Interactive Language Acquisition with One-shot Visual Concept Learning through a Conversational Game

Haichao Zhang[†], **Haonan Yu**[†], **and Wei Xu** [†§]
[†] Baidu Research - Institue of Deep Learning, Sunnyvale USA
[§] National Engineering Laboratory for Deep Learning Technology and Applications, Beijing China
{zhanghaichao,haonanyu,wei.xu}@baidu.com

Abstract

Building intelligent agents that can communicate with and learn from humans in natural language is of great value. Supervised language learning is limited by the ability of capturing mainly the statistics of training data, and is hardly adaptive to new scenarios or flexible for acquiring new knowledge without inefficient retraining or catastrophic forgetting. We highlight the perspective that conversational interaction serves as a natural interface both for language learning and for novel knowledge acquisition and propose a joint imitation and reinforcement approach for grounded language learning through an interactive conversational game. The agent trained with this approach is able to actively acquire information by asking questions about novel objects and use the just-learned knowledge in subsequent conversations in a one-shot fashion. Results compared with other methods verified the effectiveness of the proposed approach.

1 Introduction

Language is one of the most natural forms of communication for human and is typically viewed as fundamental to human intelligence; therefore it is crucial for an intelligent agent to be able to use language to communicate with human as well. While supervised training with deep neural networks has led to encouraging progress in language learning, it suffers from the problem of capturing mainly the statistics of training data, and from a lack of adaptiveness to new scenarios and being flexible for acquiring new knowledge without inefficient retraining or catastrophic forgetting. Moreover, supervised training of deep neural network models needs a large number of training samples while many interesting applications require rapid learning from a small amount of data, which poses an even greater challenge to the supervised setting.

In contrast, humans learn in a way very different from the supervised setting (Skinner, 1957; Kuhl, 2004). First, humans act upon the world and learn from the consequences of their actions (Skinner, 1957; Kuhl, 2004; Petursdottir and Mellor, 2016). While for mechanical actions such as movement, the consequences mainly follow geometrical and mechanical principles, for language, humans act by speaking, and the consequence is typically a response in the form of verbal and other behavioral feedback (*e.g.*, nodding) from the conversation partner (*i.e.*, teacher). These types of feedback typically contain informative signals on how to improve language skills in subsequent conversations and play an important role in humans' language acquisition process (Kuhl, 2004; Petursdottir and Mellor, 2016). Second, humans have shown a celebrated ability to learn new concepts from small amount of data (Borovsky et al., 2003). From even just one example, children seem to be able to make inferences and draw plausible boundaries between concepts, demonstrating the ability of one-shot learning (Lake et al., 2011).

The language acquisition process and the one-shot learning ability of human beings are both impressive as a manifestation of human intelligence, and are inspiring for designing novel settings and algorithms for computational language learning. In this paper, we leverage conversation as both an interactive environment for language learning (Skinner, 1957) and a natural interface for acquiring new knowledge (Baker et al., 2002). We propose an approach for interactive language acquisition with one-shot concept learning ability. The proposed approach allows an agent to learn grounded language from scratch, acquire the transferable skill of actively seeking and memorizing information about novel objects, and develop the one-shot learning ability, purely through conversational interaction with a teacher.

2 Related Work

Supervised Language Learning. Deep neural network-based language learning has seen great success on many applications, including machine translation (Cho et al., 2014b), dialogue generation (Wen et al., 2015; Serban et al., 2016), image captioning and visual question answering (**?**Antol et al., 2015). For training, a large amount of labeled data is needed, requiring significant efforts to collect. Moreover, this setting essentially captures the statistics of training data and does not respect the interactive nature of language learning, rendering it less flexible for acquiring new knowledge without retraining or forgetting (Stent and Bangalore, 2014).

Reinforcement Learning for Sequences. Some recent studies used reinforcement learning (RL) to tune the performance of a pre-trained language model according to certain metrics (Ranzato et al., 2016; Bahdanau et al., 2017; Li et al., 2016; Yu et al., 2017). Our work is also related to RL in natural language action space (He et al., 2016) and shares a similar motivation with Weston (2016) and Li et al. (2017), which explored language learning through pure textual dialogues. However, in these works (He et al., 2016; Weston, 2016; Li et al., 2017), a set of candidate sequences is provided and the action is to select one from the set. Our main focus is rather on learning language from scratch: the agent has to learn to *generate* a sequence action rather than to simply *select* one from a provided candidate set.

