

Reward The ultimate goal of the agent is to infer reasoning results that match the objective of the reasoning task. For REC, the objective is achieved if the final node v_{u_T} is the same as the ground-truth node v_{gt}, while the objective of VQA is achieved if the predicted answer $\hat{a}$ is exactly the ground-truth answer a_{gt}. According to whether the objective is achieved, we devise an accuracy reward

$$R_t = \begin{cases} 10, & \text{if the objective is achieved} \\ 0. & \text{otherwise} \end{cases} \tag{4}$$

Optimization

Recent methods using reinforcement learning in vision-language tasks (Nguyen and Daumé III 2019; Zhou et al. 2020; Zhao, Wu, and Luo 2021; Wang et al. 2021) reveal that warm-starting the agent with supervised learning can guarantee a relatively good policy. However, applying this strategy in path routing is non-trivial. Firstly, the supervision for reasoning dynamics is unavailable, thus we can not directly optimize the policy network. Secondly, due to the non-differentiability of the action prediction, it's infeasible to train the model with the supervision of reasoning results. To address this issue, we introduce a pre-training strategy that encourages to model to gradually learn to move from one node to another node in a supervised learning manner. Based on the pre-training strategy, a two-phase learning strategy is developed, which is illustrated in the following.

Supervised Learning The softmax function is used to compute the probability distribution over all actions in Eq. (3). Generally, given the probability distribution, an RL-based agent either uses a deterministic policy by taking the action with the highest probability or uses a stochastic policy by randomly selecting an action. Both policies lead to non-differentiability.

Thus we use a differentiable scaled softmax function $\mathrm{softmax}(\beta \boldsymbol{x})$, where $\beta > 0$ is an scaling parameter, to replace the softmax function in Eq. (3) for the navigation policy. By increasing the scaling parameter, the function will become more non-smooth and thus can be used to approximate the deterministic policy as (Hinton, Vinyals, and Dean 2015; Jang, Gu, and Poole 2017).

At the beginning of the supervised learning, we set $\beta = 1$ to generate the probability distribution. Based on the distri-

bution, the representation of the next state can be obtained by aggregating all node representations. During the training process, we gradually increase β by setting $\beta_i = (1+\gamma^i)^\lambda$, where i is the current iteration number, γ and λ are two hyper-parameters. Intuitively, we allow the agent to softly navigate to "multiple" nodes at the beginning but enforce it gradually learns to focus on only one node during the training process.

After T rounds of path routing, we compute the task-specific objective of supervised learning via a cross-entropy loss as

$$\mathcal{L}_{\text{task}}^{\text{SL}} = -\boldsymbol{y}_{\text{task}} \log\big(\boldsymbol{x}_{\text{task}}\big), \tag{5}$$

where $\text{task} \in \{\text{rec}, \text{vqa}\}$, and $\boldsymbol{y}_{\text{task}}$ is an one-hot label whose element representing the ground truth answer/object is 1 and others are 0. $\boldsymbol{x}_{\text{rec}} = \boldsymbol{\alpha}_T^{\text{nav}}$ denotes the probability distribution of the navigation policy at the final time step. $\boldsymbol{x}_{\text{vqa}} = \boldsymbol{\alpha}^{ans}$ denotes the probability distribution generated by the answer classifier.

Besides, we introduce a visual coverage loss to encourage the agent to focus on different nodes at different time steps in supervised learning, inspired by the coverage mechanism in text summarization (See, Liu, and Manning 2017). The loss penalizes the attention distribution that is similar to history distributions and is computed as

$$\mathcal{L}_{\text{cover}}^{\text{SL}} = \mu \sum_{t=0}^{T-1} \sum_{i=0}^{N-1} \min(\eta_{t,i}, \alpha_{t,i}^{nav}), \tag{6}$$

where $\boldsymbol{\eta}_t$ is the accumulated probability distribution of previous time steps. The coverage loss and the task-specific loss jointly supervise the model learning and μ is a hyper-parameter to balance the losses.

