Unified Questioner Transformer for Descriptive Question Generation in Goal-Oriented Visual Dialogue

Shoya Matsumori*†[1] , Kosuke Shingyouchi†[1], Yuki Abe[1],
Yosuke Fukuchi[1], Komei Sugiura[1], Michita Imai[1]
[1]Keio University

Abstract

Building an interactive artificial intelligence that can ask questions about the real world is one of the biggest challenges for vision and language problems. In particular, goal-oriented visual dialogue, where the aim of the agent is to seek information by asking questions during a turn-taking dialogue, has been gaining scholarly attention recently. While several existing models based on the GuessWhat?! dataset [10] have been proposed, the Questioner typically asks simple category-based questions or absolute spatial questions. This might be problematic for complex scenes where the objects share attributes, or in cases where descriptive questions are required to distinguish objects. In this paper, we propose a novel Questioner architecture, called Unified Questioner Transformer (UniQer), for descriptive question generation with referring expressions. In addition, we build a goal-oriented visual dialogue task called CLEVR Ask. It synthesizes complex scenes that require the Questioner to generate descriptive questions. We train our model with two variants of CLEVR Ask datasets. The results of the quantitative and qualitative evaluations show that UniQer outperforms the baseline.

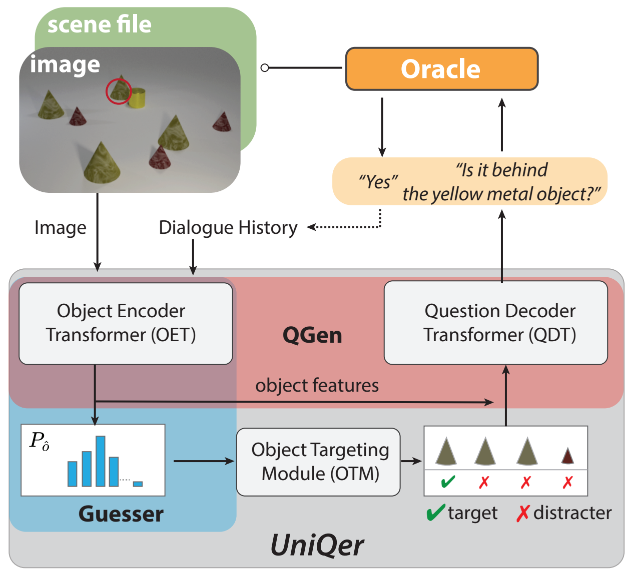

Figure 1: *UniQer* and *CLEVR Ask* task. *UniQer* unifies the Question Generator (QGen), which generates yes-no questions, and the Guesser, which guesses an *Oracle*'s reference object, into a single transformer encoder-decoder architecture. The Object Targeting Module is introduced to assign target objects which are to be set apart from distracter objects in the QGen.

1. Introduction

Information seeking through interaction is one of the most vital abilities for artificial intelligence. This is particularly true in the human-agent interaction scenario [24, 17]. For example, task-oriented agents need to understand what the users are thinking, i.e., beliefs, preferences, and intentions, in order to correctly interpret their instructions [14, 39]. In most cases, such information is not provided prior to the interaction, so the agents have to elicit it by asking questions on the fly.

To build effective information-seeking agents, several studies in the vision and language community have tackled the goal-oriented visual dialogue task [10, 28]. This task consists of two agents, called a *Questioner* and an *Oracle*, and the goal is to train the *Questioner* to guess the *Oracle*'s reference object by asking yes-no questions during a turn-taking dialogue. The *Oracle* then needs to provide an answer given the question and the target object.

In goal-oriented visual dialogue, deciding which objects and how to address will depend on the complexity of the presented image. For example, in simple scenes, where each object has different attributes and thus is easy to distinguish, the *Questioner* only needs to ask category-based questions such as "*Is it a car?*". In a more complex scene, on the other hand, the *Questioner* needs to ask descriptive questions with referring expressions [7, 31] to narrow down

*Corresponding Author: shoya@ailab.ics.keio.ac.jp
†Equally Contributed

the candidates, such as "*Is it behind a tree?*" or "*Is it a red car?*". In this paper, we particularly focus on building an agent that can facilitate such descriptive questions in goal-oriented visual dialogue.

Several models based on *GuessWhat?!* [10] have been published, most of which leveraged reinforcement learning aimed at maximizing the success reward by generating a word token as an action [28, 35, 37, 26, 27, 2, 22, 1]. However, the *Questioner*s in these models typically generate only simple category-based questions, such as "Is it a person?" or "Is it a computer?", or absolute spatial questions, such as "Is it in front?" or "Is it on the left side of the image?", which is not effective when a large number of similar-looking candidates is presented in the same place.

In response to the above issues, we propose *Unified Questioner Transformer* (*UniQer*) and a task called *CLEVR Ask* for descriptive question generation in goal-oriented visual dialogue (Fig. 1). In *UniQer*, the question generator (QGen) and the Guesser are unified into a single transformer architecture. By utilizing such architecture, both the QGen and the Guesser can make use of the same object features, and the Guesser can consider object relations more effectively thanks to the self-attention structure. We also introduce an object-targeting module, inspired by the notion of target and distracter [8], which will decide on the objects that are to be addressed in the question. With this structure, the QGen can focus on generating questions that will address the target from other objects in a supervised manner. Our task, *CLEVR Ask*, includes complex scenes in which similar-looking objects are presented; therefore, the *Questioner* needs to develop the ability to ask descriptive questions rather than simple ones. The experimental results show that our model's performance exceeds that of the baseline in terms of task success rate by around 20%. Besides, the extensive ablation studies conducted show the structural advantages of our model.

To summarize, our contributions are three-fold:

- We proposed a novel unified transformer architecture for the *Questioner* in goal-oriented visual dialogue. To the best of our knowledge, this is the first study that introduces a transformer architecture to goal-oriented visual dialogue.
- To address the limitations of *GuessWhat?!* dataset, we constructed a novel goal-oriented visual dialogue task, namely *CLEVR Ask*, which requires the *Questioner* to ask descriptive questions.
- We evaluated *UniQer* with two variants of *CLEVR Ask* datasets and found that our model outperformed the baseline and was able to ask descriptive questions with referring expressions, given complex scenes where the objects were hard to distinguish.

