Dialog without Dialog Data: Learning Visual Dialog Agents from VQA Data

Michael Cogswell[5]* **Jiasen Lu**[3]* **Rishabh Jain** [1] **Stefan Lee**[2] **Dhruv Batra**[4,1] **Devi Parikh**[4,1]
[1]Georgia Institute of Technology [2]Oregon State University
[3] Allen Institute for AI [4] Facebook AI Research [5] SRI International
michael.cogswell@sri.com jiasenl@allenai.org leestef@oregonstate.edu
{rishabhjain, dbatra, parikh}@gatech.edu

Abstract

Can we develop visually grounded dialog agents that can efficiently adapt to new tasks without forgetting how to talk to people? Such agents could leverage a larger variety of existing data to generalize to new task, minimizing expensive data collection and annotation. In this work, we study a setting we call "*Dialog without Dialog*", which requires agents to develop visually grounded dialog models that can adapt to new tasks without language level supervision. By factorizing intention and language, our model minimizes linguistic drift after fine-tuning for new tasks. We present qualitative results, automated metrics, and human studies that all show our model can adapt to new tasks and maintain language quality. Baselines either fail to perform well at new tasks or experience language drift, becoming unintelligible to humans. Code has been made available at: `https://github.com/mcogswell/dialog_without_dialog`.

1 Introduction

One goal of AI is to enable humans and computers to communicate naturally with each other in grounded language to achieve a collaborative objective. Recently the community has studied goal oriented dialog, where agents communicate for tasks like booking a flight or searching for images [1].

A popular approach to these tasks has been to observe humans engaging in dialogs like the ones we would like to automate and then train agents to mimic these human dialogs [2, 3]. Mimicking human dialogs allows agents to generate intelligible language (*i.e.*, meaningful English, not gibberish). However, these models are typically fragile and generalize poorly to new tasks. As such, each new task requires collecting new human dialogs, which is a laborious and costly process often requiring many iterations before high quality dialogs are elicited [4, 5].

A promising alternative is to use goal completion as a supervisory signal to adapt agents to new tasks. Specifically, this is realized by pre-training dialog agents via human dialog supervision on one task and then fine-tuning them on a new task by rewarding the agents for solving the task regardless of the dialog's content. This approach can indeed improve task performance, but language quality suffers even for similar tasks. It tends to drifts from human language, becoming ungrammatical and loosing human intelligible semantics – sometimes even turning into unintelligible code. Such code may allow communication with other bots, but is largely incomprehensible to humans. This trade off between task performance and language drift has been observed in prior dialog work [2, 3].

The goal of this paper is to develop visually grounded dialog models that can adapt to new tasks while exhibiting less linguistic drift, thereby reducing the need to collect new data for the new tasks. To test this we consider an image guessing game demonstrated in Fig. 1 (right). In each episode, one agent (A-bot in red) secretly selects a target image y (starred) from a pool of images. The other agent

[*] Equal contribution. Work carried out mainly at Georgia Tech.

Preprint. Under review.

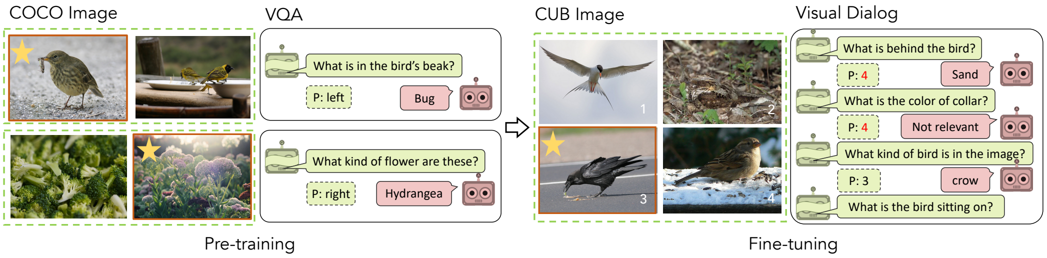

COCO Image
VQA
What is in the bird's beak?
P: left
Bug
What kind of flower are these?
P: right
Hydrangea
Pre-training
CUB Image
1
2
3
4
Visual Dialog
What is behind the bird?
P: 4
Sand
What is the color of collar?
P: 4
Not relevant
What kind of bird is in the image?
P: 3
crow
What is the bird sitting on?
Fine-tuning

Figure 1: An example of *Dialog without Dialog* (DwD) task: (Left) We pre-train questioner agent (Q-bot in green) that can discriminate between pairs of images by mimicking questions from VQAv2 [6]. (Right) Q-bot needs to generate a sequence of discriminative questions (a dialog) to identify the secret image that A-bot picked. Note that the language supervision is not available, thus we can only fine-tune Q-bot with task performance. In DwD, we can test Q-bot generalization ability by varying dialog length, pool size and image domain.

(Q-bot in green) must identify this image by asking questions for A-bot to answer in a dialog. To succeed, Q-bot needs to understand the image pool, generate discriminative questions, and interpret the answers A-bot provides to identify the secret image. The image guessing game provides the agent with a goal, and we can test Q-bot generalization ability by varying dialog length, pool size and image domain.

Contribution 1. We propose the *Dialog without Dialog* (DwD) task, which requires a Q-bot to perform our image guessing game without dialog level language supervision. As shown in in Fig. 1 (left), Q-bot in this setting first learns to ask questions to identify the secret image by mimicking single-round human-annotated visual questions. For the dialog task (right), no human dialogs are available so Q-bot can only be supervised by its image guessing performance. To measure task performance and language drift in increasingly out-of-distribution settings we consider varied pool sizes and take pool images from diverse image sources (*e.g.* close-up bird images).

Contribution 2. We propose a Q-bot architecture for the DwD task that decomposes question intent from the words used to express that intent. We model the question intent by introducing a discrete latent representation that is the only input to the language decoder. We further pair this with a *pre-train then fine-tune* learning approach that teaches Q-bot how to ask visual questions from VQA during pre-training and 'what to ask' during fine-tuning for visual dialogs.

Contribution 3. We measure Q-bot's ability to adapt to new tasks and maintain language quality. Task performance is measured with both automatic and human answerers while language quality is measured using three automated metrics and two human judgement based metrics. Our results show the proposed Q-bot both adapts to new tasks better than a baseline chosen for language quality and maintains language quality better than a baseline optimized for just task performance.

2 Dialog Based Image Guessing Game

2.1 Game Definition

Our image guessing game proceeds one round at a time, starting at round $r = 1$ and running for a fixed number of rounds of dialog R. At round r, Q-bot observes the pool of images $\mathcal{I} = \{I_1, \ldots, I_P\}$ of size P, the history of question answer pairs $q_1, a_1, \ldots q_{r-1}, a_{r-1}$, and placeholder representations q_0, a_0 that provide input for the first round. It generates a question

$$q_r = \texttt{QBot.Ask}(\mathcal{I}, q_0, a_0, \ldots q_{r-1}, a_{r-1}). \tag{1}$$

Given this question, but not the entire dialog history, A-bot answers based on the randomly selected target image $I_{\hat{y}}$ (not known to Q-bot):

$$a_r = \texttt{ABot.Answer}(I_{\hat{y}}, q_r). \tag{2}$$

Once Q-bot receives the answer from A-bot, it makes a prediction y_r guessing the target image:

$$y_r = \texttt{QBot.Predict}(\mathcal{I}, q_0, a_0, \ldots, q_r, a_r) \tag{3}$$

Comparison to GuessWhich. Our Image Guessing game is inspired by GuessWhich game of Das et al. [2], and there are two subtle but important differences. In GuessWhich, Q-bot initially observes

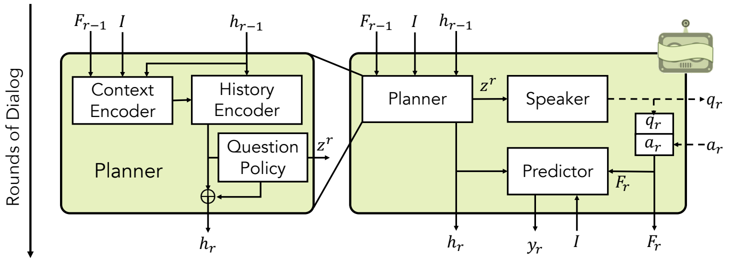


Figure 2: A single round of our Q-bot which decomposes into the modules described in Section 2.3. This factorization allows us to limit language drift while fine-tuning for task performance.

a caption describing A-bot's selected image and must predict the selected image's features to retrieve it from a large, fixed pool of images it does not observe. First, the inclusion of the caption leaves little room for the dialog to add information [7], so we omit it. Second, in our game a small pool of images is sampled for each dialog and Q-bot directly predicts the target image given those choices.

2.2 Modelling A-bot

In this work, we focus primarily on Q-bot agent rather than A-bot. We set A-bot to be a standard visual question answering agent, specifically the Bottom-up Top-down [8] model; however, we do make one modification. Q-bot may generate questions that are not well grounded in A-bot's selected image (though they may be grounded in other pool images) – e.g. asking about a surfer when none exists. To enable A-bot to respond appropriately, we augment A-bot's answer space with a `Not Relevant` token. To generate training data for this token we augment every image with an additional randomly sampled question and set `Not Relevant` as its target answer. A-bot is trained independently from Q-bot on the VQAv2 dataset and then frozen.

2.3 Modelling Q-bot

We conceptualize Q-bot as having three major modules. The *planner* encodes the state of the game to decide what to ask about. The *speaker* takes this intent and formulates the language to express it. The *predictor* makes target image predictions taking the dialog history into account. We make fairly standard design choices here, then adapt this model for the DwD task in Section 3.

