Goal-Oriented Visual Question Generation via Intermediate Rewards

Junjie Zhang[1,3] ⋆[0000−0002−0033−0494], Qi Wu[2] ⋆⋆[0000−0003−3631−256X], Chunhua Shen[2][0000−0002−8648−8718], Jian Zhang[1][0000−0002−7240−3541], Jianfeng Lu[3][0000−0002−9190−507X], and Anton van den Hengel[2][0000−0003−3027−8364]

[1]School of Electrical & Data Engineering, University of Technology Sydney, Australia
[2]Australian Insititute for Machine Learning, The University of Adelaide, Australia
[3]School of Computer Science & Engineering, Nanjing University of Science & Technology, China
{junjie.zhang@student., jian.zhang@}uts.edu.au lujf@njust.edu.cn
{qi.wu01, chunhua.shen, anton.vandenhengel}@adelaide.edu.au

Abstract. Despite significant progress in a variety of vision-and-language problems, developing a method capable of asking intelligent, goal-oriented questions about images is proven to be an inscrutable challenge. Towards this end, we propose a Deep Reinforcement Learning framework based on three new intermediate rewards, namely *goal-achieved*, *progressive* and *informativeness* that encourage the generation of succinct questions, which in turn uncover valuable information towards the overall goal. By directly optimizing for questions that work quickly towards fulfilling the overall goal, we avoid the tendency of existing methods to generate long series of inane queries that add little value. We evaluate our model on the GuessWhat?! dataset and show that the resulting questions can help a standard 'Guesser' identify a specific object in an image at a much higher success rate.

Keywords: Goal-Oriented · VQG · Intermediate Rewards

1 Introduction

Although visual question answering (VQA) [2, 23, 24] has attracted more attention, visual question generation (VQG) is a much more difficult task. Obviously, generating facile, repetitive questions represents no challenge at all, but generating a series of questions that draw out useful information towards an overarching goal, however, demands consideration of the image content, the goal, and the conversation thus far. It could, generally, also be seen as requiring consideration of the abilities and motivation of the other participant in the conversation.

A well-posed question extracts the most informative answer towards achieving a particular goal, and thus reflects the knowledge of the asker, and their

⋆ The work was done while visiting The University of Adelaide.
⋆⋆ The first two authors contributed to this work equally.

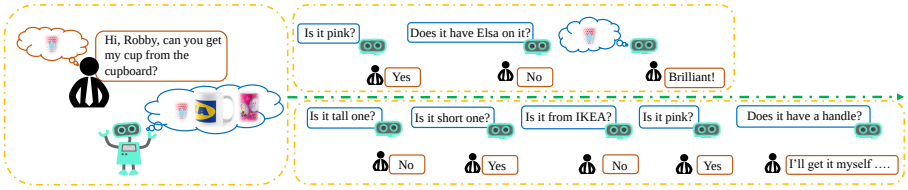

Fig. 1: Two illustrative examples of potential conversations between a human and a robot. The bottom conversation clearly makes people frustrated while the top one makes people happy because the robot achieves the goal in a quicker way via less but informative questions.

estimate of the capabilities of the answerer. Although the information would be beneficial in identifying a particular object in an image, there is little value in an agent asking a human about the exact values of particular pixels, the statistics of their gradients, or the aspect ratio of the corresponding bounding box. The fact that the answerer is incapable of providing the requested information makes such questions pointless. Selecting a question that has a significant probability of generating an answer that helps achieve a particular goal is a complex problem.

Asking questions is an essential part of the human communication. Any intelligent agent that seeks to interact flexibly and effectively with humans thus needs to be able to ask questions. The ability to ask intelligent questions is even more important than receiving intelligent, actionable answers. A robot, for example in Fig. 1, has been given a task and realized that it is missing critical information required to carry it out, needs to ask a question. It will have a limited number of attempts before the human gets frustrated and carries out the task themselves. This scenario applies equally to any intelligent agent that seeks to interact with humans, as we have surprisingly little tolerance for agents that are unable to learn by asking questions, and for those that ask too many.

As a result of the above, VQG has started to receive attention, but primarily as a vision-to-language problem [10, 13, 25]. Methods that approach the problem in this manner tend to generate arbitrary sequences of questions that are somewhat related to the image [14], but which bare no relationship to the goal. This reflects the fact that these methods have no means of measuring whether the answers generated to assist in making progress towards the goal. Instead, in this paper, we ground the VQG problem as a goal-oriented version of the game - GuessWhat?!, introduced in [22]. The method presented in [22] to play the GuessWhat game is made up of three components: the `Questioner` asks questions to the `Oracle`, and the `Guesser` tries to identify the object that the `Oracle` is referring to, based on its answers. The quality of the generated questions is thus directly related to the success rate of the final task.

Goal-oriented training that uses a game setting has been used in the visual dialog generation previously [4]. However, it focuses on generating more human-like dialogs, not on helping the agent achieve the goal through better question generation. Moreover, previous work [18] only uses the final goal as the reward

to train the dialog generator, which might be suitable for dialog generation but is a rather weak and undirected signal by which to control the quality, effectiveness, and informativeness of the generated question in a goal-oriented task. In other words, in some cases, we want to talk to a robot because we want it to finish a specific task but not to hold the meaningless boring chat. Therefore, in this paper, we use intermediate rewards to encourage the agent to ask short but informative questions to achieve the goal. Moreover, in contrast to previous works that only consider the overall goal as the reward, we assign different intermediate rewards for each posed question to control the quality.