Communication and Emergence of Language. Recent studies have examined learning to communicate (Foerster et al., 2016; Sukhbaatar et al., 2016) and invent language (Lazaridou et al., 2017; Mordatch and Abbeel, 2018). The emerged language needs to be interpreted by humans via post-processing (Mordatch and Abbeel, 2018). We, however, aim to achieve language learning from the dual perspectives of understanding and generation, and the speaking action of the agent is readily understandable without any post-processing. Some studies on language learning have used a guesser-responder setting in which the guesser tries to achieve the final goal (*e.g.*, classification) by collecting additional information through asking the responder questions (Strub et al., 2017; Das et al., 2017). These works try to optimize the question being asked to help the guesser achieve the final goal, while we focus on transferable speaking and one-shot ability.

One-shot Learning and Active Learning. One-shot learning has been investigated in some recent works (Lake et al., 2011; Santoro et al., 2016; Woodward and Finn, 2016). The memory-augmented network (Santoro et al., 2016) stores visual representations mixed with ground truth class labels in an external memory for one-shot learning. A class label is always provided following the presentation of an image; thus the agent receives information from the teacher in a passive way. Woodward and Finn (2016) present efforts toward active learning, using a vanilla recurrent neural network (RNN) without an external memory. Both lines of study focus on image classification only, meaning the class label is directly provided for memorization. In contrast, we target language and one-shot learning via conversational interaction, and the learner has to learn to extract important information from the teacher's sentences for memorization.

3 The Conversational Game

We construct a conversational game inspired by experiments on language development in infants from cognitive science (Waxman, 2004). The game is implemented with the XWORLD simulator (Yu et al., 2018; Zhang et al., 2017) and is publicly available online.[1] It provides an environment for the agent[2] to learn language and develop the one-shot learning ability. One-shot learning here means that during test sessions, no further training happens to the agent and it has to answer teacher's questions correctly about novel images of never-before-seen classes after being taught only once by the teacher, as illustrated in Figure 1. To succeed in this game, the agent has to learn to 1) speak by generating sentences, 2) extract and memorize useful information with only one exposure and use it in subsequent conversations, and 3) behave adaptively according to context and its own knowledge (*e.g.*, asking questions about unknown objects and answering questions about something known), all achieved through interacting with the

[1] `https://github.com/PaddlePaddle/XWorld`

[2] We use the term *agent* interchangeably with *learner*.

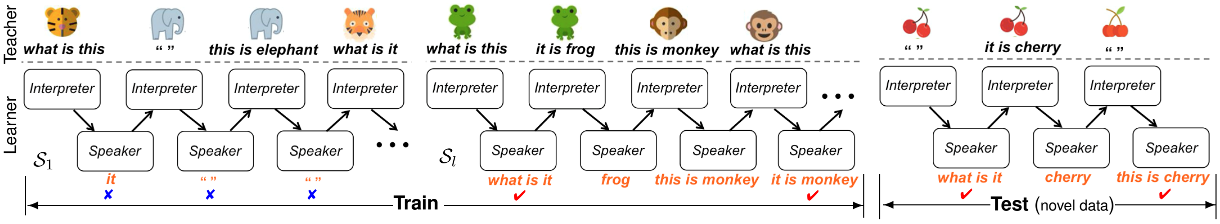

Figure 1: **Interactive language and one-shot concept learning.** Within a session $\mathcal{S}_l$, the teacher may ask questions, answer learner's questions, make statements, or say nothing. The teacher also provides reward feedback based on learner's responses as (dis-)encouragement. The learner alternates between interpreting teacher's sentences and generating a response through *interpreter* and *speaker*. **Left:** Initially, the learner can barely say anything meaningful. **Middle:** Later it can produce meaningful responses for interaction. **Right:** After training, when confronted with an image of *cherry*, which is a novel class that the learner never saw before during training, the learner can ask a question about it ("*what is it*") and generate a correct statement ("*this is cherry*") for another instance of cherry after only being taught once.

teacher. This makes our game distinct from other seemingly relevant games, in which the agent cannot speak (Wang et al., 2016) or "speaks" by *selecting* a candidate from a provided set (He et al., 2016; Weston, 2016; Li et al., 2017) rather than *generating* sentences by itself, or games mainly focus on slow learning (Das et al., 2017; Strub et al., 2017) and falls short on one-shot learning.