Pool & Image Encoding. We represent the p-th image I_p of the pool as a set of B bounding boxes such that I_b^p is the embedding of the b-th box using the same Faster R-CNN [9] embeddings as in [10]. Note that we do not assume prior knowledge about the size or composition of the pool.

Planner. The planner's role is to encode the dialog context (image pool and dialog history) into representation z^r, deciding what to ask about in each round. It also produces an encoding h_r of the dialog history. To limit clutter, we denote the question-answer pair at round r as a 'fact' $F_r = [q_r, a_r]$.

Planner – Context Encoder. Given the prior dialog state h_{r-1}, F_{r-1}, and image pool $\mathcal{I}$, the context encoder performs hierarchical attention over images in $\mathcal{I}$ and object boxes in each image to identify image regions that are most relevant for generating the next question. As we detail in Section 2 of the supplement, F_{r-1} and h_{r-1} query the image to attend to relevant regions across the pool. First a P dimensional distribution α over images in the pool is produced and then a B dimensional distribution β^p over boxes is produced for each image $p \in \{1, \dots, P\}$. The image pool encoding $\hat{v}_r$ at round r is

$$\hat{v}_r = \sum_{p=1}^{P} \alpha_p \sum_{b=1}^{B} \beta_b^p I_b^p. \tag{4}$$

This combines the levels of attention and is agnostic to pool size.

Planner – History Encoder. To track the state of the game, the planner applies an LSTM-based history encoder that takes $\hat{v}_r$ and F_{r-1} as input and produces an intermediate hidden state h_r. Here h_r includes a compact representation of question intent and dialog history, helping provide a differentiable connection between the intent and final predictions through the dialog state.

Planner – Question Policy. The question policy transforms h_r to this module's output z^r, which the speaker decodes into a question. By default z^r is equal to the hidden state h_r, but in Section 3.2 we show how a discrete representations can be used to reduce language drift.

Speaker. Given an intent vector z^r, the speaker generates a natural language question. Our speaker is a standard LSTM-based decoder with an initial hidden state equal to z^r.

Predictor. The predictor uses the dialog context generated so far to guess the target image. It takes a concatenation $F = [F_1, \ldots, F_r]$ of fact embeddings and the dialog state h_r and computes an attention pooled fact $\hat{F}$ using h_r as attention context. Along with h_r, this is used to attend to salient image features then compute a distribution over images in the pool using a softmax (see Algorithm 2 in the supplement for full details), allowing for the use of cross-entropy as the task loss. Note that the whole model is agnostic to pool size.

3 Dialog without Dialog

Aside from some abstracted details, the game setting and model presented in the previous section could be trained without any further information – a pool of images could be generated, A-bot could be assigned an image, the game could be rolled out for arbitrarily many rounds, and Q-bot could be trained to predict the correct image given A-bot's answers. While this is an interesting research direction in its own right [11, 12, 13], there is an obvious shortcoming – it would be highly improbable for Q-bot to discover a fully functional language that humans can already understand. Nobody discovers French. They have to learn it.

At the other extreme – representing standard practice in dialog problems – humans could be recruited to perform this image guessing game and provide dense supervision for what questions Q-bot should ask to perform well at this specific task. However, this requires collecting language data for every new task. It is also intellectually dissatisfying for agents' knowledge of natural language to be so inseparably intertwined with individual tasks. After all, one of the greatest powers of language is the ability to use it to communicate about many different problems.

In this section, we consider a middle ground that has two stages. Stage 1 trains our agent on one task where training data already exists (VQA; *i.e.*, single round dialog) and then stage 2 adapts it to carry out goal driven dialog (image guessing game) without further supervision.

3.1 Stage 1: Language Pre-training

We leverage the VQAv2 [6] dataset as our language source to learn how to ask questions that humans can understand. By construction, for each question in VQAv2 there exists at least one pair of images which are visually similar but have different ground truth answers to the same question. Fortuitously, this resembles our dialog game – the image pair is the pool, the question is guaranteed to be discriminative, and we can provide an answer depending on A-bot's selected image. We view this as a special case of our game that is fully supervised but contains only a single round of dialog. During stage 1 Q-bot is trained to mimic the human question (via cross-entropy teacher forcing) and to predict the correct image given the ground truth answer.

For example, in the top left of Fig. 1 outlined in dashed green we show a pair of two bird images with the question "What is in the bird's beak?" from VQAv2. Our agents engage in a single round dialog where Q-bot asks that question and A-bot provides the answer (also supervised by VQAv2).

3.2 Stage 2: Transferring to Dialog

A first approach for adapting agents would be to take the pre-trained weights from stage 1 and simply fine-tune for our full image guessing task. However, this agent would face a number of challenges. It has never had to model multiple steps of a dialog. Further, while trying to predict the target image there is little to encourage Q-bot to continue producing intelligible language. Indeed, we find our baselines do exhibit language drift. We consider four modifications to address these problems.

Discrete Latent Intention Representation z^r. Rather than a continuous vector passing from the question policy to the speaker, we pass discrete vectors. Specifically, we consider a representation composed of N different K-way Concrete variables [14]. Let $z^r_n \in \{1, \ldots, K\}$ and let the logits $l_{n,1}, \ldots, l_{n,K}$ paramterize the Concrete distribution $p(z^r_n)$. We learn a linear transformation W^z_n from the intermediate dialog state h_r to produce these logits for each variable n:

$$l_{n,k} = \text{LogSoftmax}\,(W^z_n h_r)_k \;\; \forall k \in \{1, \ldots, K\} \; \forall n \in \{1, \ldots, N\} \tag{5}$$

To provide input to the speaker, z^r is embedded using a learned dictionary of embeddings. In our case each variable in z^r has a dictionary of K learned embeddings. The value of z^r_n ($\in \{1, \ldots, K\}$) picks one of the embeddings for each variable and the final representation simply sums over all variables:

$$e_z = \sum_{n=1}^{N} E_n(z_n^r). \tag{6}$$

VAE Pre-training. When using this representation for the intent, we train stage 1 by replacing the likelihood with an ELBO (Evidence Lower BOund) loss as seen in Variational Auto-Encoders (VAEs) [15] to help disentangle intent from expression by restricting information flow through z^r. We use the existing speaker module to decode z^r into questions and train a new encoder module to encode ground truth VQAv2 question $\hat{q}_1$ into conditional distribution $q(z^1|\hat{q}_1, \mathcal{I})$ over z^r at round 1.

For the encoder we use a version of the previously described context encoder from Section 2.3 that uses the question $\hat{q}_1$ as attention query instead of F_{r-1} and h_{r-1} (which are not available in this context). The resulting ELBO loss is

$$\mathcal{L} = E_{z^1 \sim q(z^1|\hat{q}_1, \mathcal{I})} \left[\log p(\texttt{speaker}(z^1)) \right] + \frac{1}{N} \sum_{n=1}^{N} D_{KL} \left[q(z_n^1|\hat{q}_1, \mathcal{I}) || \mathcal{U}(K) \right] \tag{7}$$

This is like the Full ELBO described, but not implemented, in [16]. The first term encourages the encoder to represent and the speaker to mimic the VQA question. The second term uses the KL Divergence D_{KL} to push the distribution of z close to the K-way uniform prior $\mathcal{U}(K)$, encouraging z to ignore irrelevant information. Together, the first two terms form an ELBO on the question likelihood given the image pool [17, 18].

Fixed Speaker. Since the speaker contains only lower level information about how to generate language, we freeze it during task transfer. We want only the high level ideas represented by z and the predictor which receives direct feedback to adapt to the new task. If we updated the speaker then its language could drift given only the sparse feedback available in each new setting.

Adaptation Curriculum. As the pre-trained (stage 1) model has never had to keep track of dialog contexts beyond the first round, we fine-tune in two stages, 2.A and 2.B. In **stage 2.A** we fix the Context Encoder and Question Policy parts of the planner so the model can learn to track dialog effectively without trying to generate better dialog at the same time. This stage takes 20 epochs to train. Once Q-bot learns how to track dialog we update the entire planner in **stage 2.B** for 5 epochs.[2]

4 Experiments

We want to show that our proposed agent can adapt to new tasks while exhibiting less linguistic drift. In Section 4 and Section 4.1 we start by describing the new tasks we construct and the baselines we compare to, then the following sections demonstrate how our model adapts while preventing drift using qualitative examples (Section 4.2), automated metrics (Section 4.3), and human judgements (Section 4.4). We also summarize the model ablations (Section 12) detailed in the supplement.

Task Settings. We construct new tasks by varying four parameters of our image guessing game:

- **Number of Dialog Rounds.** The number of dialog rounds R is fixed at 1, 5, or 9.
- **Pool Size.** The number of images in a pool P to 2, 4, or 9.
- **Image Domain.** By default we use VQA images (*i.e.*, from COCO [19]), but we also construct pools using CUB (bird) images [20] and AWA (animal) images [21].
- **Pool Sampling Strategy.** We test two ways of sampling pools of images. The Constrast sampling method, required for pre-training (Section 3.1), chooses a pair of images with contrasting answers to the same question from VQAv2. This method only works for $P = 2$. The Random sampling method chooses P images at random from the images available in the split.