Reinforcement Learning We use a policy gradient method, the advantage actor-critic (A2C) algorithm (Mnih et al. 2016), to train the policy network. The gradient of reinforcement learning is calculated as

$$\nabla_\theta J(\theta) = \sum_{t=1}^{T} \nabla_\theta \log \pi_{\text{nav}}\left(a_t^p \mid s_{0:t-1}\right) \big(\sum_{i=t}^{T} R_i - b_t\big), \tag{7}$$

where θ denotes the parameters of the policy network. b_t is the expected accumulated reward learned via a value function. A single fully-connected layer is used to map the current state representation $\boldsymbol{s}_t$ to the expected reward b_t. Note that, for VQA, due to the existence of the answer classifier, the reinforcement learning is combined with the supervised answer classification loss to train the model in an end-to-end manner, as shown in Figure 2.

Experiment

We apply the proposed method on two tasks, REC and VQA, to evaluate its effectiveness. We first evaluate our method on REC, which tests the relational reasoning capability of models. We use two REC datasets: the CLEVR-Ref+ (Liu et al. 2019b) that is a synthetic diagnostic dataset, and the Ref-reasoning (Yang, Li, and Yu 2020) that contains real images. The reason to choose these two datasets is that they can provide more complex referring expressions that require strong reasoning capability. Secondly, we evaluate our method on VQA, which tests not only relational reasoning ability but also other capabilities such as question answering and commonsense reasoning. The challenging GQA dataset (Hudson and Manning 2019b) that contains compositional questions about real-world images is used. In the following, we illustrate the experimental settings and results for both tasks.

Referring Expression Comprehension

Datasets The CLEVR-Ref+ (Liu et al. 2019b) contains synthetic images and automatically generated referring expressions. There are a train split and a val split in the CLEVR-Ref+ dataset. A uniform sampling strategy is employed to guarantee the dataset is approximately unbiased. The Ref-Reasoning dataset is a large-scale real-world dataset. It includes a train split, a val split, and a test split. The expressions of these splits are generated by using diverse expression templates and functional programs over scene graphs of images to guarantee diversity. These expressions may involve multiple objects and thus require strong visual reasoning ability to solve.

Implementation details We use object-level features for the two datasets because our method explicitly navigates from one object to another. The Ref-reasoning provides the 2048-d object-level features detected by the Faster R-CNN detector (Ren et al. 2015). For the CLEVR-Ref+, we follow the settings of (Hu et al. 2019) and use 1024-d object features extracted from the ResNet-101 (He et al. 2016). The ground-truth bounding boxes are used for evaluations. For the Ref-reasoning, the hyper-parameters μ, λ and γ are set as 0.01, 0.5, and 0.01. For the CLEVR-Ref+, the three hyper-parameters are set as 0.01, 0.5, and 0.001. The max number of time steps is set as 4 for the Ref-reasoning and 3 for the CLEVR-Ref+. For both datasets, the dimensions of the spatial feature d_b and the common space d are set as 128, and 512, respectively.

Comparisons with state-of-the-art methods The results of our method and state-of-the-art methods on the Ref-Reasoning dataset and the CLEVR-Ref+ dataset are listed in Table 1 and Table 2, respectively. We found from the tables that our method outperforms the others on both datasets, which demonstrates the effectiveness of our method for REC in both synthetic and real image datasets.

For the Ref-Reasoning dataset (from Table 1), the results on the val split and the test split are presented. For the test split, the results on four subsets are also listed, where different subsets contain expressions with different numbers of objects. It can be seen from the table that our method is superior to other methods in both splits. For the test split, the improvement in the subsets with more objects is more significant than that in the subsets with fewer objects, which demonstrates the multi-step reasoning capability of our method. The DGA (Yang, Li, and Yu 2019b), the CMRIN (Yang, Li, and Yu 2019a), and the SGMN (Yang, Li, and Yu 2020) also build visual graphs to capture the relational layout of images. The DGA and the CMRIN perform language-guided visual graph convolution over visual graphs, while the SGMN performs modular reasoning over visual graphs. Benefiting from path routing, our method is

Methods	Number of Objects				Split	
	one	two	three	$\geq$ four	val	test
CNN	10.57	13.11	14.21	11.32	12.36	12.15
CNN+LSTM	75.29	51.85	46.26	32.45	42.38	42.43
DGA (Yang, Li, and Yu 2019b)	73.14	54.63	48.48	37.63	45.37	45.87
CMRIN (Yang, Li, and Yu 2019a)	79.20	56.87	50.07	35.29	45.43	45.87
SGMN (Yang, Li, and Yu 2020)	79.71	61.77	55.57	41.89	51.04	51.39
Ours	**81.62**	**62.43**	**56.60**	**43.95**	**52.22**	**53.02**

Table 1: Results of our method and the state-of-the-art methods on the val split and the test split of the Ref-Reasoning dataset.