This is achieved through fitting the goal-oriented VQG into a reinforcement learning (RL) paradigm and devising three different intermediate rewards, which are our main contributions in this paper, to explicitly optimize the question generation. The first *goal-achieved* reward is designed to encourage the agent to achieve the final goal (pick out the object that the `Oracle` is 'thinking') via asking multiple questions. However, different from only considering whether the goal is achieved, additional rewards are awarded if the agent can use fewer questions to achieve it. This is a reasonable setting because you do not need a robot that can finish a task but has to ask you hundreds of questions. The second reward we proposed is the *progressive* reward, which is established to encourage questions that generated by the agent can progressively increase the probability of the right answer. This is an intermediate reward for the individual question, and the reward is decided by the change of the ground-truth answer probability. A negative reward will be given if the probability decreases. The last reward is the *informativeness* reward, which is used to restrict the agent not to ask 'useless' questions, for example, a question that leads to the identical answer for all the candidate objects (this question cannot eliminate any ambiguous). We show the whole framework in Fig. 2.

We evaluate our model on the GuessWhat?! dataset [22], with the pre-trained standard `Oracle` and `Guesser`, we show that our novel `Questioner` model outperforms the baseline and state-of-the-art model by a large margin. We also evaluate each reward respectively, to measure the individual contribution. Qualitative results show that we can produce more informative questions.

2 Related Works

Visual Question Generation Recently, the visual question generation problem has been brought to the computer vision community, aims at generating visual-related questions. Most of the works treat the VQG as a standalone problem and follow an image captioning style framework, *i.e.*, translate an image into a sentence, in this case, a question. For example, in [13], Mora *et al.* use a CNN-LSTM model to generate questions and answers directly from the image visual content. Zhang *et al.* [25] focus on generating questions of grounded images. They use Densecap [8] as region captioning generator to guide the question generation. In [14], Mostafazadeh *et al.* propose a dataset to generate natural questions about images, which are beyond the literal description of image content. Li *et al.* [10]

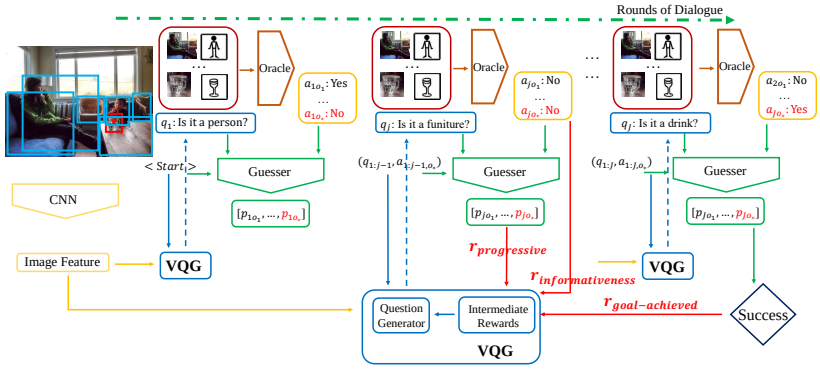

Fig. 2: The framework of the proposed VQG agent plays in the whole game environment. A target object o^* is assigned to the Oracle, but it is unknown to VQG and Guesser. Then VQG generates a series of questions, which are answered by Oracle. During training, we let Oracle answer the question based on all the objects at each round, and measure the *informativeness* reward, and we also let Guesser generate probability distribution to measure the *progressive* reward. Finally, we consider the number of rounds J and set the *goal-achieved* reward based on the status of success. These intermediate rewards are adopted for optimizing the VQG agent by the REINFORCE.

view the VQA and VQG as a dual learning process by jointly training them in an end-to-end framework. Although these works can generate meaningful questions that are related to the image, the motivation of asking these questions are rather weak since they are not related to any goals. Another issue of the previous works is that it is hard to conduct the quality measurement on this type of questions. Instead, in our work, we aim to develop an agent that can learn to ask realistic questions, which can contribute to achieving a specific goal.

Goal-Oriented Visual Dialogue generation has attracted many attentions at most recently. In [5], Das *et al.* introduce a reinforcement learning mechanism for visual dialogue generation. They establish two RL agents corresponding to question and answer generation respectively, to finally locate an unseen image from a set of images. The question agent predicts the feature representation of the image and the reward function is given by measuring how close the representation is compared to the true feature. However, we focus on encouraging the agent to generate questions that directed towards the final goal, and we adopt different kinds of intermediate rewards to achieve that in the question generation process. Moreover, the question generation agent in their model only asks questions based on the dialogue history, which does not involve visual information. In [18], Florian *et al.* propose to employ reinforcement learning to solve question generation of the GuessWhat game by introducing the final status of success as the sole reward. We share the similar backbone idea, but there are several technical differences. One of the most significant differences is that the previous work only considers using whether achieving the final goal as the reward but we assign different

intermediate rewards for each posed question to push VQG agent to ask short but informative questions to achieve the goal. The experimental results and analysis in Section 4 show that our model not only outperforms the state-of-art but also achieves higher intelligence, *ie.*, using as few questions as possible to finish the task.