In this game, sessions ($\mathcal{S}_l$) are randomly instantiated during interaction. Testing sessions are constructed with a separate dataset with concepts that never appear before during training to evaluate the language and one-shot learning ability. Within a session, the teacher randomly selects an object and interacts with the learner about the object by randomly 1) posing a question (*e.g.*, "*what is this*"), 2) saying nothing (*i.e.*, "") or 3) making a statement (*e.g.*, "*this is monkey*"). When the teacher asks a question or says nothing, i) if the learner raises a question, the teacher will provide a statement about the object asked (*e.g.*, "*it is frog*") with a *question-asking reward* (+0.1); ii) if the learner says nothing, the teacher will still provide an answer (*e.g.*, "*this is elephant*") but with an *incorrect-reply reward* (−1) to discourage the learner from remaining silent; iii) for all other incorrect responses from the learner, the teacher will provide an *incorrect-reply reward* and move on to the next random object for interaction. When the teacher generates a statement, the learner will receive no reward if a correct statement is generated otherwise an *incorrect-reply reward* will be given. The session ends if the learner answers the teacher's question correctly, generates a correct statement when the teacher says nothing (receiving a *correct-answer reward* +1), or when the maximum number of steps is reached. The sentence from teacher at each time step is generated using a context-free grammar as shown in Table 1.

A success is reached if the learner behaves correctly during the whole session: asking questions about novel objects, generating answers when asked, and making statements when the teacher says nothing about objects that have been taught within the session. Otherwise it is a failure.

Table 1: Grammar for the teacher's sentences.

start	→	question | silence | statement
question	→	Q1 | Q2 | Q3
silence	→	" "
statement	→	A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8
Q1	→	"*what*"
Q2	→	"*what*" M
Q3	→	"*tell what*" N
M	→	"*is it*" | "*is this*" | "*is there*" | "*do you see*" | "*can you see*" | "*do you observe*" | "*can you observe*"
N	→	"*it is*" | "*this is*" | "*there is*" | "*you see*" | "*you can see*" | "*you observe*" | "*you can observe*"
A1	→	G
A2	→	"*it is*" G
A3	→	"*this is*" G
A4	→	"*there is*" G
A5	→	"*i see*" G
A6	→	"*i observe*" G
A7	→	"*i can see*" G
A8	→	"*i can observe*" G
G	→	object name

4 Interactive Language Acquisition via Joint Imitation and Reinforcement

Motivation. The goal is to *learn to converse and develop the one-shot learning ability by conversing with a teacher and improving from teacher's feedback*. We propose to use a *joint imitation and reinforce* approach to achieve this goal. *Imitation* helps the agent to develop the basic ability to generate sensible sentences. As learning is done by observing the teacher's behaviors during conversion, the agent essentially imitates the teacher from a *third-person perspective* (Stadie et al., 2017) rather than imitating an expert agent who is conversing with the teacher (Das et al., 2017; Strub et al., 2017). During conversations, the agent perceives sentences and images without any explicit labeling of ground truth answers, and it has to learn to make sense of raw perceptions, extract useful information, and save it for later use when generating an answer to teacher's question. While it is tempting to purely imitate the teacher, the agent trained this way only develops *echoic behavior* (Skinner, 1957), *i.e.*, mimicry. *Reinforce* leverages confirmative feedback from the teacher for learning to converse adaptively beyond mimicry by adjusting the action policy. It enables the learner to use the acquired speaking ability and adapt it according to reward feedback. This is analogous to some views on the babies' language-learning process that babies use the acquired speaking skills by trial and error with parents and improve according to the consequences of speaking actions (Skinner, 1957; Petursdottir and Mellor, 2016). The fact that babies don't fully develop the speaking capabilities without the ability to hear (Houston and Miyamoto, 2011), and that it is hard to make a meaningful conversation with a trained parrot signifies the importance of both imitation and reinforcement in language learning.