Methods	Accuracy
Stack-NMN (Hu et al. 2018)	56.5
SLR (Yu et al. 2017)	57.7
MAttNet (Yu et al. 2018)	60.9
GroundeR (Rohrbach et al. 2016)	61.7
LCGN (Hu et al. 2019)	74.8
LCGN†	76.0
Ours†	**77.7**

Table 2: Results of our method and the state-of-the-art on the val split of the CLEVR-Ref+ dataset. † indicates this method uses ground-truth bounding boxes. The other methods use grid features.

Methods	Accuracy
Ours (w/o history attention)	48.40
Ours (w/o history decision)	52.15
Ours (w/o pre-training)	19.16
Ours (w/o scaled activation)	50.80
Ours (w/o coverage loss)	49.77
Ours (w adaptive nodes)	51.84
Ours	**53.02**

Table 3: Results of different variants of our model on the test split of the Ref-Reasoning dataset.

capable of achieving complex visual relational reasoning and thus outperforms all three methods.

In the CLEVR-Ref+ dataset (from Table 2), our method outperforms other methods. To make fair comparisons, we train the LCGN (Hu et al. 2019) model with ground truth bounding boxes. And our method also outperforms this model. These results demonstrate the effectiveness of our method for grounding complex referring expressions on synthetic images.

Ablation Studies We evaluate different variants of our model by ablating certain components to study the effectiveness of several important components of our method. The results of those models on the test split of the Ref-Reasoning dataset are shown in Table 3.

Firstly, we investigate the influence of the history states and history attention in the policy network. We remove the history attention of the policy network and obtain a model called "Ours (w/o history attention)". We observe the model performs significantly worse than our full model, which demonstrates the history states are critical in path routing for visual relational reasoning. Similarly, by removing the term about history decisions in Eq (3), we obtain "Ours (w/o history decisions)". The comparisons between the model and the full model show that the history decisions are also beneficial for path routing.

Then, we study the influence of the pre-training strategy. We train the agent from scratch and obtain a model called "Ours (w/o pre-training)". We find that the model performs significantly worse than the full model, which demonstrates that the pre-training is indispensable for path routing. The main reason is that the visual and language inputs are noisy and directly performing path routing without any initialization may lead to collapse. By using a vanilla softmax function in pre-training, we obtain the "Ours (w/o scaled activation)". Besides, we remove the coverage loss and obtain the "Ours (w/o coverage loss)". These models are also inferior to the full model, which shows gradually learning to move from one node to another in pre-training is beneficial.

Finally, we modify our model to let it adaptively focus on one to three nodes based on the outputted probability distribution. The representations of the attended nodes are summed to obtain the state representation for further path routing. The obtained model is entitled "Ours (w adaptive nodes)". We observe that although its potential capability is stronger, its performance is only comparable with the full model. The reason is that the dataset mainly focuses on relations of two objects rather than three or more objects (such as "surrounded by"). We found that current datasets about relational reasoning hardly contain samples involving relations of three or more objects.

Visual Question Answering

Datasets The GQA dataset (Hudson and Manning 2019b) is a large-scale dataset with 140K real images and 1.7M balanced questions-answer pairs. The dataset has a train split for training, a test-dev split for validation, and a test split for online testing. Each image in the GQA is associated with a manually annotated scene graph describing the classes, attributes, relations of objects in the image. Based on the scene graphs, diverse compositional questions that require multi-step reasoning are generated via a question engine.

Implementation details In our implementation, the bottom-up-attention model (Anderson et al. 2018) is used to extract 2048-d object-level features. For each image, we keep the top 48 bounding boxes ranked by confidence scores. The hyper-parameters μ, λ and γ are set as 0.001, 0.5 and 0.001,

