Multi-step Reasoning via Recurrent Dual Attention for Visual Dialog

Zhe Gan[1], Yu Cheng[1], Ahmed EI Kholy[1], Linjie Li[1], Jingjing Liu[1], Jianfeng Gao[2]
[1]Microsoft Dynamics 365 AI Research   [2]Microsoft Research
{zhe.gan, yu.cheng, ahmed.elkholy, lindsey.li, jingjl, jfgao}@microsoft.com

Abstract

This paper presents Recurrent Dual Attention Network (ReDAN) for visual dialog, using multi-step reasoning to answer a series of questions about an image. In each turn of the dialog, ReDAN infers answers progressively through multiple steps. In each step, a recurrently-updated semantic representation of the (refined) query is used for iterative reasoning over both the image and previous dialog history. Experimental results on VisDial v1.0 dataset show that the proposed ReDAN model outperforms prior state-of-the-art approaches across multiple evaluation metrics. Visualization on the iterative reasoning process further demonstrates that ReDAN can locate context-relevant visual and textual clues leading to the correct answers step-by-step.

1. Introduction

There has been a recent surge of interest in developing neural network models capable of understanding both vision and natural language, with applications ranging from image captioning [13, 50, 52] to visual question answering (VQA) [3, 14, 1] and visual dialog [9, 11, 10]. Unlike VQA, where the model can answer a single question about an image, a visual dialog system can answer a series of questions regarding an image, which requires further understanding on both the image and the dialog history.

Most recent work on visual dialog performs reasoning via attention mechanisms [4, 52], to emphasize specific regions of the image and dialog-history snippets that are relevant to the question. These attention models measure the relevance between a query and the correspondingly attended image as well as the dialog context. To generate an answer, either a discriminative decoder is used for scoring answer candidates, or a generative decoder is trained for synthesizing an answer [9, 32]. Though promising results have been reported, these models often fail to provide an accurate answer in the cases where answers are confined to particular image regions or dialog-history snippets.

One hypothesis for the cause of these failed cases is the inherent limitation of single-step reasoning. Intuitively, af-
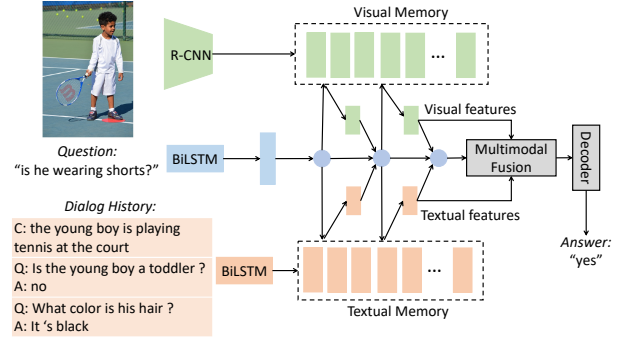
(a) Overview of the proposed ReDAN framework.
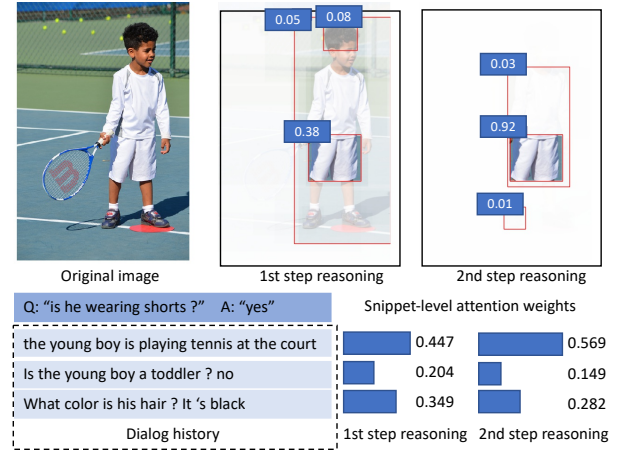
(b) An example of multi-step reasoning in ReDAN.

Figure 1: Model architecture and visualization of the learned multi-step reasoning strategies. In the first step, ReDAN first focuses on all relevant objects in the image (*e.g.*, "*boy*", "*shorts*"), and all relevant facts in the dialog history (*e.g.*, "*young boy*", "*playing tennis*", "*black hair*"). In the second step, the model then narrows down the focus to more context-relevant regions and dialog context (*i.e.*, the attention map becomes sharper) which leads to the correct answer ("*yes*"). The numbers in the bounding boxes and the histograms are the attention weights of the corresponding objects or dialog history snippets.

ter taking a first glimpse of the image and the dialog history, readers often revisit some specific sub-areas to obtain

a better understanding of the multimodal context. Inspired by this, we propose a Recurrent Dual Attention Network (ReDAN) that utilizes multi-step reasoning for visual dialog.

Figure 1a provides an overview of the model architecture of ReDAN. First, a set of visual and textual memories are created to store image features and dialog context. In each step, a semantic representation of the question is used as query to attend to the memories, in order to obtain a query-aware image representation and dialog-history representation, both of which subsequently contribute to updating the query vector via a recurrent neural network. Later reasoning steps typically provide a sharper attention distribution, focusing on regions more relevant to the answer. Finally, after several iterations of reasoning steps, the refined query vector and visual/textual clues are fused to obtain a final multimodal context vector, which is fed to the decoder to generate the answer. The described multi-step reasoning process is performed in each turn of the dialog.

Figure 1b provides an illustrative example of the iterative reasoning process. To answer the question "*is he wearing shorts?*", in the initial stage, the system needs to draw knowledge from previous dialog history to know what "*he*" refers to (*i.e.*, "*the young boy*"), as well as understanding the image to rule out irrelevant objects (e.g., *"net", "racket" and "court"*). After this, the system performs a second round of reasoning to pinpoint the image region ("*shorts*", whose attention weight increases from 0.38 to 0.92 from the 1st step to the 2nd step) and the dialog-history snippet ("*playing tennis at the court*", attention weight increasing from 0.447 to 0.569), which are most indicative of the correct answer ("*yes*").