Formulation. The agent's response can be modeled as a sample from a probability distribution over the possible sequences. Specifically, for one session, given the visual input $\mathbf{v}^t$ and conversation history $\mathcal{H}^t = \{\mathbf{w}^1, \mathbf{a}^1, \cdots, \mathbf{w}^t\}$, the agent's response $\mathbf{a}^t$ can be generated by sampling from a distribution of the speaking action $\mathbf{a}^t \sim p^{\mathrm{S}}_\theta(\mathbf{a}|\mathcal{H}^t, \mathbf{v}^t)$. The agent interacts with the teacher by outputting the utterance $\mathbf{a}^t$ and receives feedback from the teacher in the next step, with $\mathbf{w}^{t+1}$ a sentence as verbal feedback and r^{t+1} reward feedback (with positive values as encouragement while negative values as discouragement, according to $\mathbf{a}^t$, as described in Section 3). Central to the goal is learning $p^{\mathrm{S}}_\theta(\cdot)$. We formulate the problem as the minimization of a cost function as:

$$\mathcal{L}_\theta = \underbrace{\mathbb{E}_{\mathcal{W}}\left[-\textstyle\sum_t \log p^{\mathrm{I}}_\theta(\mathbf{w}^t|\cdot)\right]}_{\text{Imitation } \mathcal{L}^{\mathrm{I}}_\theta} + \underbrace{\mathbb{E}_{p^{\mathrm{S}}_\theta}\left[-\textstyle\sum_t [\gamma]^{t-1} \cdot r^t\right]}_{\text{Reinforce } \mathcal{L}^{\mathrm{R}}_\theta}$$

where $\mathbb{E}_{\mathcal{W}}(\cdot)$ is the expectation over all the sentences $\mathcal{W}$ from teacher, γ is a reward discount factor, and $[\gamma]^t$ denotes the exponentiation over γ. While the imitation term learns directly the predictive distribution $p^{\mathrm{I}}_\theta(\mathbf{w}^t|\mathcal{H}^{t-1}, \mathbf{a}^t)$, it contributes to $p^{\mathrm{S}}_\theta(\cdot)$ through *parameter sharing* between them.

Architecture. The learner comprises four major components: *external memory*, *interpreter*, *speaker*, and *controller*, as shown in Figure 2. *External memory* is flexible for storing and retrieving information (Graves et al., 2014; Santoro et al., 2016), making it a natural component of our network for one-shot learning. The *interpreter* is responsible for interpreting the teacher's sentences, extracting information from the perceived signals, and saving it to the external memory. The *speaker* is in charge of generating sentence responses with reading access to the external memory. The response could be a question asking for information or a statement answering a teacher's question, leveraging the information stored in the external memory. The *controller* modulates the behavior of the speaker to generate responses according to context (*e.g.*, the learner's knowledge status).

At time step t, the *interpreter* uses an interpreter-RNN to encode the input sentence $\mathbf{w}^t$ from the teacher as well as historical conversational information into a state vector $\mathbf{h}^t_{\mathrm{I}}$. $\mathbf{h}^t_{\mathrm{I}}$ is then passed through a residue-structured network, which is an identity mapping augmented with a learnable controller $f(\cdot)$ implemented with fully connected layers for producing $\mathbf{c}^t$. Finally, $\mathbf{c}^t$ is used as the initial state of the speaker-RNN for generating the response $\mathbf{a}^t$. The final state $\mathbf{h}^t_{\mathrm{last}}$ of the speaker-RNN will be used as the initial state of the interpreter-RNN at the next time step.

4.1 Imitation with Memory Augmented Neural Network for Echoic Behavior

The teacher's way of speaking provides a source for the agent to imitate. For example, the syntax for composing a sentence is a useful skill the agent can learn from the teacher's sentences, which could benefit both *interpreter* and *speaker*. Imitation is achieved by predicting teacher's future sentences with *interpreter* and parameter sharing between *interpreter* and *speaker*. For prediction, we can represent the probability of the next sentence $\mathbf{w}^t$ conditioned on the image $\mathbf{v}^t$ as well as previous sentences from both the teacher and the

