Visual Coreference Resolution in Visual Dialog using Neural Module Networks

Satwik Kottur[1,2*], José M. F. Moura[2], Devi Parikh[1,3], Dhruv Batra[1,3], and Marcus Rohrbach[1]

[1] Facebook AI Research, Menlo Park, USA
[2] Carnegie Mellon University, Pittsburgh, USA
[3] Georgia Institute of Technology, Atlanta, USA

Abstract. Visual dialog entails answering a series of questions grounded in an image, using dialog history as context. In addition to the challenges found in visual question answering (VQA), which can be seen as one-round dialog, visual dialog encompasses several more. We focus on one such problem called *visual coreference resolution* that involves determining which words, typically noun phrases and pronouns, *co-refer* to the same entity/object instance in an image. This is crucial, especially for pronouns (e.g., '*it*'), as the dialog agent must first link it to a previous coreference (e.g., *'boat'*), and only then can rely on the visual grounding of the coreference *'boat'* to reason about the pronoun '*it*'. Prior work (in visual dialog) models visual coreference resolution either (a) implicitly via a memory network over history, or (b) at a coarse level for the entire question; and not explicitly at a phrase level of granularity. In this work, we propose a neural module network architecture for visual dialog by introducing two novel modules—`Refer` and `Exclude`—that perform explicit, grounded, coreference resolution at a finer word level. We demonstrate the effectiveness of our model on MNIST Dialog, a visually simple yet coreference-wise complex dataset, by achieving near perfect accuracy, and on VisDial, a large and challenging visual dialog dataset on real images, where our model outperforms other approaches, and is more interpretable, grounded, and consistent qualitatively.

1 Introduction

The task of Visual Dialog [11, 40] involves building agents that 'see' (i.e. understand an image) and 'talk' (i.e. communicate this understanding in a dialog). Specifically, it requires an agent to answer a sequence of questions about an image, requiring it to reason about both the image and the past dialog history. For instance, in Fig. 1, to answer *'What color is it?'*, the agent needs to reason about the history to know what *'it'* refers to and the image to find out the color. This generalization of visual question answering (VQA) [6] to dialog takes a step closer to real-world applications (aiding visually impaired users, intelligent home

[*] Work partially done as an intern at Facebook AI Research

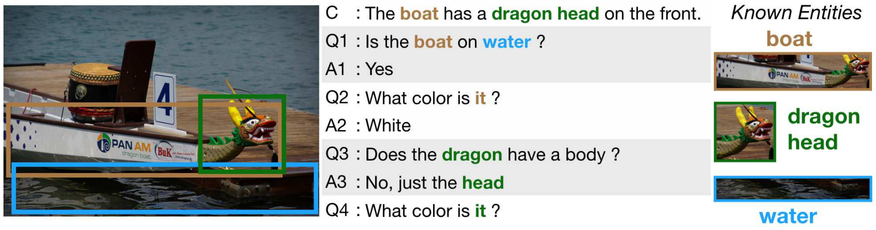

Fig. 1: Our model begins by grounding entities in the caption (C), *boat* (brown) and *dragon head* (green), and stores them in a pool for future coreference resolution in the dialog (right). When asked *'Q1: Is the boat on water?'*, it identifies that the *boat* (known entity) and *water* (unknown entity) are crucial to answer the question. It then grounds the novel entity *water* in the image (blue), but resolves *boat* by referring back to the pool and reusing the available grounding from the caption, before proceeding with further reasoning. Thus, our model explicitly resolves coreferences in visual dialog.

assistants, natural language interfaces for robots) but simultaneously introduces new modeling challenges at the intersection of vision and language. The particular challenge we focus on in this paper is that of *visual coreference resolution* in visual dialog. Specifically, we introduce a new model that performs explicit visual coreference resolution and interpretable entity tracking in visual dialog.

It has long been understood [16, 44, 31, 46] that humans use *coreferences*, different phrases and short-hands such as pronouns, to refer to the same entity or referent in a single text. In the context of visually grounded dialog, we are interested in referents which are in the image, e.g. an object or person. All phrases in the dialog which refer to the same entity or referent in the image are called visual coreferences. Such coreferences can be noun phrases such as *'a dragon head'*, *'the head'*, or pronouns such as *'it'* (Fig. 1). Especially when trying to answer a question that contains an anaphora, for instance the pronoun *'it'*, which refers to its full form (the antecedent) *'a dragon head'*, it is necessary to *resolve* the coreference on the language side and ground it to the underlying visual referent. More specifically, to answer the question *'What color is it?'* in Fig. 1, the model must correctly identify which object *'it'* refers to, in the given context. Notice that a word or phrase can refer to different entities in different contexts, as is the case with *'it'* in this example. Our approach to explicitly resolve visual coreferences is inspired from the functionality of variables or memory in a computer program. In the same spirit as how one can refer back to the contents of variables at a later time in a program without explicitly re-computing them, we propose a model which can refer back to entities from previous rounds of dialog and reuse the associated information; and in this way resolve coreferences.

Prior work on VQA [28, 13, 2] has (understandably) largely ignored the problem of visual coreference resolution since individual questions asked in isolation rarely contain coreferences. In fact, recent empirical studies [1, 20, 47, 15] suggest that today's vision and language models seem to be exploiting dataset-level statistics and perform poorly at grounding entities into the correct pixels. In contrast, our work aims to explicitly reason over past dialog interactions by referring back to previous references. This allows for increased interpretability of the model. As the dialog progresses (Fig. 1), we can inspect the pool of entities known to the model, and also visualize which entity a particular phrase in the question has been resolved to. Moreover, our explicit entity tracking model has benefits even in cases that may not strictly speaking require coreference resolution. For instance, by explicitly referring *'dragon'* in Q3 (Fig. 1) back to a known entity, the model is consistent with itself and (correctly) grounds the phrase in the image. We believe such consistency in model outputs is a strongly desirable property as we move towards human-machine interaction in dialog systems.

