

An Active Information Seeking Model for Goal-oriented Vision-and-Language Tasks

Ehsan Abbasnejad¹, Qi Wu¹, Iman Abbasnejad², Javen Shi¹, Anton van den Hengel¹

¹{ehsan.abbasnejad,qi.wu01,javen.shi,anton.vandenhengel}@adelaide.edu.au

²iman.abbasnejad@gmail.com

¹Australian Institute of Machine Learning & The University of Adelaide, Australia

²Queensland University of Technology, Australia

Abstract

As Computer Vision algorithms move from passive analysis of pixels to active reasoning over semantics, the breadth of information algorithms need to reason over has expanded significantly. One of the key challenges in this vein is the ability to identify the information required to make a decision, and select an action that will recover this information. We propose an reinforcement-learning approach that maintains an distribution over its internal information, thus explicitly representing the ambiguity in what it knows, and needs to know, towards achieving its goal. Potential actions are then generated according to particles sampled from this distribution. For each potential action a distribution of the expected answers is calculated, and the value of the information gained is obtained, as compared to the existing internal information. We demonstrate this approach applied to two vision-language problems that have attracted significant recent interest, visual dialog and visual query generation. In both cases the method actively selects actions that will best reduce its internal uncertainty, and outperforms its competitors in achieving the goal of the challenge.

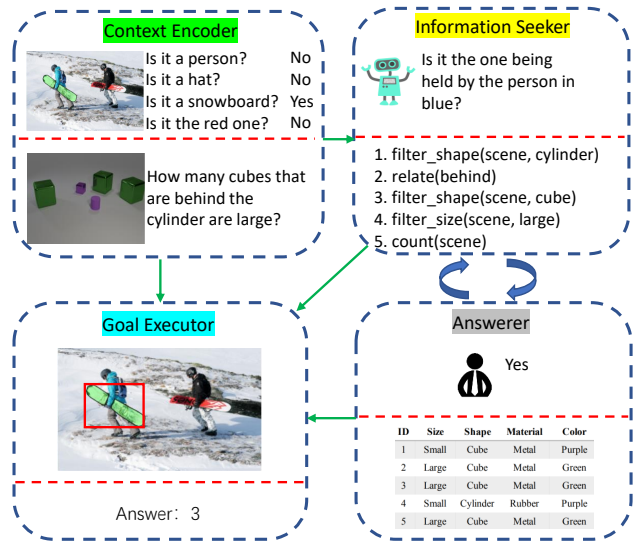


Figure 1. Two goal-oriented vision-and-language tasks, broken up into four constituent parts: a context encoder, an information seeker, an answerer and a goal executor. The given examples are chosen from a goal-oriented visual dialog dataset GuessWhat [9] (upper from the red dash-line) and, a compositional VQA dataset CLEVR [17] (lower).

1. Introduction

In most problems in Computer Vision it is assumed that all of the information required is available a-priori, and suitable to be embodied in the code or the weights of the solution. This assumption is so pervasive that it typically goes unsaid. In fact this assumption is satisfied by a small subset of problems of practical interest. Problems in this set must be self-contained, tightly specified, relate to a very prescribed form of data drawn from a static distribution, and be completely predictable. Many important problems do not satisfy these criteria, even though researchers have found many that do.

The majority of problems that computer vision might be applied to are solvable only by an agent that is capable of actively seeking the information it needs. This might be because the information required is not available at training time, or because it is too broad to be embodied in the code or weights of an algorithm. The ability to seek the information required to complete a task enables a degree of flexibility and robustness that cannot be achieved through other means, but also enables behaviors that lie towards the Artificial Intelligence end of the spectrum.

Applications that lie at the intersection of vision and language are examples of such problems. They are more challenging than conventional computer vision problems because they often require an agent (model) to acquire in-

formation on the fly to help to make decisions, such as visual dialog [6, 8] and visual question answering [20, 9, 40]. More recently, a range of tasks have been proposed that use ‘language generation’ as a mechanism to gather more information in order to achieve another specific goal. These tasks offer a particular challenge because the information involved is inevitably very broad, which makes concrete representations difficult to employ practically.

In visual dialog, and particularly goal-oriented visual question generation, an agent needs to understand the user request and complete a task via asking a limited number of questions. Similarly, compositional VQA (e.g. [17]) is a visual query generation problem that requires a model first to convert a natural language question to a sequence of ‘programs’ and then obtain the answer by running the programs on an engine. The question-to-program model represents an information ‘seeker’, while the broader goal is to generate an answer based on the information acquired.

Agents applicable to these tasks normally consist of three parts: a *context encoder*, an *information seeker* and a *goal executor*, as shown in Fig. 1. The context encoder is responsible for encoding information such as images, questions, or dialog history to a feature vector. The information seeker is a model that is able to generate new queries (such as natural language questions and programs) based on the goal of the given task and its seeking strategy. The information returned will then join the context and internal information to be sent to the goal executor model to achieve the goal. The seeker model plays a crucial role in goal-oriented vision-and-language tasks as the better seeking strategies that recovers more information, the more likely it is that the goal can be achieved. Moreover, the seeker’s knowledge of the value of additional information is essential in directing the seeker towards querying what is needed to achieve the goal. In this paper, we focus on exploring the seeker model.

The conventional ‘seeker’ models in these tasks follow a sequence-to-sequence generation architecture, that is, they translate an image to a question or translate a question to a program sequence via supervised learning. This requires large numbers of ground-truth training pairs. Reinforcement learning (RL) is thus employed in such goal-oriented vision-language to mediate this problem due to its ability to focus on achieving a goal through directed trial and error [9, 52]. A policy in RL models specifies how the seeker asks for additional information. However, these methods generally suffer from two major drawbacks: (1) they maintain a single policy that translates the input sequence to the output while disregarding the strategic diversity needed. Intuitively a single policy is not enough in querying diverse information content for various goals—we need multiple strategies. In addition, (2) the RL employed in these approaches can be prohibitively inefficient since the question generation process does not consider its effect in di-

recting the agent towards the goal. In fact, the agent does not have a notion of what information it needs and how it benefits in achieving its goal.