Reinforcement Learning for V2L Reinforcement learning [9, 20] has been adopted in several vision-to-language (V2L) problems, including image captioning [11, 16, 17], VQA [1, 7, 26], and aforementioned visual dialogue system [5, 12] *etc.* In [16], Ren *et al.* use a policy network and a value network to collaboratively generate image captions, while different optimization methods for RL in image captioning are explored in [11] and [17], called SPIDEr and self-critical sequence training. Zhu *et al.* [26] introduce knowledge source into the iterative VQA and employ RL to learn the query policy. In [1], authors use RL to learn the parameters of QA model for both images and structured knowledge bases. These works solve V2L related problems by employing RL as an optimization method, while we focus on using RL with carefully designed intermediate rewards to train the VQG agent for goal-oriented tasks.

Reward Shaping Our work is also somewhat related to the reward shaping, which focuses on solving the sparsity of the reward function in the reinforcement learning. In [19], Su *et al.* examine three RNN based approaches as potential functions for reward shaping in spoken dialogue systems. In [6], El Asri *et al.* propose two diffuse reward functions to apply to the spoken dialogue system by evaluating the states and transitions respectively. Different from these prior works that condition their model on discourse-based constraints for a purely linguistic (rather than visuo-linguistic) dataset. The tasks we target, our architectural differences, and the dataset and metrics we employ are distinct.

3 Goal-Oriented VQG

We ground our goal-oriented VQG problem on a *Guess What* game, specifically, on the GuessWhat?! dataset [22]. GuessWhat?! is a three-role interactive game, where all roles observe the same image of a rich visual scene that contains multiple objects. We view this game as three parts: `Oracle`, `Questioner` and `Guesser`. In each game, a random object in the scene is assigned to the `Oracle`, where this process is hidden to the `Questioner`. Then the `Questioner` can ask a series of yes/no questions to locate this object. The list of objects is also hidden to the `Questioner` during the question-answer rounds. Once the `Questioner` has gathered enough information, the `Guesser` can start to guess. The game is considered as successful if the `Guesser` selects the right object.

The `Questioner` part of the game is a goal-oriented VQG problem, each question is generated based on the visual information of the image and the previous rounds of question-answer pairs. The goal of VQG is to successfully finish the game, in this case, to locate the right object. In this paper, we fit the goal-oriented VQG into a reinforcement learning paradigm and propose three different

intermediate rewards, namely the *goal-achieved* reward, *progressive* reward, and *informativeness* reward, to explicitly optimize the question generation. The *goal-achieved* reward is established to lead the dialogue to achieve the final goal, the *progressive* reward is used to push the intermediate generation process towards the optimal direction, while the *informativeness* reward is used to ensure the quality of generated questions. To better express the generation process, we first introduce the notations of GuessWhat?! game.

Each game is defined as a tuple (I, D, O, o^*), where I is the observed image, D is the dialogue with J rounds of question-answer pairs $(q_j, a_j)_{j=1}^{J}$, $O = (o_n)_{n=1}^{N}$ is the list of N objects in the image I, where o^* is the target object. Each question $q_j = (w_m^j)_{m=1}^{M_j}$ is a sequence of M_j tokens, which are sampled from the predefined vocabulary V. The V is composed of word tokens, a question stop token <?> and a dialogue stop token <End>. The answer $a_j \in$ {<Yes>,<No>,<NA>} is set to be yes, no or not applicable. For each object o, it has an object category $c_o \in \{1 \dots C\}$ and a segment mask.

3.1 Learning Environment

We build the learning environment to generate visual dialogues based on the GuessWhat?! dataset. Since we focus on the goal-oriented VQG, for a fair comparison, the `Oracle` and `Guesser` are produced by referring to the original baseline models in GuessWhat?! [22]. We also introduce the VQG supervised learning model, which is referred as the baseline for the rest of the paper.

The `Oracle` requires generating answers for all kinds of questions about any objects within the image scene. The bounding box (obtained from the segment mask) of the object o are encoded to represent the spatial feature, where $o_{spa} = [x_{min}, y_{min}, x_{max}, y_{max}, x_{center}, y_{center}, w, h]$ indicates the box coordinates, width and height. The category c_o is embedded using a learned look-up table, while the current question is encoded by an LSTM. All three features are concatenated into a single vector and fed into a one hidden layer MLP followed by a softmax layer to produce the answer probability $p(a|o_{spa}, c_o, q)$.

Given an image I and a series of question-answer pairs, the `Guesser` requires predicting right object o^* from a list of objects. We consider the generated dialogue as one flat sequence of tokens and encode it with an LSTM. The last hidden state is extracted as the feature to represent the dialogue. We also embed all the objects' spatial features and categories by an MLP. We perform a dot-product between dialogue and object features with a softmax operation to produce the final prediction.

Given an image I and a history of the question-answer pairs $(q, a)_{1:j-1}$, the VQG requires generating a new question q_j. We build the VQG baseline based on an RNN generator. The RNN recurrently produces a series of state vectors $s_{1:m}^j$ by transitioning from the previous state s_{m-1}^j and the current input token w_m^j. We use an LSTM as the transition function f, that is, $s_m^j = f(s_{m-1}^j, w_m^j)$. In our case, the state vector s is conditioned on the whole image and all the previous question-answer tokens. We add a softmax operation to produce the probabil-

ity distribution over the vocabulary V, where $p(w_m^j|I,(q,a)_{1:j-1},w_{1:m-1}^j)$. This baseline is conducted by employing the supervised training. We train the VQG by minimizing the following negative log loss function:

$$\begin{aligned} L &= -\log p(q_{1:J}|I,a_{1:J}) \\ &= -\sum_{j=1}^{J}\sum_{m=1}^{M_j}\log p(w_m^j|I,w_{1:m-1}^j,(q,a)_{1:j-1}) \end{aligned} \tag{1}$$

During the test stage, the question can be sampled from the model by starting from state s_1^j; a new token w_m^j is sampled from the probability distribution, then embedded and fed back to the LSTM. We repeat this operation until the end of question token is encountered.