2. Related Works

Visual Question Answering. Visual Question Answering (VQA) [3, 15, 13] lies at the intersection of computer vision and natural language processing (NLP). Compared with image captioning tasks [12], VQA requires a comprehensive understanding of the visual object elements and the relationships between them. [15] proposed *CLEVR*, a synthetic VQA dataset, aiming to remove the difficulties in image recognition and creates a balanced dataset that requires reasoning abilities without shortcuts. Besides, extending VQA to dialogues has been challenged in Visual Dialogue [9, 25].

Transformer Architecture. Transformer architecture [30] was recently introduced in NLP tasks showing the effectiveness of the self-attention structure in language modeling, followed by large-scale pre-trained models such as BERT [11] and its successive models [20, 32]. Several studies have imported transformer architecture to the aforementioned vision and language tasks, such as Image BERT [23], Meshed Memory Transformer [6], and UNITER [4].

Referring Expression Generation. Referring expressions are language constructions used to identify specific objects in a scene [31, 7]. These specific objects here are called the targets, and a set of objects that are to be isolated from the target set is called the distracter group [8]. Various datasets have been proposed recently including both synthetic [19] and natural image datasets [21, 33] to generate and comprehend referring expressions.

Goal-oriented Visual Dialogue. Our task is grounded on the goal-oriented visual dialogue framework. This test-bed was first implemented in *GuessWhat?!* [10], which is composed of 155K goal-oriented dialogues, collected via the Amazon Mechanical Turk, and includes 822K questions, with a unique vocabulary size of 5K. The images were borrowed from the *MSCOCO* dataset and consist of up to 67k images and 134K objects.

The *GuessWhat?!* task was originally designed for supervised learning, but it was extended to fit into the reinforcement learning framework [28]. Ongoing works promote various approaches to *GuessWhat?!*. One is the rewarding in reinforcement learning, where [36] used two intermediate rewards, while another, proposed by [27], makes use of an informativeness reward based on regularized information gain. [37] improved the reinforcement learning optimization by extending a policy gradient method using a temperature for each action based on action frequencies. Moreover, other studies focus on improving the model's architecture. [2] introduced a Bayesian approach to quantify the uncertainty in the model. [22] proposed dialogue state tracking module to make use of the belief of the Guesser. [27] proposed single visually grounded dialogue encoder shared by both the Guesser and the QGen, trained with cooperative learning.

Limitations. Among such models proposed on *Guess-*

What?!, there are two major limitations. (1) First is the separated learning approach that previous methods adopt. In most of the previous methods, a *Questioner* has two major components: the QGen which generates questions based on the image presented and the current dialogue history and the Guesser which guesses the target object of the *Oracle*. Ideally, the object's features obtained during the training should be shared with both module, but these two components are often learning separately. Additionally, the Guesser in the previous studies only looks at a single object at a time to determine the probability of it being the target object and does not consider the relation between objects. This will be problematic for processing questions that refer to the other objects, i.e. referring expressions. (2) Another problem comes from the fact that the QGen was too heavily burdened. The QGen in the previous models needs to decide on which objects to refer to and how to refer to in the same architecture. Coupled with the difficulties of tuning generator with reinforcement learning, this will degrade the final performance of the task, such as losing the lexical diversity of the questions as reported in [26].

While *GuessWhat?!* has pioneered the frontier of the goal-oriented visual dialogue, it has several issues yet to be resolved in order to build a *Questioner* that can ask descriptive questions. The major issue of the *GuessWhat?!* dataset is the poor performance of the *Oracle* [10]. Since the *Oracle* is solving a 3-class classification problem—yes, no, and not applicable—, this accuracy causes substantial errors in the *Questioner*'s learning process. Another problem is that the *Oracle* generates answers to the questions without having been given the visual features; only the categorical information and spatial information of the objects are given. This means that the *Oracle* is incapable of understanding descriptive questions with referring expressions that include visual attributes.

3. *CLEVR Ask* Specifications

Notation. *CLEVR Ask* is defined by the tuple $(\mathcal{I}, \mathcal{S}, \mathcal{O}, o^*, \mathcal{D})$, where $\mathcal{I}$ is the image, $\mathcal{S}$ is the scene meta-data for the image, $\mathcal{O}$ is the set of objects appearing in the scene, o^* is the goal reference object of the *Oracle*, and $\mathcal{D}$ is the dialogue between the *Oracle* and the *Questioner*.

Formally, $\mathcal{I} \in \mathbb{R}^{H \times W}$ is the current observed image, with height H and width W. Corresponding to the image, the scene meta-data $\mathcal{S}$ provides the information of the scene, including the collections of the objects attributes (shape, size, color, and material) and position on the ground-plane. These are represented by a scene graph [16] of *CLEVR*, which provides a complete view of the environment. The objects in the scene are represented by $\mathcal{O} = \{o_i\}_{i=1}^{N}$, where o_i is the i-th object and N is the number of the objects within the scene. For each session, the goal object $o^* \in \mathcal{O}$ is arbitrarily chosen. The dialogue, where the *Questioner* seeks to identify the goal object o^*, consists of question-answer pairs $\mathcal{D} = (q^t, a^t)_{t=1}^{T}$, T denotes the number of such pairs. Each question is composed of word tokens $q^t = (w^t_\omega)_{\omega=1}^{W_t}$, that were sampled from the vocabulary $\mathcal{V}$. Finally, the answer to the question is restricted to yes, no or invalid question, that is, $a^t \in \{\text{<yes>}, \text{<no>}, \text{<invalid>}\}$.

Oracle. The central role of the *Oracle* is to provide an answer to each question generated by the *Questioner* based on the current scene. Since its performance will directly affect the learning process of the *Questioner*, the *Oracle* needs to be perfect, as the name implies.