To this end, in contrast to conventional methods that use a single policy to model a vision-and-language task, we instead maintain a *distribution of policies*. By employing a Bayesian reinforcement learning framework for learning this distribution of the seeker’s policy, our model incorporates the expected gain from a query towards achieving its goal. Our framework uses recently proposed Stein Variational Gradient Descent [24] to perform an efficient update of the posterior policies. Having a distribution over seeking policies, our agent is *capable of considering various strategies for obtaining further information*, analogous to human contemplation of various ways to ask a question. Each sample from the seeker’s policy posterior represents a policy of its own, and seeks a different piece of information. This allows the agent to further contemplate the outcome of the various strategies before seeking additional information and considers the consequence towards the goal. We then formalize an approach for the agent to *evaluate the consequence of receiving additional information* towards achieving its goal.

We apply the proposed approach to two complex vision-and-language tasks, namely GuessWhat [9] and CLEVR [17], and show that it outperforms the baselines and achieves the state-of-art results.

2. Related work

Goal-oriented Visual Dialog Visual dialog is a recently proposed vision-and-language task that began with image captioning [47, 19, 49] and, includes visual question answering [4, 33, 50]. Das *et al.* [6] proposed a visual dialogue task that requires an agent to engage in conversation with a human centred on the content of a given image.

Das *et al.* [7] establish two reinforcement learning based 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. De Vries *et al.* [9] propose a Guess-What game dataset, where one person asks questions about an image to guess which object has been selected, and the second person answers questions as yes/no/NA. Lee *et al.* [21] adapt the information theoretic approach which allows a questioner to ask appropriate consecutive questions in the GuessWhat setting.

Compositional VQA Visual question answering is an AI-complete task that requires a model to answer image-related questions. An increasingly popular research direction in this area is to consider modular architectures. This approach

involves connecting distinct modules designed for specific desired capabilities such as memory or specific types of reasoning. Neural Module Networks (NMNs) were introduced by Andreas *et al.* in [1, 2]. There the question parse tree is turned into an assembly of modules from a predefined set, which are then used to answer the question. Johnson *et al.* [17] propose a Compositional Language and Elementary Visual Reasoning (CLEVR) dataset that allows the question to be transferred to a sequence of functional problems, which can be further used to query information for the structured scene representation to help to answer the question. In this paper, our ‘seeker’ model is used as a program generator to generate functional programs from questions.

RL in Vision-and-Language Reinforcement learning (RL) [18, 45] has been adopted in several vision-and-language problems, including image captioning [25, 34, 35], VQA [1, 13, 54], and aforementioned visual dialogue system [7, 27] *etc.* Recently, some works [5, 41] make an effort to integrate the Seq2Seq model and RL. RL has also been widely used to improve dialogue managers, which manage transitions between dialogue states [39, 32]. Most recently, Wu *et al.* [51] combines reinforcement learning and generative adversarial networks (GANs) to generate more human-like dialogues. However, nearly all of the methods use a single policy that translates the input sequence to the output while disregarding the strategic diversity needed. In our work, we instead maintain a distribution of policies. And The posterior in our RL framework is updated based on its evaluation of how much it gains towards achieving its goal once the additional information is obtained. Our agent is capable of considering various strategies for obtaining further information, analogous to human contemplation of various ways to ask a question.

Intrinsic rewards The notation of intrinsic reward which focuses of the rewards beyond what is gained from the environment has recently gained attention in the RL community. These rewards are motivated by sparse nature of the environment rewards and a need for better exploration. For example, curiosity [30] is one such intrinsic reward mechanism within which agents are encouraged to visit new states. This idea has been extended to employ Bayesian methods to learn the expected improvement of the policy for taking an action [12, 15, 44]. We use the expected gain from the answer in the vision-language task as an intrinsic reward to improve our model. Nevertheless, our approach is flexible enough that can be easily integrated with any of such additional intrinsic rewards.

3. Goal-oriented Vision-Language Task

We ground our goal-oriented vision-and-language problem as an interactive game between three agents. One agent

called *seeker* takes as the input the encoded image and context feature to generate a query to seek more information from an *answerer* agent, who will generate a response. This process can be performed multiple rounds until the *seeker* gathered enough information. The queried information are sent to the third agent *executer* who will make the final prediction. The game is recognized as a success if the prediction hits the given goal.

Formally, for each game at the round of t , we have a tuple $(I, C, q^{(t)})$, where I is the observed image, C is the context information at the current round and $q^{(t)}$ is a query generated from the *seeker* agent to query the information. In the next step, the $q^{(t)}$ is sent to the *answerer*, who will generate a response $a^{(t)}$. After T rounds of this ‘seek-answer’ process, the tuple $(I, C, \{q^{(t)}\}_{t=1}^T, \{a^{(t)}\}_{t=1}^T)$ is sent to the *executer*, who will select the target from $O = \{o_1, o_2, \dots, o_N\}$ which is a candidate list. The ground truth target is denoted as o^* and the game is success if the o^* is successfully selected by the *executer*.

To be more specific, in the visual dialog (Guesswhat) setting, C is the dialog history and $q^{(t)}$ is a natural language question. O is the candidate object bounding boxes. In the VQA (CLEVR), C is a single question asked by users and $q^{(t)}$ is a functional program, while the O is a candidate answer vocabulary. In this paper, we adapt pre-trained, fixed *answerer* and *executer* in the game and only focus on training a better *seeker*, which will be illustrated in the following section.

4. Preliminaries

We introduce the background of reinforcement learning and discuss policy gradient estimation method that we will modify for a contemplation-based question generation.

4.1. Reinforcement Learning

Reinforcement learning considers agents interacting with their environment by taking a sequence of actions and assessing their effect through a scalar reward for each action. The agent’s task is to learn a *policy* that maximizes the cumulative rewards.

Consider the dialog generating agent asking questions $q^{(t)} \in \mathcal{Q}$ at each time step t given the state $\mathbf{s}^{(t)}$. Each $\mathbf{s}^{(t)}$ encompasses the history of the dialog (including past question-answer pairs) and the input image. Upon taking receiving an answer $a^{(t)} \in \mathcal{A}$, the agent then observes a new state $\mathbf{s}^{(t+1)}$ and receives a scalar reward $r(\mathbf{s}^{(t)}, q^{(t)}) \in \mathcal{R}$. The goal of dialog generating reinforcement learning is to find a questing policy $\pi(q^{(t)}|\mathbf{s}^{(t)})$ for choosing an action given state $\mathbf{s}^{(t)}$ to maximize an expected return:

$$J(\pi) = \mathbb{E}_{\mathbf{s}_0, q_0, \dots \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t r(\mathbf{s}^{(t)}, q^{(t)}) \right],$$

where $0 \leq \gamma^t \leq 1$ is a discount factor. The expected return J depending on π because $q^{(t)} \sim \pi(q^{(t)}|\mathbf{s}^{(t)})$ drawn from the policy (distribution) π . The state $\mathbf{s}^{(t+1)} \sim P(\mathbf{s}^{(t+1)}|\mathbf{s}^{(t)}, q^{(t)})$ is generated by the dialog's environmental dynamics which are unknown. The state value function $V^\pi(\mathbf{s}^{(t)}) = \mathbb{E}_{a^{(t)}, \mathbf{s}^{(t+1)}, \dots \sim \pi} [\sum_{i=0}^{\infty} \gamma^i r(\mathbf{s}^{(t+i)}, q^{(t+i)})]$ is the expected return by policy π from state $\mathbf{s}^{(t)}$. The answer $a^{(t)}$ is a response from the `answerer`, who takes $q^{(t)}$ as input. The `answerer` is a neural network that learns the potential answer by mapping the input $(\mathbf{s}^{(t)}, q^{(t)})$ to $a^{(t)}$.