The main contributions of this paper are three-fold. (i) We propose a ReDAN framework that supports multi-step reasoning for visual dialog. (ii) Our proposed model outperforms prior state of the art and achieves the best published results on the VisDial dataset. (iii) Comprehensive evaluation and visualization analysis demonstrate the effectiveness of our model in inferring answers progressively through iterative reasoning steps.

2. Related Work

Visual Dialog The visual dialog task was recently proposed by [9] and [11]. Specifically, [9] released the VisDial dataset, which contains free-form natural language questions and answers. And [11] introduced the GuessWhat?! dataset, where the dialogs provided are more goal-oriented and aimed at object discovery within an image, through a series of yes/no questions between two dialog agents.

For the VisDial task, a typical visual dialog system follows the encoder-decoder framework proposed in [48]. Different encoder models have been explored, including late fusion, hierarchical recurrent network, memory net-

work (all three proposed in [9]), early answer fusion [22], history-conditional image attention [32], and sequential co-attention [51]. The decoder model usually falls into two categories: (i) generative decoder to generate the answer with a Recurrent Neural Network (RNN) [9]; and (ii) discriminative decoder to rank answer candidates via a softmax-based cross-entropy loss [9] or a ranking-based multi-class N-pair loss [32].

Reinforcement Learning (RL) was used in [10, 5] to train two agents to play image guessing games. [32] proposed a training schema to effectively transfer knowledge from a pre-trained discriminative model to a generative dialog model. Generative Adversarial Network [16, 55, 29] was used in [51] to generate answers indistinguishable from human-generated answers, and a conditional variational autoencoder [25, 43] was developed in [33] to promote answer diversity. [39] solves visual coreference resolution implicitly via attention memory, while [26] solves the same problem, but using a more explicit reasoning procedure based on neural module networks [2]. In addition to answering questions, the ability of generating sequence of questions is also investigated in [22, 33].

For the GuessWhat?! task, various methods such as RL have been proposed to improve the performance of dialog agents, measured by task completion rate as in goal-oriented dialog [46, 40, 47, 28, 57]. Other related work includes image-grounded chitchat [35] and dialog-based image retrieval [18]. A recent survey on neural approaches to dialog modeling can be found in [15].

In this work, we focus on the VisDial task. Distinct from previous approaches to visual dialog, which all used a single-step reasoning strategy, we propose a novel multi-step reasoning framework that can boost the performance of visual dialog systems by processing the image and the dialog history iteratively.

Multi-step Reasoning The idea of multi-step reasoning has been explored in many tasks, including image classification [34], text classification [54], image generation [17], language-based image editing [6], Visual Question Answering (VQA) [53, 36], and Machine Reading Comprehension (MRC) [8, 12, 20, 44, 41, 31]. Specifically, [34] introduced an RNN for image classification, by adaptively selecting a sequence of regions and only processing the selected regions. [54] used an RNN for text classification, by learning to skip irrelevant information when reading the text input. A recurrent variational autoencoder termed DRAW was proposed in [17] for multi-step image generation. A recurrent attentive model for image editing was also proposed in [6] to fuse image and language features via multiple steps.

For VQA, Stacked Attention Network (SAN) [53] was proposed to attend the question to relevant image regions via multiple attention layers. For MRC, ReasoNet [41] was
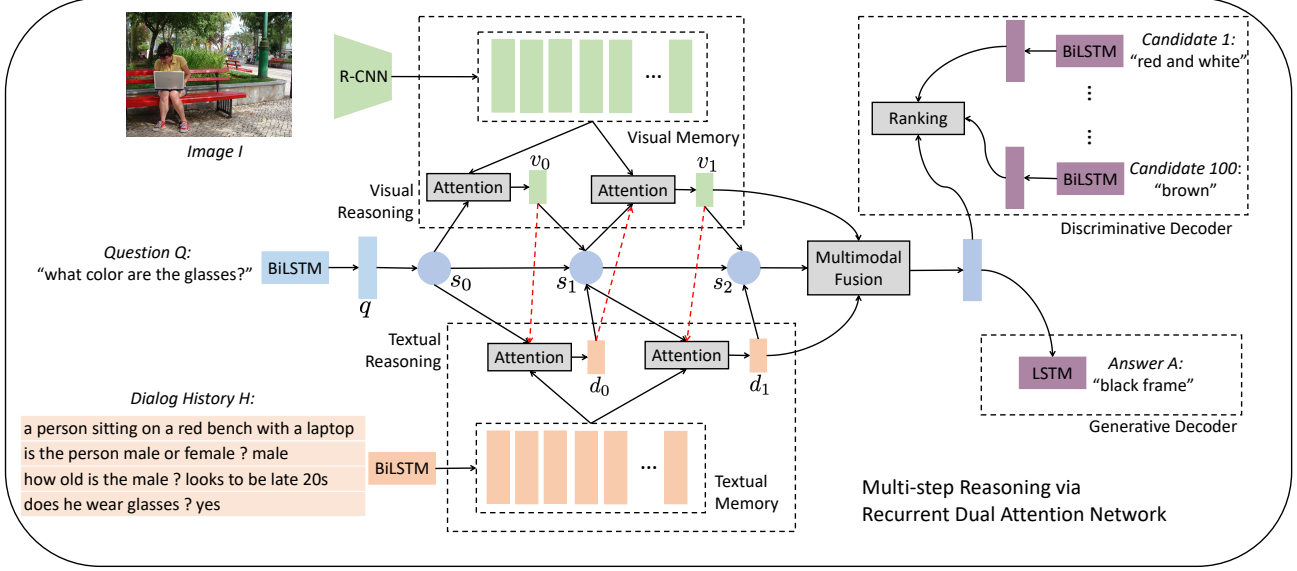
Figure 2: Model Architecture of Recurrent Dual Attention Network for visual dialog. Please see Sec. 3 for details.

developed to perform multi-step reasoning to infer the answer span based on a given passage and a question, where the number of steps can be dynamically determined via a termination gate.