Our main technical contribution is a neural module network architecture for visual dialog. Specifically, we propose two novel modules—`Refer` and `Exclude`—that perform explicit, grounded, coreference resolution in visual dialog. In addition, we propose a novel way to handle captions using neural module networks at a word-level granularity finer than a traditional sentence-level encoding. We show quantitative benefits of these modules on a reasoning-wise complicated but visually simple MNIST dialog dataset [37], where achieve near perfect accuracy. On the visually challenging VisDial dataset [11], our model not only outperforms other approaches but also is more interpretable by construction and enables word-level coreference resolution. Furthermore, we qualitatively show that our model is (a) more interpretable (a user can inspect which entities were detected and tracked as the dialog progresses, and which ones were referred to for answering a specific question), (b) more grounded (where the model looked to answer a question in the dialog), (c) more consistent (same entities are considered across rounds of dialog).

2 Related Work

We discuss: (a) existing approaches to visual dialog, (b) related tasks such as visual grounding and coreference resolution, and (c) neural module networks.

Visual Dialog. Though the origins of visual dialog can be traced back to [43, 14], it was largely formalized by [11, 40] who collected human annotated datasets for the same. Specifically, [11] paired annotators to collect free-form natural-language questions and answers, where the questioner was instructed to ask questions to help them imagine the hidden scene (image) better. On the other hand, dialogs from [40] are more goal driven and contain yes/no questions directed towards identifying a secret object in the image. The respective follow up works used reinforcement learning techniques to solve this problem [12, 39]. Other approaches to visual dialog include transferring knowledge from a discriminatively trained model to a generative dialog model [27], using attention networks to solve visual coreferences [37], and more recently, a probabilistic treatment of dialogs using conditional variational autoencoders [30]. Amongst these, [37] is the closest to this work, while [27, 30] are complementary. To solve visual coreferences, [37] relies on global visual attentions used to answer previous questions. They store these attention maps in a memory against keys based on textual representations of the entire question and answer, along with the history. In contrast, operating at a finer word-level granularity within each question, our model can resolve different phrases of a question, and ground them to different parts of the image, a core component in correctly understanding and grounding coreferences. E.g., *'A man and woman in a car. Q: Is he or she driving?'*, which requires resolving *'he'* and *'she'* individually to answer the question.

Grounding language in images and video. Most works in this area focus on the specific task of localizing a textual referential expression in the image [19, 22, 29, 32, 35, 41, 46] or video [34, 24, 45, 5]. Similar to these works, one component of our model aims to localize words and phrases in the image. However, the key difference is that if the phrase being grounded is an anaphora (e.g., *'it'*, *'he'*, *'she'*, etc.), our model first resolves it explicitly to a known entity, and then grounds it by borrowing the referent's visual grounding.

Coreference resolution. The linguistic community defines coreference resolution as the task of clustering phrases, such as noun phrases and pronouns, which refer to the same entity in the world (see, for example, [8]). The task of visual coreference resolution links the coreferences to an entity in the visual data. For example, [33] links character mentions in TV show descriptions with their occurrence in the video, while [22] links text phrases to objects in a 3D scene. Different from these works, we predict a program for a given natural language question about an image, which then tries to resolve any existing coreferences, to then answer the question. An orthogonal direction is to generate language while jointly grounding and resolving coreferences – e.g., [36] explore this for movie descriptions. While out of scope for this work, it is an interesting direction for future work in visual dialog, especially when generating questions.

Neural Module Networks [4] are an elegant class of models where an instance-specific architecture is composed from neural 'modules' (or building blocks) that are shared across instances. The high-level idea is inspired by 'options' or subtasks in hierarchical RL. They have been shown to be successful for visual question answering in real images and linguistic databases [3] and for more complex reasoning tasks in synthetic datasets [21, 18]. For this, [21, 18] learn program prediction and module parameters jointly, end-to-end. Within this context, our work generalizes the formulation in [18] from VQA to visual dialog by introducing a novel module to perform explicit visual coreference resolution.

3 Approach

Recall that visual dialog [11] involves answering a question Q_t at the current round t, given an image I, and the dialog history (including the image caption) $H = (\underbrace{C}_{H_0}, \underbrace{(Q_1, A_1)}_{H_1}, \cdots, \underbrace{(Q_{t-1}, A_{t-1})}_{H_{t-1}})$, by ranking a list of 100 candidate answers $\mathcal{A}_t = \{A_t^{(1)}, \cdots, A_t^{(100)}\}$. As a key component for building better visual dialog agents, our model explicitly resolves visual coreferences in the current question, if any.

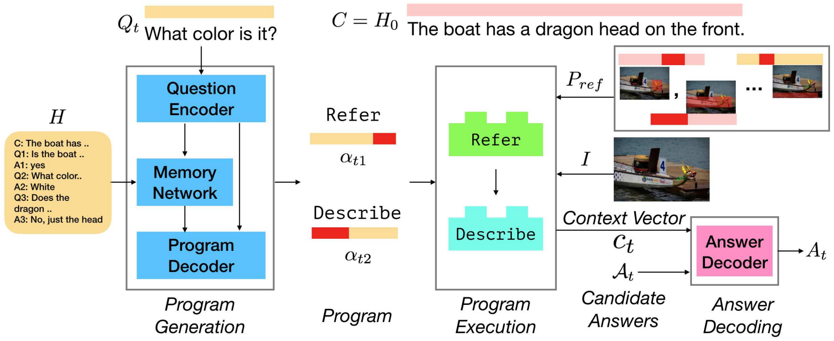

Fig. 2: Overview of our model architecture. The question Q_t (orange bar) is encoded along with the history H through a memory augmented question encoder, using which a program (`Refer Describe`) is decoded. For each module in the program, an attention α_{ti} over Q_t is also predicted, used to compute the text feature x_{txt}. For Q_t, attention is over *'it'* for `Refer` and *'What color'* for `Describe`, respectively (orange bars with red attention). `Refer` module uses the coreference pool P_{ref}, a dictionary of all previously seen entities with their visual groundings, resolves *'it'*, and borrows the referent's visual grounding (*boat* in this case). Finally, `Describe` extracts the 'color' to produce c_t used by a final decoder to pick the answer A_t from the candidate pool $\mathcal{A}_t$.

Towards this end, our model first identifies relevant words or phrases in the current question that refer to entities in the image (typically objects and attributes). The model also predicts whether each of these has been mentioned in the dialog so far. Next, if these are novel entities (unseen in the dialog history), they are localized in the image before proceeding, and for seen entities, the model predicts the (first) relevant coreference in the conversation history, and retrieves its corresponding visual grounding. Therefore, as rounds of dialog progress, the model collects unique entities and their corresponding visual groundings, and uses this *reference pool* to resolve any coreferences in subsequent questions.