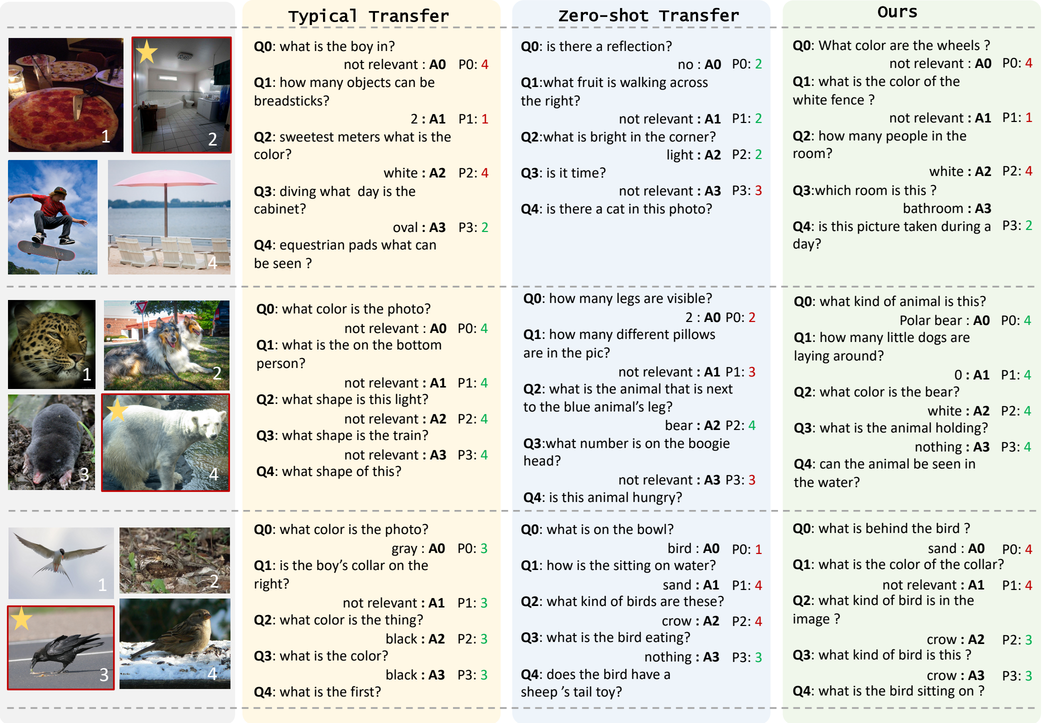
For example, consider the 'VQA - 2 Contrast - 5 Round' setting. These pools are constructed from 2 VQA images with the Contrast sampling strategy and dialogs are rolled out to 5 rounds.

4.1 Baselines

We compare our proposed approach to two baselines – **Zero-shot Transfer** and **Typical Transfer** – ablating aspects of our model that promote adaptation to new tasks or prevent language drift.

[2] We find 5 epochs stops training early enough to avoid overfitting on our val set.

The **Zero-shot Transfer** baseline is our model after the single round fully supervised pre-training. Improvements over this model represent gains made from task based fine-tuning. The **Typical Transfer** baseline is our model under standard encoder-decoder dialog model design choices – *i.e.*, a continuous latent variable, maximum likelihood pre-training, and fine-tuning the speaker module. Improvements over this model represent gains made from the modifications aimed at preventing language drift described in Section 3.2 – specifically, the discrete latent variable, ELBO pre-training, and frozen speaker module.



Typical Transfer	Zero-shot Transfer	Ours
Q0: what is the boy in? not relevant : **A0** P0: 4 **Q1**: how many objects can be breadsticks? 2 : **A1** P1: 1 **Q2**: sweetest meters what is the color? white : **A2** P2: 4 **Q3**: diving what day is the cabinet? oval : **A3** P3: 2 **Q4**: equestrian pads what can be seen ?	**Q0**: is there a reflection? no : **A0** P0: 2 **Q1**:what fruit is walking across the right? not relevant : **A1** P1: 2 **Q2**:what is bright in the corner? light : **A2** P2: 2 **Q3**: is it time? not relevant : **A3** P3: 3 **Q4**: is there a cat in this photo?	**Q0**: What color are the wheels ? not relevant : **A0** P0: 4 **Q1**: what is the color of the white fence ? not relevant : **A1** P1: 1 **Q2**: how many people in the room? white : **A2** P2: 4 **Q3**:which room is this ? bathroom : **A3** **Q4**: is this picture taken during a day? P3: 2
Q0: what color is the photo? not relevant : **A0** P0: 4 **Q1**: what is the on the bottom person? not relevant : **A1** P1: 4 **Q2**: what shape is this light? not relevant : **A2** P2: 4 **Q3**: what shape is the train? not relevant : **A3** P3: 4 **Q4**: what shape of this?	**Q0**: how many legs are visible? 2 : **A0** P0: 2 **Q1**: how many different pillows are in the pic? not relevant : **A1** P1: 3 **Q2**: what is the animal that is next to the blue animal's leg? bear : **A2** P2: 4 **Q3**:what number is on the boogie head? not relevant : **A3** P3: 3 **Q4**: is this animal hungry?	**Q0**: what kind of animal is this? Polar bear : **A0** P0: 4 **Q1**: how many little dogs are laying around? 0 : **A1** P1: 4 **Q2**: what color is the bear? white : **A2** P2: 4 **Q3**: what is the animal holding? nothing : **A3** P3: 4 **Q4**: can the animal be seen in the water?
Q0: what color is the photo? gray : **A0** P0: 3 **Q1**: is the boy's collar on the right? not relevant : **A1** P1: 3 **Q2**: what color is the thing? black : **A2** P2: 3 **Q3**: what is the color? black : **A3** P3: 3 **Q4**: what is the first?	**Q0**: what is on the bowl? bird : **A0** P0: 1 **Q1**: how is the sitting on water? sand : **A1** P1: 4 **Q2**: what kind of birds are these? crow : **A2** P2: 4 **Q3**: what is the bird eating? nothing : **A3** P3: 3 **Q4**: does the bird have a sheep 's tail toy?	**Q0**: what is behind the bird ? sand : **A0** P0: 4 **Q1**: what is the color of the collar? not relevant : **A1** P1: 4 **Q2**: what kind of bird is in the image ? crow : **A2** P2: 3 **Q3**: what kind of bird is this ? crow : **A3** P3: 3 **Q4**: what is the bird sitting on ?

Figure 3: Qualitative comparison of dialogs generated by our model with those generated by Typical Transfer and Zero-shot Transfer baselines. Top / middle / bottom rows are image pool from COCO / AWA / CUB images respectively. Our model is pre-trained on VQA (COCO images) and generates more intelligible questions on out-of-domain images.

4.2 Qualitative Results

Figure 3 shows example outputs of the Typical Transfer and Zero-shot Transfer baselines alongside our Q-bot on VQA, AWA and CUB images using size 4 Randomly sampled pools and 5 rounds of dialog. Both our model and the Typical Transfer baseline tend to guess the target image correctly, but it is much easier to tell what the questions our model asks mean and how they might help with guessing the target image. On the other hand, questions from the Zero-shot Transfer baseline are clearly grounded in the images, but they do not seem to help guess the target image and the Zero-shot Transfer baseline indeed fails to guess correctly. This is a pattern we will reinforce with quantitative results in Section 4.4 and Section 4.3.

These examples and others we have observed suggest interesting patterns that highlight A-bot. Our automated A-bot based on [10] does not always provide accurate answers, limiting the questions Q-bot can usefully ask. When there is signal in the answers, it is not necessarily intelligible, providing an opportunity for Q-bot's language to drift.

4.3 Automated Evaluation

We consider metrics addressing both **Task** performance and **Language** quality. While task performance is straightforward (did Q-bot guess the correct target image?), language quality is harder to measure. We describe three automated metrics here and further investigate language quality using human evaluations in Section 4.4.

Task – Guessing Game Accuracy. To measure task performance so we report the accuracy of Q-bot's target image guess at the final round of dialog.

Language – Question Relevance via A-bot. To be human understandable, the generated questions should be relevant to at least one image in the pool. We measure question relevance as the maximum question image relevance across the pool as measured by A-bot, i.e., $1 - p(\texttt{Not Relevant})$. We note that this is only a proxy for actual question relevance as A-bot may report `Not Relevant` erroneously if it fails to understand Q-bot's question; however, in practice we find A-bot does a fair job in determining relevance. We also provide human relevance judgements in Section 4.4.

Language – Fluency via Perplexity. To evaluate Q-bot's fluency, we train an LSTM-based language model on the corpus of questions in VQA. This allows us to evaluate the perplexity of the questions generated by Q-bot for dialogs on its new tasks. Lower perplexity indicates the generated questions are similar to VQA questions in terms of syntax and content. However, we note that questions generated for the new tasks could have lower perplexity because they have drifted from English or because different things must be asked for the new task, so lower perplexity is not always better [22].

Language – Diversity via Distinct n-grams. This considers the set of all questions generated by Q-bot across all rounds of dialog on the val set. It counts the number of n-grams in this set, G_n, and the number of distinct n-grams in this set, D_n, then reports $\frac{G_n}{D_n}$ for each value of $n \in \{1, 2, 3, 4\}$. Note that instead of normalizing by the number of words as in previous work [23, 24], we normalize by the number of n-grams so that the metric represents a percentage for values of n other than $n = 1$. Generative language models frequently produce safe standard outputs [23], so diversity is a sign this problem is decreasing, but diversity by itself does not make language meaningful or useful.