To satisfy this demand, we built a robust *Oracle* function that meets the following two requirements. First, the *Oracle* needs to understand the scene completely to answer questions. For this purpose, we take advantage of the fully structured environment of *CLEVR*. Since the *CLEVR* world is built on a structured ground-truth representation [16], the complete and exhaustive information about the image is available in the scene file. We introduced such information to the *Oracle* so that it can answer any kind of detailed expressions. Second, the *Oracle* is required to completely understand what the *Questioner* says. Here, we standardized the language used by the *Questioner* to a pseudo-language that can be directly interpreted by the *Oracle*. This language is an executable functional program; given the scene-file, it yields the objects that match the query in a deterministic manner, which enables the *Oracle* to deduce the answer to the question. These settings will liberate the *Oracle* from having to interpret the question based on the model obtained via learning, which is likely to produce some errors.

Dataset Generation. The *CLEVR Ask* dataset consists of images including scene files and questions that are bound to images. In the original *CLEVR* dataset, the object attributes are three object shapes, two absolute sizes, two materials, and eight colors. Notably, the numbers of entities in each attribute in the *CLEVR* dataset are not the same. This will likely cause undesirable shortcuts for the goal-oriented visual dialogue task; for example, only asking questions about attributes with fewer entities, (e.g., sizes and materials) can result in the candidate objects being bisected.

To circumvent this issue, we prepared two new balanced datasets, Ask3 and Ask4, which contain three and four entities for each attribute, respectively. Both Ask3 and Ask4 datasets consist of 70K training, 7.5K validation, and 7.5K test images. All images were generated by randomly sampling a scene graph and rendered by Blender [5]. We followed the original sampling procedure in *CLEVR* except that for each generation step we randomly chose either to copy the existing object or to create a new object by sampling attributes. This enables us to intentionally place an identical object in the scene.

As presented in [26], the questions that appear in goal-oriented visual dialogue can be roughly categorized into the

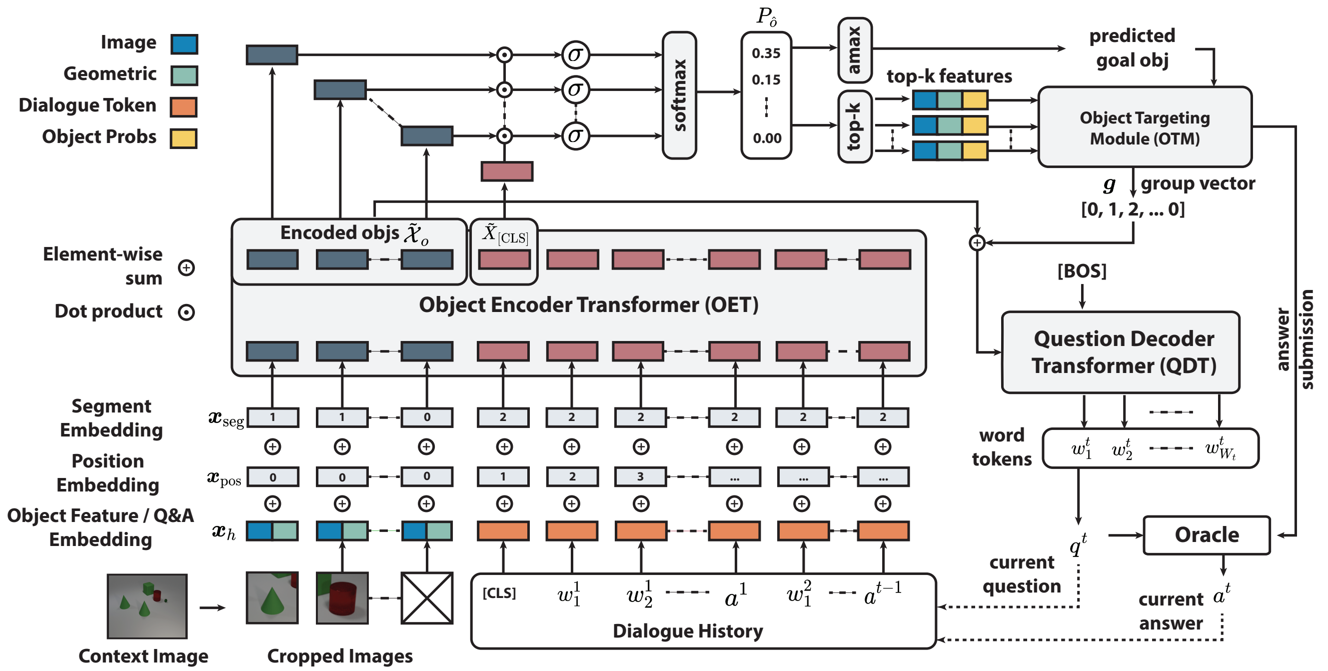

Figure 2: Network structure of *UniQer*. Given the cropped image features, geometrical features, [CLS] token, and past question and answer tokens, with the segment and sequence position embedding injected, embedding vectors are fed into *OET*. By comparing the encoded [CLS] token with the encoded objects and passing them to softmax, goal object probability $P_{\hat{o}}$ is obtained. The top-k objects with high $P_{\hat{o}}$ are encoded and input to the *OTM* module along with the image features and geometrical features, and each object is assigned a group ID. Finally, *QDT* takes [BOS] as an input, generates the next question by taking the sum of the encoded objects and the group vector as memory, and the *Oracle* answers to the question.

following question types and its combinations thereof: entity, attribute, location, and action. Among these, we focus on attribute questions and location questions with referring expressions. The attribute question inquires about the attributes of the object and the location questions ask about the spatial relations, such as "Is it the object in front of the purple cube?".

4. *Unified Questioner Transformer*

We propose *Unified Questioner Transformer* (*UniQer*) as a *Questioner* model for goal-oriented visual dialogue tasks. Conceptually, *UniQer* can be divided into the following three major components:

- Object Encoder Transformer (*OET*)
- Object Targeting Module (*OTM*)
- Question Decoder Transformer (*QDT*)

as shown in Fig. 2. The *OET* encodes objects and serves as a Guesser, which infers the goal object o^*. The *OTM* is trained to learn which object should be addressed in the next question based on the *OET*'s current inference. The *QDT* decodes object features and generates questions, by addressing the object to set apart from the other objects that have different group IDs from the *OTM*. When it is confident enough, the *OTM* decides to submit its answer to the oracle. Both the *OET* and *QDT* are trained jointly in a supervised manner, and thus they can benefit from each other. The *OTM* is trained with reinforcement learning.