Our model has three broad components: (a) *Program Generation* (Sec. 3.3), where a reasoning pathway, as dictated by a *program*, is predicted for the current question Q_t, (b) *Program Execution* (Sec. 3.4), where the predicted program is executed by dynamically connecting neural modules [3, 4, 18] to produce a *context* vector summarizing the semantic information required to answer Q_t from the context (I, H), and lastly, (c) *Answer Decoding* (Sec. 3.4), where the context vector c_t is used to obtain the final answer $\hat{A}_t$. We begin with a general characterization of neural modules used for VQA in Sec. 3.1 and then discuss our novel modules for coreference resolution (Sec. 3.2) with details of the reference pool. After describing the inner working of the modules, we explain each of the above three components of our model.

3.1 Neural Modules for Visual Question Answering

The main technical foundation of our model is the neural module network (NMN) [4]. In this section, we briefly recap NMNs and more specifically, the attentional modules from [18]. In the next section, we discuss novel modules we propose to handle additional challenges in visual dialog.

For a module m, let x_{vis} and x_{txt} be the input image and text embeddings, respectively. In particular, the image embeddings x_{vis} are spatial activation maps of the image I from a convolutional neural network. The text embedding x_{txt} is computed as a weighted sum of embeddings of words in the question Q_t using the soft attention weights α predicted by a program generator for module m (more details in Sec. 3.3). Further, let $\{a_i\}$ be the set of n_m single-channel spatial maps corresponding to the spatial image embeddings, where n_m is the number of attention inputs to m. Denoting the module parameters with θ_m, a neural module m is essentially a parametric function $y = f_m(x_{vis}, x_{txt}, \{a_i\}_{i=1}^{n_m}; \theta_m)$. The output from the module y can either be a spatial image attention map (denoted by a) or a context vector (denoted by c), depending on the module. The output spatial attention map a feeds into next level modules while a context vector c is used to obtain the final answer A_t. The upper part of Tab. 1 lists modules we adopt from prior work, with their functional forms. We shortly summarize their behavior. A `Find` module localizes objects or attributes by producing an attention over the image. The `Relocate` module takes in an input image attention and performs necessary spatial relocations to handle relationships like *'next to'*, *'in front of'*, *'beside'*, etc. Intersection or union of attention maps can be obtained using `And` and `Or`, respectively. Finally, `Describe`, `Exist`, and `Count` input an attention map to produce the context vector by describing an attribute, checking for existence, or counting, respectively, in the given input attention map. As noted in [18], these modules are designed and named for a potential 'atomic' functionality. However, we do not enforce this explicitly and let the modules discover their expected behavior by training in an end-to-end manner.

3.2 Neural Modules for Coreference Resolution

We now introduce novel components and modules to handle visual dialog.

Reference Pool (P_{ref}). The role of the reference pool is to keep track of entities seen so far in the dialog. Thus, we design P_{ref} to be a dictionary of key-value pairs (x_{txt}, a) for all the `Find` modules instantiated while answering previous questions $(Q_i)_{i=1}^{t-1}$. Recall that `Find` localizes objects/attributes specified by x_{txt}, and thus by storing each output attention map y, we now have access to all the entities mentioned so far in the dialog with their corresponding visual groundings. Interestingly, even though x_{txt} and y are intermediate outputs from our model, both are easily interpretable, making our reference pool a *semantic dictionary.* To the best of our knowledge, our model is the first to attempt explicit, interpretable coreference resolution in visual dialog. While [37] maintains a dictionary similar to P_{ref}, they do not consider word/entity level coreferences nor do their keys lend similar interpretability as ours. With $P_{ref} = \{(x_p^{(i)}, a_p^{(i)})\}_i$ as input to `Refer`, we can now resolve references in Q_t.

`Refer` **Module.** This novel module is responsible for resolving references in the question Q_t and ground them in the conversation history H. To enable grounding

Name	Inputs	Output	Function
Neural Modules for VQA [18]			
`Find`	x_{vis}, x_{txt}	attention	$y = \text{conv}_2(\text{conv}_1(x_{vis} \odot W x_{txt}))$
`Relocate`	a, x_{vis}, x_{txt}	attention	$\tilde{y} = W_1 \text{sum}(a \odot x_{vis})$ $y = \text{conv}_2(\text{conv}_1(x_{vis}) \odot \tilde{y} \odot W_2 x_{txt})$
`And`	a_1, a_2	attention	$y = \min\{a_1, a_2\}$
`Or`	a_1, a_2	attention	$y = \max\{a_1, a_2\}$
`Exist`	a, x_{vis}, x_{txt}	context	$y = W^T \text{vec}(a)$
`Describe`	a, x_{vis}, x_{txt}	context	$y = W_1^T (W_2 \text{sum}(a \odot x_{vis}) \odot W_3 x_{txt})$
`Count`	a, x_{vis}, x_{txt}	context	$y = W_1^T ([\text{vec}(a), \max\{a\}, \min\{a\}])$
Neural Modules for Coreference resolution (Ours)			
`Not`	a	attention	$y = \text{norm}_{L_1}(1 - a)$
`Refer`	x_{txt}, P_{ref}	attention	(see text for details, (3))
`Exclude`	a, x_{vis}, x_{txt}	attention	$y =$ `And`[`Find`$[x_{vis}, x_{txt}]$, `Not`$[a]$]

Table 1: Neural modules used in our work for visual dialog, along with their inputs, outputs, and function formulations. The upper portion contains modules from prior work used for visual question answering, while the bottom portion lists our novel modules designed to handle additional challenges in visual dialog.

in dialog history, we generalize the above formulation to give the module access to a pool of references P_{ref} of previously identified entities. Specifically, `Refer` only takes the text embedding x_{txt} and the reference pool P_{ref} as inputs, and resolves the entity represented by x_{txt} in the form of a soft attention α over Q_t. in this section after introducing P_{ref}. For the example shown in Fig. 2, α for `Refer` attends to *'it'*, indicating the phrase it is trying to resolve.