Results. Table 1 presents results on our val set for our model and baselines across the various settings described in Section 4. Agents are tasked with generalizing further and further from their source language data. Setting `A` is the same as for stage 1 pre-training. In that same column, `B` and `C` require generalization to multiple rounds of dialog and Randomly sampled image pairs instead of pools sampled with the Contrast strategy. In the right side of Tab. 1 we continue to test generalization farther from the language source using more images and rounds of dialog (`D`) and then using different types of images (`E` and `F`). Our model performs well on both task performance and language quality across the different settings in terms of these automatic evaluation metrics. Other notable findings are:

			Accuracy ↑	Perplexity↓	Relevance ↑	Diversity↑				Accuracy ↑	Perplexity↓	Relevance ↑	Diversity↑
VQA 2 Contrast 1 Round	`A1`	Zero-shot Transfer	0.73	2.62	0.87	0.50	**VQA** 9 Random 9 Rounds	`D1`	Zero-shot Transfer	0.18	2.72	**0.77**	1.11
	`A2`	Typical Transfer	0.71	10.62	0.66	**5.55**		`D2`	Typical Transfer	**0.78**	40.66	**0.77**	**2.57**
	`A3`	Ours	**0.82**	**2.6**	**0.88**	0.54		`D3`	Ours	0.53	**2.55**	0.75	0.95
VQA 2 Contrast 5 Rounds	`B1`	Zero-shot Transfer	0.67	2.62	0.87	0.50	**AWA** 9 Random 9 Rounds	`E1`	Zero-shot Transfer	0.47	2.49	**0.96**	0.24
	`B2`	Typical Transfer	0.74	10.62	0.66	**5.55**		`E2`	Typical Transfer	0.48	12.56	0.64	**2.21**
	`B3`	Ours	**0.87**	**2.60**	**0.88**	0.54		`E3`	Ours	**0.74**	**2.41**	**0.96**	0.28
VQA 2 Random 5 Rounds	`C1`	Zero-shot Transfer	0.64	**2.64**	0.75	1.73	**CUB** 9 Random 9 Rounds	`F1`	Zero-shot Transfer	0.36	2.56	**1.00**	0.04
	`C2`	Typical Transfer	0.86	16.95	0.62	**8.13**		`F2`	Typical Transfer	0.38	20.92	0.47	**2.16**
	`C3`	Ours	**0.95**	2.69	**0.77**	2.34		`F3`	Ours	**0.74**	**2.47**	**1.00**	0.04

Table 1: Performance of our models and baselines in different experimental settings. From setting `A` to setting `F`, agents are tasked with generalizing further from the source data. Our method strikes a balance between guessing game performance and interpretability.

Ours *vs*. Zero-shot Transfer. To understand the relative importance of the proposed stage 2 training which transferring to dialog for DwD task, we compared the task accuracy of our model with that of Zero-shot Transfer. In setting, `A` which matches the training regime, our model outperforms Zero-shot Transfer by 9% (`A3` *vs*. `A1`) on task performance. As the tasks differ in settings B-F, we see further gains with our model consistently outperforming Zero-shot Transfer by 20-38%. Despite these gains, our model maintains similar language perplexity, A-bot relevance, and diversity.

Ours *vs*. Typical Transfer. Our discrete latent variable, variational pre-training objective, and fixed speaker play an important role in avoiding language drift. Compared to the Typical Transfer model without these techniques, our model achieves over 4x (`A2` / `A3`) lower perplexity and 10-53% better A-bot Relevance. Our model also improves in averaged accuracy, which means more interpretable language also improves the task performance. Note that Typical Transfer has 2-100x higher diversity compared to our model, which is consistent with the gibberish we observe from that model (*e.g.*, in Fig. 3) and further suggests its language is drifting away from English.

Results from Game Variations. We consider the following variations on the game:

– **Dialog Rounds.** Longer dialogs (more rounds) achieve better accuracy (A3 vs B3).

- **Pool Sampling Strategy.** As expected, Random pools are easier compared to Contrast pools (B3 vs C3 accuracy), however language fluency and relevance drop on the Random pools (B3 vs C3 perplexity and a-bot relevance).
- **Image Source.** CUB and AWA pools are harder compared to COCO image domain (D3 vs E3 vs F3). Surprisingly, our models maintains similar perplexity and high a-bot relevance even on these out-of-domain image pools. The Zero-shot Transfer and Typical Transfer baselines generalize poorly to these different image domains – reporting task accuracies nearly half our model performance.

4.4 Human Studies

We also evaluate our models by asking if humans can understand Q-bot's language. Specifically, we use workers (turkers) on Amazon Mechanical Turk to evaluate the relevance, fluency, and task performance of our models. Section 3 of the supplement details these studies, but we briefly summarize the results here.

Turkers from Amazon's Mechanical Turk considered the questions from our model more relevant to the image pool than those from the Typical Transfer model and about equally as relevant as the Zero-shot Transfer model's questions. Similarly, they considered our model's questions more fluent than the Typical Transfer model questions and equally as fluent as the Zero-shot Transfer model's questions. However, when we used used turkers to answer Q-bot's questions – replacing the automated A-bot – our Q-bot was able to guess the correct image 69% of the time while the Typical Transfer only achieved 45% accuracy and the Zero-shot Transfer model achieved 23% accuracy. This again confirms that our model can adapt to new tasks with minimal sacrifice to language quality.

4.5 Model Ablations

We investigate the impact of our modelling choices from Section 3.2 by ablating these choices in Section 5 of the supplement, summarizing the results here. The choice of discrete instead of continuous z^r helps maintain language quality, as does the use of variational (ELBO) pre-training instead of maximum likelihood. Surprisingly, the ELBO loss probably has more impact than the discreteness of z^r. Fixing the speaker module during stage 2 also had a minor role in discouraging language drift. Finally, we find that improvements in task performance are due more to learning to track the dialog in stage 2.A than they are due to asking more discriminative questions.

5 Related Work

Our interest comes from language drift problems encountered when using models comparable to the Zero-shot Transfer baseline. In [3] a dataset is collected with question supervision then fine-tuning is used in an attempt to increase task performance, but the resulting utterances are unintelligible. Similarly, [2] takes a very careful approach to fine-tuning for task performance but finds that language also diverges, becoming difficult for humans to understand. Neither approach uses a discrete latent variables or a multi stage training curriculum, as in our proposed model. Furthermore, these models need to be adapted to work in our new setting, and doing so would yield models very similar to our Typical Transfer baseline.

More recently, [25] observe language drift in a translation game from French to German. They reduce drift by supervising communications between agents with auxiliary translations to English and grounding in images. This setting is somewhat different than ours since grounding is directly necessary to solve our task. The approach also requires direct supervision on the communication channel, which is not practical for a multiple round dialog game like ours.

We used a visual reference game to study question generation, improving the quality of generated language using concepts related to latent action spaces. Some works like [26] and [27] also aim to ask visual questions with limited question supervision. Other works represent dialog using latent action spaces [16, 28, 29, 30, 31, 32, 30, 33, 34, 35, 36]. Finally, reference games are generally popular for studying language [37, 5, 38]. Section 6 of the supplement describes the relationship between our approach and these works in more detail.

6 Conclusion

In this paper we proposed the Dialog without Dialog (DwD) task along with a model designed to solve this task and an evaluation scheme that takes its goals into account. The task is to perform the image guessing game, maintaining language quality without dialog level supervision. This balance is hard to strike, but our proposed model manages to strike it. Our model approaches this task by representing dialogs with a discrete latent variable and carefully transfering language information via multi stage training. While baseline models either adapt well to new tasks or maintain language quality and intelligibility, our model is the only one to do both according to both automated metrics and human judgements. We hope these contributions help inspire useful dialog agents that can also interact with humans.

7 Broader Impact

We think the main ethical aspects of this work and their consequences for society have to do with fairness. There is an open research problem around existing deep learning models often reflecting and amplifying undesirable biases that exist in society.

While visual question answering and visual dialog models do not currently work well enough to be relied on in the real world (largely because of the aforementioned proneness to bias), they could be deployed in applications where these biases could have negative impacts on fairness in the future. For example, visually impaired users might use these models to understand visual aspects of their world [39]. If these models are not familiar with people in certain contexts (e.g., men shopping) or are only used to interacting with certain users (e.g., native English speakers) then they might fail for some sub-groups (e.g., non-native English speakers who go shopping with men) but not others.

Our research model may be prone to biases, though it was trained on the VQAv2 dataset [6], which aimed to be more balanced than its predecessor. However, by increasing the intelligibility of generated language our work may help increase the overall interpretability of models. This may help by making bias easier to measure and providing additional avenues for correcting it.

8 Acknowledgements

The Georgia Tech effort was supported in part by NSF, AFRL, DARPA, ONR YIPs, ARO PECASE, Amazon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government, or any sponsor.

References

[1] A. H. Miller, W. Feng, A. Fisch, J. Lu, D. Batra, A. Bordes, D. Parikh, and J. Weston. Parlai: A dialog research software platform. *arXiv preprint arXiv:1705.06476*, 2017.

[2] Abhishek Das, Satwik Kottur, José M.F. Moura, Stefan Lee, and Dhruv Batra. Learning cooperative visual dialog agents with deep reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[3] Mike Lewis, Denis Yarats, Yann Dauphin, Devi Parikh, and Dhruv Batra. Deal or no deal? end-to-end learning for negotiation dialogues. In *EMNLP*, 2017.

[4] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José M. F. Moura, Devi Parikh, and Dhruv Batra. Visual dialog. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1080–1089, 2017.

[5] Harm de Vries, Florian Strub, A. P. Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron C. Courville. Guesswhat?! visual object discovery through multi-modal dialogue. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4466–4475, 2016.

[6] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[7] Mircea Mironenco, Dana Kianfar, Ke Tran, Evangelos Kanoulas, and Efstratios Gavves. Examining cooperation in visual dialog models. *arXiv preprint arXiv:1712.01329*, 2017.

