Community Regularization of Visually-Grounded Dialog

Akshat Agarwal, Swaminathan Gurumurthy, Vasu Sharma, Mike Lewis, Katia Sycara∗
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

aa7@cmu.edu, sgurumur@andrew.cmu.edu, vasus@andrew.cmu.edu, katia@cs.cmu.edu

ABSTRACT

The task of conducting visually grounded dialog involves learning goal-oriented cooperative dialog between autonomous agents who exchange information about a scene through several rounds of questions and answers in natural language. We posit that requiring artificial agents to adhere to the rules of human language, while also requiring them to maximize information exchange through dialog is an ill-posed problem. We observe that humans do not stray from a common language because they are social creatures who live in communities, and have to communicate with many people everyday, so it is far easier to stick to a common language even at the cost of some efficiency loss. Using this as inspiration, we propose and evaluate a multi-agent community-based dialog framework where each agent interacts with, and learns from, multiple agents, and show that this community-enforced regularization results in more relevant and coherent dialog (as judged by human evaluators) without sacrificing task performance (as judged by quantitative metrics).

KEYWORDS

Visual Dialog; Multi Agent Reinforcement Learning; Curriculum Learning; Emergent Communication

ACM Reference Format:

1 INTRODUCTION

AI is increasingly becoming an important part of our daily lives, be it in the household, the workplace or in public places. In order for humans to interact with and understand the AI system, it needs to learn how to communicate with us about our environment using the languages that we speak. This requires the AI system to visually interpret the world, and communicate descriptions of the physical world. While such a task would have been considered impossible a few years ago, the recent progress in the fields of Computer Vision and Natural Language Processing, which are important building blocks for this task, have reinvigorated interest in the community. Several problems like image captioning ([12], [36], [30], [10], [20], [37]), image classification ([14], [29], [8], [34]), object detection ([17], [23], [24]), image segmentation ([18], [9], [22]), dialog ([28], [31], [5]), question answering ([38], [27], [35]) etc. have received immense amounts of attention from the research community. The paradigm of reinforcement learning has also shown promising results in several problems including learning to play Go [26] and Atari games [21], among others, at superhuman levels.

Capitalizing on the growth in all these different domains, it now seems plausible to build more advanced dialog systems capable of reasoning over multiple modalities while also learning from one another. Such systems will allow humans to have a meaningful dialog with intelligent systems containing visual as well as textual content. Use cases include assistive systems for the visually impaired, smart multimodal dialog agents (unlike current versions of Siri and Alexa which are primarily audio based and cannot make effective use of multimodal data) and even large scale visual retrieval systems. However, as these systems become more advanced, it will become increasingly common to have two agents interact with each other to achieve a particular goal [15]. We want these conversations to be interpretable to humans for the sake of transparency and ease of debugging. This motivates our work on goal-driven agents which interact in coherent language understandable to humans.

This paper presents work on Goal driven Visual Dialog Agents. Most prior work on visual dialog ([2], [4]) has approached the problem using supervised learning where, conditioned on the question - answer pair dialog history, a caption $c$ and the image $I$, the agent is required to answer a given question $q$. The model is trained in a supervised learning framework using ground truth supervision from a human-human dialog dataset.

Some recent work [3] has tried to approach the problem using reinforcement learning, with two agents, namely the Question (Q-)
We address this problem of improving the agents’ performance we were to make the agents more social, and make them interact while increasing dialog quality by taking inspiration from humans. This, however, reduces transparency of the agents’ dialog to any observer (human or AI), and is hence undesirable. Prior work \cite{2, 3} has focused on improving performance as measured by the Euclidean distance between the target description and the predicted description. While the generated sentences stop making sense to human observers, the two agents are able to understand each other much better, and the Q-Bot’s understanding of the image improves. This is similar to how twins often develop a private language \cite{25}, an idiosyncratic and exclusive form of communication understandable only to them. This, however, reduces transparency of the agents’ dialog to any observer (human or AI), and is hence undesirable. Prior work \cite{2, 3} which has focused on improving performance as measured by the Q-Bot’s image retrieval rank has suffered from incoherent dialog. We address this problem of improving the agents’ performance while increasing dialog quality by taking inspiration from humans. We observe that humans continue to speak in commonly spoken languages, and hypothesize that this is because they need to communicate with an entire community, and having a private language for each person would be extremely inefficient. With this idea, we let our agents learn in a similar setting, by making them talk to (ask questions of, get answers from) multiple agents, one by one. We claim that if, instead of allowing a single pair of agents to interact, we were to make the agents more social, and make them interact and learn from multiple other agents, they would be disincentivized to develop a private language, and would have to conform to the common language. We call this Community Regularization.