At a high level, `Refer` treats x_{txt} as a 'query' and retrieves the most likely match from P_{ref} as measured by some similarity with respect to keys $\{x_p^{(i)}\}_i$ in P_{ref}. The associated image attention map of the best match is used as the visual grounding for the phrase that needed resolution (i.e. *'it'*). More concretely, we first learn a *scoring network* which when given a query x_{txt} and a possible candidate $x_p^{(i)}$, returns a scalar value s_i indicating how likely these text features refer to the same entity (1). To enable `Refer` to consider the sequential nature of dialog when assessing a potential candidate, we additionally provide $\Delta_i t$, a measure of the 'distance' of a candidate $x_p^{(i)}$ from x_{txt} in the dialog history, as input to the scoring network. $\Delta_i t$ is formulated as the absolute difference between the round of x_{txt} (current round t) and the round when $x_p^{(i)}$ was first mentioned. Collecting these scores from all the candidates, we apply a softmax function to compute contributions $\tilde{s}_i$ from each entity in the pool (2). Finally, we weigh the corresponding attention maps via these contributions to obtain the visual grounding a_{out} for x_{txt} (3).

$$s_i = \mathrm{MLP}([x_{txt}, x_p^{(i)}, \mathit{\Delta}_i t]) \quad (1)$$

$$\tilde{s}_i = \mathrm{Softmax}(s_i) \quad (2)$$

$$a_{out} = \sum_{i=1}^{|P_{ref}|} \tilde{s}_i a_p^{(i)} \quad (3)$$

`Not` **Module.** Designed to focus on regions of the image **not** attended by the input attention map a, it outputs $y = \mathrm{norm}_{L_1}(1-a)$, where $\mathrm{norm}_{L_1}(.)$ normalizes the entries to sum to one. This module is used in `Exclude`, described next.

`Exclude` **Module.** To handle questions like *'What other red things are present?'*, which seek other objects/attributes in the image than those specified by an input attention map a, we introduce yet another novel module – `Exclude`. It is constructed using `Find`, `Not`, and `And` modules as $y = \texttt{And}[\texttt{Find}[x_{txt}, x_{vis}], \texttt{Not}[a]]$, where x_{txt} is the text feature input to the `Exclude` module, for example, *'red things'*. More explicitly, `Find` first localizes all objects instances/attributes in the image. Next, we focus on regions of the image other than those specified by a using $\texttt{Not}[a]$. Finally, the above two outputs are combined via `And` to obtain the output y of the `Exclude` module.

3.3 Program Generation

A *program* specifies the network layout for the neural modules for a given question Q_t. Following [18], it is serialized through the reverse polish notation (RPN) [9]. This serialization helps us convert a hard, structured prediction problem into a more tractable sequence prediction problem. In other words, we need a program predictor to output a series of module tokens in order, such that a valid layout can be retrieved from it. There are two primary design considerations for our predictor. First, in addition to the program, our predictor must also output a soft attention α_{ti}, over the question Q_t, for every module m_i in the program. This attention is responsible for the *correct* module instantiation in the current context. For example, to answer the question *'What color is the cat sitting next to the dog?'*, a `Find` module instance attending to *'cat'* qualitatively serves a different purpose than one attending to *'dog'*. This is implemented by using the attention over Q_t to compute the text embedding x_{txt} that is directly fed as an input to the module during execution. Second, to decide whether an entity in Q_t has been seen before in the conversation, it must be able to 'peek' into the history H. Note that this is unique to our current problem and does not exist in [18]. To this effect, we propose a novel augmentation of attentional recurrent neural networks [7] with memory [42] to address both the requirements (Fig. 2).

The program generation proceeds as follows. First, each of the words in Q_t are embedded to give $\{w_{ti}\}_{i=1}^{T}$, where T denotes the number of tokens in Q_t. We then use a *question encoder*, a multi-layer LSTM, to process w_{ti}'s, resulting in a sequence of hidden states $\{\hat{w}_{ti}\}_{i=1}^{T}$ (4). Notice that the last hidden state h_T is the question encoding, which we denote with q_t. Next, each piece of history $(H_i)_{i=0}^{t-1}$ is processed in a similar way by a *history encoder*, which is a multi-layer LSTM akin to the question encoder. This produces encodings $(h_i)_{i=0}^{t-1}$ (5) that serve as memory units to help the program predictor 'peek' into the conversation history. Using the question encoding q_t, we attend over the history encodings $(h_i)_{i=0}^{t-1}$, and obtain the history vector $\hat{h}_t$ (6). The history-agnostic question encoding q_t is then fused with the history vector $\hat{h}_t$ via a fully connected layer to give a history-aware question encoding $\hat{q}_t$ (7), which is fed into the *program decoder*.

Question Encoder

$$\{\hat{w}_{ti}\} = \text{LSTM}(\{w_{ti}\}) \tag{4}$$

$$q_t = \hat{w}_{tT}$$

History Memory

$$\hat{h}_i = \text{LSTM}(h_i) \tag{5}$$

$$\beta_{ti} = \text{Softmax}(q_t^T \hat{h}_i)$$

$$\hat{h}_t = \sum_{i=0}^{t-1} \beta_{ti} \hat{h}_i \tag{6}$$

$$\hat{q}_t = \text{MLP}([q_t, \hat{h}_t]) \tag{7}$$

Program Decoder

$$\tilde{u}_{ti}^{(j)} = \text{Linear}([\hat{w}_{tj}, d_{ti}])$$

$$u_{ti}^{(j)} = v^T \tanh(\tilde{u}_{ti}^{(j)})$$

$$\alpha_{ti}^{(j)} = \text{Softmax}(u_{ti}^{(j)})$$

$$e_{ti} = \sum_{j=1}^{T} \alpha_{ti}^{(j)} \hat{w}_{tj} \tag{8}$$

$$\tilde{e}_{ti} = \text{MLP}([e_{ti}, d_{ti}]) \tag{9}$$

$$p(m_i | \{m_k\}_{k=1}^{i-1}, Q_t, H) = \text{Softmax}(\tilde{e}_{ti}) \tag{10}$$

The decoder is another multi-layer LSTM network (with hidden states $\{d_{ti}\}$) which, at every time step i, produces a soft attention map α_{ti} over the input sequence (Q_t) [7]. This soft attention map for each module is used to compute the corresponding text embedding, $x_{txt} = \sum_j \alpha_{ti}^{(j)} w_{tj}$. Finally, to predict a module token m_i at time step i, a weighted sum of encoder hidden states e_{ti} (8) and the history-aware question vector $\hat{q}_t$ are combined via another fully-connected layer (9), followed by a softmax to give a distribution $P(m_i|\{m_k\}_{k=1}^{i-1}, Q_t, H)$ over the module tokens (10). During training, we minimize the cross-entropy loss $\mathcal{L}_Q^{prog}$ between this predicted distribution and the ground truth program tokens. Fig. 2 outlines the schematics of our program generator.