4.2. Policy Gradient Estimation

In policy-based reinforcement learning approaches, the policy is parameterized by θ as $\pi(q|\mathbf{s}; \theta)$. Then the objective is to iteratively update θ to optimize $J(\theta) := J(\pi(q|\mathbf{s}; \theta))$. We use the shorthanded notation $J(\theta)$ for clarity from now on. In policy gradient algorithms [46] such as the well-known REINFORCE [48], the gradient $\nabla_\theta J(\theta)$ is estimated by samples from the policy $\pi(q|\mathbf{s}; \theta)$ using the likelihood ratio trick. Specifically, REINFORCE uses the following approximator of the policy gradient

$$\nabla_\theta J(\theta) \approx \sum_{t=0}^{\infty} \nabla_\theta \log \pi(q^{(t)}|\mathbf{s}^{(t)}; \theta) r(\mathbf{s}^{(t)}, q^{(t)}) \quad (1)$$

based on a single rollout trajectory, where $r(\mathbf{s}^{(t)}, q^{(t)}) = \sum_{i=0}^{\infty} \gamma^i r(\mathbf{s}^{(t+i)}, q^{(t+i)})$ is the accumulated return from time step t . It was shown that this gradient is an unbiased estimation of $\nabla_\theta J(\theta)$. Typically a baseline function $b(\mathbf{s}^{(t)})$ is considered to reduce the variance of this estimator, then: $\nabla_\theta J(\theta) \approx \sum_{t=0}^{\infty} \nabla_\theta \log \pi(q^{(t)}|\mathbf{s}^{(t)}; \theta) (r(\mathbf{s}^{(t)}, q^{(t)}) - b(\mathbf{s}^{(t)}))$. It is common to use the value function $V^\pi(\mathbf{s}^{(t)})$ as the baseline where $r(\mathbf{s}^{(t)}, q^{(t)}) - V^\pi(\mathbf{s}^{(t)})$ is known as the advantage function.

5. Information Seeker and Answerer Imitation

This section introduces our main framework. We discuss three main parts of the framework: answerer imitation (Subsection 5.1), the seeker's belief (Subsection 5.2) and the seeker's update (Subsection 5.3).

In the `answerer` imitation, the agent models a belief over the potential answers and evaluates the gain from querying at a given time. The `seeker` considers a belief over the policies that could generate queries to obtain additional information. In sharp contrast to existing methods, we are interested in not only finding a right policy, but more importantly we model a multi-modal distribution of policies, to enable learning diverse seeking policies in analogy to human contemplation using multiple strategies.

Finally, in seeker's belief update, the agent updates its belief over the distribution of the policies incorporating the feedback from the environment.

5.1. Answerer's Imitation

In our approach the agent keeps a model of the `answerer` to be able to predict which question worth asking. The agent uses this model to imitate the behavior of the goal `executer` and anticipate its potential response to the question. Utilizing this imitating model, the agent asks the questions whose answers bring it closer to achieving its goal. In particular, the agent asks a question $q^{(t)}$ at time t only if it believes the answer $a^{(t)}$ it receives ultimately maximizes the *gain* in achieving its goal at state $\mathbf{s}^{(t)}$:

$$\begin{aligned} \mathcal{G}_\omega(\mathbf{s}^{(t)}, q^{(t)}) &= \mathbb{E}_a[u(a^*|q^{(t)}, \mathbf{s}^{(t)}, C, I, \omega)] \\ &\quad - \mathbb{E}_a[u(a|q^{(t)}, \mathbf{s}^{(t)}, C, I, \omega)] \\ &= U_{a^*, \omega}(\mathbf{s}^{(t)}, q^{(t)}) - U_\omega(\mathbf{s}^{(t)}, q^{(t)}) \end{aligned} \quad (2)$$

where u is the utility function that measures the performance of the `answerer`. In addition, ω is the set of parameters of the `answerer` that we learn. Particularly we find ω such that $\mathbb{E}_{a \sim p(a|\mathbf{s}^{(t)}, C, I, \omega)}[p(o|a, \mathbf{s}^{(t)})]$, where

$$\begin{aligned} p(a|\mathbf{s}^{(t)}, C, I, \omega) &= \int p(a|q^{(t)}, \mathbf{s}^{(t)}; \omega) \\ &\quad \times \pi(q^{(t)}|\mathbf{s}^{(t)}; \theta) \pi(\theta|C, I) d\theta \end{aligned} \quad (3)$$

is maximized. Here, $p(a^{(t)}|q^{(t)}, \mathbf{s}^{(t)}; \omega)$ is the probability that the query seeker produced, yields a particular answer. Note that we have taken the distribution of the parameter θ which indicates the `answerer` has to imitate the goal `executioner` for all possible questions produces by the policies. Since policies and `answerer's` response are uncertain, we devise a bound on the gain. Using Chebyshev's inequality, we have:

$$p\left(|\mathcal{G}_\omega(\mathbf{s}^{(t)}, q^{(t)}) - \hat{\mu}_\omega(\mathbf{s}^{(t)}, q^{(t)})| < \beta \hat{\sigma}_\omega(\mathbf{s}^{(t)})\right) \geq 1 - \frac{1}{\beta^2} \quad (4)$$

where $\hat{\mu}_\omega(\mathbf{s}^{(t)}, q^{(t)}) = (\hat{U}_{a^*, \omega}(\mathbf{s}^{(t)}, q^{(t)}) - \hat{U}_\omega(\mathbf{s}^{(t)}, q^{(t)}))$ is the empirical difference of the expected utilities in Eq. 2 and $\hat{\sigma}_\omega(\mathbf{s})$ is its standard deviation. Moreover, $\beta > 0$ is an appropriate constant. With high probability we have $|\mathcal{G}_\omega(\mathbf{s}^{(t)}, q^{(t)}) - \hat{\mu}_\omega(\mathbf{s}^{(t)}, q^{(t)})| < \beta \hat{\sigma}_\omega(\mathbf{s}^{(t)}, q^{(t)})$ which in turn defines an *optimistic upper bound* on the gain from asking a question:

$$\hat{\mathcal{G}}_\omega(\mathbf{s}^{(t)}, q^{(t)}) = \hat{\mu}_\omega(\mathbf{s}^{(t)}, q^{(t)}) + \beta^2 \hat{\sigma}_\omega(\mathbf{s}^{(t)}, q^{(t)}) \quad (5)$$