3.2 Reinforcement Learning of VQG

We use our established `Oracle`, `Guesser` and VQG baseline model to simulate a complete GuessWhat?! game. Given an image I, an initial question q_1 is generated by sampling from the VQG baseline until the stop question token is encountered. Then the `Oracle` receives the question q_1 along with the assigned object category o^* and its spatial information o_{spa}^*, and output the answer a_1, the question-answer pair (q_1,a_1) is appended to the dialogue history. We repeat this loop until the end of the dialogue token is sampled, or the number of questions reaches the maximum. Finally, the `Guesser` takes the whole dialogue D and the object list O as inputs to predict the object. We consider the goal reached if o^* is selected. Otherwise, it failed.

To more efficiently optimize the VQG towards the final goal and generate informative questions, we adopt three intermediate rewards (which will be introduced in the following sections) into the RL framework.

`State, Action & Policy` We view the VQG as a Markov Decision Process (MDP), the `Questioner` is noted as the agent. For the dialogue generated based on the image I at time step t, the state of agent is defined as the image visual content with the history of question-answer pairs and the tokens of current question generated so far: $S_t=(I,(q,a)_{1:j-1},(w_1^j,\ldots,w_m^j))$, where $t=\sum_{k=1}^{k=j-1}M_k+m$. The action A_t of agent is to select the next output token w_{m+1}^j from the vocabulary V. Depending on the actions that agent takes, the transition between two states falls into one of the following cases:

1) w_{m+1}^j =<?>: The current question is finished, the `Oracle` from the environment will answer a_j, which is appended to the dialogue history. The next state $S_{t+1}=(I,(q,a)_{1:j})$.

2) w_{m+1}^j =<End>: The dialogue is finished, the `Guesser` from the environment will select the object from the list O.

3) Otherwise, the new generated token w_{m+1}^j keeps appending to the current question q_j, the next state $S_{t+1}=(I,(q,a)_{1:j-1},(w_1^j,\ldots,w_m^j,w_{m+1}^j))$.

The maximum length of question q_j is M_{max}, and the maximum rounds of the dialogue is J_{max}. Therefore, the number of time steps T of any dialogue are $T\leq M_{max}*J_{max}$. We model the VQG under the stochastic policy $\pi_\theta(A|S)$,

where θ represents the parameters of the deep neural network we used in the VQG baseline that produces the probability distributions for each state. The goal of the policy learning is to estimate the parameter θ.

After we set up the components of MDP, the most significant aspect of the RL is to define the appropriate reward function for each state-action pair (S_t, A_t). As we emphasized before, the goal-oriented VQG aims to generate the questions that lead to achieving the final goal. Therefore, we build three kinds of intermediate rewards to push the VQG agent to be optimized towards the optimal direction. The whole framework is shown in Fig. 2.

`Goal-Achieved Reward` One basic rule of the appropriate reward function is that it cannot conflict with the final optimal policy [15]. The primary purpose of the VQG agent is to gather enough information as soon as possible to help `Guesser` to locate the object. Therefore, we define the first reward to reflect whether the final goal is achieved. But more importantly, we take the number of rounds into consideration to accelerate the questioning part and let the reward be nonzero when the game is successful.

Given the state S_t, where the <End> token is sampled or the maximum round J_{max} is reached, the reward of the state-action pair is defined as:

$$r_g(S_t, A_t) = \begin{cases} 1+\lambda \cdot J_{max}/J, & \text{If } \texttt{Guesser}(S_t) = o^* \\ 0, & \text{Otherwise} \end{cases} \tag{2}$$

We set the reward as one plus the weighted maximum number of rounds J_{max} against the actual rounds J of the current dialogue if the dialogue is successful, and zero otherwise. This is based on that we want the final goal to motivate the agent to generate useful questions. The intermediate process is considered into the reward as the rounds of the question-answer pairs J, which guarantees the efficiency of the generation process; the fewer questions are generated, the more reward VQG agent can get at the end of the game (if and only if the game succeed). This is a quite useful setting in the realistic because we do want to use fewer orders to guide the robot to finish more tasks. λ is a weight to balance between the contribution of the successful reward and the dialogue round reward.

`Progressive Reward` Based on the observation of the human interactive dialogues, we find that the questions of a successful game, are ones that progressively achieve the final goal, *i.e.* as long as the questions being asked and answered, the confidence of referring to the target object becomes higher and higher. Therefore, at each round, we define an intermediate reward for state-action pair as the improvement of target probability that `Guesser` outputs. More specific, we interact with the `Guesser` at each round to obtain the probability of predicting the target object. If the probability increases, it means that the generated question q_j is a positive question that leads the dialogue towards the right direction.