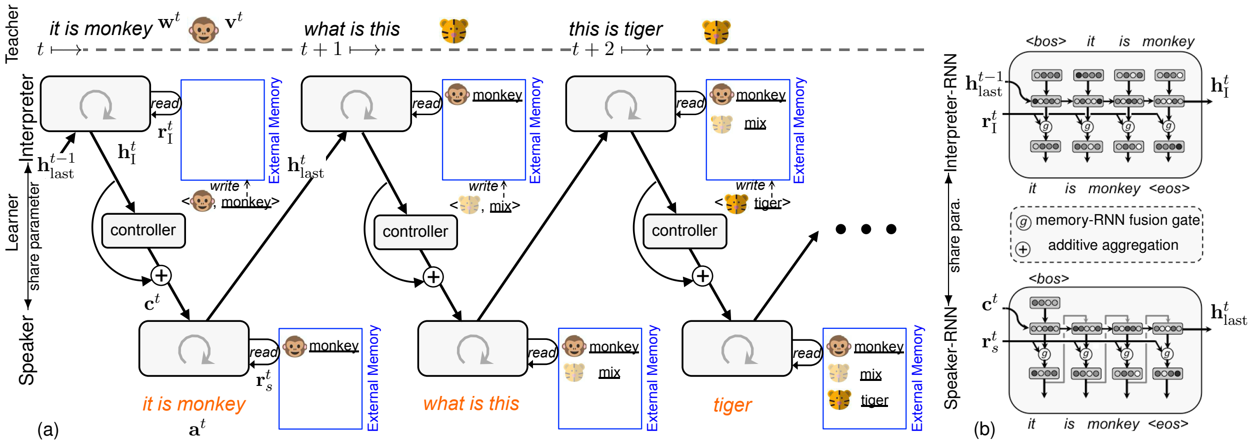

Figure 2: **Network structure.** (a) Illustration of the overall architecture. At each time step, the learner uses the interpreter module to encode the teacher's sentence. The visual perception is also encoded and used as a key to retrieve information from the external memory. The last state of the interpreter-RNN will be passed through a controller. The controller's output will be added to the input and used as the initial state of the speaker-RNN. The interpreter-RNN will update the external memory with an importance (illustrated with transparency) weighted information extracted from the perception input. 'Mix' denotes a mixture of word embedding vectors. (b) The structures of the interpreter-RNN (top) and the speaker-RNN (bottom). The interpreter-RNN and speaker-RNN share parameters.

learner $\{\mathbf{w}^1, \mathbf{a}^1, \cdots, \mathbf{w}^{t-1}, \mathbf{a}^{t-1}\}$ as

$$\begin{aligned} & p^{\mathrm{I}}_\theta(\mathbf{w}^t|\mathcal{H}^{t-1}, \mathbf{a}^{t-1}, \mathbf{v}^t) \\ & = \textstyle\prod_i p^{\mathrm{I}}_\theta(w^t_i|w^t_{1:i-1}, \mathbf{h}^{t-1}_{\text{last}}, \mathbf{v}^t), \end{aligned} \quad (1)$$

where $\mathbf{h}^{t-1}_{\text{last}}$ is the last state of the RNN at time step t–1 as the summarization of $\{\mathcal{H}^{t-1}, \mathbf{a}^{t-1}\}$ (*c.f.*, Figure 2), and i indexes words within a sentence.

It is natural to model the probability of the i-th word in the t-th sentence with an RNN, where the sentences up to t and words up to i within the t-th sentence are captured by a fixed-length state vector $\mathbf{h}^t_i = \text{RNN}(\mathbf{h}^t_{i-1}, w^t_i)$. To incorporate knowledge learned and stored in the external memory, the generation of the next word is *adaptively* based on i) the predictive distribution of the next word from the state of the RNN to capture the *syntactic structure of sentences*, and ii) the information from the external memory to represent the previously learned knowledge, via a fusion gate g:

$$p^{\mathrm{I}}_\theta(w^t_i|\mathbf{h}^t_i, \mathbf{v}^t) = (1-g)\cdot p_{\mathbf{h}} + g \cdot p_{\mathbf{r}}, \quad (2)$$

where $p_{\mathbf{h}} = \text{softmax}\big(\mathbf{E}^{\mathsf{T}} f_{\text{MLP}}(\mathbf{h}^t_i)\big)$ and $p_{\mathbf{r}} = \text{softmax}\big(\mathbf{E}^{\mathsf{T}}\mathbf{r}\big)$. $\mathbf{E}\in\mathbb{R}^{d\times k}$ is the word embedding table, with d the embedding dimension and k the vocabulary size. $\mathbf{r}$ is a vector read out from the external memory using a visual key as detailed in the next section. $f_{\text{MLP}}(\cdot)$ is a multi-layer Multi-Layer Perceptron (MLP) for bridging the semantic gap between the RNN state space and the word embedding space. The fusion gate g is computed as $g = f(\mathbf{h}^t_i, c)$, where c is the confidence score $c{=}\max(\mathbf{E}^{\mathsf{T}}\mathbf{r})$, and a well-learned concept should have a large score by design (Appendix A.2).