4.1. Embedding Preparation

The input of the *UniQer* is the cropped object images and dialogue history. The embedding layer prepares the feature vectors for both inputs by

$$\boldsymbol{x}_e(i) = \boldsymbol{x}_h(i) + \boldsymbol{x}_{\text{seg}}(i) + \boldsymbol{x}_{\text{pos}}(i), \tag{1}$$

where i denotes the index of the object, $\boldsymbol{x}_h(i)$ is either the object feature embedding $\boldsymbol{x}_v(i)$ or the dialog token embedding $\boldsymbol{x}_l(i)$, $\boldsymbol{x}_{\text{seg}}(i)$ is a segment embedding, and $\boldsymbol{x}_{\text{pos}}(i)$ is a sequence position embedding.

Object Feature Embedding. Each cropped image corresponding to the objects in the scenes is first processed to $\boldsymbol{o}_v(i)$ using an arbitrary image feature extractor. As in [34], two types of 5D geometric feature for each object are introduced as well: a source geometric feature represented by $\boldsymbol{o}_{sg}(i) = [\frac{x_{ci}}{W}, \frac{y_{ci}}{H}, \frac{w_{bi}}{W}, \frac{h_{bi}}{H}, \frac{w_{bi}h_{bi}}{WH}]$, and relative geometric feature $\boldsymbol{o}_{rg}(i,j) = [\frac{x_{cj}}{w_{bi}}, \frac{y_{cj}}{h_{bi}}, \frac{w_{bj}}{w_{bi}}, \frac{h_{bj}}{h_{bi}}, \frac{w_{bj}h_{bj}}{w_{bi}h_{bi}}]_{j\in\mathcal{N}, j\neq i}$, where (x_{ci}, y_{ci}) and (w_{bi}, h_{bi}) denote the center coordinates and the width and height of the bounding box for the object o_i, respectively, (W, H) is the scene image size, and $\mathcal{N}$ is the set of object indices in the scene. Thus, we define the geometrical feature as follows:

$$\boldsymbol{o}_g^{\Phi}(i) = [\boldsymbol{o}_{sg}(i), \{\boldsymbol{o}_{rg}(i,j) | j \in \Phi, j \neq i\}], \tag{2}$$

where Φ is a set of object indices. Then the image feature and the geometrical feature are concatenated and fed into the linear transformation function $\mathcal{F}_v$ with ReLU activation as follows: $\boldsymbol{x}_v(i) = \mathcal{F}_v([\boldsymbol{o}_v(i), \boldsymbol{o}_g^{\mathcal{N}}(i)])$.

Dialogue Token Embedding. Questions and answers in the current dialogue history $\mathcal{D}^t = (q^\tau, a^\tau)_{\tau=1}^t$ are transformed into dialogue token embedding $\boldsymbol{x}_l^t$ with a one-hot embedding layer. Additionally, the classification token ([CLS]) [11], is introduced at the very front of the dialogue token.

Segment embedding and sequence position embedding. Segment embedding $\boldsymbol{x}_{\text{seg}}$ is introduced to distinguish the modality of the input. There are three tokens that represent the cropped object image input, dialogue token input, and mask (padding) token input. Sequence position embedding $\boldsymbol{x}_{\text{pos}}$ is introduced to indicate the position of the token in the sequence. We use the same value for all object embeddings since there is no numerical order in objects, while for tokens in the dialogue history, we add the position ID incrementally.

4.2. Object Encoder Transformer

Given the input embeddings $\boldsymbol{x}_e$, the *OET* plays two important roles; it extracts object features that are then used in the *QDT* and the *OTM* and it predicts a goal object. The *OET* is built on a transformer encoder architecture [30].

The output feature of the *OET* is written as $\tilde{\mathcal{X}} = \{\tilde{\mathcal{X}}_o, \tilde{X}_{[\text{CLS}]}, \tilde{\mathcal{X}}_l\}$, where $\tilde{\mathcal{X}}_o = (\tilde{X}_o^1, \ldots, \tilde{X}_o^{N_{\max}})$ is a set of object embeddings with the maximum number of objects $N_{\max}$, $\tilde{X}_{[\text{CLS}]}$ is the embeddings of the [CLS] token, and $\tilde{\mathcal{X}}_l = (\tilde{X}_l^1, \ldots, \tilde{X}_l^{\Omega})$ is a set of dialogue question and answer token embeddings with its current length Ω. From the encoder output $\tilde{\mathcal{X}}$, only the object embeddings $\tilde{\mathcal{X}}_o$ are passed to the *QDT*.

As for the goal object prediction, the model needs to find the objects that match the current dialogue history $\mathcal{D}^t$. First, the inner products of linear-transformed object embeddings $\tilde{\mathcal{X}}_o$ and [CLS] token embeddings $\tilde{X}_{[\text{CLS}]}$ are computed as

$$\sigma_o = [\text{sigmoid}(\mathcal{F}_o(\tilde{X}_o^i) \cdot \mathcal{F}_c(\tilde{X}_{[\text{CLS}]}))]_{i \in \mathcal{N}}, \tag{3}$$

where $\mathcal{F}_o$ and $\mathcal{F}_c$ are the linear transformation functions. Then, we apply the softmax function to acquire the goal object probability

$$P_{\hat{o}} = \text{softmax}(\sigma_o). \tag{4}$$

Similar to the role of the dialogue state encoder in previous studies, $\tilde{X}_{[\text{CLS}]}$ is expected to capture the summary of the dialogue that is comparable to the object features in order to produce the similarity.

4.3. Object Targeting Module

The *OTM* has two roles; it determines which objects should be addressed by *QDT* and it decides when to submit the answer, the *OET*'s prediction, to the *Oracle*.

The *OTM* determines which objects should be addressed in the question by assigning any of the following three types of property to each object: a target object property for the object to be addressed, a distracter object property for an object that will not be addressed but is distinguishable from the target objects, and a masked object property for objects that do not exist or will be ignored.