We set an intermediate reward called *progressive* reward to encourage VQG agent to progressively generate these positive questions. At each round j, we record the probability $p_j(o^*|I, (q, a)_{1:j})$ returned by `Guesser`, and compare it with the last round $j-1$. The difference between the two probabilities is used

as the intermediate reward. That is:

$$r_p(S_t, A_t) = p_j(o^*|I, (q,a)_{1:j}) - p_{j-1}(o^*|I, (q,a)_{1:j-1}) \tag{3}$$

Despite the total reward summed over all time steps are the initial and final states due to the cancellation of intermediate terms, during the REINFORCE optimization, the state-action value function that returns the cumulative rewards of each step are different. In this way, the question is considered high-quality and has a positive reward, if it leads to a higher probability to guess the right object. Otherwise, the reward is negative.

`Informativeness Reward` When we human ask questions (especially in a guess what game), we expect an answer that can help us to eliminate the confusion and distinguish the candidate objects. Hence, imagine that if a posed question that leads to the same answer for all the candidate object, this question will be useless. For example, all the candidate objects are 'red' and if we posed a question that 'Is it red?', we will get the answer 'Yes.' However, this question-answer pair cannot help us to identify the target. We want to avoid this kind of questions because they are non-informative. In this case, we need to evaluate the question based on the answer from the `Oracle`.

Given generated question q_j, we interact with the `Oracle` to answer the question. Since the `Oracle` takes the image I, the current question q_j, and the target object o^* as inputs, and outputs the answer a_j, we let the `Oracle` answer question q_j for all objects in the image. If more than one answer is different from others, we consider q_j is useful for locating the right object. Otherwise, it does not contribute to the final goal. Therefore, we set the reward positive, which we called *informativeness* reward, for these useful questions.

Formally, during each round, the `Oracle` receives the image I, the current question q_j and the list of objects O, and then outputs the answer set $a_{jO} = \{a_{jo_1}, \ldots, a_{jo_N}\}$, where each element corresponds to each object. Then the *informativeness* reward is defined as:

$$r_i(S_t, A_t) = \begin{cases} \eta, & \text{If all } a_{jo_n} \text{ are not identical} \\ 0, & \text{Otherwise} \end{cases} \tag{4}$$

By giving a positive reward to the state-action pair, we improve the quality of the dialogue by encouraging the agent to generate more informative questions.

`Training with Policy Gradient` Now we have three different kinds of rewards that take the intermediate process into consideration, for each state-action pair (S_t, A_t), we add three rewards together as the final reward function:

$$r(S_t, A_t) = r_g(S_t, A_t) + r_p(S_t, A_t) + r_i(S_t, A_t) \tag{5}$$

Considering the large action space in the game setting, we adopt the policy gradient method [21] to train the VQG agent with proposed intermediate rewards. The goal of policy gradient is to update policy parameters with respect to the expected return by gradient descent. Since we are in the episodic environment, given the policy π_θ, which is the generative network of the VQG agent, in this case, the policy objective function takes the form:

$$J(\theta) = E_{\pi_\theta}[\sum_{t=1}^{T} r(S_t, A_t)] \tag{6}$$

Algorithm 1 Training procedure of the VQG agent.

Input: `Oracle`(Ora), `Guesser`(Gus), VQG, batch size H

for Each update **do**
  # Generate episodes τ
  for $h = 1$ to H **do**
    select image I_h and one target object $o_h^* \in O_h$
    # Generate question-answer pairs $(q, a)_{1:j}^h$
    for $j = 1$ to J_{max} **do**
      $q_j^h = VQG(I_h, (q, a)_{1:j-1}^h)$
      # N is the number of total objects
      for $n = 1$ to N **do**
        $a_{jo_{hn}}^h = Ora(I_h, q_j^h, o_{hn})$
      if all $a_{jo_{hn}}^h$ are not identical **then**
        $r_i(S_t, A_t) = \eta$
      else $r_i(S_t, A_t) = 0$
      $r(S_t, A_t) = r_i(S_t, A_t)$
      $p_j(o_h^*|\cdot) = Gus(I_h, (q, a)_{1:j}^h, O_h)$
      if $j > 1$ **then**
        $r_p(S_t, A_t) = p_j(o_h^*|\cdot) - p_{j-1}(o_h^*|\cdot)$
        $r(S_t, A_t) = r(S_t, A_t) + r_p(S_t, A_t)$
      if <End>$\in q_j^h$ **then**
        break;
    $p(o^h|\cdot) = Gus(I_h, (q, a)_{1:j}^h, O_h)$
    if $argmax_{o_h} p(o^h|\cdot) = o_h^*$ **then**
      $r_g(S_t, A_t) = 1 + \lambda \cdot J_{max}/j$
    else $r_g(S_t, A_t) = 0$
    $r(S_t, A_t) = r(S_t, A_t) + r_g(S_t, A_t)$
  Define $\tau = (I_h, (q, a)_{1:j_h}^h, r_h)_{1:H}$
  Evaluate $\nabla J(\theta)$ as Eq. 9 and update VQG agent
  Evaluate $\nabla L(\varphi)$ as Eq. 10 and update b_φ baseline