This measure defines the expected upper bound on the gain at state t for potentially seeking $q^{(t)}$. Note that this is not simply a difference of expected utilities, because a simple difference will be too greedy and does not consider that the estimate of the distribution of the answers for the agent might not be correct. As such, the agent needs to gather more information about the potential answers it is not confident about.

In order to integrate this measure into an RL framework, inspired by curiosity-driven and information maximizing

exploration [12, 30], we modify the reward function to incorporate this intrinsic motivation to consider the gain, i.e.

$$r^{\text{new}}(\mathbf{s}^{(t)}, q^{(t)}) = r(\mathbf{s}^{(t)}, q^{(t)}) + \eta \hat{\mathcal{G}}_{\omega}(\mathbf{s}^{(t)}, q^{(t)}) \quad (6)$$

$$J(\theta) = \mathbb{E}_{\pi(\mathbf{s}, q|\theta)} \left[\sum_{t=0}^{\infty} \gamma^t r^{\text{new}}(\mathbf{s}^{(t)}, q^{(t)}) \right], \quad (7)$$

for $\eta \geq 0$ that controls the intrinsic reward. In this new reward, an agent’s anticipation of the answer is taken into account when updating the policy. When the seeker knows the answer and its gain is small, the parameters are not changed significantly. In other words, there is no need for further changes to the questions where the answer is known. On the other hand, when the agent anticipates a large gain from the answer and receives a large reward, the policy has to be adjusted by a larger change in the parameters. Similarly, if the agent expects a large gain and is not rewarded, there has to be significant update in the policy.

The advantage of this approach is twofold: (1) it helps deal with sparse rewards and (2) we encourage the method to ask the most informative questions. This allows the agent to learn to mimic the behavior of the goal executor and generalize to unseen cases.

5.2. Information Seeker’s Belief

Considering Eq. 3 and the need for the `answerer` to consider the distribution of the policies, instead of finding a single policy as parameterized by θ , we model the *seeker’s policy distribution*. Each sample of the parameter θ gives rise to a different questioning policy allowing us to model policy distribution.

This distribution allows for the agent to consider alternatives, or contemplates, various question policies to improve the overall dialog performance. As such, here we consider the policy parameter θ as a random variable (leading to random policies that we can model their distribution) and seek a distribution to optimize the expected return. We incorporate a prior distribution over the policy parameter, for instance, for when we have no answer for question-answer pairs or to incorporate prior domain knowledge of parameters. We formulate the optimization of π as the following regularized problem:

$$\max_{\pi} \left\{ \mathbb{E}[J(\theta)] - \alpha \text{KL}(\pi \| \pi_0) \right\}, \quad (8)$$

where π maximizes the expected cumulative reward regularized by a relative entropy $\text{KL}(\pi \| \pi_0)$,

$$\text{KL}(\pi \| \pi_0) = \mathbb{E}_{\pi(\theta|a^*, q^{(t)}, C, I)} [\log \pi(\theta|a^*, q^{(t)}, C, I) - \log \pi_0(\theta|C, I)].$$

Effectively we seek a parameter distribution that gives rise to policies that both maximize the expected reward

while remain close to the prior. It is easy to see if we use an uninformative prior such as a uniform distribution, the second KL term is simplified to the entropy of π . Then optimization in Eq. 8 becomes $\max_{\pi} \left\{ \mathbb{E}_{\pi(\theta)}[J(\theta)] + \alpha \mathbf{H}(\pi) \right\}$ which explicitly encourages exploration in the parameter space θ . This exploration yields diverse policies that result in varied queries.

By taking the derivative of the objective function in (8) and setting it to zero, the optimal distribution of policy parameter θ is obtained as

$$\pi(\theta|a^*, q^{(t)}, C, I) \propto \exp(J(\theta)/\alpha) \pi_0(\theta|C, I). \quad (9)$$

In this formulation, $\pi(\theta|a^*, q, C, I)$ is similar to the “posterior” of the parameters θ in the conventional Bayesian approach. Here, $\exp(J(\theta)/\alpha)$ is effectively the “likelihood” function. The coefficient α is the parameter that controls the strength of exploration in the parameter space and how far the posterior is from the prior. As $\alpha \rightarrow 0$, samples drawn from $\pi(\theta|a^*, q^{(t)}, C, I)$ will be concentrated on a single policy around the optimum of $\mathbb{E}[J(\theta)]$ and lead to less diverse seekers.

Remember from Eq. 7 that the “likelihood” here considers the agent’s anticipation of the answer. If its reward is higher, then a larger change to the parameter is needed to allow exploitation of new knowledge about the effect of the current policy on the goal.

Similar ideas of entropy regularization has been investigated in other reinforcement learning methods [29, 37, 26]. However, in our approach we use the regularization to obtain the posterior for the policy parameters in the information seeking framework where the imitation of the `answerer` refines the policy distribution.

5.3. Seeker’s Posterior Update

A conventional method to utilize the posterior in Eq. 9 is MCMC where samples from this distribution are used. However, MCMC methods are computationally expensive, suffer from slow convergence and have high-variance due to stochastic nature of estimating $J(\theta)$. Since estimating $J(\theta)$ by itself is a computationally demanding task and may vary significantly for each policy, we look for an efficient alternative. Thus, rather than $J(\theta)$, we use the gradient information $\nabla_{\theta} J(\theta)$ that points to the direction for seeker’s policy change using the *Stein variational gradient descent* (SVGD) for Bayesian inference [24, 23]. SVGD is a non-parametric variational inference algorithm that leverages efficient deterministic dynamics to transport a set of particles $\{\theta_i\}_{i=1}^n$ to approximate given target posterior distributions $\pi(\theta|a^*, q^{(t)}, C, I)$. Unlike traditional variational inference methods, SVGD does not confine the approximation within a parametric families, which means the seeker’s policy does not need to be approximated by another. Further, SVGD

Algorithm 1 Seeker

Input: Learning rate $\epsilon_\theta, \epsilon_\omega$, kernel $k(\theta, \theta')$, initial policy particles $\{\theta_i\}$, context history C , image I .

for iteration $t = 0, 1, \dots, T$ **do**

for particle $i = 1, \dots, n$ **do**

 Sample $q^{(t)} \sim \pi(q|s^{(t)}, C, I; \theta_i)$

 Sample $a^{(t)} \sim \pi(a|s^{(t)}, C, I; \omega)$

 Compute $\mathcal{G}(s^{(t)}, q^{(t)})$ from Eq. (5).