Modules on captions. As the image caption C is also a part of the dialog (history H_0 at round 0), it is desirable to track entities from C via the coreference pool P_{ref}. To this effect, we propose a novel extension of neural module networks to captions by using an auxiliary task that checks the alignment of a (caption, image) pair. First, we learn to predict a program from C, different from those generated from Q_t, by minimizing the negative log-likelihood $\mathcal{L}_C^{prog}$, akin to $\mathcal{L}_Q^{prog}$, of the ground truth caption program. Next, we execute the caption program on two images $I^+ = I$ and I^- (a random image from the dataset), to produce caption context vectors c_C^+ and c_C^-, respectively. Note that c_C^+ and c_C^- are different from the context vector c_t produced from execution of the question program. Finally, we learn a binary classifier on top to output classes $+1/-1$ for c_C^+ and c_C^-, respectively, by minimizing the binary cross entropy loss $\mathcal{L}_C^{aux}$. The intuition behind the auxiliary task is: to rightly classify aligned (C, I^+) from misaligned (C, I^-), the modules will need to localize and focus on salient entities in the caption. These entities (specifically, outputs from `Find` in the caption program) are then collected in P_{ref} for explicit coreference resolution on Q_t.

Entities in answers. Using an analogous argument as above, answers from the previous rounds $\{A_i\}_{i=1}^{t-1}$ could have entities necessary to resolve coreferences in Q_t. For example, *'Q: What is the boy holding? A: A ball. Q: What color is it?'*

requires resolving ‘it’ with the ‘ball’ mentioned in the earlier answer. To achieve this, at the end of round $t-1$, we encode $H_{t-1} = (Q_{t-1}, A_{t-1})$ as h_t^{ref} using a multi-layer LSTM, obtain the last image attention map a fed to the last module in the program that produced the context vector c_t, and add (h^{ref}, a) as an additional candidate to the reference pool P_{ref}. Notice that h^{ref} contains the information about the answer A_{t-1} in the context of the question Q_{t-1}, while a denotes the image attention which was the last crucial step in arriving at A_{t-1} in the earlier round. In resolving coreferences in Q_t, if any, all the answers from previous rounds now become potential candidates by virtue of being in P_{ref}.

3.4 Other Model Components

Program Execution. This component takes the generated program and associated text features x_{txt} for each participating module, and executes it. To do so, we first deserialize the given program from its RPN to a hierarchical module layout. Next, we arrange the modules dynamically according to the layout, giving us the network to answer Q_t. At this point, the network is a simple feed-forward neural network, where we start the computation from the leaf modules and feed outputs activations from modules at one layer as inputs into modules at the next layer (see Fig. 2). Finally, we feed a context vector c_t produced from the last module into the next answer decoding component.

Answer Decoding. This is the last component of our model that uses the context vector c_t to score answers from a pool of candidates $\mathcal{A}_t$, based on their correctness. The answer decoder: (a) encodes each candidate $A_t^{(i)} \in \mathcal{A}_t$ with a multi-layer LSTM to obtain $o_t^{(i)}$, (b) computes a score via a dot product with the context vector, i.e., $c_t^T o_t^{(i)}$, and (c) applies a softmax activation to get a distribution over the candidates. During training, we minimize the negative log-likelihood $\mathcal{L}_A^{dec}$ of the ground truth answer A_t^{gt}. At test time, the candidate with the maximum score is picked as $\mathcal{A}_t$. Using nomenclature from [11], this is a *discriminative* decoder. Note that our approach is not limited to a discriminative decoder, but can also be used with a *generative* decoder (see supplement).

Training Details. Our model components have fully differentiable operations within them. Thus, to train our model, we combine the supervised loss terms from both program generation $\{\mathcal{L}_Q^{prog}, \mathcal{L}_C^{prog}, \mathcal{L}_C^{aux}\}$ and answer decoding $\{\mathcal{L}_A^{dec}\}$, and minimize the sum total loss $\mathcal{L}^{total}$.

4 Experiments

We first show results on the synthetic MNIST Dialog dataset [37], designed to contain complex coreferences across rounds while being relatively easy textually and visually. It is important to resolve these coreferences accurately in order to do well on this dataset, thus stress testing our model. We then experiment with a large visual dialog dataset on real images, VisDial [11], which offers both linguistic and perceptual challenge in resolving visual coreferences and grounding them in the image. Implementation details are in the supplement.

Model	Acc.
I [37]	20.2
Q [37]	36.6
AMEM\Seq [37]	89.2
AMEM [37]	96.4
NMN [18]	23.8
CorefNMN\Seq	88.7
CorefNMN	**99.3**

Table 2: Answer accuracy on MNIST Dialog dataset. Higher the better. Our CorefNMN outperforms all other models with a near perfect accuracy on test set.