Multimodal Associative Memory. We use a multimodal memory for storing visual (v) and sentence (s) features with each modality while preserving the correspondence between them (Baddeley, 1992). Information organization is more structured than the single modality memory as used in Santoro et al. (2016) and cross modality retrieval is straightforward under this design. A visual encoder implemented as a convolutional neural network followed by fully connected layers is used to encode the visual image $\mathbf{v}$ into a visual key $\mathbf{k}_v$, and then the corresponding sentence feature can be retrieved from the memory as:

$$\mathbf{r} \leftarrow \textsc{Read}(\mathbf{k}_v, \mathbf{M}_v, \mathbf{M}_s). \quad (3)$$

$\mathbf{M}_v$ and $\mathbf{M}_s$ are memories for visual and sentence modalities with the same number of slots (columns). Memory read is implemented as $\mathbf{r} = \mathbf{M}_s\boldsymbol{\alpha}$ with $\boldsymbol{\alpha}$ a soft reading weight obtained through the visual modality by calculating the cosine similarities between $\mathbf{k}_v$ and slots of $\mathbf{M}_v$.

Memory write is similar to Neural Turing Machine (Graves et al., 2014), but with a content importance gate g_{mem} to adaptively control whether the content $\mathbf{c}$ should be written into memory:

$$\mathbf{M}_m \leftarrow \textsc{Write}(\mathbf{M}_m, \mathbf{c}_m, g_{\text{mem}}), \;\; m \in \{v, s\}.$$

For the visual modality $\mathbf{c}_v \triangleq \mathbf{k}_v$. For the sentence modality, $\mathbf{c}_s$ has to be selectively extracted from the sentence generated by the teacher. We use an attention mechanism to achieve this by $\mathbf{c}_s = \mathbf{W}\boldsymbol{\eta}$, where $\mathbf{W}$ denotes the matrix with columns being the embedding vectors of all the words in the sentence. $\boldsymbol{\eta}$ is a normalized attention vector representing the relative importance of each word in the sentence as measured by the cosine similarity between the sentence representation vector and each word's context vector, computed using a bidirectional-RNN. The scalar-valued content importance gate g_{mem} is computed as a function of the sentence from the teacher, meaning that the importance of the content to be written into memory depends on the content itself (*c.f.*, Appendix A.3 for more details). The memory write is achieved with an erase and an add operation:

$$\begin{aligned}\tilde{\mathbf{M}}_m &= \mathbf{M}_m - \mathbf{M}_m \odot (g_{\text{mem}} \cdot \mathbf{1} \cdot \boldsymbol{\beta}^{\mathsf{T}}),\\ \mathbf{M}_m &= \tilde{\mathbf{M}}_m + g_{\text{mem}} \cdot \mathbf{c}_m \cdot \boldsymbol{\beta}^{\mathsf{T}},\ m \in \{v, s\}.\end{aligned}$$

$\odot$ denotes Hadamard product and the write location $\boldsymbol{\beta}$ is determined with a Least Recently Used Access mechanism (Santoro et al., 2016).

4.2 Context-adaptive Behavior Shaping through Reinforcement Learning

Imitation fosters the basic language ability for generating echoic behavior (Skinner, 1957), but it is not enough for conversing adaptively with the teacher according to context and the knowledge state of the learner. Thus we leverage reward feedback to shape the behavior of the agent by optimizing the policy using RL. The agent's response $\mathbf{a}^t$ is generated by the *speaker*, which can be modeled as a sample from a distribution over all possible sequences, given the conversation history $\mathcal{H}^t = \{\mathbf{w}^1, \mathbf{a}^1, \cdots, \mathbf{w}^t\}$ and visual input $\mathbf{v}^t$:

$$\mathbf{a}^t \sim p_\theta^{\mathrm{S}}(\mathbf{a}|\mathcal{H}^t, \mathbf{v}^t). \qquad (4)$$

As $\mathcal{H}^t$ can be encoded by the interpreter-RNN as $\mathbf{h}_{\mathrm{I}}^t$, the action policy can be represented as $p_\theta^{\mathrm{S}}(\mathbf{a}|\mathbf{h}_{\mathrm{I}}^t, \mathbf{v}^t)$. To leverage the language skill that is learned via imitation through the *interpreter*, we can generate the sentence by implementing the *speaker* with an RNN, sharing parameters with the interpreter-RNN, but with a conditional signal modulated by a controller network (Figure 2):

$$p_\theta^{\mathrm{S}}(\mathbf{a}^t|\mathbf{h}_{\mathrm{I}}^t, \mathbf{v}^t) = p_\theta^{\mathrm{I}}(\mathbf{a}^t|\mathbf{h}_{\mathrm{I}}^t + f(\mathbf{h}_{\mathrm{I}}^t, c), \mathbf{v}^t). \qquad (5)$$