Our proposed ReDAN is conceptually similar to SAN for VQA [53] and ReasoNet for MRC [41]. However, different from these two models that reason over a single type of input (either images or text), ReDAN incorporates multimodal context that contains both visual information and textual dialog history. This presents a mutual enhancement between image and text inputs for a better understanding on both: on the one hand, the attended image regions can provide additional information for better dialog-history interpretation; on the other hand, the attended history snippets can be used for better image understanding.

3. Recurrent Dual Attention Network

The visual dialog task [9] is formulated as follows: given a question Q_i grounded in an image I, and the dialog history (including the image caption C) $H_i = \{C, (Q_1, A_1), \cdots, (Q_{i-1}, A_{i-1})\}$ in the current turn i as additional context, the goal is to generate an answer by ranking a list of N candidate answers $\mathcal{A}_i = \{A_i^{(1)}, \ldots, A_i^{(N)}\}$.

As illustrated in Figure 2, a Recurrent Dual Attention Network (ReDAN) is designed to infer visual dialog answers progressively via multi-step reasoning. Specifically, ReDAN consists of three components: (*i*) *Memory Generation* (Sec. 3.1), where a set of visual and textual memories are generated to provide the grounding for reasoning; (*ii*) *Multi-step Reasoning* (Sec. 3.2), where recurrent dual

attention is applied to jointly encode question, image and dialog history as a multimodal context vector for decoding; and (*iii*) *Answer Decoding* (Sec. 3.3), where the multimodal context vector is used to derive the final answer. The following sub-sections describe the details of these three components.

3.1. Memory Generation Module

In this module, the image I and the dialog history H_i are transformed into a set of memory vectors (visual and textual).

Visual Memory   We use a pre-trained Faster R-CNN [38, 1] to extract image features, to enable attention at object-level and other salient region-level, each associated with a feature vector. Compared to the image features extracted from VGG-Net [42] and ResNet [19], this type of features has achieved state-of-the-art performance in both image captioning and VQA [1, 49] tasks. Specifically, the image features $\mathbf{F}_I$ for a raw image I are represented by:

$$\mathbf{F}_I = \text{R-CNN}(I) \in \mathbb{R}^{n_f \times M}, \qquad (1)$$

where $M = 36$ is the number of detected objects in an image, and $n_f = 2048$ is the dimension of the feature vector. A single-layer perceptron is used to transform each feature into a new vector that has the same dimension as the query vector (described in Sec. 3.2):

$$\mathbf{M}_v = \tanh(\mathbf{W}_I \mathbf{F}_I) \in \mathbb{R}^{n_h \times M}, \qquad (2)$$

where $\mathbf{W}_I \in \mathbb{R}^{n_h \times n_f}$. All the bias terms in this paper are omitted for simplicity. $\mathbf{M}_v$ is the visual memory, and its m-

th column corresponds to the visual feature vector for the region of the object indexed by m.

Textual Memory In the i-th dialogue turn, the dialog history H_i consists of the caption C and $i - 1$ rounds of QA pairs (Q_j, A_j) $(j = 1, \ldots, i - 1)$. We first concatenate all these text segments in sequential order, resulting in a "*passage*" denoted as $H = (u_1, \ldots, u_L)$, where the subscript i for H_i is omitted for ease of discussion. u_ℓ is the ℓ-word in the history, and L is the maximum length of the dialog history. If the number of words in the history is less than L, zero padding is added. We embed each word u_ℓ into a vector through a learned word embedding matrix, then a bidirectional Long Short-Term Memory (LSTM) network [21] is applied to obtain a contextualized representation for each word. Specifically,

$$x_\ell = \mathbf{W}_e[u_\ell], \quad \ell \in \{1, 2, \ldots, L\}, \tag{3}$$

$$h_\ell^{'} = \text{LSTM}_f(x_\ell, h_{\ell-1}^{'}), \quad \ell \in \{1, 2, \ldots, L\}, \tag{4}$$

$$h_\ell^{''} = \text{LSTM}_b(x_\ell, h_{\ell+1}^{''}), \quad \ell \in \{L, L-1, \ldots, 1\}, \tag{5}$$

$$h_\ell = [h_\ell^{'}, h_\ell^{''}], \tag{6}$$

where $[\cdot, \cdot]$ denotes concatenation, and $\mathbf{W}_e$ is the word embedding matrix to be learned. LSTM_f and LSTM_b represents the forward and backward LSTM, respectively. The textual memory is then represented as $\mathbf{M}_d = [h_1, \ldots, h_L] \in \mathbb{R}^{n_h \times L}$.

3.2. Multi-step Reasoning Module

The multi-step reasoning framework is implemented via an RNN, where the hidden state s_t represents the current representation of the question, and acts as a query to retrieve visual and textual memories. The initial state s_0 is a self-attended question vector q. Let v_t and d_t denote the attended image representation and dialog-history representation in the t-th step, respectively, a one-step reasoning pathway is then illustrated as $s_t \rightarrow v_t \rightarrow d_t \rightarrow s_{t+1}$, which is performed T times starting from $t = 0$ to achieve T-step reasoning. Details are described below.

Self-attended Question Similar to how we generate the textual memory, a question Q (the subscript i for Q_i is omitted to reduce confusion) is first represented as a matrix $\mathbf{M}_q = [q_1, \ldots, q_K] \in \mathbb{R}^{n_h \times K}$ via a BiLSTM, where K is the maximum length of the question. Based on this, a self-attention mechanism is applied to learn the attention weight of every word in the question, identifying the key words and ruling out irrelevant information. Specifically,

$$\boldsymbol{\alpha} = \text{softmax}(p_\alpha^T \cdot \tanh(\mathbf{W}_q \mathbf{M}_q)), \quad q = \boldsymbol{\alpha}\mathbf{M}_q^T, \tag{7}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^{1 \times K}, p_\alpha \in \mathbb{R}^{n_h \times 1}$, and $\mathbf{W}_q \in \mathbb{R}^{n_h \times n_h}$. The resulting self-attended question vector $q \in \mathbb{R}^{1 \times n_h}$ then serves as the initial hidden state of the RNN, *i.e.*, $s_0 = q$.