[8] Damien Teney, Peter Anderson, Xiaodong He, and Anton van den Hengel. Tips and tricks for visual question answering: Learnings from the 2017 challenge. *CoRR*, abs/1708.02711, 2017.

[9] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015.

[10] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018.

[11] Yi Ren, Shangmin Guo, Matthieu Labeau, Shay B. Cohen, and Simon Kirby. Compositional languages emerge in a neural iterated learning model. *ICLR*, 2020.

[12] Rahma Chaabouni, Eugene Kharitonov, Diane Bouchacourt, Emmanuel Dupoux, and Marco Baroni. Compositionality and generalization in emergent languages. *ArXiv*, abs/2004.09124, 2020.

[13] Paul Pu Liang, Junqiao Chen, Ruslan Salakhutdinov, Louis-Philippe Morency, and Satwik Kottur. On emergent communication in competitive multi-agent teams. *ArXiv*, abs/2003.01848, 2020.

[14] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations (ICLR)*, 2017.

[15] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *ICLR*, 2014.

[16] Tiancheng Zhao, Kaige Xie, and Maxine Eskénazi. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. In *NAACL-HLT*, 2019.

[17] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparametrization with gumble-softmax. In *International Conference on Learning Representations (ICLR)*, 2017.

[18] Lukasz Kaiser, Aurko Roy, Ashish Vaswani, Niki Parmar, Samy Bengio, Jakob Uszkoreit, and Noam Shazeer. Fast decoding in sequence models using discrete latent variables. In *ICML*, 2018.

[19] T.Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014.

[20] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[21] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[22] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *CoRR*, abs/1511.01844, 2015.

[23] Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. Diverse beam search: Decoding diverse solutions from neural sequence models. *AAAI*, 2018.

[24] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B. Dolan. A diversity-promoting objective function for neural conversation models. In *HLT-NAACL*, 2015.

[25] Jason Lee, Kyunghyun Cho, and Douwe Kiela. Countering language drift via visual grounding. In *EMNLP/IJCNLP*, 2019.

[26] Ishan Misra, Ross B. Girshick, Rob Fergus, Martial Hebert, Abhinav Gupta, and Laurens van der Maaten. Learning by asking questions. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11–20, 2017.

[27] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Visual curiosity: Learning to ask questions to learn visual recognition. In *CoRL*, 2018.

[28] Tiancheng Zhao, Kyusong Lee, and Maxine Eskénazi. Unsupervised discrete sentence representation learning for interpretable neural dialog generation. In *ACL*, 2018.

[29] Tiancheng Zhao and Maxine Eskénazi. Zero-shot dialog generation with cross-domain latent actions. In *SIGDIAL Conference*, 2018.

[30] Denis Yarats and Mike Lewis. Hierarchical text generation and planning for strategic dialogue. *arXiv preprint arXiv:1712.05846*, 2017.

[31] Tsung-Hsien Wen, Yishu Miao, Phil Blunsom, and Steve Young. Latent intention dialogue models. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3732–3741. JMLR. org, 2017.

[32] Iulian V Serban, II Ororbia, G Alexander, Joelle Pineau, and Aaron Courville. Piecewise latent variables for neural variational text processing. *arXiv preprint arXiv:1612.00377*, 2016.

[33] Hengyuan Hu, Denis Yarats, Qucheng Gong, Yuandong Tian, and Mike Lewis. Hierarchical decision making by generating and following natural language instructions. *arXiv preprint arXiv:1906.00744*, 2019.

[34] Dongyeop Kang, Anusha Balakrishnan, Pararth Shah, Paul Crook, Y-Lan Boureau, and Jason Weston. Recommendation as a communication game: Self-supervised bot-play for goal-oriented dialogue. *arXiv preprint arXiv:1909.03922*, 2019.

[35] Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[36] Jason D Williams, Kavosh Asadi, and Geoffrey Zweig. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. *arXiv preprint arXiv:1702.03274*, 2017.

[37] David Lewis. *Convention: A philosophical study*. John Wiley & Sons, 2008.

[38] Prithvijit Chattopadhyay, Deshraj Yadav, Viraj Prabhu, Arjun Chandrasekaran, Abhishek Das, Stefan Lee, Dhruv Batra, and Devi Parikh. Evaluating visual conversational agents via cooperative human-ai games. In *Proceedings of the Fifth AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*, 2017.

[39] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. *CVPR*, 2018.

[40] Jiasen Lu, Anitha Kannan, Jianwei Yang, Devi Parikh, and Dhruv Batra. Best of both worlds: Transferring knowledge from discriminative learning to a generative visual dialog model. In *NIPS*, 2017.

9 Supplement Overview

This document contains supplementary material for "Dialog without Dialog Data: Learning Visual Dialog Agents from VQA Data". The main paper excludes some details which we provide here. Section 10 describes the Q-bot proposed in the main paper in more detail, including algorithms that show how it executes one round of dialog. Section 11 describes the human studies we use to evaluate our model and reports those results in detail. Section 12 reports the ablations we use to evaluate the effects of different aspects of the proposed Q-bot. Section 13 reports how various models we consider perform at different rounds of dialog. Finally, Section 14 explores in more depth how our work relates to other relevant work in the literature.

10 Architecture Details

This section describes our architecture in more detail. Algorithm 1 summarizes our complete `QBot()` implementation and subsequent algorithms define the subroutines used inside `QBot()` along with the encoder we use for variational pre-training. The planner module is described in Algorithm 3, the predictor is described in Algorithm 2, and the speaker is described in Algorithm 4. Algorithm 5 describes the encoder used for the ELBO loss.

Note that the number of bounding boxes per image is B, the number of images in a pool is P, and the max question length is T.

There is a minor notation difference between this section and the main paper. In this section there is an additional hidden state $\bar{h}_r$ that parallels h_r and is used only inside the planner. While h_r is the hidden state of an LSTM, $\bar{h}_r$ is computed in the same way except it uses a different output gate (see line 11 of Algorithm 3). This is essentially a second LSTM output that allows the context coder query to forget dialog history information irrelevant to the current round, and allowing h_r to focus on representing the entire dialog state.

Algorithm 1: Question Bot

```
Function QBot($\mathcal{I}, h_{r-1}, \bar{h}_{r-1}, q_0, a_0, \ldots, q_{r-1}, a_{r-1}$)
    Input: $\mathcal{I}, h_{r-1}, \bar{h}_{r-1}, q_0, a_0, \ldots, q_{r-1}, a_{r-1}$
    Output: $q_r, h_r, \bar{h}_r, y_r$
    $y_r \leftarrow$ Predictor($\mathcal{I}, h_{r-1}, q_0, a_0, \ldots, q_{r-1}, a_{r-1}$)
    $h_r, \bar{h}_r, z^r \leftarrow$ Planner($\mathcal{I}, q_{r-1}, a_{r-1}, h_{r-1}, \bar{h}_{r-1}$)
    $q_r \leftarrow$ Speaker($z^r$)
    return $q_r, h_r, \bar{h}_r, y_r$
```

Algorithm 2: Predictor

```
Function Predictor($\mathcal{I}, h_{r-1}, q_0, a_0, \ldots, q_{r-1}, a_{r-1}$)
    Input: $\mathcal{I}$ ($I_p^b \in \mathbb{R}^{2048}$), $h_{r-1}, q_0, a_0, \ldots, q_{r-1}, a_{r-1}$
    Output: $y_r$
    $\text{Attention}(Q, K, V) = \text{Softmax}\left(g_3(g_1(Q) \odot g_2(K))\right) V$
    $f_{r-1} \leftarrow [E_q(q_{r-1}), E_a(a_{r-1})]$ /* fact                         */
    $F \leftarrow [f_0, \ldots, f_{r-1}]$
    /* Attention over rounds                                                      */
    $e_F \leftarrow \text{Attention}(h_{r-1}, F, F)$
    $Q_y \leftarrow [h_{r-1}, e_F]$
    /* Attention over bounding boxes                                              */
    $e_I \leftarrow \text{Attention}(Q_y, \mathbf{x}, \mathbf{x}) \in \mathbb{R}^{P \times 2048}$
    $e_I \leftarrow g_1(e_I)$
    $Q_p \leftarrow g_2(Q_y)$
    $l_y \leftarrow g_3(Q_p \odot e_I)$
    $y_r \leftarrow \text{argmax}\,\text{Softmax}\,(l_y)$
    return $y_r$
```

In the planner Algorithm 3 at lines 5 and 6 g, f_1, f_3 are all two layer MLPs with ReLU output and weight norm. Both f_2 and f_4 are linear transformations with weight norm applied (no activation function). f_5 is a linear transformation without weight norm purely for dimensionality reduction. To compute $\bar{h}_r$ we also add new linear weights W_1 and W_2 as for a standard LSTM output gate.

Note that for the planner there is an additional residual connection at line 16 which augments the hidden state. This allows gradients to flow through the question policy parameters W^z at line 13 when we fine-tune for task performance without fully supervised dialogs.

In Algorithm 2 g_1, g_2 are both 2-layer ReLU nets with weight norm. Also g_3 is a 2-layer net with ReLU and Dropout on the hidden activation and weight norm on both layers.

In Algorithm 4 β is an LSTM decoder.