In the subsequent sections we describe the Visual Dialog task and the neural network architectures of our Q-Bots and A-Bots in detail. We then describe the training process of the agents sequentially: (a) in isolation (via supervision), (b) while interacting with one partner agent (via reinforcement), and (c) our proposed multi-agent setup where each agent interacts with multiple other agents (via reinforcement). We compare the performance of the agents trained in these different settings, both quantitatively using image retrieval ranks, and qualitatively evaluating the overall coherence, grammar and relevance of the dialog generated, as judged by impartial human evaluators. We make the following contributions: We propose a multi-agent dialog setup in a natural language setting and show that it results in community regularization which ensures that the interactions between the agents remain grounded in the rules and grammar of natural language, are coherent and human-interpretable without compromising on task performance. The code can be found in the following repository https://github.com/agakshat/visualdialog-pytorch.

## 2 PROBLEM STATEMENT

We begin by defining the problem of Visually Grounded Dialog for the co-operative image guessing game on the VisDial dataset.

**Players and Roles:** The game involves two collaborative agents — a question bot (Q-bot) and an answer bot (A-bot). The A-bot has access to an image and caption, while the Q-bot has access to the image’s caption, but not the image itself. Both the agents share a common objective, which is for the Q-bot to form a good “mental representation” of the unseen image which can be used to retrieve, rank or generate that image. This is facilitated by the exchange of 10 pairs of questions and answers between the two agents, using a shared vocabulary, where the Q-bot asks the A-bot a question about the image, and the A-bot answers the question, hence improving the Q-Bot’s understanding of the image scene.

**General Game Objective:** At each round, in addition to communicating with the A-bot, the Q-bot also provides the learning algorithm with its best estimate $y_t$ of the unknown image $I$ based on the dialog history and caption. Both agents receive a common reward from the environment which is inversely proportional to the error in this description under some metric $L(y_t, y_{gt})$. We note that this is a general setting where the ‘description’ $y_t$ can take on varying levels of specificity – from image feature embeddings extracted by deep neural networks to textual descriptions and pixel-level image re-generations.

**Specific Instantiation:** In our experiments, we focus on the setting where the Q-bot is tasked with estimating a vector embedding of the image $I$, which is later used to retrieve a similar image from the dataset. Given a feature extractor (say, a pretrained CNN model like VGG \cite{29}), the target ‘description’ $y_{gt}$ of the image, can be obtained by simply forward propagating through the VGG model, without the requirement of any human annotation. Reward/error can be measured by the Euclidean distance between the target description $y_{gt}$ and the predicted description $y_t$, and any image may be used as the visual grounding for a dialog. Thus, an unlimited number of games may be simulated without human supervision, motivating the use of reinforcement learning in this framework.

Our primary focus for this work is to ensure that the agents’ dialog remains coherent and understandable while also being informative and improving task performance. For concreteness, consider an example of dialog that is informative yet incoherent: **question:** “do you recognize the guy and age is the adult?”, **answered with:** “you couldn’t be late teens, his”. The example shows that the bots try to extract and convey as much information as possible in a single question/answer (sometimes by incorporating multiple questions or answers into a single statement). But in doing so they lose basic semantic and syntactic structure. We also provide a sample of the dialogs in Figure 2.
Figure 2: A randomly selected image from the VisDial dataset followed by the ground truth (human) and generated dialog about that image for each of our 4 systems (SL, RL-1Q,1A, RL-1Q,3A, RL-3Q,1A). These examples were 2 of the 102 used in the human evaluation results shown in Table 2

3 RELATED WORK

Most of the major works which combine vision and language have traditionally focused on the problem of image captioning ([12], [36], [30], [10], [20], [37]) and visual question answering ([1], [39], [7]). The problem of visual dialog is relatively new and was first introduced by Das et al. [2] who also created the VisDial dataset to advance the research on visually grounded dialog. The dataset was collected by pairing two annotators on Amazon Mechanical Turk to chat about an image. They formulated the task as a ‘multi-round’ VQA task and evaluated individual responses at each round in an image guessing setup. In subsequent work by Das et al. [3] they proposed a reinforcement learning based setup where they allowed the Question bot and the Answer bot to have a dialog with each other with the goal of correctly predicting the image unseen to the Question bot. However, in their work they noticed that the reinforcement learning based training quickly led the bots to diverge from natural language. In fact Kottur et al. [13] recently showed that language emerging from two agents interacting with each other might not even be interpretable or compositional. We use community regularization to alleviate this problem. Recent work has also proposed using such goal driven dialog agents for other related tasks including negotiation [16] and collaborative drawing [11]. We believe that our work can easily extend to those settings as well. Lu et al. [19] proposed a generative-discriminative framework...
We describe all the different components of the agent architectures (c) question decoder and (d) image regression network. Our Q-Bot, shown in Fig. 3a, consists of 4 parts, (a) fact encoder, (b) state-history encoder, (c) question decoder and (d) image regression network. Similar to the original architecture, our Q-Bot, shown in Fig. 3a, also consists of 4 parts, (a) fact encoder, (b) state-history encoder, (c) question decoder and (d) image regression network.