 Compute new rewards r^{new} from Eq. (6).

 Compute $\nabla_{\theta_i} J(\theta_i)$ in Eq. (7).

$\Delta\omega \leftarrow \Delta\omega + \nabla_\omega \log(p(o|a^{(t)}, s^{(t)}))$

end for

$\omega \leftarrow \omega + \epsilon_\omega \Delta\omega$

for particle $i = 0, 1, \dots, n$ **do**

$J_{\text{new}}(\theta_j) = \frac{1}{\alpha} J(\theta_j) + \log \pi_0(\theta_j)$

$\Delta\theta_i \leftarrow \frac{1}{n} \sum_{j=1}^n [\nabla_{\theta_j} J_{\text{new}}(\theta_j) k(\theta_j, \theta_i) + \nabla_{\theta_j} k(\theta_j, \theta_i)]$

$\theta_i \leftarrow \theta_i + \epsilon \Delta\theta_i$

end for

end for

converges faster than MCMC due to the deterministic updates that efficiently leverage gradient information of the seeker’s policy posterior. This inference is efficiently performed by iteratively updating multiple “particles” $\{\theta_i\}$ as $\theta_i = \theta_i + \epsilon_\theta \psi^*(\theta_i)$, where ϵ_θ is a step size and $\psi(\cdot)$ is a function in the unit ball of a reproducing kernel Hilbert space (RKHS). Here, ψ^* is chosen as the solution to minimizing KL divergence between the particles and the target distribution. It was shown that this function has a closed form solution, and an empirical estimate [24]:

$$\hat{\psi}(\theta_i) = \frac{1}{n} \sum_{j=1}^n [\nabla_{\theta_j} \log \pi(\theta_j | a^*, q^{(t)}, C, I) k(\theta_j, \theta_i) + \nabla_{\theta_j} k(\theta_j, \theta_i)]. \quad (10)$$

where k is the the positive definite kernel associated with the RKHS space. In this update rule $\hat{\psi}$, the first term involves the gradient $\nabla_{\theta} \log \pi(\theta | a^*, q^{(t)}, C, I)$ which moves the seeker’s policy particles θ_i towards the high probability regions by sharing information across similar particles. The second term $\nabla_{\theta_j} k(\theta_j, \theta_i)$ utilizes the curvature of the parameter space to push the particles away from each other, which leads to diversification of the seeker’s policies.

An example of the landscape of the policies is shown in Fig. 2. Each initial sample from the policy distribution can move towards one of the modes of a highly multi-modal distribution. These moves are governed by the gradient of the policy that in our case consists of the agent’s belief about the answer and its consequence once its response is known. In addition, kernel k controls the distance between the parameters to deter from collapsing to a single point in multi-modal distribution. It is intuitive from the figure that a better

gradient from the rewards by incorporating the answers and considering the distribution of policies improves the performance of the seeker by guiding the parameter updates.

It is important to note that diversification in the parameter space allows for an accurate modeling of a highly multimodal policy space. Otherwise, the policy distribution collapses to a single point which is the same as the conventional maximum a posteriori (MAP) estimate. This MAP estimate only considers a single policy that in the highly complex task of vision-language is inadequate.

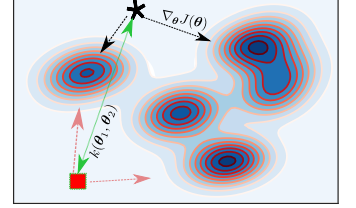


Figure 2. A multi-modal distribution of the policies. Unlike conventional policy gradient only explores nearest mode ignoring other modes, our novel approach always use a number of initial points (i.e. the policy parameters) to explore multiple modes collaboratively in analogy of human contemplation of multiple strategies. We only show two initial points, a red rectangle and black asterisk, for the ease of visualization.

6. Experiments

To evaluate the performance of the proposed approach we conducted experiments on two different goal-oriented vision-and-language datasets: CLEVR [17] and GuessWhat [9]. The former one is a visual question answering task

while the later is a visual dialog task. In both experiments we pre-train the networks using the supervised model and refine using reinforcement learning, which is a normal practice in this area [8, 9, 22]. Without using supervised learning first, the dialog model may diverge from human language. In both cases we generate the policies by sampling from the policy posterior $\theta \sim \pi(\theta | a^*, q^{(t)}, C, I)$ and generate the query with the highest gain measured by the answerer. Our approach outperforms the baseline and previous state-of-art in both cases, which will be detailed discussed in the following sections.

6.1. CLEVR

CLEVR [17] is a synthetically generated dataset containing 700K (image, question, answer, program) tuples. Images are 3D-rendered objects of various shapes, materials, colors, and sizes. Questions are compositional in nature and range from counting questions to comparison questions and can be 40+ words long. An answers is a word from a set of 28 choices. For each image and question, a program consists of step-by-step instructions, on how to answer the question. During the test, the programs are not given, which need to be generated conditioned on the input question.