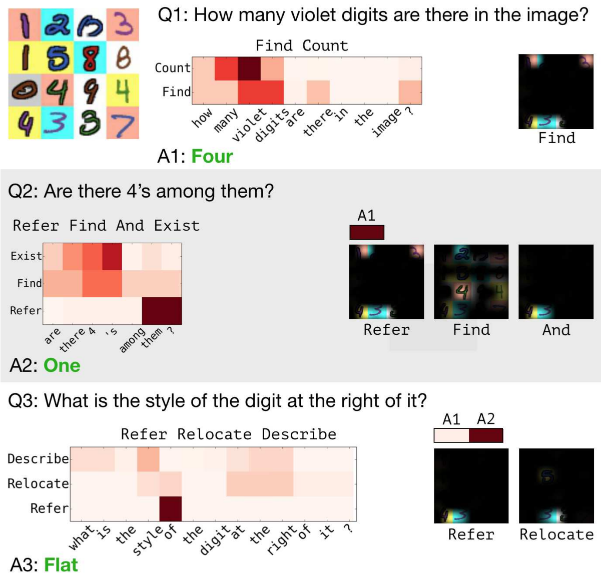
Q1: How many violet digits are there in the image?
Find Count
Count
Find
how many violet digits are there in the image ?
Find
A1: Four
Q2: Are there 4's among them?
Refer Find And Exist
A1
Exist
Find
Refer
are there 4 's among them ?
Refer
Find
And
A2: One
Q3: What is the style of the digit at the right of it?
A1 A2
Refer Relocate Describe
Describe
Relocate
Refer
what is the style of the digit at the right of it ?
Refer
Relocate
A3: Flat

Fig. 3: Illustration of explicit coreference resolution reasoning of our model on the MNIST dialog dataset. For each question, a program and corresponding attentions (α's) over question words (hot matrix on the left) is predicted. A layout is unpacked from the program, and modules are connected to form a feed-forward network used to answer the question, shown in green to indicate correctness. We also visualize output attention maps (right) from each participating module. Specifically, in Q1 and Q2, `Find` localizes all violet digits and 4's, respectively (indicated by the corresponding α). In Q2, `Refer` resolves *'them'* and borrows the visual grounding from previous question.

4.1 MNIST Dialog Dataset

Dataset. The dialogs in the MNIST dialog dataset [37] are grounded in images composed from a 4×4 grid of MNIST digits [23]. Digits in the grid have four attributes—digit class $(0 - 9)$, color, stroke, and background color. Each dialog has 10 question-answer pairs, where the questions are generated through language templates, and the answers are single words. Further, the questions are designed to query attributes of target digit(s), count digits with similar attributes, etc., all of which need tracking of the target digits(s) by resolving references across dialog rounds. Thus, coreference resolution plays a crucial part in the reasoning required to answer the question, making the MNIST dataset both interesting and challenging (Fig. 3). The dataset contains $30k$ training, $10k$ validation, and $10k$ test images, with three 10-round dialogs for each image.

Models and baselines. Taking advantage of single-word answers in this dataset, we simplify our answer decoder to be a N-way classifier, where N is the number of possible answers. Specifically, the context vector c_t now passes through a fully connected layer of size N, followed by softmax activations to give us a distribution over possible answer classes. At training time, we minimize the cross-entropy $\mathcal{L}_A^{dec}$ of the predicted answer distribution with the ground truth answer, at every round. Note that single-word answers also simplify evaluation as answer accuracy can now be used to compare different models. We further simplify our model by removing the memory augmentation to the program generator, i.e., $\hat{q}_t = q_t$ (7), and denote it as CorefNMN. In addition to the full model, we also evaluate an ablation, CorefNMN\Seq, without $\Delta_i t$ that additionally captured sequential nature of dialog (see `Refer` description). We compete against the explicit reasoning model (NMN) [18] and a comprehensive set of baselines AMEM, image-only (I), and question-only (Q), all from [37].

Supervision. In addition to the ground truth answer, we also need program supervision for questions to learn the program generation. For each of the 5 'types' of questions, we manually create one program which we apply as supervision for all questions of the corresponding type. The type of question is provided with the question. Note that our model needs program supervision only while training, and uses predictions from program generator at test time.

Results. Tab. 2 shows the results on MNIST dataset. The following are the key observations: (a) The text-only Q (36.6%) and image-only I (20.2%) do not perform well, perhaps as expected as MNIST Dialog needs resolving strong coreferences to arrive at the correct answer. For the same reason, NMN [18] has a low accuracy of 23.8%. Interestingly, Q outperforms NMN by around 13% (both use question and image, but not history), possibly due to the explicit reasoning nature of NMN prohibiting it from capturing the statistic dataset priors. (b) Our CorefNMN outperforms all other models with near perfect accuracy of 99.3%. Examining the failure cases reveals that most of the mistakes made by CorefNMN was due to misclassifying qualitatively hard examples from the original MNIST dataset. (c) Factoring the sequential nature of the dialog additionally in the model is beneficial, as indicated by the 10.6% improvement in CorefNMN, and 7.2% in AMEM. Intuitively, phrases with multiple potential referents, more often than not, refer to the most recent referent, as seen in Fig. 1, where *'it'* has to be resolved to the closest referent in history. Fig. 3 shows a qualitative example.

4.2 VisDial Dataset

Dataset. The VisDial dataset [11] is a crowd-sourced dialog dataset on COCO images [25], with free-form answers. The publicly available VisDial v0.9 contains 10-round dialogs on around $83k$ training images, and $40k$ validation images. VisDial was collected from pairs of human workers, by instructing one of them to ask questions in a live chat interface to help them imagine the scene better. Thus, the dialogs contain a lot of coreferences in natural language, which need to be resolved to answer the questions accurately.

Models and baselines. In addition to the CorefNMN model described in Sec. 3, we also consider ablations without the memory network augmented program generator (CorefNMN\Mem) or the auxiliary loss $\mathcal{L}_C^{aux}$ to train modules on captions (CorefNMN\$\mathcal{L}_C^{aux}$), and without both (CorefNMN\Mem\$\mathcal{L}_C^{aux}$). As strong baselines, we consider: (a) neural module network without history [18] with answer generation, (b) the best *discriminative* model based on memory networks

Model	**MRR**	**R@1**	**R@5**	**R@10**	**Mean**
MN-QIH-D [11]	0.597	45.55	76.22	85.37	5.46
HCIAE-D-MLE [27]	0.614	47.73	77.50	86.35	5.15
AMEM+SEQ-QI [37]	0.623	48.53	78.66	87.43	4.86
NMN[18]	0.616	48.24	77.54	86.75	4.98
CorefNMN\Mem	0.618	48.56	77.76	86.95	4.92
CorefNMN\\$\mathcal{L}_C^{aux}$	**0.636**	**50.49**	79.56	88.30	4.60
CorefNMN\Mem\\$\mathcal{L}_C^{aux}$	0.617	48.47	77.54	86.77	4.99
CorefNMN	**0.636**	50.24	**79.81**	**88.51**	**4.53**
CorefNMN (ResNet-152)	0.641	50.92	80.18	88.81	4.45

Table 3: Retrieval performance on the validation set of VisDial dataset [11] (discriminative models) using VGG [38] features (except last row). Higher the better for mean reciprocal rank (MRR) and recall@k (R@1, R@5, R@10), while lower the better for mean rank. Our CorefNMN model outperforms all other models across all metrics.