4 AGENT ARCHITECTURES

We describe all the different components of the agent architectures in this section. Note that the overall architecture is mostly borrowed from Das et al. [3], Lu et al. [19] but the individual units have been modified to provide more useful representations. Similar to the original architecture, our Q-Bot, shown in Fig. 3a, also consists of 4 parts, (a) fact encoder, (b) state-history encoder, (c) question decoder and (d) image regression network.

4.1 Question Bot Architecture

The question bot architecture we use is inspired by the answer bot architecture in Das et al. [3], Lu et al. [19] but the individual units have been modified to provide more useful representations. Similar to the original architecture, our Q-Bot, shown in Fig. 3a, also consists of 4 parts, (a) fact encoder, (b) state-history encoder, (c) question decoder and (d) image regression network.

1. **Fact Encoder**: The fact encoder is a unidirectional LSTM which is given the previous question-answer pair \((q_{t-1}, a_{t-1})\) as input. The LSTM generates a fact embedding \(F_t \in \mathbb{R}^{512}\).

2. **State/History Encoder**: We modify the state-history encoder to incorporate a two-level hierarchical encoding of the dialog. The encoder first computes the fact embeddings \(H_t^Q = (F_1, F_2, F_3, ..., F_{t-1})\), using an LSTM akin to the fact encoder described above. We pass these embeddings and \(F_t\) computed by the Fact Encoder through a fully connected layer, generating attention weights which are used to attend over \(H_t^Q\), producing the history embedding \(S_t^Q \in \mathbb{R}^{512}\). Notice that this results in a two-level hierarchical encoding of the dialog \((q_t, a_t) \rightarrow F_t \) and \((F_1, F_2, F_3, ..., F_t) \rightarrow S_t^Q\).

3. **Caption Encoder**: This is a unidirectional LSTM which is given the image caption \(c\) as input. The LSTM generates a caption embedding \(C_t \in \mathbb{R}^{512}\).

4. **Feature Regression Network**: \((F_t^Q, S_t^Q, C_t)\) are concatenated to produce an embedding \(e_t^Q\). This is then passed through 2 fully connected layers with dropout to produce \(\hat{y}_t\) from the current encoded state \(\hat{y}_t = f(S_t^Q)\).

5. **Question Decoder**: The hidden state of this LSTM is initialized with the hidden state of the fact encoder. \(E_t^Q\) is passed through a fully connected layer to generate \(e_t^Q\), which is used to update the hidden state of the LSTM of the question decoder. The question \(q_t\) is then generated by sequentially sampling words (either via teacher forcing during supervised pretraining or via autoregressive generation during RL and evaluation).

Note that we use a dropout of 0.5 in all the LSTMs during training. All LSTM hidden layers sizes are 512, and the image embedding size is 4096. The input word embedding size is 300.

4.2 Answer Bot Architecture

The architecture for A-Bot, also inspired from Lu et al. [19], shown in Fig. 3b, is similar to that of the Q-Bot. It has 3 components: (a) question encoder, (b) state-history encoder and (c) answer decoder.

1. **Question Encoder**: The question encoder is a unidirectional LSTM which is given the current question \(q_t\) generated by the Q-Bot as input. The LSTM generates a question embedding \(Q_t^A \in \mathbb{R}^{512}\).

2. **State/History/Image Encoder**: The encoder first computes the fact embeddings \(H_t^A = (F_1, F_2, F_3, ..., F_{t-1})\), using an LSTM akin to the fact encoder described above. By passing these embeddings and the \(Q_t^A\) computed by the Question Encoder through a fully connected layer, attention weights are generated which are used to attend over \(H_t^A\), producing the history embedding \(S_t^A \in \mathbb{R}^{512}\). Notice that this results in a two-level hierarchical encoding of the dialog \((q_t, a_t) \rightarrow F_t \) and \((F_1, F_2, F_3, ..., F_t) \rightarrow S_t^A\). \((Q_t^A, S_t^A, y_{gt})\) are then concatenated to produce an embedding \(E_t^A\).