The reasoning pathway $s_t \rightarrow v_t \rightarrow d_t \rightarrow s_{t+1}$ includes the following steps: (i) $(s_t, d_{t-1}) \rightarrow v_t$; (ii) $(s_t, v_t) \rightarrow d_t$; and (iii) $(v_t, d_t) \rightarrow s_{t+1}$.

Query and History Attending to Image Given $s_t \in \mathbb{R}^{1 \times n_h}$ and the previous attended dialog history representation $d_{t-1} \in \mathbb{R}^{1 \times n_h}$, we update v_t as follow:

$$\boldsymbol{\beta} = \text{softmax}(p_\beta^T \cdot \tanh(\mathbf{W}_v \mathbf{M}_v + \mathbf{W}_s s_t^T + \mathbf{W}_d d_t^T)),$$
$$v_t = \boldsymbol{\beta} \cdot \mathbf{M}_v^T, \tag{8}$$

where $\boldsymbol{\beta} \in \mathbb{R}^{1 \times M}, p_\beta \in \mathbb{R}^{n_h \times 1}, \mathbf{W}_v \in \mathbb{R}^{n_h \times n_h}, \mathbf{W}_s \in \mathbb{R}^{n_h \times n_h}$ and $\mathbf{W}_d \in \mathbb{R}^{n_h \times n_h}$. The updated v_t, together with s_t, is used to attend to the dialog history.

Query and Image Attending to History Given $s_t \in \mathbb{R}^{1 \times n_h}$ and the attended image representation $v_t \in \mathbb{R}^{1 \times n_h}$, we update d_t as follows:

$$\boldsymbol{\gamma} = \text{softmax}(p_\gamma^T \cdot \tanh(\mathbf{W}_d^{'} \mathbf{M}_d + \mathbf{W}_s^{'} s_t^T + \mathbf{W}_v^{'} v_t^T)),$$
$$d_t = \boldsymbol{\gamma} \cdot \mathbf{M}_d^T, \tag{9}$$

where $\boldsymbol{\gamma} \in \mathbb{R}^{1 \times L}, p_\gamma \in \mathbb{R}^{n_h \times 1}, \mathbf{W}_v^{'} \in \mathbb{R}^{n_h \times n_h}, \mathbf{W}_s^{'} \in \mathbb{R}^{n_h \times n_h}$ and $\mathbf{W}_d^{'} \in \mathbb{R}^{n_h \times n_h}$. The updated d_t is fused with v_t and then used to update the RNN query state.

Multimodal Fusion Given the query vector s_t, we have thus far obtained the updated image representation v_t and the dialog-history representation d_t. Now, we use Multimodal Factorized Bilinear pooling (MFB) [56] to fuse v_t and d_t together. Specifically,

$$z_t = \text{SumPooling}(\mathbf{U}_v v_t^T \circ \mathbf{U}_d d_t^T, k), \tag{10}$$
$$z_t = \text{sign}(z_t)|z_t|^{0.5}, \tag{11}$$
$$z_t = z_t^T / ||z_t||, \tag{12}$$

where $\mathbf{U}_v \in \mathbb{R}^{n_h k \times n_h}, \mathbf{U}_d \in \mathbb{R}^{n_h k \times n_h}$. The function $\text{SumPooling}(x, k)$ in (10) means using a one-dimensional non-overlapped window with the size k to perform sum pooling over x. (11) and (12) represent power normalization and ℓ_2 normalization, respectively. The whole process above is denoted in short as:

$$z_t = \text{MFB}(v_t, d_t) \in \mathbb{R}^{1 \times n_h}. \tag{13}$$

There are also other methods for multimodal fusion, such as MCB [14] and MLB [23]. We use MFB in this paper due to its superior performance in VQA.

Image and History Updating RNN State The initial state s_0 is set to q, which represents the initial understanding of the question. The question representation is then updated based on the current dialogue history and the image,

via an RNN with Gated Recurrent Unit (GRU) [7]:

$$s_{t+1} = \text{GRU}(s_t, z_t). \qquad (14)$$

This process forms a cycle completing one reasoning step. After performing T steps of reasoning, we obtain a set of $\{s_t, v_t, d_t\}_{t=1}^T$. Multimodal fusion is then performed to obtain the final context vector:

$$c_t = [\text{MFB}(s_t, v_t), \text{MFB}(s_t, d_t), \text{MFB}(v_t, d_t)]. \quad (15)$$

Hence, we have obtained a set of T multimodal context vectors $\{c_t\}_{t=1}^T$, each from one reasoning step, which will be used in the decoding module to infer the answer.

3.3. Answer Decoding Module

Discriminative Decoder The context vectors $\{c_t\}_{t=1}^T$ are used to score answers from a pool of candidates $\mathcal{A}$ (the subscript i for $\mathcal{A}_i$ is omitted). Similar to how we obtain the self-attended question vector in Sec. 3.2, a BiLSTM, together with the self-attention mechanism, is used to obtain a vector representation for each candidate $A_j \in \mathcal{A}$, resulting in $a_j \in \mathbb{R}^{1 \times n_h}$, for $j = 1, \ldots, N$. Based on this, a similarity score is computed via a dot product with the context vectors, resulting in a similarity matrix $\mathbf{S} \in \mathbb{R}^{T \times N}$, where $\mathbf{S}[t, j] = c_t a_j^T$.