The *OTM* only pays attention to top-k high-scored candidate objects calculated from the object probability $P_{\hat{o}}$, and ignores the others. More specifically, the input of the *OTM* is the set of top-k feature vectors $\{\boldsymbol{x}_k(i)\}_{i \in \mathcal{K}}$, where $\mathcal{K}$ is the set of top-k object indices and $\boldsymbol{x}_k(i) = [\mathcal{F}_A(\boldsymbol{o}_v(i)), \mathcal{F}_B(\boldsymbol{o}_g^{\mathcal{K}}(i)), \mathcal{F}_C(P_{\hat{o}}(i))]$, where $\mathcal{F}_A, \mathcal{F}_B$, and $\mathcal{F}_C$ are different linear layers. Given such an input, the *OTM* will decide how to assign these properties to each top-k objects by producing a targeting action $\hat{g}_k$ as

$$\hat{g}_k \sim \mathcal{F}_{\text{RL}}(\{\boldsymbol{x}_k(i)\}_{i \in \mathcal{K}}), \tag{5}$$

where $\hat{g}_k$ is an integer ranging from zero to $3^k - 1$, which corresponds to the number of combinations to respectively allot the three property groups to k objects. $\mathcal{F}_{\text{RL}}$ is a parameterized function trained with reinforcement learning defined as

$$\mathcal{F}_{\text{RL}} = \text{softmax}(\mathcal{F}_l([\mathcal{F}_{\text{GRU}}(\{\boldsymbol{x}_k(i)\}_{i \in \mathcal{K}})])), \tag{6}$$

where $\mathcal{F}_{\text{GRU}}$ is the two-layered bi-directional GRU to encode input vectors and $\mathcal{F}_l$ is the linear transformation function with the ReLU activation.

To make the *OTM*'s output compatible with the *QDT*'s input, the targeting action $\hat{g}_k$ is then reformatted with base-10 to base-3 conversion to a top-k group vector $\boldsymbol{g}_k \in \mathbb{R}^{k \times 3}$, whose elements each represent the group ID –zero, one, or two– assigned to an object. Finally, a group vector $\boldsymbol{g} \in \mathbb{R}^{N_{\max} \times 3}$ is obtained by filling the masked property group ID to the index for the ignored objects on top-k.

We treat the cases $\hat{g}_k = 0$ and $\hat{g}_k = 3^k - 1$ as a submission action, which submits the current answer $\hat{o} = \text{argmax}(\text{P}_{\hat{\text{o}}})$ to the oracle. Examples of the OTM are demonstrated in the appendix section.

4.4. Question Decoder Transformer

The role of the *QDT* is to generate a question that will distinguish the target objects from the distracter objects as specified by the group vector $\boldsymbol{g}$ produced by the *OTM*. The *QDT* consists of a standard transformer decoder [30]. The input of the *QDT* is the beginning of sentence token ([BOS]) and as a memory we put the element-wise sum of the object embeddings $\tilde{\mathcal{X}}_o$ and the embeddings of group vector $\boldsymbol{g}$, described as:

$$\mathcal{M} = \{\tilde{X}_o^{(i)} + \mathcal{F}_s(\boldsymbol{g}^{(i)})\}_{i\in\mathcal{N}}, \tag{7}$$

where $\mathcal{F}_s$ is an embedding function and $\boldsymbol{g}^{(i)}$ is the i-th element of $\boldsymbol{g}$, which represents the property group of the object o_i. Given [BOS] token and $\mathcal{M}$, the *QDT* generates a tokenized word in an auto-regressive manner.

5. Training Process

5.1. Supervised Learning.

Both the *OET* and the *QDT* are jointly trained with supervised learning to predict the goal object and generate questions. For this purpose, the overall loss function can be expressed as the sum of the object prediction loss and the target-wise question generation loss, as $L = \alpha L_{\text{pred}} + L_{\text{gen}}$, where α is a constant that modifies the ratio between the two loss functions.

Object Prediction Loss. To predict the goal object, we can simply classify each object as to whether or not it is a candidate at the end of each question and answer pair. By applying the $\mathrm{softmax}$ function during reinforcement learning, we can acquire the goal object probability. Therefore, the loss for candidate object prediction is formalized as a multi-label classification problem, where the binary cross-entropy is applied to the sigmoid-activated result in Eq. (3) as follows:

$$L_{\text{pred}} = \sum_{t=1}^{T} -y_o^t \log(\sigma_o^t) - (1-y_o^t)\log(1-\sigma_o^t), \tag{8}$$

where T is the number of questions in a dialogue, y_o^t represents the ground-truth binary labels for the object in the t-th question.

Target-Wise Question Generation Loss. Question generation loss is defined as the negative log-likelihood, as

$$L_{\text{gen}} = -\sum_{t=1}^{T}\sum_{l=1}^{W_t} \log p(w_{l+1}^t | w_l^t, \dots, w_2^t, w_1^t, \mathcal{M}), \tag{9}$$

where W_t is the number of tokens included in the t-th question and $\mathcal{M}$ is expressed as $\{\tilde{X}_o^{(i)} + \mathcal{F}_s(\boldsymbol{g}^{(i)})\}_{i\in\mathcal{N}}$. During supervised learning, the $\boldsymbol{g}$ is computed from the ground-truth question. Here, the target group is assigned to the object that matches with the question when the answer to it is true, otherwise the distracter group is assigned. Note that, $N_{\max} - k$ objects are assigned to a masked group in order to reproduce the reinforcement training conditions, where k is a pre-defined constant for top-k objects.

5.2. Reinforcement Learning

The *OTM* is trained with reinforcement learning to generate the group vector $\boldsymbol{g}$ and decide when to submit its answer. During RL, the *OET* and the *QDT* are frozen.

Formalization. We formalize *CLEVR Ask* as an MDP problem given the tuple (S, A, P, R, γ), where S is the set of states, A is the finite set of actions, P is the state transition function, R is the reward function, and γ is the discount factor. We define each state, action, and reward on the timestep t. The set of actions A corresponds to $\hat{g}_k$ produced by an action function F_{RL} defined in Eqs. (5, 6). The states of the game are defined as $S_t = (\mathcal{I}, \{(q^\tau, a^\tau)\}_{1:t-1})$. Agent choose actions until the number of questions reaches the pre-defined limit question count, T. Among the 3^k actions, we treat $A_t = 0$ or $A_t = 3^k - 1$ as end of dialogue (EOD) cases. If one of these actions is selected, the dialogue is considered finished and the object with the highest probability $P_{\hat{o}}$ at that time is submitted to the *Oracle* as the final prediction. The *Oracle* compares the *Questioner*'s prediction with the ground truth o^* and returns the reward. The model is trained with policy gradient optimization using REINFORCE algorithm [29].