The parameters θ then can be optimized by following the gradient update rule. In REINFORCE algorithm [9], the gradient of $J(\theta)$ can be estimated from a batch of episodes τ that are sampled from the policy π_θ:

$$\nabla J(\theta) \approx \left\langle \sum_{t=1}^{T} \sum_{A_t \in V} \nabla_\theta \log \pi_\theta(S_t, A_t)(Q^{\pi_\theta}(S_t, A_t) - b_\varphi) \right\rangle_\tau \tag{7}$$

where $Q^{\pi_\theta}(S_t, A_t)$ is the state-action value function that returns the expectation of cumulative reward at (S_t, A_t):

$$Q^{\pi_\theta}(S_t, A_t) = E_{\pi_\theta}[\sum_{t'=t}^{T} r(S_{t'}, A_{t'})] \tag{8}$$

by substituting the notations with VQG agent, we have the following policy gradient:

$$\begin{aligned} \nabla J(\theta) \approx \Big\langle \sum_{j=1}^{J} \sum_{m=1}^{M_j} & \nabla_\theta \log \pi_\theta(w_m^j | I, (q, a)_{1:j-1}, w_{1:m-1}^j) \\ & (Q^{\pi_\theta}(I, (q, a)_{1:j-1}, w_{1:m-1}^j, w_m^j) - b_\varphi) \Big\rangle_\tau \end{aligned} \tag{9}$$

b_φ is a baseline function to help reduce the gradient variance, which can be chosen arbitrarily. We use a one-layer MLP that takes state S_t as input in VQG

agent and outputs the expected reward. The baseline b_φ is trained with mean squared error as:

$$\min_{\varphi} L(\varphi) = \left\langle [b_\varphi(S_t) - \sum_{t'=t}^{T} r(S_{t'}, A_{t'})]^2 \right\rangle_\tau \tag{10}$$

The whole training procedure is shown in Alg.1.

4 Experiment

In this section, we present our VQG results and conduct comprehensive ablation analysis about each intermediate reward. As mentioned above, the proposed method is evaluated on the GuessWhat?! game dataset [22] with pre-trained standard `Oracle` and `Guesser`. By comparing with the baseline and the state-of-the-art model, we show that the proposed model can efficiently generate informative questions, which serve the final goal.

4.1 Dataset & Evaluation Metric

The GuessWhat?! Dataset [22] is composed of 155,281 dialogues grounded on the 66,537 images with 134,074 unique objects. There are 821,955 question-answer pairs in the dialogues with vocabulary size 4,900. We use the standard split of training, validation and test in [22, 18]. Following [18], we report the accuracies of the games as the evaluation metric. Given a J-round dialogue, if the target object o^* is located by `Guesser`, the game is noted as successful, which indicates that the VQG agent has generated the qualified questions to serve the final goal. There are two kinds of test runs on the training set and test set respectively, named `NewObject` and `NewImage`. NewObject is randomly sampling target objects from the training images (but we restrict only to use new objects that are not seen before), while NewImage is sampling objects from the test images (unseen). We report three inference methods namely sampling, greedy and beam-search (beam size is 5) for these two test runs.

4.2 Implementation Details

The standard `Oracle`, `Guesser` and VQG baseline are reproduced by referring to [18]. The error of trained `Oracle`, `Guesser` on test set are 21.1% and 35.8% respectively. The VQG baseline is referred as Baseline in Tab.1. [1]

We initialize the training environment with the standard `Oracle`, `Guesser` and VQG baseline, then start to train the VQG agent with proposed reward functions. We train our models for 100 epochs with stochastic gradient descent (SGD) [3]. The learning rate and batch size are 0.001 and 64, respectively. The baseline function b_φ is trained with SGD at the same time. During each epoch, each training image is sampled once, and one of the objects inside it is randomly assigned as the target. We set the maximum round $J_{max} = 5$ and maximum

[1] These results are reported on https://github.com/GuessWhatGame by original authors.

Table 1: Results on training images (NewObject) and test images (NewImage).

Method	NewObject			NewImage		
	Sampling	Greedy	Beam-Search	Sampling	Greedy	Beam-Search
Baseline [22]	41.6	43.5	47.1	39.2	40.8	44.6
Sole-r [18]	58.5	60.3	60.2	56.5	58.4	58.4
VQG-r_g	60.6	61.7	61.4	58.2	59.3	59.4
VQG-r_g+r_p	62.1	62.9	63.1	59.3	60.6	60.5
VQG-r_g+r_i	61.3	62.4	62.7	58.5	59.7	60.1
VQG-r_g+r_p+r_i	63.2	63.6	63.9	59.8	60.7	60.8

length of question $M_{max} = 12$. The weight of the dialog round reward is set to $\lambda = 0.1$. The progressive reward is set as $\eta = 0.1$[2].

4.3 Results & Ablation Analysis

In this section, we give the overall analysis on proposed intermediate reward functions. To better show the effectiveness of each reward, we conduct comprehensive ablation studies. Moreover, we also carry out a human interpretability study to evaluate whether human subjects can understand the generated questions and how well the human can use these question-answer pairs to achieve the final goal. We note VQG agent trained with *goal-achieved* reward as VQG-r_g, trained with *goal-achieved* and *progressive* rewards as VQG-r_g+r_p, trained with *goal-achieved* and *informativeness* rewards as VQG-r_g+r_i. The final agent trained with all three rewards is noted as VQG-r_g+r_p+r_i.

`Overall Analysis` Tab. 1 show the comparisons between VQG agent optimized by proposed intermediate rewards and the state-of-the-art model proposed in [18] noted as Sole-r, which uses indicator of whether reaching the final goal as the sole reward function. As we can see, with proposed intermediate rewards and their combinations, our VQG agents outperform both compared models on all evaluation metrics. More specifically, our final VQG-r_g+r_p+r_i agent surpasses the Sole-r 4.7%, 3.3% and 3.7% accuracy on NewObject sampling, greedy and beam-search respectively, while obtains 3.3%, 2.3% and 2.4% higher accuracy on NewImage sampling, greedy and beam-search respectively. Moreover, all of our agents outperform the supervised baseline by a significant margin.