3. **Answer Decoder**: The hidden state of this LSTM is initialized with the hidden state of the question encoder. \(E_t^A\) is passed through a fully connected layer to generate \(e_t^A\), which
is used to update the hidden state of the LSTM of the answer decoder. The answer \( a_t \) is then generated by sequentially sampling words (either via teacher forcing during supervised pretraining or via autoregressive generation during RL and evaluation).

5 TRAINING

We follow the training process proposed in Das et al. [3]. Two agents, a Q-Bot and an A-Bot are first trained in isolation, by supervision from the VisDial dataset. After this supervised pretraining for 15 epochs over the data, we smoothly transition the agents to learn from each other via reinforcement learning. The individual phases of training will be described in more detail below. Note that the key novelty of the work is the multi-agent dialog framework proposed in Section 5.3

5.1 Supervised pre-training

In the supervised part of training, both the Q-Bot and A-Bot are trained separately, using a Maximum Likelihood Estimation (MLE) loss computed using the ground truth questions and answers, respectively, for every round of dialog. The Q-Bot simultaneously optimizes another objective: minimizing the Mean Squared Error (MSE) loss between the true \((y_q)\) and predicted \((y_t)\) image embeddings. The ground truth dialogs and image embeddings are from the VisDial dataset.

Given the true dialog history, image features and a question from the dataset, the A-Bot generates an answer to that question. Given the true dialog history and the previous question-answer pair from the dataset, the Q-Bot is made to generate the next question to ask the A-Bot. Both agents receive only ground truth questions and answers, never what the other agent generated - so the two agents never actually interact during this phase of training. However, there are multiple problems with this training scheme. First, MLE is known to result in models that generate repetitive dialogs and often produce generic responses. Second, since the agents are never allowed to interact during training, they end up encountering out-of-distribution questions and answers when made to interact during evaluation, which reduces the task performance. This can be observed in Figure 4. The performance of the agents trained via supervised learning dips after each successive dialog round.

5.2 Reinforcement Learning Setup

To alleviate the issues pointed out with supervised training, we let the two bots interact with each other via self-play (no ground-truth except images and captions). This interaction is also in the form of questions asked by the Q-Bot, and answered in turn by the A-Bot, using a shared vocabulary. The state space is partially observed and asymmetric, with the Q-Bot observing \( \{c, q_1, a_1 \ldots q_{10}, a_{10}\} \) and the A-Bot observing the same, plus the image \( I \). Here, \( c \) is the caption, and \( q_i, a_i \) is the \( i^{th} \) dialog pair exchanged where \( i = 1 \ldots 10 \). The action space for both bots consists of all possible output sequences of a specified maximum length (Q-Bot: 16, A-Bot: 9) under a fixed vocabulary (size 8645). Each action involves predicting words sequentially until a stop token is predicted, or the generated statement reaches the maximum length. Additionally, Q-Bot also produces a guess of the visual representation of the input image (VGG fc-7 embedding of size 4096). Both Q-Bot and A-Bot share the same objective and get the same reward to encourage cooperation. Information gain in each round of dialog is incentivized by setting the reward as the change in distance of the predicted image embedding to the ground-truth image representation. This means that a QA pair is of high quality only if it helps the Q-Bot make a better prediction of the image representation. Both policies are modeled by neural networks, as discussed in Section 4.

A dialog round at time \( t \) consists of the following steps: 1) the Q-Bot, conditioned on the state encoding, generates a question \( q_t \). 2) A-Bot updates its state encoding with \( q_t \) and then generates an answer \( a_t \). 3) Both Q-Bot and A-Bot encode the completed exchange as a fact embedding. 4) Q-Bot updates its state encoding to incorporate this fact and finally 5) Q-Bot predicts the image representation for the unseen image conditioned on its updated state encoding.

Similar to Das et al. [2], we use the REINFORCE [32] algorithm that updates policy parameters in response to experienced rewards. The per-round rewards that are used to calculate the discounted returns follow:

\[
    r_t(s_t^Q, q_t, a_t, y_t) = l(y_{t-1}, y^{Q\theta}) - l(y_t, y^{a\theta})
\]

This reward is positive if the distance between image representation generated at time \( t \) is closer to the ground truth than the representation at time \( t - 1 \), hence incentivizing information gain at each round of dialog. The REINFORCE update rule ensures that a \((q_t, a_t)\) exchange that is informative has its probabilities pushed up. Do note that the image feature regression network \( f \) is trained directly via supervised gradient updates on the L-2 loss.