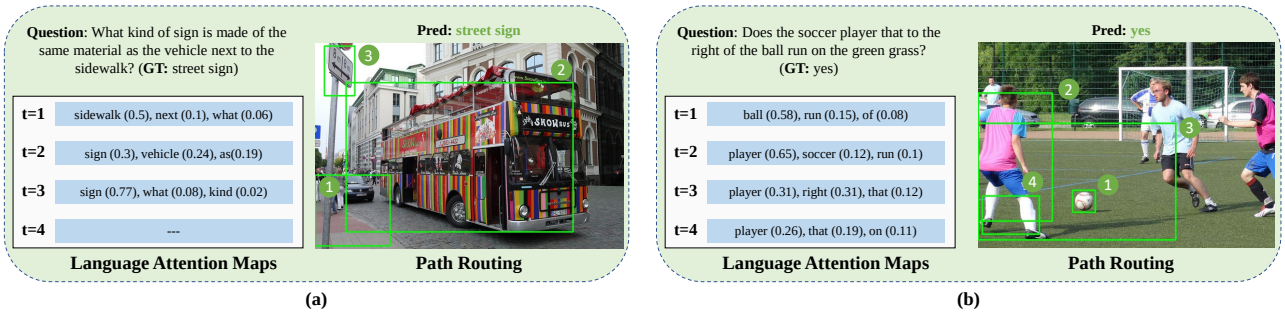

Figure 4: Qualitative examples from the test-dev split of the GQA. For each example, the upper-left corner shows a question and the answer of it. The lower-left corner shows the language attention maps of our method. For each language attention map, the top-3 words with the highest attention values are shown. The right side shows the nodes during the path routing and the predicted answer for the question. The node in the t-th time step is marked by a green rectangle with the corresponding number.

Methods	Required Inputs	Accuracy
Bottom-Up (Anderson et al. 2018)	V+L	49.74
MAC (Hudson and Manning 2018)	V+L	54.06
NMN (Andreas et al. 2016)	V+L+Program	55.70
BAN (Kim, Jun, and Zhang 2018)	V+L	57.10
GRN (Guo, Xu, and Tao 2019)	V+L	57.04
LCGN (Hu et al. 2019)	V+L	57.07
LXMERT (Tan and Bansal 2019)	V+L	60.33
MMN (Chen et al. 2021)	V+L+Program	60.83
NSM (Hudson and Manning 2019a)	V+L+SceneModel	63.17
Ours	V+L	59.43

Table 4: Results of our method and the state-of-the-art methods on the test split of the GQA dataset.

respectively. The max number of time steps is set as 4. The dimensions of the spatial feature and the common space are set as 96, and 512, respectively. In the supervised learning stage, we train the model with the "all" split of questions of the GQA and fine-tune it with the "balanced" split as (Chen et al. 2021). In supervised learning, we also perform step-wise supervision as (Chen et al. 2021) by training a model with ground-truth scene graphs of the GQA dataset and executing knowledge distillation. The obtained model is then used to initialize the agent for reinforcement learning.

Comparisons with state-of-the-art methods The results of the proposed method and the state-of-the-art methods on the test split are listed in Table 4. We observe that our method achieves comparable performance with other methods, which demonstrates the effectiveness of our method for answering compositional questions through complex relational reasoning. Our method outperforms two graph-network-based visual reasoning methods, the GRN (Guo, Xu, and Tao 2019) and the LCGN (Hu et al. 2019), thanks to the learning of dynamics. The main reason that we do not surpass all the state-of-the-art is that the GQA evaluates not only relational reasoning capability but also the question answering. But in this paper, we mainly focus on visual relational reasoning. By contrast, most previous models are tailored for the VQA task and cannot be applied in other tasks such as REC. The NSM (Hudson and Manning 2019a) relies on a scene graph generation model (Yang et al. 2018) and the LXMERT (Tan and Bansal 2019) uses multiple datasets to pre-train their model.

Qualitative Results We visualize the reasoning processes of our method to demonstrate its transparency. Figure 4 shows two qualitative examples from the test-dev split of the GQA. For each example, we provide the language attention map and the current node of each time step t in path routing. It is shown that our method can relatively accurately focus on nouns in questions and further localize the corresponding objects in images. In the left example, it gradually localizes the sidewalk, the vehicle, and the sign and then figures out the correct answer. The overall reasoning process is almost faithful and close to the thinking process of humans.

Conclusion and Future work

In this paper, we have presented a reinforced path routing method for visual relational reasoning, providing a new point of view for this area. Our method learns the dynamics of reasoning by introducing a reasoning model to explore paths over a visual graph based on an input sentence to infer reasoning results. Extensive experiments demonstrate that our method is capable of achieving accurate visual relational reasoning with transparent reasoning processes.