Reasoning Strategy We explore different reasoning strategies to achieve the final logit vector $o \in \mathbb{R}^N$ used to predict the answer, including (i) prediction from the final step, *i.e.*, $o = \mathbf{S}[T, :]$; (ii) prediction averaged from all the steps, *i.e.*, $o = \frac{1}{T} \sum_{t=1}^T \mathbf{S}[t, :]$; and ($iii$) stochastic dropout. Inspired by [31], in the third strategy, during training, we randomly dropout the results of some steps with a fixed probability, and use the average of the remaining steps for prediction. During test, the average from all the steps is used for prediction.

Training and Evaluation Given $o \in \mathbb{R}^N$, we also consider two training loss functions: (i) a standard softmax-based cross-entropy loss; and (ii) a ranking-based multiclass N-pair loss [32]. For the cross-entropy loss, o is fed into a softmax to compute the probability over candidates. During training, we maximize the log-likelihood of the correct answer candidate. For the N-pair loss, assuming the g-th candidate is the ground-truth, the loss is defined as:

$$\mathcal{L}_{n-pair} = \log \left(1 + \sum_{j=1}^N \exp(o_j - o_g)\right). \qquad (16)$$

The above objective corresponds to a single turn in a dialog. In training, we average over all the dialog turns for training. During evaluation, the answer candidates are simply ranked based on the logit vector o.

Ensemble Ensemble is used to further boost the model performance. Three different ensembling strategies are explored here: (i) average over scores; (ii) average over ranks; and (iii) average over reciprocal ranks.

Specifically, in a dialog session, assuming $o_1, \ldots, o_K$ represents the scores (*i.e.*, the logit vector) obtained from K trained models. In the first ensembling strategy, the average scores $\frac{1}{K} \sum_{k=1}^K o_k$ are used for ranking. In the second one, we first obtain the ranking results of each individual model, then use the average ranks $\frac{1}{K} \sum_{k=1}^K \text{rank}(o_k)$ to re-rank the candidates, where $\text{rank}(o)$ represents the ranking result based on o. In the third strategy, we use the average of the reciprocal ranks of each individual model $\frac{1}{K} \sum_{k=1}^K 1/\text{rank}(o_k)$ for re-ranking.

Generative Decoder Besides the discriminative decoder, following [9], we also consider a generative decoder, where another LSTM is used to decode the context vector into the answer. During training, we maximize the log-likelihood of the ground-truth answers. During evaluation, we use the log-likelihood scores to rank answer candidates.

4. Experiments

In this section, we explain in details our experiments on the VisDial dataset. We compare our ReDAN model with state-of-the-art baselines, and conduct ablation studies to validate the effectiveness of our proposed model.

4.1. Experimental Setup

Dataset We evaluate our proposed approach on the recently released VisDial v1.0 dataset[1]. Specifically, the training and validation splits from v0.9 are combined together to form the new training data in v1.0, which contains dialogs on $123,287$ images from COCO dataset [30]. Each dialog is equipped with 10 turns, resulting in a total of around 1.2M question-answer pairs. An additional $10,064$ COCO-like images are further collected from Flickr, of which $2,064$ images are used as the validation set (val v1.0), and the rest 8K are used as the test set (test-std v1.0), hosted on an evaluation server[2] (the ground-truth answers for this split are not publicly available). Each image in the val v1.0 split is associated with a 10-turn dialog, while a dialog with a flexible number of turns is provided for each image in test-std v1.0. Each question-answer pair in the VisDial dataset is accompanied by a list of 100 answer candidates, and the goal is to find out the correct answer among all the candidates.

Preprocessing We truncate captions/questions/answers that are longer than 40/20/20 words, respectively. And we

---

[1] As suggested in https://visualdialog.org/data, results should be reported on v1.0, instead of v0.9.
[2] https://evalai.cloudcv.org/web/challenges/challenge-page/103/overview

Model	MRR	R@1	R@5	R@10	Mean
MN [9]†	60.29	46.14	77.68	87.57	4.84
HCIAE [32]†	61.96	48.25	78.97	88.43	4.61
CoAtt [51]†	62.77	49.38	78.99	88.49	4.56
ReDAN (T=1)	63.16	49.54	79.85	89.24	4.41
ReDAN (T=2)	63.72	50.31	80.29	89.51	4.30
ReDAN (T=3)	63.59	49.88	80.53	89.63	4.23
ReDAN+GloVe	64.29	50.65	81.29	90.17	4.10
Ensemble	65.54	52.02	82.29	90.77	3.91
Ensemble+GloVe	**65.92**	**52.49**	**82.77**	**91.10**	**3.80**

Table 1: Comparison of the propose model (ReDAN) to state-of-the-art methods on VisDial v1.0 validation set. Higher score is better for MRR and Recall@k, while lower score is better for mean rank. (†) We re-implemented all the models with bottom-up-attention features for fair comparison.

build a vocabulary of words that occur at least 5 times in train v1.0, resulting in $11,319$ words in the vocabulary. For word embeddings, we use the pre-trained GloVe vectors [37] for all the captions, questions and answers, concatenating with the learned word embedding from the BiLSTM encoders to further boost the performance. For image representation, we use the bottom-up-attention features [1] extracted from Faster R-CNN [38] pre-trained on Visual Genome [27]. A set of 36 features is created for each image. Each feature is a 2048-dimentional vector.

Evaluation Similar to [9], we use a set of ranking metrics for evaluation, to measure the performance of retrieving the ground-truth answer from a pool of 100 candidates. These evaluation metrics are: Recall@k for $k = \{1, 5, 10\}$, mean rank, and mean reciprocal rank (MRR). Furthermore, the Normalized Discounted Cumulative Gain (NDCG) score is used for evaluation in the visual dialog challenge 2018. Since this requires dense human annotations, the NDCG score is only available for evaluation on test-std v1.0.

Training details All the 3 BiLSTMs used in the model are single-layer with 512 hidden units. The number of factors used in MFB is set to 5, and we use mini-batches of size 100. The maximum number of epochs to run is set to 20. We do not perform any dataset-specific tuning and regularization other than dropout [45] and early stopping on validation sets. The dropout ratio is set to 0.2. The Adam algorithm [24] with learning rate 4×10^{-4} is utilized for optimization. The learning rate is halved every 10 epochs. By default, we use reasoning strategy (i), training loss (ii), and ensemble method (i) introduced in Sec. 3.3 for experiments, and investigate the other options for ablation study. Code for our experiments will be available at https://www.github.com/author-name/ReDAN.