Implementation Details We follow the experiment setup of [16, 17] in which a ResNet [10] is used to encode

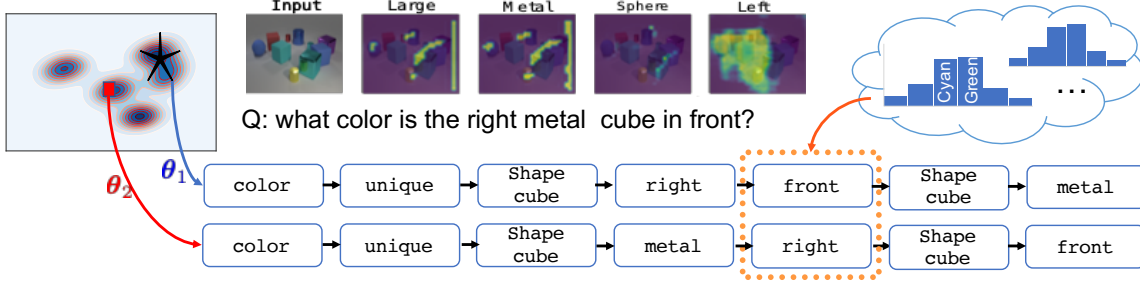


Figure 3. An example of a question and the programs generated using samples from the posterior in CLEVR. Samples from the policy distribution take the input image and the question and generate its corresponding programs. As observed, these two samples produce different program sequences which enable to explore multiple distributions over the goal (final answer) shown on top in the cloud. Expected utility of each question $\hat{U}_\omega(s^{(t)}, q^{(t)})$ gives us an indication of which one is better to ask.

the given images and a standard Long short-term memory (LSTM) [11] for the Seq2Seq model to generate programs in the context encoder. Note that our context encoder is a pixel level model that does not extract objects explicitly from the given image. For the goal executor, we use a modular network [3]. We use 10 particles in Algorithm 1 to model the policy distribution using samples from the pre-trained model with added noise so that they correspond to different initial policies. For more efficient implementation, we use two sets of shard parameters for the encoder in the underlying Seq2Seq model and use independent parameters for the LSTM decoder. In addition to efficiency, this parameter sharing ensures there are common latent representations that particles learn. We use our information seeker model in Section 5 to generate samples or programs for each question and consider the consequence of that program using the answerer internally to choose one. Once a program is generated, it is then executed by the goal executor to obtain the feedback and compute the corresponding rewards. The computed reward is then used to update the policy distributions as discussed. We use the Adam optimization method with learning rate set to 10^{-5} to update both the seeker and the answerer’s parameters. The testing procedure thus takes an image and question pairs, produces a program, then the goal executor produces an answer. The goal executor then evaluates the quality of the generated program. For $\eta = 0.1 \times \frac{\text{epoch}_{\max} - \text{epoch}}{\text{epoch}_{\max}}$ to encourage the policies to explore more in the initial stages. We set $\beta = 1.0$ and $\alpha = 0.01$. We use the median trick from [24] to compute the RBF-kernel’s hyper-parameter which essentially ensures $\sum_j k(\theta_i, \theta_j) \approx 1$.

Overall Results For the answerer and the goal executor, we consider two alternative baselines: (G)eneric similar to [16] where each module follows a generic architecture; and, (D)esigned similar to [28] where each module is specifically designed based on the desired operation. We report the accuracy of the goal executor. Since the later case provides a better representation on each module, we

Model	Overall	Count	Compare Numbers	Exist	Query Attribute	Compare Attribute
NMN [3]	72.1	52.5	72.7	79.3	79.0	78.0
N2NMN [13]	88.8	68.5	84.9	85.7	90.0	88.8
Human [17]	92.6	86.7	86.4	96.6	95.0	96.0
LSTM+RN [36]	95.5	90.1	93.6	97.8	97.1	97.9
PG+EE (9k) [16]	88.6	79.7	79.7	89.7	92.6	96.0
PG+EE (18k) [16]	95.4	90.1	96.2	95.3	97.3	97.9
PG+EE (700k) [16]	96.9	92.7	98.6	97.1	98.1	98.9
FiLM [31]	97.6	94.5	93.8	99.2	99.2	99.0
DDRprog [42]	98.3	96.5	98.4	98.8	99.1	99.0
MAC [14]	98.9	97.2	99.4	99.5	99.3	99.5
TbD-net [28]	98.7	96.8	99.1	98.9	99.4	99.2
TbD-net++ [28]	99.1	97.6	99.4	99.2	99.5	99.6
Ours+G+entropy (9k)	91.4	86.4	93.6	89.8	93.2	96.2
Ours+G+entropy (18k)	95.6	93.3	96.8	95.4	97.8	98.1
Ours+G+entropy (700k)	97.4	96.8	98.1	98.2	96.2	98.1
Ours+D+entropy (9k)	94.7	92.2	95.6	93.2	95.1	97.7
Ours+D+entropy (18k)	96.6	94.6	96.1	95.6	98.1	98.6
Ours+D+entropy (700k)	98.3	98.1	99.1	97.1	98.6	98.8
Ours+G+exp (9k)	91.8	87.5	93.7	90.2	93.1	96.5
Ours+G+exp (18k)	96.3	93.3	96.8	95.4	97.8	98.1
Ours+G+exp (700k)	98.0	96.2	98.6	98.0	98.0	99.0
Ours+D+exp (9k)	95.2	91.5	96.7	93.8	95.7	98.7
Ours+D+exp (18k)	97.1	94.5	98.2	96.1	98.3	98.6
Ours+D+exp (700k)	98.9	97.8	99.2	98.9	99.5	99.3
Ours+D+exp++ (700k)	99.2	97.8	99.5	99.4	99.6	99.6

Table 1. Performance comparison of state-of-the-art models on the CLEVR dataset. "Ours+G+entropy" is our seeker when used with the generic architecture and entropic gain; "Ours+D+entropy" is the same except for using designed architecture. Similarly, "Ours+G+exp" is generic architecture with u_{exp} ; and, "Ours+D+exp" is its designed counterpart. We achieve state of the art performance, especially using smaller ground-truth programs. The '++' indicator shows a model was trained using higher-resolution 28×28 feature maps rather than 14×14 .

expect it to perform better. Further, we use two functions $u_{\text{entropy}}(\cdot) = -\log(\cdot)$ (corresponding to the information-theoretic notion of gain in the expectation) and $u_{\text{exp}}(\cdot) = 1/(1 + \exp(\cdot))$ to operate on the output of the answerer’s score to compute the gain and ultimately the new reward in Eq. 2 and 7. The results are shown in Tab. 1. As shown our approach outperforms the baselines by a significant margin. In particular, our approach almost achieves the same performance as that of [16] with half the programs used for train-

ing with the same neural architecture. Moreover, choices of u affects the policies found, for instance using u_{entropy} generally leads to outperforming in the "count" function. As shown in Figure 3, each sample from the policy can generate a different program. In addition, we are able to utilize the attention mechanism in the model to *reason* about where in the image the information seeker focuses.

Furthermore, Fig. 4 plots the average reward at each iteration (on top) and the average distance between the particles in the policies (in the bottom). If the problem was indeed unimodal (as conventional methods assume), all the particles would collapse to a single point indicated by a zero average distance. However, as is observed, while the distance between the particles decrease in early stages, they soon increase indicating each one is converging to a different mode.