However, as noted above, this RL optimization problem is ill-posed, since rewarding the agents for information exchange does not motivate them to stick to the rules and conventions of the English language. Thus, we follow an elaborate curriculum scheme described in [2]. Specifically, for the first \( K \) rounds of dialog for each image, the agents are trained using supervision from the VisDial dataset. For the remaining \( 10-K \) rounds, however, they are trained via reinforcement learning. \( K \) starts at 9 and is linearly annealed to 0 over 10 epochs. Despite these modifications the bots are still observed to diverge from natural language and produce non-grammatical and incoherent dialog. Thus, we propose a multi-bot architecture to help the agents interact in diverse and coherent, yet informative, dialog.

5.3 Multi-Agent Dialog Framework (MADF)

In this section we describe our proposed Multi-Agent Dialog architecture in detail. We claim that if, instead of allowing a single pair of agents to interact, we were to make the agents more social, and make them interact and learn from multiple other agents, they would be disincentivized to develop a private language, and would have to conform to the common language. We call this Community Regularization.

In particular, we create either multiple Q-bots to interact with a single A-bot, or multiple A-bots to interact with a single Q-bot. All these agents are initialized with the learned parameters from the supervised pretraining as described in Section 5.1. Then, for each batch of images from the VisDial dataset, we randomly choose a Q-bot to interact with the A-bot, or randomly choose an A-bot to interact with the Q-bot, as the case may be. The two chosen agents...
Table 1: Comparison of answer retrieval metrics with previously published work. SL has the best scores. The scores drop drastically in RL-1Q,1A, but MADF agents (RL-3Q,1A and RL-1Q,3A) are able to retain the same language quality as the SL agent.

<table>
<thead>
<tr>
<th>Model</th>
<th>MRR</th>
<th>Mean Rank</th>
<th>R@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Answer Prior [2]</td>
<td>0.3735</td>
<td>26.50</td>
<td>53.23</td>
</tr>
<tr>
<td>MN-QIH-G [2]</td>
<td>0.5259</td>
<td>17.06</td>
<td>68.88</td>
</tr>
<tr>
<td>HCIAE-G-DIS [19]</td>
<td>0.547</td>
<td>14.23</td>
<td>71.55</td>
</tr>
<tr>
<td>Frozen-Q-Multi [3]</td>
<td>0.437</td>
<td>21.13</td>
<td>60.48</td>
</tr>
<tr>
<td>CoAtt-GAN [33]</td>
<td>0.5578</td>
<td>14.4</td>
<td>71.74</td>
</tr>
<tr>
<td>SL(Ours)</td>
<td>0.610</td>
<td>5.323</td>
<td>72.68</td>
</tr>
<tr>
<td>RL - 1Q,1A(Ours)</td>
<td>0.459</td>
<td>7.097</td>
<td>72.34</td>
</tr>
<tr>
<td>RL - 1Q,3A(Ours)</td>
<td>0.601</td>
<td>5.495</td>
<td>72.48</td>
</tr>
<tr>
<td>RL - 3Q,1A(Ours)</td>
<td>0.590</td>
<td>5.56</td>
<td>72.61</td>
</tr>
</tbody>
</table>

then have a complete dialog consisting of 10 question-answer pairs about each of those images, and update their respective weights based on the rewards received (as per Equation 1) during the conversation, using the REINFORCE algorithm. This process is repeated for each batch of images, over the entire VisDial dataset. It is important to note that histories are not shared across batches. MADF can be understood in detail using the pseudocode in Algorithm 1.

**Connection to Regularization:** It is interesting to note that the MADF setting can actually be seen as a regularizer for the model. To establish this more formally, we look at the total loss minimized by each agent. The Total loss (TL) being minimized during the RL phase = $A_1 + A_2$ at time t, where $A_1$ is negative of the RL reward as described in section 5.2, and $A_2$ is the L-2 loss between predicted and true image embeddings. Consider a setting with N Abots and 1 QBot. The $A_2$ Loss can be written as:

$$A_2 = \sum_{i=1}^{N} (y^{(i)} - y^{gt})^2 = (y^{(1)} - y^{gt})^2 + \sum_{i=2}^{N} (y^{(i)} - y^{gt})^2$$

From the equation, we observe that $A_2$ is a sum of 2 terms, where the first term is the standard regression loss which would apply for the 1Q,1A case. The second term can be viewed as a regularization imposed by the other A-bots with the Q-bot, hence we can rewrite $A_2$ as:

$$A_2 = (y^{(1)} - y^{gt})^2 + R_{A_2}(\theta)$$

(2)

where $R_{A_2}(\theta)$ represents regularization imposed by the other agents. Similarly, $A_1$ can also be broken down into a likelihood and a regularization term as follows:

$$A_1 = -G_1^{(1)} \log p(y^{(1)} | d^{(1)}) + R_{A_1}(\theta)$$

(3)

where $G_1^{(1)}$ is the monte-carlo return calculated using the first pair of agents at time t. Thus, both the terms in the total loss can be broken down into a loss term akin to the 1Q,1A case and a regularization term. This regularization term comes from the regularization imposed by pairing each Q-Bot with multiple A-bots or vice versa. This clearly shows that the multi-bot framework can be seen as a form of regularization. In the experiments we show that the regularization helps with the language quality by ensuring that the bots don’t deviate much from natural language.