4.2. Quantitative Results

Baselines We compare our proposed approach with state-of-the-art discriminative models, including Memory Network (MN) [9], History-Conditioned Image Attentive Encoder (HCIAE) [32] and Sequential Co-Attention model (CoAtt) [51]. In their original papers, all these models used VGG-Net [42] for image feature extraction, and reported results on VisDial v0.9. Since bottom-up-attention features provide better performance than VGG-Net in several tasks, for fair comparison, we re-implemented all these models with bottom-up-attention features, and used the same ranking-based N-pair loss for training. We choose the best three models on VisDial v0.9 as the baselines:

- **MN** [9]: (i) mean pooling is performed over the bottom-up-attention features for image representation; (ii) image and question attend to the dialog history.

- **HCIAE** [32]: (i) question attends to dialog history; (ii) then, question and the attended history attend to the image.

- **CoAtt** [51]: (i) question attends to the image; (ii) question and image attend to the history; (iii) image and history attend to the question; (iv) question and history attend to the image again.

Results on VisDial val v1.0 We first present results on val v1.0, shown in Table 1. Using only one reasoning step, our ReDAN model already achieves better performance than CoAtt, which is the best-performing model among the three baselines. Using two reasoning steps further boosts the performance. When 3 steps are used, we observe higher R@5, R@10 and mean rank scores. However, MRR and R@1 are slightly lower. Based on ReDAN with T=2, we further investigate the impact of using pre-trained GloVe embeddings. As shown in the table, incorporating GloVe into the model further boosts the performance, increasing MRR from 63.72 to 64.29.

We also report results on an ensemble of 10 ReDAN models, using reasoning steps ranging from 1 to 5. Each reasoning step is associated with 2 identical models trained with different initializations. Significant improvements are observed via this ensemble of diversified models, boosting MRR from 64.29 to 65.92.

Results on VisDial test-std v1.0 We also evaluate the proposed ReDAN on the blind test-std v1.0 set, by submitting results to the online evaluation server. Table 2 shows the comparison to the state-of-the-art visual dialog models. ReDAN outperforms the published state-of-the-art method, CorefNMN [26], by a significant margin, lifting NDCG from 54.70 to 57.63, and MRR from 61.50 to 64.75. Recent state-of-the-art has reached up to 57.88 (NDCG) from

Model	NDCG	MRR	R@1	R@5	R@10	Mean
ReDAN	57.63	**64.75**	**51.10**	**81.73**	**90.90**	**3.89**
Unpublished						
DL-61	**57.88**	63.42	49.30	80.77	90.68	3.97
USTC-YTH	56.47	61.44	47.65	78.13	87.88	4.65
MS ConvAI	55.35	63.27	49.53	80.40	89.60	4.15
Published						
CorefNMN*	54.70	61.50	47.55	78.10	88.80	4.40
LF-Att	49.76	57.07	42.08	74.83	85.05	5.41
MN-Att	49.58	56.90	42.43	74.00	84.35	5.59
MN	47.50	55.49	40.98	72.30	83.30	5.92
HRE	45.46	54.16	39.93	70.45	81.50	6.41
LF	45.31	55.42	40.95	72.45	82.83	5.95

Table 2: Comparison of ReDAN to state-of-the-art visual dialog models on the blind test-std v1.0 set as reported by the test server. (*) results are from [26].

Model	MRR	R@1	R@5	R@10	Mean
MN-G [9]†	47.99	38.18	57.54	64.32	18.60
HCIAE-G [32]†	49.10	39.35	58.49	64.70	18.46
CoAtt-G [51]†	49.25	39.66	58.83	65.38	18.15
ReDAN-G (T=1)	49.24	39.63	58.89	65.27	18.21
ReDAN-G (T=2)	**49.69**	**40.19**	**59.35**	**66.06**	**17.92**
ReDAN-G (T=3)	49.54	39.88	59.10	65.84	18.00

Table 3: Comparison of ReDAN with a generative decoder to state-of-the-art generative models on VisDial val v1.0. (†) All the models are re-implemented with bottom-up-attention features for fair comparison.

team DL-61; however, the approach and its details are unpublished at this time. Our model outperforms DL-61 in all the metrics except NDCG.

Results using generative decoders The results presented above focus on discriminative decoders. In addition, we also evaluate our model with a generative decoder. Results are summarized in Table 3. Similar to Table 1, ReDAN with T=2 achieves the best performance.

4.3. Ablation Study

Impact of image features Since VGG-Net [42] was widely used in previous visual dialog models, we first compare the performance of using VGG-Net and the bottom-up-attention features based on HCIAE [32]. For VGG-Net, we take the output of the last pooling layer (size of $512 \times 7 \times 7$) as image features. As shown in Table 4, bottom-up-attention features achieve significant better performance than VGG-Net.

Impact of reasoning strategies and steps We also conduct an ablation study to investigate the impact of different reasoning strategies and steps. Results are summarized

Model	MRR	R@1	R@5	R@10	Mean
HCIAE (VGG)	58.79	44.65	75.67	85.87	5.34
HCIAE (Bottom-Up)	61.96	48.25	78.97	88.43	4.61

Table 4: Ablation study on different image features.
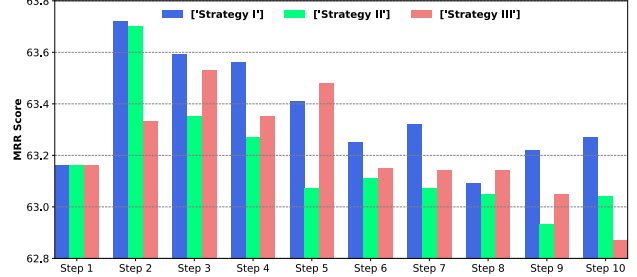
Figure 3: Ablation study on different reasoning strategies and steps.