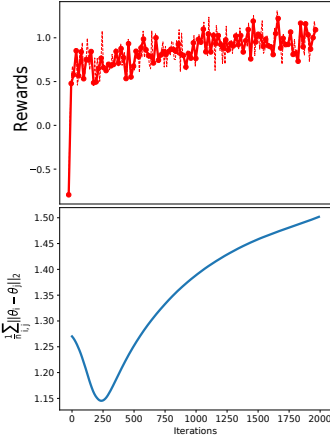


Figure 4. Average reward for the agent at each iteration (Top); and, average distance between particles in the posterior for CLEVR (Bottom).

6.2. GuessWhat

GuessWhat [9] is a classical goal-oriented visual dialog game which consists of three players. In each game, a random object in the scene is assigned to the answerer, where this process is hidden to the questioner (*i.e.* our information seeker). 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 (*i.e.* our goal executor) can start to guess. If the guesser guesses the correct object the game is successfully concluded. The dataset includes 155, 281 dialog of 821, 955 pairs of question/answers with vocabulary size 11, 465 on 66, 537 unique images and 134, 074 objects.

Implementation Details In our model, the information seeker is a recurrent neural network (RNN) that produces a sequence of state vectors for a given input sequence by applying LSTM as a transition function. The output of this LSTM network is the internal estimate of the reward with size 1024. To obtain a distribution over tokens, a softmax is applied to this output. The image representation is obtained using a VGG [38]. The concatenation of the image and history features are given to the LSTM in the Seq2Seq model for question generation where each word is sampled conditioned on its previous word. We use the u_{exp} from the CLEVR experiment to compute all the rewards.

Model	New Object	New Image
Supervised-S [9]	41.6	39.2
Supervised-G [9]	43.5	40.8
RL-S [40]	56.5	58.5
RL-G [40]	60.3	58.4
Tempered [53]	62.6	-
Tempered-Seq2Seq [53]	63.5	-
Tempered-MemoryNet [53]	68.3	-
Ours	64.2	62.1
Ours+MemoryNet (Single)	70.1	67.9
Ours+MemoryNet	74.4	72.1

Table 2. Accuracy in identifying the goal object in the GuessWhat dataset (higher is better). The "S" indicator is for sampling for words method vs "G" which is greedy. Ours+MemoryNet is the method with modified answerer that employs Memory network and Attention. Further, (Single) indicates training our method with a single particle.

We set η , β and the RBF-kernel's hyper-parameter similar to the experiments in CLEVR, however we set $\alpha = 0.001$ here using grid-search.

Overall Results We compare two cases, labeled *New Object* and *New Image*. In the former the object sought is new, but the image has been seen previously. In the latter the image is also previously unseen. We report the prediction accuracy for the guessed objects. It is clear that the accuracies are generally higher for the new objects as they are obtained from the already seen images.

The results are summarized in Table 2. As shown, using the conventional REINFORCE [40] by either sampling each word (RL-S) or greedily selecting one (RL-G) improves the performance compared to the supervised baseline significantly. Since our approach explore and exploits the space of policies for question generation better, it achieves better performance. Furthermore, this performance is improved when a better goal seeker or answerer model is employed. Better answerer leads to more realistic intrinsic rewards that corresponds to true gains and guide the policy distribution to the better posterior. For instance, employing a Memory network [43] within the answerer improves its performance that in turn is reflected in the quality of the questions and consequently agent's ability to achieve goals more accurately. This is because our policy update depends on the reward and the expected gain of the agent from its answer.

7. Conclusion

The ability to identify the information needed to support a conclusion, and the actions required to obtain it, is a critical capability if agents are to move beyond carrying out low-level prescribed tasks towards achieving flexible high semantic level goals. The method we describe is capable of reasoning about the information it holds, and the information it will need to achieve its goal, in order to identify the action that will best enable it to fill the gap between the two.

Our approach thus actively seeks the information it needs to achieve its goal on the basis of a model of the uncertainty in its own understanding. If we are to enable agents that actively work towards a high-level goal the capability our approach demonstrates will be critical.

References

- [1] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Learning to compose neural networks for question answering. In *Proc. Conf. of North American Chapter of Association for Computational Linguistics*, 2016. 3
- [2] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Neural Module Networks. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016. 3
- [3] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Neural module networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 39–48, 2016. 7
- [4] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: Visual Question Answering. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 2425–2433, 2015. 2
- [5] N. Asghar, P. Poupart, J. Xin, and H. Li. On-line sequence-to-sequence reinforcement learning for open-domain conversational agents. *arXiv preprint arXiv:1612.03929*, 2016. 3
- [6] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh, and D. Batra. Visual dialog. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2017. 2
- [7] A. Das, S. Kottur, J. M. Moura, S. Lee, and D. Batra. Learning cooperative visual dialog agents with deep reinforcement learning. *arXiv preprint arXiv:1703.06585*, 2017. 2, 3
- [8] A. Das, S. Kottur, J. M. F. Moura, S. Lee, and D. Batra. Learning cooperative visual dialog agents with deep reinforcement learning. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2970–2979, 2017. 2, 6
- [9] H. de Vries, F. Strub, S. Chandar, O. Pietquin, H. Larochelle, and A. C. Courville. Guesswhat?! visual object discovery through multi-modal dialogue. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 6, 8
- [10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 6
- [11] S. Hochreiter and J. Schmidhuber. Long short-term memory. 9:1735–80, 12 1997. 7
- [12] R. Houthoofd, X. Chen, X. Chen, Y. Duan, J. Schulman, F. De Turck, and P. Abbeel. Vime: Variational information maximizing exploration. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1109–1117. Curran Associates, Inc., 2016. 3, 5
- [13] R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko. Learning to reason: End-to-end module networks for visual question answering. *arXiv preprint arXiv:1704.05526*, 2017. 3, 7
- [14] D. A. Hudson and C. D. Manning. Compositional attention networks for machine reasoning. In *International Conference on Learning Representations*, 2018. 7
- [15] L. Itti and P. F. Baldi. Bayesian surprise attracts human attention. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 547–554. MIT Press, 2006. 3
- [16] J. Johnson, B. Hariharan, L. v. Maaten, J. Hoffman, L. Fei-Fei, C. L. Zitnick, and R. Girshick. Inferring and executing programs for visual reasoning. In *2017 IEEE International Conference on Computer Vision (ICCV)*, volume 00, pages 3008–3017, Oct. 2018. 6, 7
- [17] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017. 1, 2, 3, 6, 7
- [18] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *J. Arti. Intell. Research*, 4:237–285, 1996. 3
- [19] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 3128–3137, 2015. 2
- [20] S. Lee, Y. Heo, and B. Zhang. Answerer in questioner’s mind for goal-oriented visual dialogue. *CoRR*, abs/1802.03881, 2018. 2
- [21] S.-W. Lee, Y.-J. Heo, and B.-T. Zhang. Answerer in questioner’s mind for goal-oriented visual dialogue. *arXiv preprint arXiv:1802.03881*, 2018. 2
- [22] M. Lewis, D. Yarats, Y. N. Dauphin, D. Parikh, and D. Batra. Deal or No Deal? End-to-End Learning for Negotiation Dialogues. *ArXiv e-prints*, 2017. 6
- [23] Q. Liu. Stein variational gradient descent as gradient flow. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3115–3123. Curran Associates, Inc., 2017. 5