Algorithm 3: Planner

```
Function Planner($(\mathcal{I}, q_{r-1}, a_{r-1}, h_{r-1}, \bar{h}_{r-1})$)
    Input: $\mathcal{I}$ ($I_b^p \in \mathbb{R}^{2048}$), $q_{r-1}$, $a_{r-1}$, $h_{r-1}$, $\bar{h}_{r-1}$
    Output: $h_r$, $\bar{h}_r$, $z^r$
    /* Context Coder                                                  */
    $e_q \leftarrow E_q(q_{r-1})$
    $e_a \leftarrow E_a(a_{r-1})$
    $e_c \leftarrow f_5([\bar{h}_{r-1}, e_q, e_a])$
    $\alpha_p \leftarrow \mathrm{Softmax}\left(f_2(g(e_c) \odot f_1(I_b^p))\right)$
    $\beta_b^p \leftarrow \mathrm{Softmax}\left(f_4(g(e_c) \odot f_3(I_b^p))\right)$
    $\hat{v}_{r-1} \leftarrow \sum_{p=1}^{P} \sum_{b=1}^{B} \alpha_p \beta_b^p I_b^p$
    $x_{r-1}^{context} \leftarrow [\hat{v}_{r-1}, e_q, e_a]$
    /* Dialog RNN                                                     */
    $h_r, c_r \leftarrow \gamma(x_{r-1}^{context}, h_{r-1})$
    $h_r \leftarrow \mathrm{Dropout}(h_r)$
    $\bar{h}_r \leftarrow \sigma(W_1^T x_{r-1}^{context} + W_2^T h_{r-1}) \odot \tanh(c_r)$
    $\bar{h}_r \leftarrow \mathrm{Dropout}(\bar{h}_r)$
    /* Question Policy                                                */
    $l_{n,k} \leftarrow \mathrm{LogSoftmax}\left(W_n^z h_r\right)_k \;\; \forall k \in \{1, \ldots, K\} \; \forall n \in \{1, \ldots, N\}$
    $z_n^r \leftarrow \mathrm{GumbelSoftmax}(l_n) \; \forall n \in \{1, \ldots, N\}$
    $h_r \leftarrow h_r + \mathrm{ReLU}\left(W^l l\right)$
    return $h_r, \bar{h}_r, z^r$
```

Algorithm 4: Speaker

```
Function Speaker($(z)$)
    Input: $z$
    Output: $q_{r+1}$
    $e_z \leftarrow \sum_{n=0}^{N-1} E_n^z(z_n)$
    $q_{r+1} \leftarrow \beta(e_z)$
    return $q_{r+1}$
```

11 Human Evaluation

As summarized in Section 4.5 of the main paper, we also evaluate our models by asking if humans can understand Q-bot's language. Specifically, we use workers (turkers) on Amazon Mechanical Turk to evaluate the relevance, fluency, and task performance of our models. We discuss each study below and report results for all studies in Fig. 4.

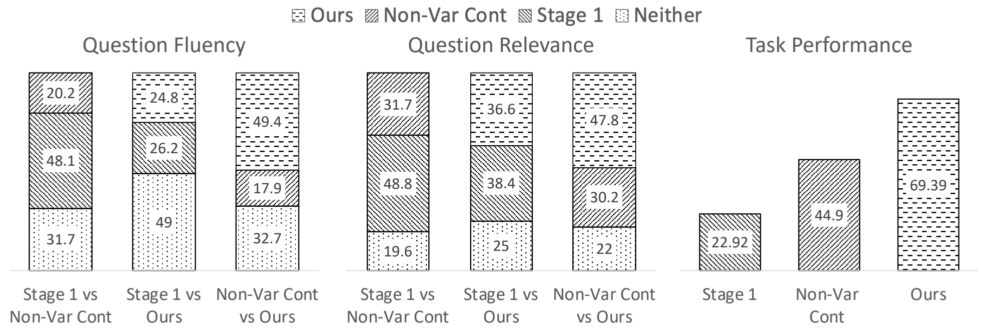


Figure 4: Human evaluation of language quality – question fluency (left), relevance (middle) and task performance (right). Question fluency and relevance compare a pair of agent-generated questions, asking users which (or possibly neither) is more fluent/relevant. Task performance is to have humans interact dynamically with Q-bot in real time.

Algorithm 5: Encoder

```
Function Encoder($\mathcal{I}, q_r$)
    Input: $\mathcal{I}(I_p^b \in \mathbb{R}^{2048})$, $q_r$
    Output: $z$ (sample or distribution parameters)
    /* Context Coder                                         */
    $e_q \leftarrow E_q(q_r)$
    $\alpha_p \leftarrow \mathrm{Softmax}\left(f_2(g(e_q) \odot f_1(I_b^p))\right)$
    $\beta_b^p \leftarrow \mathrm{Softmax}\left(f_4(g(e_q) \odot f_3(I_b^p))\right)$
    $\hat{v} \leftarrow \sum_{p=1}^{P} \sum_{b=1}^{B} \alpha_p \beta_b^p I_b^p$
    $h \leftarrow W_z^T \hat{v}$
    $l_{n,k} \leftarrow \mathrm{LogSoftmax}\left(W_n^z h\right)_k \ \forall k \in \{1, \ldots, K\} \ \forall n \in \{1, \ldots, N\}$
    $z_n \leftarrow \mathrm{GumbelSoftmax}(l_n) \ \forall n \in \{1, \ldots, N\}$
    return $z$
```

Human Study for Question Relevance. To get a more accurate measure of question relevance, we asked humans to evaluate questions generated by our model and the baselines (Zero-shot Transfer & Typical Transfer). We curated 300 random, size 4 pools where all three models predicted the target correctly at round 5. Size 9 pools require longer dialogs, so they take more effort for humans to analyze. Humans can analyze more size 4 pools in the same time, so we use size 4 pools here. For a random round, we show turkers the questions from a pair of models and ask *'Which question is most relevant to the images?'* Answering the question is a forced choice between three options: one of the two models or an *'Equally relevant'* option. See Fig. 5 for an example of the interface we presented to turkers. All model pairs were evaluated for each pool of images and the questions were presented in a random order, though the *'Equally relevant'* option was always last.

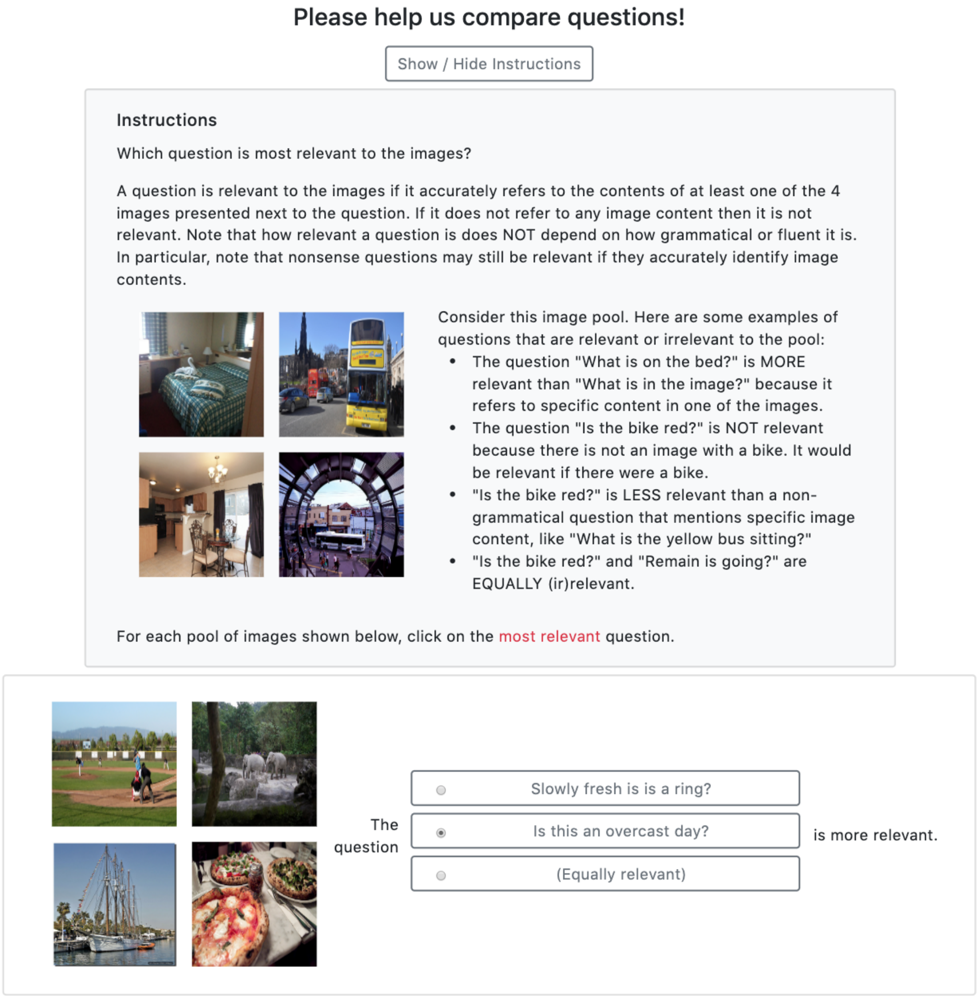
The results in Fig. 4 (middle) show the frequency with which each option was chosen for each model pair. Our model was considered more relevant than the Typical Transfer model (47.8% *vs*. 30.2%) and about the same as the Zero-shot Transfer model (36.6% *vs*. 38.4%).

Human Study for Question Fluency. We also evaluate fluency by asking humans to compare questions. In particular, we presented the same pairs of questions to turkers as in the relevance study, but this time we did *not* present the pool of images and asked them *'Which question is more understandable?'* As before, there was a forced choice between two models and an *'Equally understandable'* option. This captures fluency because humans are more likely to report that they understand grammatical and fluent language. An example interface is in Fig. 6. We used the same pairs of questions as in the relevance interface but turkers were not given image pools with which to associate the questions. As in the relevance study, questions were presented in a random order.