Training objective	Step 2	Step 3	Step 4	Step 5
Cross-entropy loss	63.71	63.58	63.56	63.46
N-pair loss	63.72	63.59	63.67	63.48

Table 5: Performance (MRR) of different training objectives.

Ensemble	MRR	R@1	R@5	R@10	Mean
Strategy I	65.92	52.49	82.77	91.10	3.80
Strategy II	65.90	52.54	82.63	91.02	3.81
Strategy III	65.99	52.61	82.68	91.03	3.79

Table 6: Ablation study on using different ensemble strategies.

in Figure 3. Across reasoning steps 1 to 10, strategy (i) achieves the best performance in most cases, while strategy (ii) performs the worst. Results also show that the performance of our model is robust to the number of reasoning steps. For best performance, we recommend setting reasoning steps to 2. However, we also use steps in the range of $(2, 5]$ for model ensemble.

Impact of training objective Previous work shows that using a N-pair loss as the objective function results in better performance than a softmax-based cross-entropy loss [32]. However, as shown in Table 5, our experiments indicate that similar performance can be achieved by both loss functions.

Impact of ensemble strategies We also investigate different ensemble methods. Table 6 shows that different strategies result in similar performance, with strategy I performs the best on R@5 and R@10, while strategy III performs the best on other metrics.
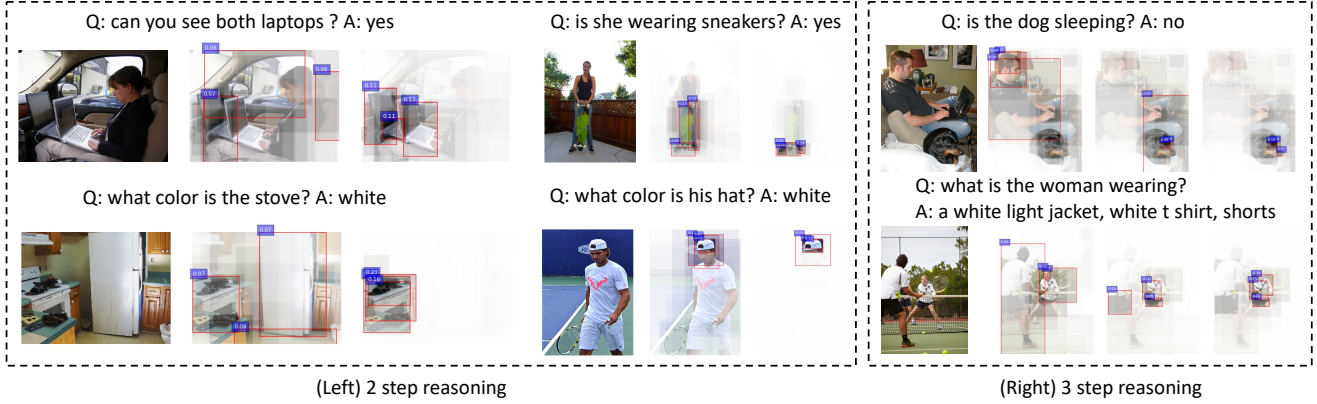
Figure 4: Visualization of learned attention maps in multiple reasoning steps.

Figure 5: Example dialogs generated by our model and by the CoAtt baseline. Our model provides answers that are more accurate than CoAtt (*green* denotes correct answers, and *red* denotes wrong answers). Results from our model are also more natural and comprehensive (highlighted in *orange*).

4.4. Qualitative Analysis

Besides the example illustrated in Figure 1b, we provide another six examples in Figure 4 to visualize the learned attention maps. The accompanying dialog histories are omitted for simplicity. Typically, the attention maps become sharper and more focused during the reasoning process. Through multi-step reasoning, the model learns to narrow down to the image regions corresponding to the key objects ("*laptops*", "*stove*", "*sneakers*", "*hat*", "*dog's eyes*" and "*woman's clothes*") to achieve more clarity on the answer. For instance, in the top-right example, ReDAN focuses on the wrong region ("*man*") in the 1st step, but gradually shifts its focus to the correct regions ("*dog's eyes*").

Figure 5 also provides two dialog examples, as a comparison of the end-to-end results between ReDAN and the state-of-the-art CoAtt model. By analyzing a set of randomly selected examples, we have two observations. First,

ReDAN generally provides more accurate answers. Interestingly, even in cases where both models retrieve the correct answers, our model tends to provide longer and more natural human-like answers, while the answers from CoAtt are often single words. More examples are provided in the Supplementary Material.

5. Conclusion

We have presented Recurrent Dual Attention Network (ReDAN), a new framework to perform multi-step reasoning for visual dialog. ReDAN attends to the image and dialog history via a recurrently-updated query vector. This multi-step process provides a fine-grained understanding of the multimodal context, thus boosting question answering performance. Experiments conducted on the VisDial dataset validate the effectiveness of the proposed approach.

References

[1] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018. 1, 3, 6

[2] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Neural module networks. In *CVPR*, 2016. 2

[3] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *ICCV*, 2015. 1

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

[5] P. Chattopadhyay, D. Yadav, V. Prabhu, A. Chandrasekaran, A. Das, S. Lee, D. Batra, and D. Parikh. Evaluating visual conversational agents via cooperative human-ai games. In *HCOMP*, 2017. 2

[6] J. Chen, Y. Shen, J. Gao, J. Liu, and X. Liu. Language-based image editing with recurrent attentive models. In *CVPR*, 2018. 2

[7] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. 5

[8] Y. Cui, Z. Chen, S. Wei, S. Wang, T. Liu, and G. Hu. Attention-over-attention neural networks for reading comprehension. In *ACL*, 2017. 2

[9] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh, and D. Batra. Visual dialog. In *CVPR*, 2017. 1, 2, 3, 5, 6, 7