Figure 4: Comparison of Task Performance: Image Retrieval Percentile scores. This refers to the percentile scores of the ground truth image compared to the entire test set of 40k images, as ranked by distance from the Q-Bot’s estimate of the image. The X-axis denotes the dialog round number (from 1 to 10), while the Y-axis denotes the image retrieval percentile score. The percentile score decreases monotonically for SL, but is stable for all versions using RL. This shows that the MADF agents are able to capitalize on the benefits of interactive learning.

6 EXPERIMENTS AND RESULTS

6.1 Dataset Description

We use the VisDial 0.9 dataset for our task introduced by Das et al. [2]. The data is collected using Amazon Mechanical Turk by pairing 2 annotators and asking them to chat about the image as a multi round VQA setup. One of the annotators acts as the questioner and has access to only the caption of the image and has to ask questions from the other annotator who acts as the ‘answerer’ and must answer the questions based on the visual information from the actual image. This dialog repeats for 10 rounds at the end of which the questioner has to guess what the image was. We perform our experiments on VisDial v0.9 (the latest available release) containing 83k dialogs on COCO-train and 40k on COCO-val images, for a total of 1.2M dialog question-answer pairs. We split the 83k into 82k for train, 1k for validation, and use the 40k as test, in a manner consistent with [2]. The caption is considered to be the first round in the dialog history.

6.2 Evaluation Metrics

We evaluate the performance of our model’s individual responses by using 4 metrics, proposed by [3]: 1) Mean Reciprocal Rank (MRR). 2) Mean Rank. 3) Recall@10 and 4) Image Retrieval Percentile. Mean Rank and MRR compute the average rank (and its reciprocal, respectively) assigned to the ground truth answer, over a set of 100 candidate answers for each question (also averaged over all the 10 rounds). Recall@10 computes the percentage of answers with rank less (better) than 10. Intuitively all these language metrics are trying to measure similar things, i.e, how highly does
Algorithm 1 Multi-Agent Dialog Framework (MADF)

1: procedure MULTI-BOT-TRAIN
2: while train_iter < max_train_iter do
3:     Qbot ← random_select (Q1, Q2, Q3, ..., Qk)
4:     Abot ← random_select (A1, A2, A3, ..., A_L)
5:     history ← (0, 0, ..., 0)
6:     fact ← (0, 0, ..., 0)
7:     QZ1 ← Ques_enc(Qbot, fact, history, caption)
8:     for t in 1:10 do
9:         ques_t ← Ques_gen(Qbot, QZ_t)
10:        A_z_t ← Ans_enc(Abot, fact, history, image, ques_t, caption)
11:        ans_t ← Ans_gen(Abot, A_z_t)
12:        fact ← [ques_t, ans_t]
13:        history ← concat(history, fact)
14:        QZ_t ← Ques_enc(Qbot, fact, history, caption)
15:        image_pred ← image_regress(Qbot, fact, history, caption)
16:        Rt ← (target_image − image_pred)^2 − ∆image_pred
17:        ∆image_pred ← (target_image − image_pred)^2
18:        end for
19:    ∆w_{Qbot} ← \frac{1}{10} \sum_{t=1}^{10} \nabla_{\theta_{Qbot}} [G_t \log p(q_{\theta_{Qbot}}) − ∆image_pred]
20:    ∆w_{Abot} ← \frac{1}{10} \sum_{t=1}^{10} G_t \nabla_{\theta_{Abot}} \log p(ans_t, \theta_{Abot})
21:    w_{Qbot} ← w_{Qbot} + ∆w_{Qbot}
22:    w_{Abot} ← w_{Abot} + ∆w_{Abot}
23: end while
24: end procedure

Table 2: Human Evaluation Results - Mean Rank (Lower is better): Results show that RL-3Q,1A outperforms RL-1Q,3A for A-relevance and overall coherence but otherwise SL (Supervised Learning), RL-1Q,3A, and RL-3Q,1A showed equivalent performance indicating that community regularization can effectively eliminate any losses to human intelligibility introduced through RL.