- [24] Q. Liu and D. Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2378–2386. Curran Associates, Inc., 2016. 2, 5, 6, 7
- [25] S. Liu, Z. Zhu, N. Ye, S. Guadarrama, and K. Murphy. Optimization of image description metrics using policy gradient methods. *arXiv preprint arXiv:1612.00370*, 2016. 3
- [26] Y. Liu, P. Ramachandran, Q. Liu, and J. Peng. Stein variational policy gradient. *arXiv preprint arXiv:1704.02399*, 2017. 5
- [27] J. Lu, A. Kannan, J. Yang, D. Parikh, and D. Batra. Best of both worlds: Transferring knowledge from discriminative learning to a generative visual dialog model. *arXiv preprint arXiv:1706.01554*, 2017. 3
- [28] D. Mascharka, P. Tran, R. Soklaski, and A. Majumdar. Transparency by Design: Closing the Gap Between Performance and Interpretability in Visual Reasoning. *ArXiv e-prints*, Mar. 2018. 7
- [29] P. A. Ortega and D. A. Braun. Information, utility and bounded rationality. In *Proceedings of the 4th International Conference on Artificial General Intelligence*, AGI'11, pages 269–274, 2011. 5
- [30] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning (ICML)*, 2017. 3, 5
- [31] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. C. Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018. 7
- [32] O. Pietquin, M. Geist, S. Chandramohan, and H. Frezza-Buet. Sample-efficient batch reinforcement learning for dialogue management optimization. *ACM Transactions on Speech and Language Processing (TSLP)*, 7(3):7, 2011. 3
- [33] M. Ren, R. Kiros, and R. Zemel. Image Question Answering: A Visual Semantic Embedding Model and a New Dataset. In *Proc. Advances in Neural Inf. Process. Syst.*, volume 1, page 5, 2015. 2
- [34] Z. Ren, X. Wang, N. Zhang, X. Lv, and L.-J. Li. Deep reinforcement learning-based image captioning with embedding reward. *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2017. 3
- [35] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. Self-critical sequence training for image captioning. *arXiv preprint arXiv:1612.00563*, 2016. 3
- [36] A. Santoro, D. Raposo, D. G. T. Barrett, M. Malinowski, R. Pascanu, P. W. Battaglia, and T. P. Lillicrap. A simple neural network module for relational reasoning. In *NIPS*, 2017. 7
- [37] J. Schulman, S. Levine, P. Moritz, M. Jordan, and P. Abbeel. Trust region policy optimization. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 1889–1897, 2015. 5
- [38] K. "Simonyan and A. Zisserman. "very deep convolutional networks for large-scale image recognition". "CoRR", "abs/1409.1556", "2014". 8
- [39] S. Singh, D. Litman, M. Kearns, and M. Walker. Optimizing dialogue management with reinforcement learning: Experiments with the njfun system. *Journal of Artificial Intelligence Research*, 16:105–133, 2002. 3
- [40] F. Strub, H. De Vries, J. Mary, B. Piot, A. Courville, and O. Pietquin. End-to-end optimization of goal-driven and visually grounded dialogue systems. *arXiv preprint arXiv:1703.05423*, 2017. 2, 8
- [41] P.-H. Su, M. Gasic, N. Mrksic, L. Rojas-Barahona, S. Ultes, D. Vandyke, T.-H. Wen, and S. Young. Continuously learning neural dialogue management. *arXiv preprint arXiv:1606.02689*, 2016. 3
- [42] J. Suarez, J. Johnson, and F.-F. Li. DDRprog: A CLEVR differentiable dynamic reasoning programmer, 2018. 7
- [43] S. Sukhbaatar, a. szlam, J. Weston, and R. Fergus. End-to-end memory networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2440–2448. Curran Associates, Inc., 2015. 8
- [44] Y. Sun, F. Gomez, and J. Schmidhuber. Planning to be surprised: Optimal bayesian exploration in dynamic environments. In *Proceedings of the 4th International Conference on Artificial General Intelligence*, AGI'11, pages 41–51, Berlin, Heidelberg, 2011. Springer-Verlag. 3
- [45] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998. 3
- [46] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, Cambridge, MA, USA, 1999. MIT Press. 4
- [47] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 3156–3164, 2014. 2

- [48] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8, May 1992. 4
- [49] Q. Wu, C. Shen, A. v. d. Hengel, L. Liu, and A. Dick. What Value Do Explicit High Level Concepts Have in Vision to Language Problems? In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 203–212, 2016. 2
- [50] Q. Wu, P. Wang, C. Shen, A. Dick, and A. v. d. Hengel. Ask Me Anything: Free-form Visual Question Answering Based on Knowledge from External Sources. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 4622–4630, 2016. 2
- [51] Q. Wu, P. Wang, C. Shen, I. Reid, and A. van den Hengel. Are you talking to me? reasoned visual dialog generation through adversarial learning. *arXiv preprint arXiv:1711.07613*, 2017. 3
- [52] J. Zhang, Q. Wu, C. Shen, J. Zhang, J. Lu, and A. Van Den Hengel. Goal-oriented visual question generation via intermediate rewards. In *European Conference on Computer Vision*, pages 189–204. Springer, 2018. 2
- [53] R. Zhao and V. Tresp. Improving goal-oriented visual dialog agents via advanced recurrent nets with tempered policy gradient. In *IJCAI*, 2018. 8
- [54] Y. Zhu, J. J. Lim, and L. Fei-Fei. Knowledge acquisition for visual question answering via iterative querying. *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2017. 3