Figure 4 (left) shows the frequency with which each option was chosen for each model pair. Our model is considered more fluent than the Typical Transfer model (49.4% *vs*. 17.9%) and about the same as the Zero-shot Transfer model (24.8% *vs*. 26.2%).

Human Study for Task Performance. What we ultimately want in the long term is for humans to be able to collaborate with bots to solve tasks. Therefore, the most direct evaluation of our the DwD task is to have humans interact dynamically with Q-bot. We implemented an interface that allowed turkers to interact with Q-bot in real time, depicted in Fig. 7. Q-bot asks a question. A human answers it after looking at the target image. Q-bot asks a new question in response to the human answer and the human responds to that question. After the 4th answer Q-bot makes a guess about which target image the human was answering based on.

We perform this study for the same pools for each model and find our approach achieves an accuracy of 69.39% – significantly higher than Typical Transfer at 44.90% and Zero-shot Transfer at 22.92% as shown in Fig. 4 (right). This study shows that our model learns language for this task that is amenable to human-AI collaboration. This is in contrast to prior work [38] that showed that improvements captured by task-trained models for similar image-retrieval tasks did not transfer when paired with human partners.



Please help us compare questions!

Instructions

Which question is most relevant to the images?

A question is relevant to the images if it accurately refers to the contents of at least one of the 4 images presented next to the question. If it does not refer to any image content then it is not relevant. Note that how relevant a question is does NOT depend on how grammatical or fluent it is. In particular, note that nonsense questions may still be relevant if they accurately identify image contents.

Consider this image pool. Here are some examples of questions that are relevant or irrelevant to the pool:

- The question "What is on the bed?" is MORE relevant than "What is in the image?" because it refers to specific content in one of the images.
- The question "Is the bike red?" is NOT relevant because there is not an image with a bike. It would be relevant if there were a bike.
- "Is the bike red?" is LESS relevant than a non-grammatical question that mentions specific image content, like "What is the yellow bus sitting?"
- "Is the bike red?" and "Remain is going?" are EQUALLY (ir)relevant.

For each pool of images shown below, click on the most relevant question.

The question

○ Slowly fresh is is a ring?

● Is this an overcast day?

○ (Equally relevant)

is more relevant.

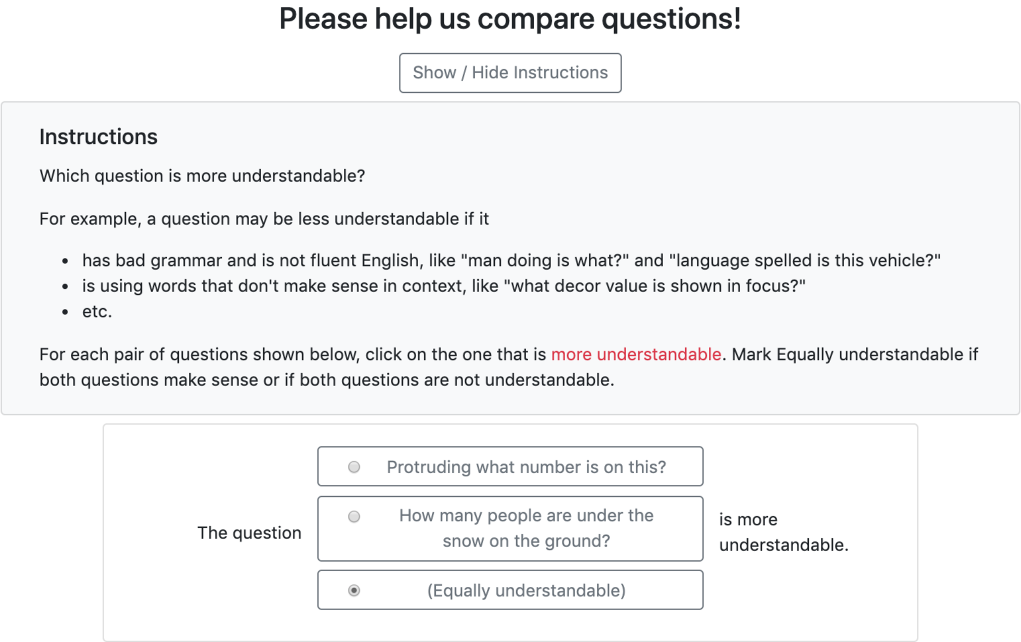
Figure 5: Human study instructions for question relevancy.

12 Model Ablations

We investigate the impact of our modelling choices from Section 3.2 of the main paper. In Tab. 2 we report the mean of all four automated metrics averaged over pool sizes, pool sampling strategies, and datasets.[3]Next we explain how we vary each of these model dimensions

– Our 128 4-way Concrete variables $z_1^r, \ldots, z_{128}^r$ require 512 logits (**Discrete**). Thus we compare to the standard Gaussian random variable common throughout VAEs with 512 dimensions (**Continuous**).
– In both discrete and continuous cases we train with an ELBO loss (**ELBO**), so we compare to a maximum likelihood only model (**MLE**) that uses an identity function as in the default option for the Question Policy (see Section 2.3.1 of the main paper). The MLE model essentially removes the KL term (2nd term of Eq. 7 of the main paper) and ignores the encoder during pre-training.
– We consider checkpoints after each step of our training curriculum: **Stage 1**, **Stage 2.A**, and **Stage 2.B**. For some approaches we skip Stage 2.A and go straight to fine-tuning everything except the speaker as in Stage 2.B. This is denoted by **Stage 2**.
– We consider 3 variations on how the speaker is fine-tuned. The first is our proposed approach of fixing the speaker (**Fixed**). The next fine-tunes the speaker (**Fine-tuned**). To evaluate the impact of fine-tuning we also consider a version of the speaker which can not learn to ask better questions

[3] This includes 10 settings: {random 2, 4, 9 pools }× {VQA, AWA, CUB} and 2 contrats pools on VQA



Please help us compare questions!

Instructions

Which question is more understandable?

For example, a question may be less understandable if it

- has bad grammar and is not fluent English, like "man doing is what?" and "language spelled is this vehicle?"
- is using words that don't make sense in context, like "what decor value is shown in focus?"
- etc.

For each pair of questions shown below, click on the one that is more understandable. Mark Equally understandable if both questions make sense or if both questions are not understandable.

The question

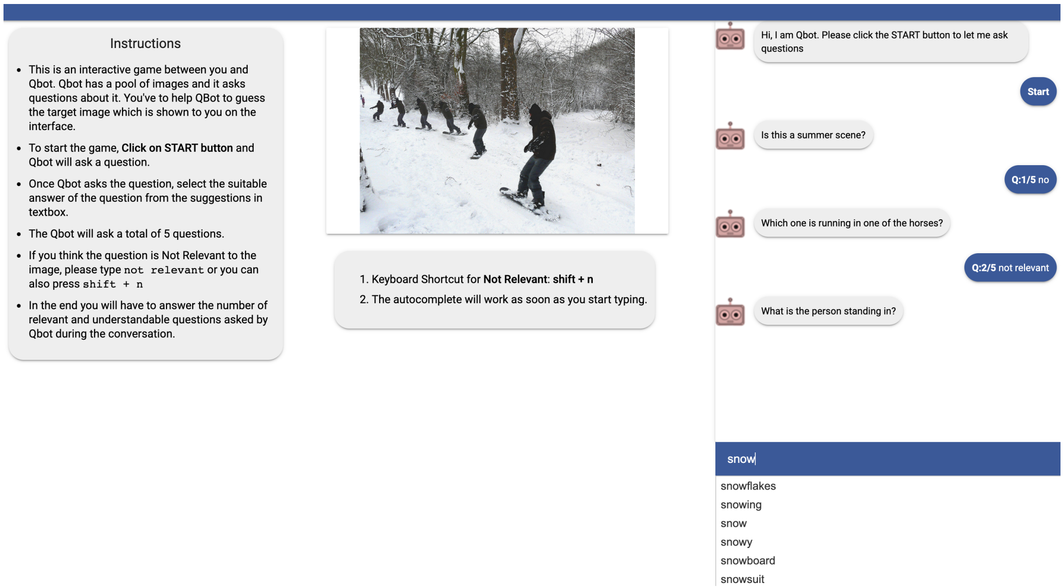
○ Protruding what number is on this?

○ How many people are under the snow on the ground?

● (Equally understandable)

is more understandable.

Figure 6: Human study instructions for question fluency.



Instructions

- This is an interactive game between you and Qbot. Qbot has a pool of images and it asks questions about it. You've to help QBot to guess the target image which is shown to you on the interface.
- To start the game, **Click on START button** and Qbot will ask a question.
- Once Qbot asks the question, select the suitable answer of the question from the suggestions in textbox.
- The Qbot will ask a total of 5 questions.
- If you think the question is Not Relevant to the image, please type `not relevant` or you can also press `shift + n`
- In the end you will have to answer the number of relevant and understandable questions asked by Qbot during the conversation.

1. Keyboard Shortcut for **Not Relevant**: **shift + n**
2. The autocomplete will work as soon as you start typing.

Hi, I am Qbot. Please click the START button to let me ask questions

Start

Is this a summer scene?

Q:1/5 no

Which one is running in one of the horses?

Q:2/5 not relevant

What is the person standing in?

snow

snowflakes
snowing
snow
snowy
snowboard
snowsuit

Figure 7: Interactive task performance MTurk interface.

by using a parallel version of the same model (**Parallel**). This last version will be described more below.