Reward function. The basic reward for reinforcement learning is the zero-one task success reward r_c, similar to [28], which is given when the stop-action is produced and the predicted goal object is correct. Note well that submitting the answer at the first action step is treated as invalid and the success reward will not be given, so as to prevent the random predictions without asking questions. We also introduced a turn discount factor r_d, similar to the goal-achieved reward proposed by [35], which will give a discount to the success reward depending on the number of questions asked to reach the answer.

6. Supervised Learning

6.1. Settings

Datasets. We trained and evaluated our model with two different datasets: CLEVR Ask3 and CLEVR Ask4.

Implementation details. Supervised learning was early-stopped with 50-epoch patience using the AdaBelief optimizer [38] with the learning rate 1e-4, and 20 epochs for the warmup, and a batch size of 1024. Additional details are available in the appendix.

Metrics. In supervised learning, we evaluated the *OET* and the *QDT* with three metrics: F1 score, perfect address ratio, and correct address ratio.

F1 score was used to evaluate if the *OET* can find out candidate objects that match a dialogue $\mathcal{D}$. The F1 score was computed from predicted candidate objects $\widehat{\mathcal{O}}_{\mathcal{D}}$ and ground-truth candidate objects $\mathcal{O}_{\mathcal{D}}$. We obtained $\widehat{\mathcal{O}}$ from the result of σ_i in Eq. (3) as $\widehat{\mathcal{O}} = \{o_i \mid i \in \mathcal{N} \wedge 0.5 < \sigma_i\}$.

Model	F1 score↑	Perfect Addr↑	Correct Addr↑
UniQer (Ask3)	0.994	57.67	86.91
UniQer (Ask4)	0.994	43.20	69.79

Table 1: The results for supervised training for both Ask3 and Ask4 datasets. F1 score measures the *OET*'s ability to find out the object candidates given a dialogue. Perfect Addr and Correct Addr stand for "perfect address ratio" and "correct address ratio", respectively, which measures the *QDT*'s ability to generate a question that correctly addresses the target objects instructed by a group vector $\boldsymbol{g}$.

Perfect address ratio and correct address ratio were used to evaluate the *QDT*'s ability to generate questions that will distinguish the target objects from the distracter objects as instructed by a group vector $\boldsymbol{g}$. The group vector $\boldsymbol{g}$ groups objects $\mathcal{O}$ into three groups: the target object group $\mathcal{O}_{\mathrm{t}}$, the distracter object group $\mathcal{O}_{\mathrm{d}}$, and the masked object group $\mathcal{O}_{\mathrm{m}}$. If the generated question succeeds to distinguish $\mathcal{O}_{\mathrm{t}}$ from $\mathcal{O}_{\mathrm{d}}$, it is perfect. If it succeeds to do so but mixes $\mathcal{O}_{\mathrm{m}}$ with $\mathcal{O}_{\mathrm{t}}$, it is only deemed correct. The detailed definition of these metrics is available in the appendix section.

6.2. Results

The results of supervised learning are summarized in Tab. 1. For both Ask3 and Ask4 datasets, the F1 scores yield a near-perfect 0.994. The question generation achieved a fairly high probability, almost 87%, of generating correct questions in Ask3 and even in Ask4, which has an increased dataset complexity, as it is still able to generate nearly 70% of the questions correctly. The scores on perfect address were around 58% and 43% for Ask3 and Ask4 respectively. Although they were lower than the scores of correct address, the model still shows the capability of addressing the target objects considering the irrelevant masked objects.

7. Reinforcement Learning

7.1. Settings

Baseline model. We compared the proposed model with a model proposed in [28]. This baseline model consists of a Guesser-QGen architecture connected with a dialogue state encoder, where the Guesser is a multi-label classifier with a single weight-shared MLP and the QGen is a recurrent neural network. As in the previous studies, the QGen and Guesser are pre-trained separately and then tuned with reinforcement learning using policy gradient optimization. Details of the baseline model can be found in the appendix.

Implementation details. In the reinforcement learning, all experiments were trained for 150 epochs using the Adam optimizer [18] with the learning rate 5e-4, and a batch size of 1024. Additional details are available in the appendix.

	Ask3		Ask4	
Model	New Img↑	New Obj↑	New Img↑	New Obj↑
Baseline	$60.00_{\pm 6.35}$	$59.60_{\pm 6.87}$	$64.75_{\pm 0.82}$	$64.21_{\pm 0.34}$
Ours (v)	$72.98_{\pm 3.13}$	$72.88_{\pm 3.47}$	$67.38_{\pm 4.18}$	$67.01_{\pm 4.34}$
Ours (num)	$69.43_{\pm 2.75}$	$69.50_{\pm 2.99}$	$72.89_{\pm 5.95}$	$72.35_{\pm 5.94}$
Ours (nu)	$50.61_{\pm 6.51}$	$50.37_{\pm 6.02}$	$65.15_{\pm 3.33}$	$64.25_{\pm 3.01}$
Ours (full)	$\mathbf{84.10}_{\pm 4.41}$	$\mathbf{83.96}_{\pm 4.70}$	$\mathbf{83.47}_{\pm 1.25}$	$\mathbf{83.81}_{\pm 0.94}$

Table 2: Quantitative results on comparison and ablation study. The average and standard deviation of five runs for the task success ratio are shown. The bold numbers represent the best performance. The model "Ours (full)" represents our proposed UniQer and the other "Ours" models are ablated UniQer models; "Ours (v)" is the *Vanilla* model, "Ours (num)" is the *Not Unified MLP Guesser* model, and "Ours (nu)" is the *Not Unified* model.

Metrics and Conditions. Following [28], we used the task success ratio, defined as the rate of correct predictions submitted by a questioner, as our primary metric. As with the previous studies, the training was conducted in two different settings: new image and new object. We conducted five experimental runs across different seeds.