MN-QIH-D from [11], (c) history-conditioned image attentive encoder (HCIAE-D-MLE) [26], and (d) Attention-based visual coreference model (AMEM+SEQ-QI) [37]. We use ImageNet pretrained VGG-16 [38] to extract x_{vis}, and also ResNet-152 [17] for CorefNMN. Further comparisons are in supplement.

Evaluation. Evaluation in visual dialog is via retrieval of the ground truth answer A_t^{gt} from a pool of 100 candidate answers $\mathcal{A}_t = \{A_t^{(1)}, \cdots A_t^{(100)}\}$. These candidates are ranked based the discriminative decoder scores. We report Recall@k for $k = \{1, 5, 10\}$, mean rank, and mean reciprocal rank (MRR), as suggested by [11], on the set of 40k validation images (there is not test available for v0.9).

Supervision. In addition to the ground truth answer A_t^{gt} at each round, our model gets program supervision for Q_t, to train the program generator. We automatically obtain (weak) program supervision from a language parser on questions (and captions) [19] and supervision to predict for `Refer` from an off-the-shelf text coreference resolution tool[4], based on [10]. For questions that are a part of coreference chain, we replace `Find` with `Refer` in the parser supervised program. Our model predicts everything from the questions at test time.

Results. We summarize our observations from Tab. 3 below: (a) Our CorefNMN outperforms all other approaches across all the metrics, highlighting the importance of explicitly resolving coreferences for visual dialog. Specifically, our R@k ($k = 1, 2, 5$) is at least 1 point higher than the best prior work (AMEM+SEQ-QI), and almost 2 points higher than NMN. (b) Removing memory augmentation (CorefNMN\Mem) hurts performance uniformly over all metrics, as the model is unable to peek into history to decide when to resolve coreferences via the `Refer` module. Modules on captions seems to have varied effect on the full model, with decrease in R@1, but marginal increase or no effect in other metrics. (c) Fig. 4 illustrates the interpretable and grounded nature of our model.

[4] https://github.com/huggingface/neuralcoref

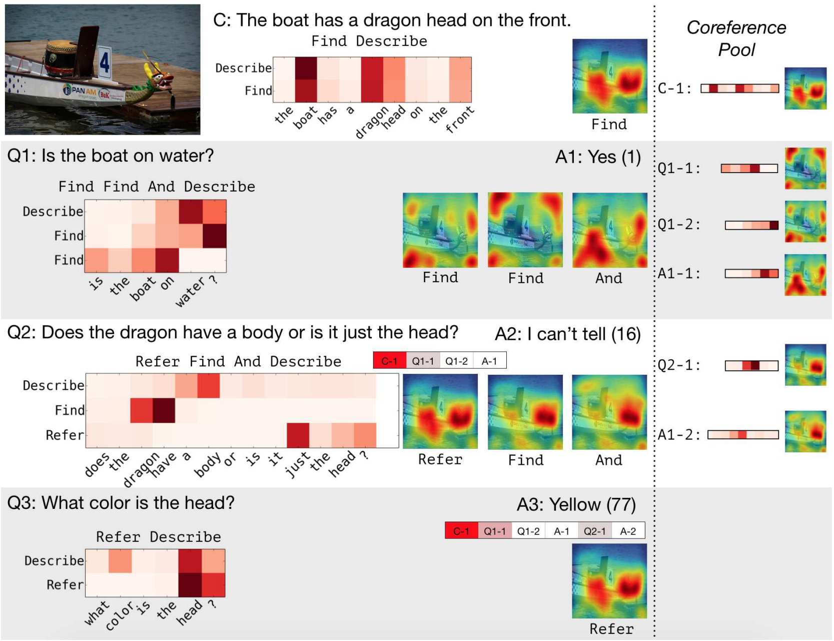

Fig. 4: Example to demonstrate explicit coreference resolution by our CorefNMN model. It begins by grounding *'dragon head'* from the caption C (shown on top), and saves it in the coreference pool P_{ref} (right). At this point however, it does not consider the entity *'boat'* important, and misses it. Next, to answer Q1, it localizes *'boat'* and *'water'*, both of which are 'unseen', and rightly answers with *Yes*. The ground truth rank (1 for Q1) is shown in the brackets. Additionally, it also registers these two entities in P_{ref} for coreference resolution in future dialog. For Q2, it refers the phrase *'the head'* to the referent registered as C-1, indicated by attention on the bar above `Refer`.

5 Conclusions

We introduced a novel model for visual dialog based on neural module networks that provides an introspective reasoning about visual coreferences. It explicitly links coreferences and grounds them in the image at a word-level, rather than implicitly or at a sentence-level, as in prior visual dialog work. Our CorefNMN outperforms prior work on both the MNIST dialog dataset (close to perfect accuracy), and on VisDial dataset, while being more interpretable, grounded, and consistent by construction.

Acknowledgements. This work was supported in part by NSF, AFRL, DARPA, Siemens, Google, Amazon, ONR YIPs and ONR Grants N00014-16-1-{2713,2793}, N000141210903. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government, or any sponsor.