<table>
<thead>
<tr>
<th>Metric</th>
<th>N</th>
<th>SK</th>
<th>RL 1Q,1A</th>
<th>RL 1Q,3A</th>
<th>RL 3Q,1A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question Relevance</td>
<td>49</td>
<td>1.97</td>
<td>3.57</td>
<td>2.20</td>
<td>2.24</td>
</tr>
<tr>
<td>Question Grammar</td>
<td>49</td>
<td>2.16</td>
<td>3.67</td>
<td>2.24</td>
<td>1.91</td>
</tr>
<tr>
<td>Overall Dialog Coherence</td>
<td>49</td>
<td>2.08</td>
<td>3.73</td>
<td>2.34</td>
<td>1.83</td>
</tr>
<tr>
<td>Answer Relevance</td>
<td>53</td>
<td>2.09</td>
<td>3.77</td>
<td>2.28</td>
<td>1.84</td>
</tr>
<tr>
<td>Answer Grammar</td>
<td>53</td>
<td>2.20</td>
<td>3.75</td>
<td>2.05</td>
<td>1.98</td>
</tr>
<tr>
<td>Overall Dialog Coherence</td>
<td>53</td>
<td>2.09</td>
<td>3.64</td>
<td>2.35</td>
<td>1.90</td>
</tr>
</tbody>
</table>

the model rank the ground truth answer over a set of 100 candidate responses. Thus, if the model gives a lower rank to the ground truth answer, then we can say that the model is likely highly to produce the ground truth response to the question. But at the same time they do have some qualitative differences. For example, MRR and Rank@10 are more robust to outliers but Mean Rank is not (but it is more interpretable). The fourth metric, i.e., the Image Retrieval percentile, is different from the first 3 metrics. It is a measure of how close the image prediction generated by the Q-bot is to the ground truth. All the images in the test set are ranked according to their distance from the predicted image embedding, and the rank of the ground truth embedding is used to calculate the image retrieval percentile. This gives a measure of the quality of the information exchange. All results are reported after 15 epochs of supervised learning and 10 epochs of curriculum learning as described in Section 5. Consequently, the training time of all 3 systems is equal.

Table 1 compares the Mean Rank, MRR, and Recall@10 of our agent architecture and dialog framework (below the horizontal line) with previously proposed architectures (above the line). SL refers to the agents after only the isolated, supervised training of Section 5.1. RL-1Q,1A refers to a single, idiosyncratic pair of agents trained via reinforcement as in Section 5.2. RL-1Q,3A and RL-3Q,1A refer to social agents trained via our Multi-Agent Dialog framework in Section 5.3, with 1Q,3A referring to 1 Q-Bot and 3 A-Bots, and 3Q,1A referring to 3 Q-Bots and 1 A-Bot.

It can be seen that our agent architectures clearly outperform all previously published results using generative architectures in MRR, Mean Rank and R@10. This indicates that our approach produces consistently good answers (as measured by MRR, Mean Rank and R@10). It is important to note that the point here is not to demonstrate the superiority of our architecture compared to other architectures. The point here is instead to show that the MADF framework (RL-3Q,1A and RL-1Q,3A) is able to maintain the same
We evaluate 4 Visual Dialog systems, trained via: 1) Supervised with single Q-Bot and A-Bot. Kruskal-Wallis tests found strong
There are no quantitative metrics to comprehensively evaluate di-
The scores drop drastically in RL-1Q,1A, but RL-3Q,1A and RL-
1Q,3A obtain scores comparable to SL. This shows that the agents
trained by MADF are able to maintain the language quality of SL
agents without sacrificing much on the task performance (image
retrieval percentile). Fig. 4 shows the change in image retrieval
percentile scores over dialog rounds. The percentile score decreases
monotonically for SL, however it is stable for all versions using
RL. The decrease in image retrieval score over dialog rounds for
SL is because the test set questions and answers are not perfectly
in-distribution (compared to the training set), and the SL system
can’t adapt to these samples as well as the systems trained with
RL. As the dialog rounds increase, the out-of-distribution nature
of dialog exchange increases, hence leading to a decrease in SL
scores. Interestingly, despite having strictly more information in
later rounds, the scores of RL agents do not increase - which we
think is because of the nature of recurrent networks to forget.

The results in Fig. 4 and Table 1 show that the MADF setup
allows the agents to achieve consistent task performance without
sacrificing on language quality. We further support this claim in
the next section where we show that human evaluators rank the
language quality of MADF agents to be much better than the agents
trained via reinforcement without community regularization.