7.2. Results

The results on our full model are presented in Tab. 2. In the table, the average and the standard deviation of task success ratio is shown. The results demonstrate that *UniQer* outperformed the baseline with a large magnitude in both datasets and conditions, showing *UniQer*'s ability to discover a goal object in the task which requires descriptive question generation. In the new image setting, the task success rate of the baseline model was 60.00% and 64.75% for Ask3 and Ask4 datasets, respectively, while *UniQer* achieved 84.10% and 81.20%. Surprisingly, the results with the new object setting was on par with that of the new image setting, which is likely due to the fact that their objects share the same attributes.

7.3. Ablation Studies

Ablation studies were conducted to determine the effectiveness of *UniQer*. We performed testing using the following three ablation models:

- **Vanilla:** The simplest model, which utilizes bidirectional GRU as the object encoder and LSTM as the question decoder. The Guesser architecture, a MLP module, is trained with the LSTM dialogue history encoder. However, it is trained separately from the object encoder, the same as the baseline model.
- **Not Unified MLP Guesser:** The model that substitutes the object encoder and the question decoder in the Vanilla model with a standard transformer. Note that the object encoder in this model does not include

	Ask3		Ask4	
Model	New Img↑	New Obj↑	New Img↑	New Obj↑
Ours (r)	$1.73_{\pm 0.12}$	$1.64_{\pm 0.21}$	$1.61_{\pm 0.07}$	$1.72_{\pm 0.10}$
Ours (fsr)	$68.78_{\pm 0.27}$	$68.49_{\pm 0.38}$	$68.99_{\pm 0.37}$	$69.73_{\pm 0.45}$
Ours (k=4)	$81.51_{\pm 3.72}$	$82.27_{\pm 3.77}$	$\mathbf{83.47}_{\pm 1.25}$	$\mathbf{83.81}_{\pm 0.94}$
Ours (k=5)	$\mathbf{84.10}_{\pm 4.41}$	$\mathbf{83.96}_{\pm 4.70}$	$81.20_{\pm 4.37}$	$80.50_{\pm 4.86}$
Ours (k=6)	$83.20_{\pm 3.37}$	$83.82_{\pm 3.27}$	$80.84_{\pm 4.77}$	$81.18_{\pm 4.79}$
Ours (k=7)	$82.66_{\pm 3.44}$	$83.39_{\pm 3.12}$	$76.47_{\pm 4.90}$	$76.63_{\pm 4.57}$

Table 3: Ablated results on the Object Targeting Module (OTM). In this study we changed the number of k and substituted the OTM with a random action module. The *r* represents the random condition and the *fsr* represents the force-stop random condition, which forcefully submits the prediction at the end of the dialogue.

the Guesser architecture; as an alternative, the Guesser is implemented with a multi-layer perceptron.

- **Not Unified:** The model that substitutes the Guesser module with the transformer. This model is the disassembled version of *UniQer*, whose QGen and Guesser modules are built on a single transformer encoder-decoder architecture.

The ablated results are presented in Tab. 2. As shown in the table, all of the ablated conditions drop performances compared with *UniQer*, showing the advantage of our architectural design. Notably, the *Not Unified* model significantly decreases the performance among the others in both datasets. This indicates the effectiveness of unifying the Guesser and the QGen.

Additionally, we conducted an ablation to gain an understanding of the OTM. In the ablation, we substituted the OTM with a random action model, which chooses an action A_t randomly on every step. Besides, we tested this random model with force-stop condition, where the model automatically submits the answer at the end of a dialogue, which means the submission action is not required. We also investigated the hyper-parameter settings for top-k.

The ablated results on the OTM are shown in Tab. 3. The random condition scored the lowest as we expected, while the force-stop condition performs much better than it. This is because in the random condition, a chance of choosing a submission action is quite low. This indicates that learning when to submit the answer is vital function in the OTM.

We also found that there is a trade-off in the settings of k. By increasing the size of k, the OTM can consider a larger number of objects when choosing target objects. However, it becomes demanding for the QDT, since it needs to generate questions that will distinguish larger number of objects. Decreasing the size of k makes the model microscopic; it can only handle a small number of objects at a time, which makes it difficult to explore other candidate objects.

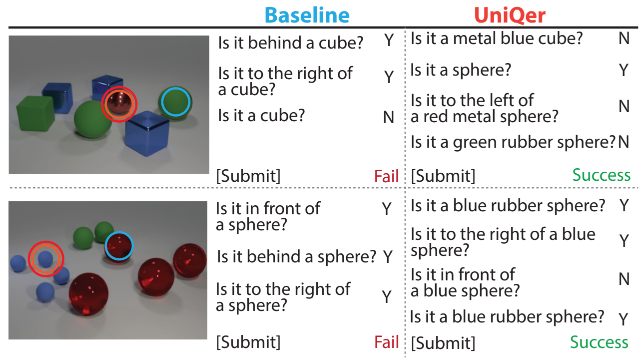

Figure 3: Samples of generated dialogues converted to natural language. The red circles indicate the goal objects, the orange ones the *UniQer* predictions, and the blue ones the baseline model predictions.

7.4. Qualitative Results

We also inspected some of the dialogue samples and found that *UniQer* was actually capable of effectively generating descriptive questions to find out a goal object. Fig. 3 shows the representative dialogue examples for the test set images. In the upper example, the goal object was the red metallic sphere. The baseline generated questions regarding "a cube", however none of the referring expressions were used for narrowing down a referenced object among the cubes. Because there are multiple cubes in the image, this question is considered to be a non-informative question. On the other hand, ours generated expressions such as "a metal blue cube" and "a red metal sphere". These questions are appropriate to distinguish the goal object among others, and considered to be informative.

In the lower example, the goal object was a small blue sphere located in a group of similar objects. The baseline generated referring expressions such as "in front of a sphere" and "behind a sphere". These questions could not distinguish the objects, since all the objects in the image were spheres. On the other hand, ours generated referring expressions such as "right of a blue sphere" and "in front of a blue sphere." Although these expressions are not very human-friendly, they are sufficient and short enough to specify the relative relationships among the blue spheres.