[10] A. Das, S. Kottur, J. M. Moura, S. Lee, and D. Batra. Learning cooperative visual dialog agents with deep reinforcement learning. In *ICCV*, 2017. 1, 2

[11] H. De Vries, F. Strub, S. Chandar, O. Pietquin, H. Larochelle, and A. C. Courville. Guesswhat?! visual object discovery through multi-modal dialogue. In *CVPR*, 2017. 1, 2

[12] B. Dhingra, H. Liu, Z. Yang, W. W. Cohen, and R. Salakhutdinov. Gated-attention readers for text comprehension. In *ACL*, 2017. 2

[13] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, et al. From captions to visual concepts and back. In *CVPR*, 2015. 1

[14] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *EMNLP*, 2016. 1, 4

[15] J. Gao, M. Galley, and L. Li. Neural approaches to conversational ai. *arXiv preprint arXiv:1809.08267*, 2018. 2

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014. 2

[17] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra. Draw: A recurrent neural network for image generation. In *ICML*, 2015. 2

[18] X. Guo, H. Wu, Y. Cheng, S. Rennie, and R. S. Feris. Dialog-based interactive image retrieval. In *NIPS*, 2018. 2

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[20] F. Hill, A. Bordes, S. Chopra, and J. Weston. The goldilocks principle: Reading children's books with explicit memory representations. In *ICLR*, 2016. 2

[21] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 1997. 4

[22] U. Jain, S. Lazebnik, and A. G. Schwing. Two can play this game: visual dialog with discriminative question generation and answering. In *CVPR*, 2018. 2

[23] J.-H. Kim, K.-W. On, W. Lim, J. Kim, J.-W. Ha, and B.-T. Zhang. Hadamard product for low-rank bilinear pooling. In *ICLR*, 2017. 4

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

[25] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 2

[26] S. Kottur, J. M. Moura, D. Parikh, D. Batra, and M. Rohrbach. Visual coreference resolution in visual dialog using neural module networks. In *ECCV*, 2018. 2, 6, 7

[27] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2017. 6

[28] S.-W. Lee, Y.-J. Heo, and B.-T. Zhang. Answerer in questioner's mind for goal-oriented visual dialogue. In *NIPS*, 2018. 2

[29] J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter, and D. Jurafsky. Adversarial learning for neural dialogue generation. In *EMNLP*, 2017. 2

[30] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 5

[31] X. Liu, Y. Shen, K. Duh, and J. Gao. Stochastic answer networks for machine reading comprehension. In *ACL*, 2018. 2, 5

[32] J. Lu, A. Kannan, J. Yang, D. Parikh, and D. Batra. Best of both worlds: Transferring knowledge from discriminative learning to a generative visual dialog model. In *NIPS*, 2017. 1, 2, 5, 6, 7

[33] D. Massiceti, N. Siddharth, P. K. Dokania, and P. H. Torr. Flipdial: A generative model for two-way visual dialogue. In *CVPR*, 2018. 2

[34] V. Mnih, N. Heess, A. Graves, et al. Recurrent models of visual attention. In *NIPS*, 2014. 2

[35] N. Mostafazadeh, C. Brockett, B. Dolan, M. Galley, J. Gao, G. P. Spithourakis, and L. Vanderwende. Image-grounded conversations: Multimodal context for natural question and response generation. *arXiv preprint arXiv:1701.08251*, 2017. 2

[36] H. Nam, J.-W. Ha, and J. Kim. Dual attention networks for multimodal reasoning and matching. In *CVPR*, 2017. 2

[37] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014. 6

[38] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 3, 6

[39] P. H. Seo, A. Lehrmann, B. Han, and L. Sigal. Visual reference resolution using attention memory for visual dialog. In *NIPS*, 2017. 2

[40] R. Shekhar, T. Baumgartner, A. Venkatesh, E. Bruni, R. Bernardi, and R. Fernandez. Ask no more: Deciding when to guess in referential visual dialogue. In *COLING*, 2018. 2

[41] Y. Shen, P.-S. Huang, J. Gao, and W. Chen. Reasonet: Learning to stop reading in machine comprehension. In *KDD*, 2017. 2, 3

[42] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3, 6, 7

[43] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. In *NIPS*, 2015. 2

[44] A. Sordoni, P. Bachman, A. Trischler, and Y. Bengio. Iterative alternating neural attention for machine reading. *arXiv preprint arXiv:1606.02245*, 2016. 2

[45] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 2014. 6

[46] F. Strub, H. De Vries, J. Mary, B. Piot, A. Courville, and O. Pietquin. End-to-end optimization of goal-driven and visually grounded dialogue systems. In *IJCAI*, 2017. 2

[47] F. Strub, M. Seurin, E. Perez, H. de Vries, P. Preux, A. Courville, O. Pietquin, et al. Visual reasoning with multihop feature modulation. In *ECCV*, 2018. 2

[48] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014. 2

[49] D. Teney, P. Anderson, X. He, and A. van den Hengel. Tips and tricks for visual question answering: Learnings from the 2017 challenge. In *CVPR*, 2018. 3

[50] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015. 1

[51] Q. Wu, P. Wang, C. Shen, I. Reid, and A. van den Hengel. Are you talking to me? reasoned visual dialog generation through adversarial learning. In *CVPR*, 2018. 2, 6, 7

[52] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 1

[53] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In *CVPR*, 2016. 2, 3

[54] A. W. Yu, H. Lee, and Q. V. Le. Learning to skim text. *arXiv preprint arXiv:1704.06877*, 2017. 2

[55] L. Yu, W. Zhang, J. Wang, and Y. Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, 2017. 2

[56] Z. Yu, J. Yu, J. Fan, and D. Tao. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In *ICCV*, 2017. 4

[57] J. Zhang, Q. Wu, C. Shen, J. Zhang, J. Lu, and A. Van Den Hengel. Goal-oriented visual question generation via intermediate rewards. In *ECCV*, 2018. 2