To fully show the effectiveness of our proposed intermediate rewards, we train three VQG agents using r_g, r_g+r_p, and r_g+r_i rewards respectively, and conduct ablation analysis. As we can see, the VQG-r_g already outperforms both the baseline and the state-of-the-art model, which means that controlling dialogue round can push the agent to ask more wise questions. With the combination of r_p and r_i reward respectively, the performance of VQG agent further improved. We find that the improvement gained from r_p reward is higher than r_i reward, which suggests that the intermediate *progressive* reward contributes more in our experiment. Our final agent combines all rewards and achieves the best results. Fig. 3 shows some qualitative results. More results can be found in the supplementary material, including some fail cases.

[2] We use a grid search to select the hyper-parameters λ and η, we find 0.1 produces the best results.

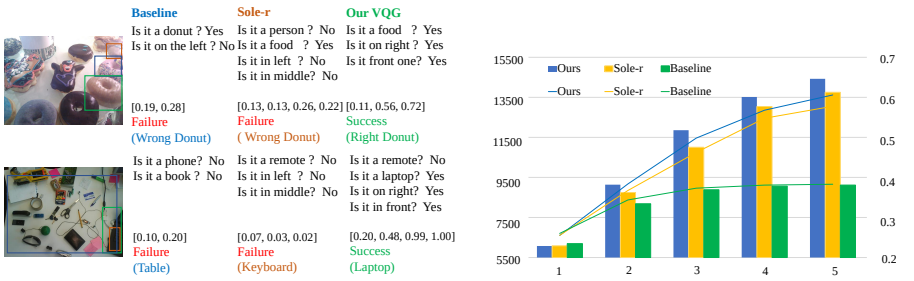

Fig. 3: Left figure: Some qualitative results of our agent (green), and the comparisons with the baseline (blue) and Sole-r model (brown). The elements in the middle array indicate the successful probabilities after each round. Right figure: The comparisons of success ratio between our agent and Sole-r, as well the baseline model, at the different dialogue round. The left and right y-axises indicate the number and ratio of successful dialogues respectively, which corresponds to the bar and line charts.

`Dialogue Round` We conduct an experiment to investigate the relationship between the dialogue round and the game success ratio. More specifically, we let `Guesser` to select the object at each round and calculate the success ratio at the given round, the comparisons of different models are shown in Fig. 3. As we can see, our agent can achieve the goal at fewer rounds compared to the other models, especially at the round three.

`Progressive Trend` To prove our VQG agent can learn a progressive trend on generated questions, we count the percentage of the successful game that has a progressive (ascending) trend on the target object, by observing the probability distributions generated by `Guesser` at each round. Our agent achieves 60.7%, while baseline and Sole-r are 50.8% and 57.3% respectively, which indicates that our agent is better at generating questions in a progressive trend considering we introduce the progressive reward r_p. Some qualitative results of the 'progressive trend' are shown in the Fig. 3, *i.e.*, the probability of the right answer is progressively increasing. Moreover, we also compute the target probability differences between the initial and final round and then divided by the number of rounds J, *i.e.*, $(p_J(o^*) - p_1(o^*))/J$. This value is the 'slope' of the progress, which reflects whether an agent can make progress in a quicker way. Our model achieves 0.10 on average, which outperforms the baseline 0.05 and Sole-r 0.08. This shows that with the proposed reward, our agent can reach the final goal with a higher 'jump' on the target probability. By combining the progressive reward with other two rewards, the agent is designed to reach the final goal in a progressive manner within limited rounds, which eliminates the infinitesimal increase case.

`Question Informativeness` We investigate the informativeness of the questions generated by different models. We let `Oracle` answer questions for all the objects at each round, and count the percentage of high-quality questions in the successful game. We define that a high-quality question is a one does not lead to the same answer for all the candidate objects. The experimental results show

that our VQG agent has 87.7% high-quality questions, which is higher than the baseline 84.7% and Sole-r 86.3%. This confirms the contribution of the r_i reward.

4.4 Human Study

We conduct human studies to see how well the human can benefit from the questions generated by these models. We show 100 images with generated question-answer pairs from different agents to eight human subjects.

For the *goal-achieved* reward, we let human subjects guess the target object, i.e., replacing the Guesser as a human. Eight subjects are asked to play on the same split, and the game is successful if more than half of the subjects give the right answer. Subjects achieve the highest success rate 75% based on our agent, while achieving 53% and 69% on the baseline and Sole-r respectively. The human study along with the ablation studies validate the significance of our proposed goal-achieved reward. For the *progressive* reward, each game generated by different agents is rated by the human subjects on a scale of 1 to 5, if the generated questions gradually improve the probability of guessing the target object from the human perspective, i.e., it can help human progressively achieve the final goal, the higher score will be given by the subject. We then compute the average scores from the eight subjects. Based on the experimental results, our agent achieves 3.24 on average, which is higher than baseline 2.02 and Sole-r 2.76. This indicates that the questions generated by our agent can lead to the goal in a more progressive way. For the *informativeness* reward, we evaluate the informativeness of each generated question by asking human subjects to rate it on a scale of 1 to 5, if this question is useful for guessing the target object from the human perspective, i.e., it can eliminate the confusion and distinguish the candidate objects for the human, the higher score will be given by the subject. We then average the scores from eight subjects for each question. Based on the experimental results, our agent achieves 3.08 on average, while baseline and Sole-r achieves 2.45 and 2.76 respectively. The advanced result shows that our agent can generate more informative questions for the human.