The results of this work illustrate that learning the dynamics is an effective and promising way to simulate human reasoning ability. It is worth mentioning that there are two avenues for further studies to learn paths consistent with the thinking processes of humans. Firstly, we would like to introduce a rollback mechanism to enable the reasoning model back to one previous node when it navigates to a wrong node. Secondly, we plan to curate a visual relational reasoning dataset with human-annotated trajectories of attention to supervise and evaluate path routing.

Acknowledgments

This work was supported by the Natural Science Foundation of China (NSFC) under Grants No. 62172041 and No. 62176021.

References

Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6077–6086.

Andreas, J.; Rohrbach, M.; Darrell, T.; and Klein, D. 2016. Neural module networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 39–48.

Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Lawrence Zitnick, C.; and Parikh, D. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2425–2433.

Chen, W.; Gan, Z.; Li, L.; Cheng, Y.; Wang, W.; and Liu, J. 2021. Meta module network for compositional visual reasoning. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 655–664.

Deng, J.; Yang, Z.; Chen, T.; Zhou, W.; and Li, H. 2021. TransVG: End-to-End Visual Grounding With Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 1769–1779.

Engelfriet, J.; and Treur, J. 1994. Temporal theories of reasoning. In *European Workshop on Logics in Artificial Intelligence*, 279–299. Springer.

Guo, D.; Xu, C.; and Tao, D. 2019. Graph reasoning networks for visual question answering. *arXiv preprint arXiv:1907.09815*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Hong, R.; Liu, D.; Mo, X.; He, X.; and Zhang, H. 2019. Learning to Compose and Reason with Language Tree Structures for Visual Grounding. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Honnibal, M.; and Montani, I. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1): 411–420.

Hu, R.; Andreas, J.; Darrell, T.; and Saenko, K. 2018. Explainable neural computation via stack neural module networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 53–69. Springer.

Hu, R.; Andreas, J.; Rohrbach, M.; Darrell, T.; and Saenko, K. 2017. Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 804–813.

Hu, R.; Rohrbach, A.; Darrell, T.; and Saenko, K. 2019. Language-Conditioned Graph Networks for Relational Reasoning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 10294–10303.

Hu, R.; Xu, H.; Rohrbach, M.; Feng, J.; Saenko, K.; and Darrell, T. 2016. Natural language object retrieval. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4555–4564.

Hudson, D.; and Manning, C. D. 2019a. Learning by abstraction: The neural state machine. In *Advances in Neural Information Processing Systems (NeurIPS)*, 5901–5914.

Hudson, D. A.; and Manning, C. D. 2018. Compositional Attention Networks for Machine Reasoning.

Hudson, D. A.; and Manning, C. D. 2019b. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6700–6709.

Jang, E.; Gu, S.; and Poole, B. 2017. Categorical reparameterization with gumbel-softmax.

Jing, C.; Wu, Y.; Pei, M.; Hu, Y.; Jia, Y.; and Wu, Q. 2020a. Visual-Semantic Graph Matching for Visual Grounding. In *Proceedings of the 28th ACM International Conference on Multimedia*, 4041–4050.

Jing, C.; Wu, Y.; Zhang, X.; Yunde, J.; and Wu, Q. 2020b. Overcoming Language Priors in VQA via Decomposed Linguistic Representations. In *Thirty-Forth AAAI Conference on Artificial Intelligence*, 11181–11188.

Johnson, J.; Hariharan, B.; van der Maaten, L.; Hoffman, J.; Fei-Fei, L.; Lawrence Zitnick, C.; and Girshick, R. 2017. Inferring and executing programs for visual reasoning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2989–2998.

Jonker, C. M.; and Treur, J. 2002. Analysis of the dynamics of reasoning using multiple representations. *Artificial Intelligence Preprint Series*, 36.

Jonker, C. M.; and Treur, J. 2003. Modelling the dynamics of reasoning processes: Reasoning by assumption. *Cognitive Systems Research*, 4(2): 119–136.

Kim, J.-H.; Jun, J.; and Zhang, B.-T. 2018. Bilinear Attention Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1571–1581.

Liao, Y.; Liu, S.; Li, G.; Wang, F.; Chen, Y.; Qian, C.; and Li, B. 2020. A Real-Time Cross-modality Correlation Filtering Method for Referring Expression Comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 10880–10889.

Liu, D.; Zhang, H.; Zha, Z.-J.; and Wu, F. 2019a. Learning to Assemble Neural Module Tree Networks for Visual Grounding. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 4673–4682.