Discrete Outperforms Continuous z^r. By comparing our model in row 1 of Tab. 2 to row 7 we see that our discrete model outperforms the corresponding continuous model in terms of task performance (higher Accuracy) and about matches it in interpretability (similar Perplexity and higher Relevance). This may be a result of discreteness constraining the optimization problem to prevent overfitting and is consistent with previous work that used a discrete latent variable to model dialog [16].

Stage 2.B Less Important than Stage 2.A Comparing rows 4, 5, and 1 of Tab. 2, we can see that each additional step, Stage 2.A (row 4 −> 5) and Stage 2.B (row 5 −> 1), increases task performance and stays about the same in terms of interpretability. However, most gains in task performance happen

	z Structure	Loss	Curriculum	Speaker	Accuracy	Perplexity	Relevance	Diversity
1	Discrete	ELBO	Stage 2.B	Fixed	0.81	2.57	0.89	**0.86**
2	Discrete	ELBO	Stage 2	Fine-tuned	**0.82**	2.54	0.85	0.59
3	Discrete	ELBO	Stage 2	Parallel	0.78	2.60	0.88	0.73
4	Discrete	ELBO	Stage 1	Fixed	0.72	2.60	**0.91**	0.48
5	Discrete	ELBO	Stage 2.A	Fixed	0.80	2.59	0.89	0.81
6	Discrete	ELBO	Stage 2	Fixed	0.80	**2.53**	0.85	0.62
7	Continuous	ELBO	Stage 2.B	Fixed	0.75	2.45	0.66	0.23
8	Continuous	MLE	Stage 2.B	Fixed	0.78	4.27	0.83	4.33

Table 2: Various ablations of our training curriculum.

between Stage 1 and Stage 2. This indicates that improvements in task performance are mainly from learning to incorporate information over multiple rounds of dialog.

Better Predictions, Slightly Better Questions To further investigate whether Q-bot is asking better questions or just understanding dialog context for prediction better we considered a **Parallel** speaker model. This model loaded two copies of Q-bot, A and B both starting at the model resulting from Stage 1. Copy A was fine-tuned for task performance, but every z^r it generated was ignored and replaced with the z^r generated by copy B, which was not updated at all. The result was that copy A of the model could not incorporate dialog context into its questions any better than the Stage 1 model, so all it could do was track the dialog better for prediction purposes. By comparing the performance of copy A (row 3 of Tab. 2) to our model (row 1) we can see a 3 point different in accuracy, so the question content of our model has improved after fine-tuning, but not by a lot. Again, this indicates most improvements are from dialog tracking for prediction (row 3 accuracy is much higher than row 4 accuracy).

Fine-tuned Speaker During both Stage 2.A and Stage 2.B we fix the Speaker module because it is intended to capture low level language details and we do not want it to change its understanding of English. Row 2 of Tab. 2 does not fix the Speaker during Stage 2 fine-tuning. Instead, it uses each softmax at each step of the LSTM decoder to parameterize one Concrete variable [17] per word. This allows gradients to flow through the decoder during fine-tuning, allowing the model to tune low-level signals. This is similar to previous approaches which either used this technique [40] or REINFORCE [2] This model is competitive with DwD in terms of task performance. However, when we inspect its output we see somewhat less understandable language.

Variational Prior Helps Interpretability We found the most important factor for maintaining interpretability to be the ELBO loss we applied during pre-training. Comparing the continuous Gaussian variable (row 7) to a similar hidden state (row 8) trained without the KL prior term we see drastically different perplexity and diversity. In the main paper these metrics dropped when a model had drifted from English (*e.g.*, for Typical Transfer). This suggests the model without the ELBO in row 8 has experienced similar language drift.

13 Performance by Round

Experiments in the main paper considered dialog performance after the first round (top of Table 1) and at the final round of dialog (either 5 or 9). This does not give much sense for how dialog performance increases over rounds of dialog, so we report `QBot()`'s guessing game performance at each round of dialog in Fig. 8. For all fine-tuned models performance goes up over multiple rounds of dialog, though some models benefit more than others. Stage 1 models decrease in performance after round 1 because it is too far from the training data such models have been exposed to.

14 Related Work

We used a visual reference game to study question generation, improving the quality of generated language using concepts related to latent action spaces. This interest is mainly inspired by problems encountered when using models comparable to the Zero-shot Transfer baseline. In [3] a dataset is collected with question supervision then fine-tuning is used in an attempt to increase task performance, but the resulting utterances are not intelligible. Similarly, [2] takes a very careful approach to fine-tuning for task performance but finds that language also diverges, becoming difficult for humans to understand.

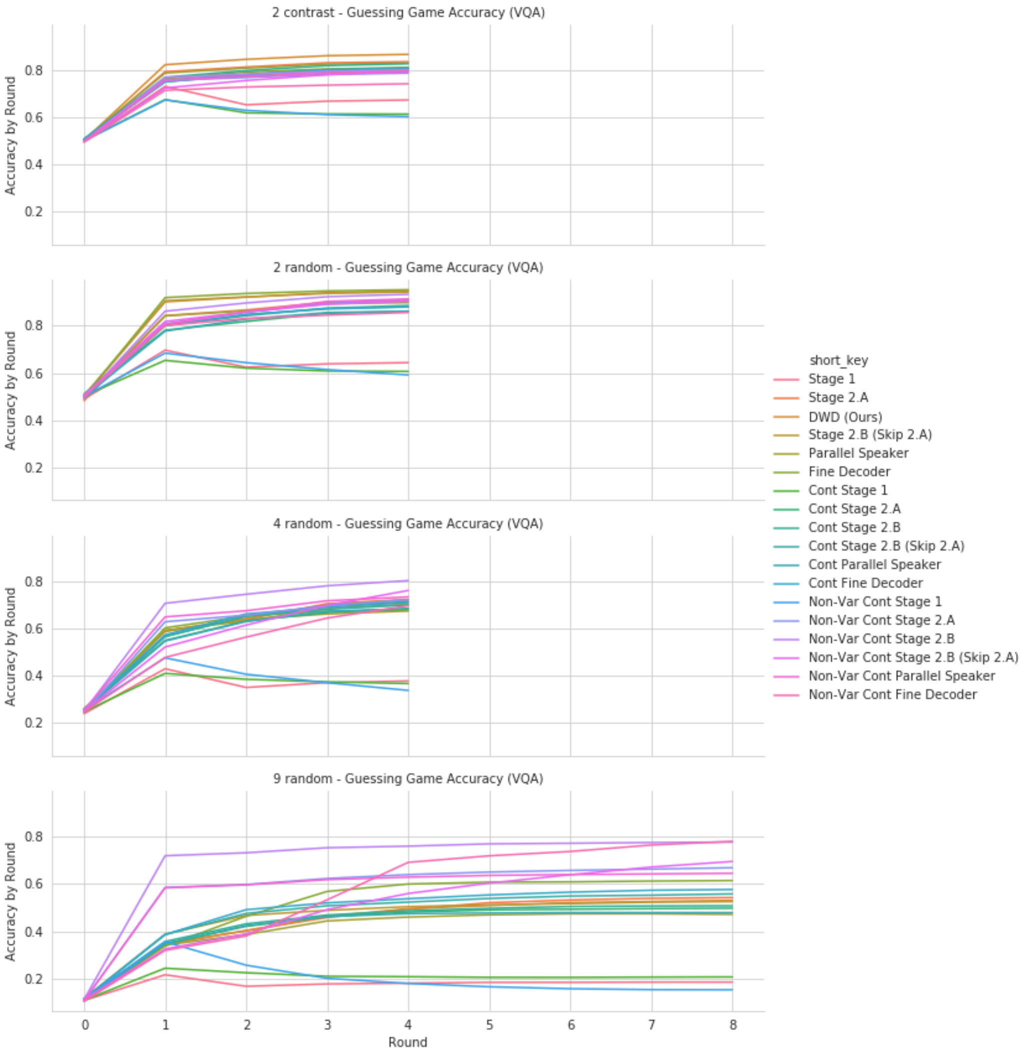


Figure 8: Task performance (guessing game accuracy) over rounds of dialog. Performance increases over rounds for all models except the Stage 1 models.

Visual Question Generation. Other approaches like [26] and [27] also aim to ask questions with limited question supervision. They give Q-bot access to an oracle to which it can ask any question and get a good answer back. This feedback allows these models to ask questions that are more useful for teaching A-bot [26] or generating scene graphs [27], but they require a domain specific oracle and do not take any measures to encourage interpretability. We are also interested in generalizing with limited supervision, using a standard VQAv2 [6] trained A-bot as a flawed oracle, but we focus on maintaining interpretability of generated questions and not just their usefulness.

Latent Action Spaces. Of particular interest to us is a line of work that uses represents dialogs using latent action spaces [28, 29, 30, 31, 32, 30, 33, 34, 35, 36]. Recent work uses these representations to discover intelligible language [28] and to perform zero-shot dialog generation [29], though neither works consider visually grounded language as in our approach. Most relevant is [16], which focuses on the difference between word level feedback and latent action level feedback. Like us, they use a variationally constrained latent action space (like our z) to generate dialogs and find that by providing feedback to the latent actions instead of the generated words (as opposed to the approaches in [2] and [3]) they achieve better dialog performance. Our variational prior is similar to the Full ELBO considered in [16], but we consider generalization from non-dialog data and generalization to new modalities.

Reference Games. The task we use to study question generation follows a body of work that uses reference games to study language and its interaction with other modalities [37]. Our particular task is most similar to those in [5] and [38]. In particular, [5] collects a dataset for goal oriented visual dialog using a similar image reference game and [38] uses a similar guessing game we use to evaluate how well humans can interact with A-bot.