5 Conclusions

The ability to devise concise questions that lead to two parties to a dialog satisfying a shared goal as effectively as possible has important practical applications and theoretical implications. By introducing suitably crafted intermediate rewards into a deep reinforcement learning framework, we have shown that it is possible to achieve this result, at least for a particular class of goal. The method we have devised not only achieves the final goal reliably and succinctly but also outperforms the state-of-art. The technique of intermediate rewards we proposed here can also be applied to related goal-oriented tasks, for example, in the robot navigation, we want the robot to spend as few movements as possible to reach the destination, or in a board game, we design AI to win quickly. Our intermediate rewards can be used in these scenarios to develop an efficient AI agent.

References

1. Andreas, J., Rohrbach, M., Darrell, T., Klein, D.: Learning to compose neural networks for question answering. In: NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016. pp. 1545–1554 (2016)
2. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., Parikh, D.: Vqa: Visual question answering. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2425–2433 (2015)
3. Bottou, L.: Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT'2010, pp. 177–186. Springer (2010)
4. Das, A., Kottur, S., Gupta, K., Singh, A., Yadav, D., Moura, J.M.F., Parikh, D., Batra, D.: Visual dialog. CoRR **abs/1611.08669** (2016)
5. Das, A., Kottur, S., Moura, J.M.F., Lee, S., Batra, D.: Learning cooperative visual dialog agents with deep reinforcement learning. In: Proc. IEEE Int. Conf. Comp. Vis. pp. 2970–2979 (2017)
6. El Asri, L., Laroche, R., Pietquin, O.: Reward shaping for statistical optimisation of dialogue management. In: International Conference on Statistical Language and Speech Processing. pp. 93–101. Springer (2013)
7. Hu, R., Andreas, J., Rohrbach, M., Darrell, T., Saenko, K.: Learning to reason: End-to-end module networks for visual question answering. In: Proc. IEEE Int. Conf. Comp. Vis. pp. 804–813 (2017)
8. Johnson, J., Karpathy, A., Fei-Fei, L.: Densecap: Fully convolutional localization networks for dense captioning. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. pp. 4565–4574 (2016)
9. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: A survey. J. Arti. Intell. Research **4**, 237–285 (1996)
10. Li, Y., Duan, N., Zhou, B., Chu, X., Ouyang, W., Wang, X.: Visual question generation as dual task of visual question answering. CoRR **abs/1709.07192** (2017)
11. Liu, S., Zhu, Z., Ye, N., Guadarrama, S., Murphy, K.: Optimization of image description metrics using policy gradient methods. CoRR **abs/1612.00370** (2016)
12. 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: Proc. Advances in Neural Inf. Process. Syst. pp. 313–323 (2017)
13. Mora, I.M., de la Puente, S.P., Giro-i Nieto, X.: Towards automatic generation of question answer pairs from images (2016)
14. Mostafazadeh, N., Misra, I., Devlin, J., Mitchell, M., He, X., Vanderwende, L.: Generating natural questions about an image. In: Proc. Conf. Association for Computational Linguistics (2016)
15. Ng, A.Y., Harada, D., Russell, S.: Policy invariance under reward transformations: Theory and application to reward shaping. In: Proc. Int. Conf. Mach. Learn. vol. 99, pp. 278–287 (1999)
16. Ren, Z., Wang, X., Zhang, N., Lv, X., Li, L.J.: Deep reinforcement learning-based image captioning with embedding reward. Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2017)
17. Rennie, S.J., Marcheret, E., Mroueh, Y., Ross, J., Goel, V.: Self-critical sequence training for image captioning. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. pp. 1179–1195 (2017)

18. 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: Proc. Int. Joint Conf. Artificial Intell. (2017)
19. Su, P., Vandyke, D., Gasic, M., Mrksic, N., Wen, T., Young, S.J.: Reward shaping with recurrent neural networks for speeding up on-line policy learning in spoken dialogue systems. In: Proceedings of the SIGDIAL 2015 Conference, The 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue. pp. 417–421 (2015)
20. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction, vol. 1. MIT press Cambridge (1998)
21. Sutton, R.S., McAllester, D.A., Singh, S.P., Mansour, Y.: Policy gradient methods for reinforcement learning with function approximation. In: Advances in neural information processing systems. pp. 1057–1063 (2000)
22. de Vries, H., Strub, F., Chandar, S., Pietquin, O., Larochelle, H., Courville, A.C.: Guesswhat?! visual object discovery through multi-modal dialogue. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2017)
23. Wu, Q., Wang, P., Shen, C., Dick, A., van den Hengel, A.: Ask me anything: Free-form visual question answering based on knowledge from external sources. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (June 2016)
24. Xu, H., Saenko, K.: Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In: Proc. Eur. Conf. Comp. Vis. pp. 451–466. Springer (2016)
25. Zhang, S., Qu, L., You, S., Yang, Z., Zhang, J.: Automatic generation of grounded visual questions. In: Proc. Int. Joint Conf. Artificial Intell. pp. 4235–4243 (2017)
26. Zhu, Y., Lim, J.J., Fei-Fei, L.: Knowledge acquisition for visual question answering via iterative querying. Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2017)