### 6.3 Human Evaluation

There are no quantitative metrics to comprehensively evaluate di-
alog quality, hence we do a human evaluation of the generated
dialog. There are 6 metrics we evaluate on: 1) Q-Bot Relevance, 2)
Q-Bot Grammar, 3) A-Bot Relevance, 4) A-Bot Grammar, 5) Q-Bot
Overall Dialog Coherence and 6) A-Bot Overall Dialog Coherence.
We evaluate 4 Visual Dialog systems, trained via: 1) Supervised
Learning (SL), 2) Reinforce for 1 Q-Bot, 1 A-Bot (RL-1Q,1A), 3)
Reinforce for 1 Q-Bot, 3 A-Bots (RL-1Q,3A) and 4) Reinforce
for 3 Q-Bots, 1 A-Bot (RL-3Q,1A). We asked a total of 61 people
to evaluate the 10 QA-pairs generated by each system for a total
of 102 randomly chosen images, requiring them to give an ordi-
nal ranking (from 1 to 4) for each metric. All the evaluators were
provided with the caption from the dataset. Evaluators taking the
perspective of the A-Bot were provided with the image and asked
to evaluate answer relevance and grammar, while those taking the
perspective of the Q-Bot evaluated question relevance and grammar.
Both groups rated dialogs for overall coherence. Table 2 contains
the average ranks obtained on each metric (lower is better).

The results convincingly validate our hypothesis that having
multiple A-Bots/Q-Bots improves the language quality as compared
with single Q-Bot and A-Bot. Kruskal-Wallis tests found strong
differences among rankings (p<.0001) across all measures. Pairwise
comparisons using the Mann-Whitney U test found a consistent
pattern in which RL 1Q,1A performed substantially worse than
other methods across all measures: for

1. **Q-relevance:** SL: U=348, p<.0001; RL-1Q3A: U=2235, p<
.0001; RL-3Q1A U=2221, p<.0001;
2. **Q-grammar:** SL: U=256, p<.0001; RL-1Q3A U=2741, p<
.0001; RL-3Q1A U=2909, p<.0001;
3. **A-relevance:** SL: U=301, p<.0001; RL-1Q3A U=2741, p<
.0001; RL-3Q1A U=2763, p<.0001;
4. **A-grammar:** SL: U=122, p<.0001; RL-1Q3A U=1945, p<
.0001; RL-3Q1A U=10052, P<.0001.

Results showed that RL 3Q,1A outperformed RL 1Q,3A for A-relevance
U=1889, p<.02 and overall coherence U=6543, p<.006 but other-
wise SL, RL-1Q3A, and RL-3Q,1A showed equivalent performance
indicating that community regularization can effectively eliminate
any losses to human intelligibility introduced through reinforce-
ment learning. These results further support the claims made in
the previous section that the MADF setup allows the agents to show
consistent task performance (image retrieval percentile) while main-
taining the language quality of the supervised agents.

We show a couple of randomly chosen examples from the set
shown to the human evaluators in Fig. 2. The trends observed in
the scores given by human evaluators are also clearly visible in
this example. MADF agents are able to model the human responses
much better than RL 1Q,1A and are about as well as (if not better)
than SL trained agents. It can also be seen that although the RL-
1Q,1A system has greater diversity in its responses, the quality
of those responses is greatly degraded, with the A-Bot’s answers
especially being both non-grammatical and irrelevant.

### 7 DISCUSSION AND CONCLUSION

In this paper we propose a novel community regularization tech-
nique, the Multi-Agent Dialog Framework (MADF), to improve the
dialog quality of artificial agents. We show that training 2 agents
with supervised learning does not ensure good task performance
(measured by the image retrieval percentile scores) at test time, and
it only deteriorates as the agents exchange more information about
the image. We hypothesize that this is because the agents were
trained in isolation and never allowed to interact during supervised
learning, which leads to failure during testing when they encounter
out of distribution samples (generated by the other agent, instead of
ground truth) for the first time. We show how allowing a single pair
of agents to interact and learn from each other via reinforcement
learning dramatically improves their percentile scores, which addi-
tionally does not deteriorate over multiple rounds of dialog, since
the agents have interacted with one another and been exposed to
the other’s generated questions or answers. However, in an attempt
to improve task performance, the agents end up developing their
own private language which does not adhere to the rules and con-
victions of human languages. As a result, the dialog system loses
interpretabiltiy and sociability. To alleviate this issue, we propose
a multi-agent dialog framework to provide regularization. In this
framework, a single A-Bot is allowed to interact with multiple Q-
Bots and vice versa. We first show mathematically that this has
direct connections to regularization. We then back it up with mul-
tiple empirical experiments including a human evaluation study,
and show that MADF leads to significant improvements in dialog
quality measured by relevance, grammar and overall coherence,
without compromising the task performance.