Liu, R.; Liu, C.; Bai, Y.; and Yuille, A. L. 2019b. Clevr-ref+: Diagnosing visual reasoning with referring expressions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4185–4194.

Liu, Y.; Wan, B.; Zhu, X.; and He, X. 2020. Learning Cross-modal Context Graph for Visual Grounding. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 11645–11652.

Mascharka, D.; Tran, P.; Soklaski, R.; and Majumdar, A. 2018. Transparency by design: Closing the gap between performance and interpretability in visual reasoning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4942–4950.

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous Methods for Deep Reinforcement Learning. Proceedings of Machine Learning Research, 1928–1937. PMLR.

Nguyen, K.; and Daumé III, H. 2019. Help, Anna! Visual Navigation with Natural Multimodal Assistance via Retrospective Curiosity-Encouraging Imitation Learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 684–695.

Perez, E.; Strub, F.; De Vries, H.; Dumoulin, V.; and Courville, A. 2018. Film: Visual reasoning with a general conditioning layer. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Port, R. F.; and Van Gelder, T. 1995. *Mind as motion: Explorations in the dynamics of cognition*. MIT press.

Qiao, Y.; Deng, C.; and Wu, Q. 2020. Referring Expression Comprehension: A Survey of Methods and Datasets. *arXiv preprint arXiv:2007.09554*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 91–99.

Rohrbach, A.; Rohrbach, M.; Hu, R.; Darrell, T.; and Schiele, B. 2016. Grounding of textual phrases in images by reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 817–834. Springer.

Schuster, M.; and Paliwal, K. K. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11): 2673–2681.

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Shi, J.; Zhang, H.; and Li, J. 2019. Explainable and Explicit Visual Reasoning over Scene Graphs. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 8376–8384.

Tan, H.; and Bansal, M. 2019. LXMERT: Learning Cross-Modality Encoder Representations from Transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 5100–5111. Association for Computational Linguistics.

Tang, K.; Zhang, H.; Wu, B.; Luo, W.; and Liu, W. 2019. Learning to compose dynamic tree structures for visual contexts. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6619–6628.

Wang, H.; Wang, W.; Liang, W.; Xiong, C.; and Shen, J. 2021. Structured Scene Memory for Vision-Language Navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8455–8464.

Wang, P.; Wu, Q.; Cao, J.; Shen, C.; Gao, L.; and Hengel, A. v. d. 2019. Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1960–1968.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4): 229–256.

Wu, F.; Xu, Z.; and Yang, Y. 2017. An end-to-end approach to natural language object retrieval via context-aware deep reinforcement learning. *arXiv preprint arXiv:1703.07579*.

Wu, Q.; Teney, D.; Wang, P.; Shen, C.; Dick, A.; and van den Hengel, A. 2017. Visual question answering: A survey of methods and datasets. *Computer Vision and Image Understanding*, 163: 21–40.

Yang, J.; Lu, J.; Lee, S.; Batra, D.; and Parikh, D. 2018. Graph r-cnn for scene graph generation. In *Proceedings of the European conference on computer vision (ECCV)*, 670–685.

Yang, S.; Li, G.; and Yu, Y. 2019a. Cross-Modal Relationship Inference for Grounding Referring Expressions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4145–4154.

Yang, S.; Li, G.; and Yu, Y. 2019b. Dynamic graph attention for referring expression comprehension. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 4644–4653.

Yang, S.; Li, G.; and Yu, Y. 2020. Graph-Structured Referring Expression Reasoning in The Wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 9952–9961.

Yu, L.; Lin, Z.; Shen, X.; Yang, J.; Lu, X.; Bansal, M.; and Berg, T. L. 2018. Mattnet: Modular attention network for referring expression comprehension. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1307–1315.

Yu, L.; Tan, H.; Bansal, M.; and Berg, T. L. 2017. A joint speaker-listener-reinforcer model for referring expressions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7282–7290.

Zhao, W.; Wu, X.; and Luo, J. 2021. Multi-modal Dependency Tree for Video Captioning. *Advances in Neural Information Processing Systems*.

Zhou, Y.; Wang, M.; Liu, D.; Hu, Z.; and Zhang, H. 2020. More Grounded Image Captioning by Distilling Image-Text Matching Model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4777–4786.