8. Conclusion

In this research, we presented *UniQer*, a novel *Questioner* architecture for descriptive question generation in goal-oriented visual dialogue. Experimental results demonstrated that *UniQer* surpasses the baseline on the *CLEVR Ask* datasets. We also validated the components of *UniQer* with ablation studies, showing the structural advantages of *UniQer*. Finally, we investigated the generated samples qualitatively and found that *UniQer* successfully generated descriptive questions and discovered the goal objects.

References

[1] E. Abbasnejad, I. Abbasnejad, Q. Wu, J. Shi, and A. v. d. Hengel. Gold seeker: Information gain from policy distributions for goal-oriented vision-and-langauge reasoning. In *CVPR*, pages 13450–13459, 2020. 2

[2] E. Abbasnejad, Q. Wu, Q. Shi, and A. v. d. Hengel. What's to know? uncertainty as a guide to asking goal-oriented questions. In *CVPR*, pages 4155–4164, 2019. 2

[3] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *ICCV*, pages 2425–2433, 2015. 2

[4] Y.-C. Chen, L. Li, L. Yu, A. El Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu. Uniter: Universal image-text representation learning. In *ECCV*, pages 104–120. Springer, 2020. 2

[5] B. O. Community. Blender–a 3d modelling and rendering package, 2016. 3

[6] M. Cornia, M. Stefanini, L. Baraldi, and R. Cucchiara. Meshed-memory transformer for image captioning. In *CVPR*, pages 10578–10587, 2020. 2

[7] R. Dale. Cooking up referring expressions. In *ACL*, pages 68–75, 1989. 1, 2

[8] R. Dale and E. Reiter. Computational interpretations of the gricean maxims in the generation of referring expressions. *Cognitive science*, 19(2):233–263, 1995. 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. 2

[10] H. De Vries, F. Strub, S. Chandar, O. Pietquin, H. Larochelle, and A. Courville. Guesswhat?! visual object discovery through multi-modal dialogue. In *CVPR*, pages 5503–5512, 2017. 1, 2, 3

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, 2019. 2, 5

[12] 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*, pages 1473–1482, 2015. 2

[13] D. A. Hudson and C. D. Manning. Compositional attention networks for machine reasoning. In *ICLR*, 2018. 2

[14] N. Iwahashi, K. Sugiura, R. Taguchi, T. Nagai, and T. Taniguchi. Robots that learn to communicate: A developmental approach to personally and physically situated human-robot conversations. In *AAAI*, 2010. 1

[15] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, pages 2901–2910, 2017. 2

[16] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei. Image retrieval using scene graphs. In *CVPR*, pages 3668–3678, 2015. 3

[17] T. Kanda and H. Ishiguro. *Human-robot interaction in social robotics*. CRC Press, 2017. 1

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

[19] R. Liu, C. Liu, Y. Bai, and A. L. Yuille. Clevr-ref+: Diagnosing visual reasoning with referring expressions. In *CVPR*, pages 4185–4194, 2019. 2

[20] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 2

[21] R. Luo and G. Shakhnarovich. Comprehension-guided referring expressions. In *CVPR*, pages 7102–7111, 2017. 2

[22] W. Pang and X. Wang. Visual dialogue state tracking for question generation. In *AAAI*, pages 11831–11838, 2020. 2

[23] D. Qi, L. Su, J. Song, E. Cui, T. Bharti, and A. Sacheti. Imagebert: Cross-modal pre-training with large-scale weak-supervised image-text data. *arXiv preprint arXiv:2001.07966*, 2020. 2

[24] S. Satake, T. Kanda, D. F. Glas, M. Imai, H. Ishiguro, and N. Hagita. How to approach humans? strategies for social robots to initiate interaction. In *HRI*, pages 109–116, 2009. 1

[25] I. Schwartz, S. Yu, T. Hazan, and A. G. Schwing. Factor graph attention. In *CVPR*, pages 2039–2048, 2019. 2

[26] R. Shekhar, A. Venkatesh, T. Baumgärtner, E. Bruni, B. Plank, R. Bernardi, and R. Fernández. Beyond task success: A closer look at jointly learning to see, ask, and guesswhat. In *NAACL-HLT*, pages 2578–2587, 2019. 2, 3

[27] P. Shukla, C. Elmadjian, R. Sharan, V. Kulkarni, M. Turk, and W. Y. Wang. What should i ask? using conversationally informative rewards for goal-oriented visual dialog. In *ACL*, pages 6442–6451, 2019. 2

[28] F. Strub, H. De Vries, J. Mary, B. Piot, A. Courvile, and O. Pietquin. End-to-end optimization of goal-driven and visually grounded dialogue systems. In *IJCAI*, pages 2765–2771, 2017. 1, 2, 6, 7

[29] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, pages 1057–1063, 2000. 6

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017. 2, 5, 6

[31] T. Winograd. Understanding natural language. *Cognitive psychology*, 3(1):1–191, 1972. 1, 2

[32] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*, pages 5753–5763, 2019. 2

[33] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg. Modeling context in referring expressions. In *ECCV*, pages 69–85. Springer, 2016. 2

[34] L. Yu, H. Tan, M. Bansal, and T. L. Berg. A joint speaker-listener-reinforcer model for referring expressions. In *CVPR*, pages 7282–7290, 2017. 4

[35] 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*, pages 186–201, 2018. 2, 6

[36] S. Zhang, L. Qu, S. You, Z. Yang, and J. Zhang. Automatic generation of grounded visual questions. In *IJCAI*, pages 4235–4243, 2017. 2

[37] R. Zhao and V. Tresp. Improving goal-oriented visual dialog agents via advanced recurrent nets with tempered policy gradient. In *IJCAI*, 2018. 2

[38] J. Zhuang, T. Tang, Y. Ding, S. C. Tatikonda, N. Dvornek, X. Papademetris, and J. Duncan. Adabelief optimizer: Adapting stepsizes by the belief in observed gradients. *NeurIPS*, 33, 2020. 6

[39] X. Zuo, N. Iwahashi, K. Funakoshi, M. Nakano, R. Taguchi, S. Matsuda, K. Sugiura, and N. Oka. Detecting robot-directed speech by situated understanding in physical interaction. *Information and Media Technologies*, 5(4):1314–1326, 2010. 1