References

1. Agrawal, A., Batra, D., Parikh, D.: Analyzing the behavior of visual question answering models. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) (2016)
2. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and vqa. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
3. Andreas, J., Rohrbach, M., Darrell, T., Klein, D.: Learning to Compose Neural Networks for Question Answering (2016)
4. Andreas, J., Rohrbach, M., Darrell, T., Klein, D.: Neural module networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
5. Anne Hendricks, L., Wang, O., Shechtman, E., Sivic, J., Darrell, T., Russell, B.: Localizing moments in video with natural language. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017)
6. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C.L., Parikh, D.: Vqa: Visual question answering. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2015)
7. Bahdanau, D., Cho, K., Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate. In: Proceedings of the International Conference on Learning Representations (ICLR) (2015)
8. Bergsma, S., Lin, D.: Bootstrapping path-based pronoun resolution. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL). Association for Computational Linguistics (2006)
9. Burks, A.W., Warren, D.W., Wright, J.B.: An analysis of a logical machine using parenthesis-free notation. Mathematical Tables and Other Aids to Computation **8**(46), 53–57 (1954), http://www.jstor.org/stable/2001990
10. Clark, K., Manning, C.D.: Deep reinforcement learning for mention-ranking coreference models. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) (2016)
11. Das, A., Kottur, S., Gupta, K., Singh, A., Yadav, D., Moura, J.M., Parikh, D., Batra, D.: Visual Dialog. In: CVPR (2017)
12. Das, A., Kottur, S., Moura, J.M., Lee, S., Batra, D.: Learning Cooperative Visual Dialog Agents with Deep Reinforcement Learning. arXiv preprint arXiv:1703.06585 (2017)
13. Fukui, A., Park, D.H., Yang, D., Rohrbach, A., Darrell, T., Rohrbach, M.: Multimodal compact bilinear pooling for visual question answering and visual grounding. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) (2016)
14. Geman, D., Geman, S., Hallonquist, N., Younes, L.: Visual Turing Test for computer vision systems. In: Proceedings of the National Academy of Sciences (2015)
15. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
16. Grice, H.P.: Logic and conversation. In: Cole, P., Morgan, J.L. (eds.) Syntax and Semantics: Vol. 3: Speech Acts, pp. 41–58. Academic Press, New York (1975), http://www.ucl.ac.uk/ls/studypacks/Grice-Logic.pdf

17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
18. Hu, R., Andreas, J., Rohrbach, M., Darrell, T., Saenko, K.: Learning to reason: End-to-end module networks for visual question answering. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017)
19. Hu, R., Xu, H., Rohrbach, M., Feng, J., Saenko, K., Darrell, T.: Natural language object retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
20. Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C.L., Girshick, R.: Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2017)
21. Johnson, J., Hariharan, B., van der Maaten, L., Hoffman, J., Fei-Fei, L., Zitnick, C.L., Girshick, R.: Inferring and executing programs for visual reasoning. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017)
22. Kong, C., Lin, D., Bansal, M., Urtasun, R., Fidler, S.: What are you talking about? text-to-image coreference. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
23. LeCun, Y., Cortes, C.: MNIST handwritten digit database (2010), http://yann.lecun.com/exdb/mnist/
24. Lin, D., Fidler, S., Kong, C., Urtasun, R.: Visual semantic search: Retrieving videos via complex textual queries. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
25. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common Objects in Context. In: Proceedings of the European Conference on Computer Vision (ECCV) (2014)
26. Lu, J., Kannan, A., Yang, J., Parikh, D., Batra, D.: Best of both worlds: Transferring knowledge from discriminative learning to a generative visual dialog model. In: Advances in Neural Information Processing Systems (NIPS) (2017)
27. Lu, J., Yang, J., Batra, D., Parikh, D.: Hierarchical Question-Image Co-Attention for Visual Question Answering. In: Advances in Neural Information Processing Systems (NIPS) (2016)
28. Malinowski, M., Rohrbach, M., Fritz, M.: Ask your neurons: A neural-based approach to answering questions about images. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2015)
29. Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A., Murphy, K.: Generation and comprehension of unambiguous object descriptions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
30. Massiceti, D., Siddharth, N., Dokania, P.K., Torr, P.H.S.: Flipdial: A generative model for two-way visual dialogue (2018)
31. Mitchell, M., van Deemter, K., Reiter, E.: Generating expressions that refer to visible objects. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1174–1184. Association for Computational Linguistics, Atlanta, Georgia (June 2013), http://www.aclweb.org/anthology/N13-1137
32. Plummer, B., Wang, L., Cervantes, C., Caicedo, J., Hockenmaier, J., Lazebnik, S.: Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2015)

33. Ramanathan, V., Joulin, A., Liang, P., Fei-Fei, L.: Linking people in videos with "their" names using coreference resolution. In: Proceedings of the European Conference on Computer Vision (ECCV) (2014)
34. Regneri, M., Rohrbach, M., Wetzel, D., Thater, S., Schiele, B., Pinkal, M.: Grounding Action Descriptions in Videos. Transactions of the Association for Computational Linguistics (TACL) **1**, 25–36 (2013)
35. Rohrbach, A., Rohrbach, M., Hu, R., Darrell, T., Schiele, B.: Grounding of textual phrases in images by reconstruction. In: Proceedings of the European Conference on Computer Vision (ECCV) (2016)
36. Rohrbach, A., Rohrbach, M., Tang, S., Oh, S.J., Schiele, B.: Generating descriptions with grounded and co-referenced people. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
37. Seo, P.H., Lehrmann, A., Han, B., Sigal, L.: Visual reference resolution using attention memory for visual dialog. In: Advances in Neural Information Processing Systems (NIPS) (2017)
38. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Proceedings of the International Conference on Learning Representations (ICLR) (2015)
39. Strub, F., de Vries, H., Mary, J., Piot, B., Courville, A.C., Pietquin, O.: End-to-end optimization of goal-driven and visually grounded dialogue systems. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) (2017)
40. de Vries, H., Strub, F., Chandar, S., Pietquin, O., Larochelle, H., Courville, A.C.: Guesswhat?! visual object discovery through multi-modal dialogue. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
41. Wang, L., Li, Y., Lazebnik, S.: Learning deep structure-preserving image-text embeddings. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
42. Weston, J., Chopra, S., Bordes, A.: Memory networks. In: Proceedings of the International Conference on Learning Representations (ICLR) (2015)
43. Winograd, T.: Procedures as a representation for data in a computer program for understanding natural language. Tech. rep., DTIC Document (1971)
44. Winograd, T.: Understanding Natural Language. Academic Press, Inc., Orlando, FL, USA (1972)
45. Yu, H., Siskind, J.M.: Grounded language learning from videos described with sentences. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL) (2013)
46. Yu, L., Poirson, P., Yang, S., Berg, A.C., Berg, T.L.: Modeling context in referring expressions. In: Proceedings of the European Conference on Computer Vision (ECCV) (2016)
47. Zhang, P., Goyal, Y., Summers-Stay, D., Batra, D., Parikh, D.: Yin and Yang: Balancing and Answering Binary Visual Questions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)