The reason for using a controller $f(\cdot)$ for modulation is that the basic language model only offers the learner the echoic ability to generate a sentence, but not necessarily the adaptive behavior according to context (*e.g.* asking questions when facing novel objects and providing an answer for a previously learned object according to its own knowledge state). Without any additional module or learning signals, the agent's behaviors would be the same as those of the teacher because of parameter sharing; thus, it is difficult for the agent to learn to speak in an adaptive manner.

To learn from consequences of speaking actions, the policy $p_\theta^{\mathrm{S}}(\cdot)$ is adjusted by maximizing expected future reward as represented by $\mathcal{L}_\theta^{\mathrm{R}}$. As a non-differentiable sampling operation is involved in Eqn.(4), policy gradient theorem (Sutton and Barto, 1998) is used to derive the gradient for updating $p_\theta^{\mathrm{S}}(\cdot)$ in the reinforce module:

$$\nabla_\theta \mathcal{L}_\theta^{\mathrm{R}} = \mathbb{E}_{p_\theta^{\mathrm{S}}}\left[\textstyle\sum_t A^t \cdot \nabla_\theta \log p_\theta^{\mathrm{S}}(\mathbf{a}^t|\mathbf{c}^t)\right], \qquad (6)$$

where $A^t = V(\mathbf{h}_{\mathrm{I}}^t, c^t) - r^{t+1} - \gamma V(\mathbf{h}_{\mathrm{I}}^{t+1}, c^{t+1})$ is the advantage (Sutton and Barto, 1998) estimated using a value network $V(\cdot)$. The imitation module contributes by implementing $\mathcal{L}_\theta^{\mathrm{I}}$ with a cross-entropy loss (Ranzato et al., 2016) and minimizing it with respect to the parameters in $p_\theta^{\mathrm{I}}(\cdot)$, which are shared with $p_\theta^{\mathrm{S}}(\cdot)$. The training signal from imitation takes the shortcut connection without going through the controller. More details on $f(\cdot)$, $V(\cdot)$ are provided in Appendix A.2.

5 Experiments

We conduct experiments with comparison to baseline approaches. We first experiment with a word-level task in which the teacher and the learner communicate a single word each time. We then investigate the impact of image variations on concept learning. We further perform evaluation on the more challenging sentence-level task in which the teacher and the agent communicate in the form of sentences with varying lengths.

Setup. To evaluate the performance in learning a transferable ability, rather than the ability of fitting a particular dataset, we use an Animal dataset for training and test the trained models on a Fruit dataset (Figure 1). More details on the datasets are provided in Appendix A.1. Each session consists of two randomly sampled classes, and the maximum number of interaction steps is six.

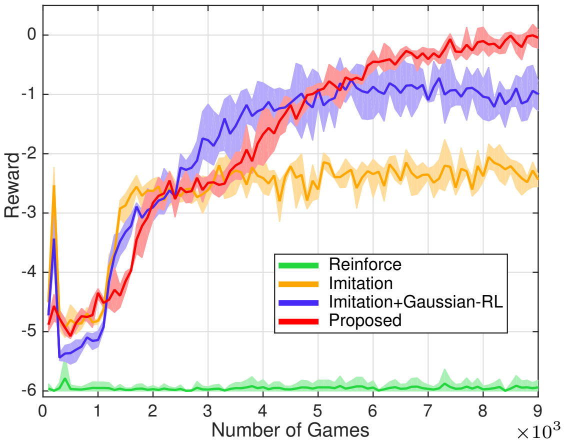

Figure 3: **Evolution of reward** during training for the word-level task without image variations.

Baselines. The following methods are compared:

- **Reinforce**: a baseline model with the same network structure as the proposed model and trained using RL only, *i.e.* minimizing $\mathcal{L}_\theta^{\mathrm{R}}$;
- **Imitation**: a recurrent encoder decoder (Serban et al., 2016) model with the same structure as ours and trained via imitation (minimizing $\mathcal{L}_\theta^{\mathrm{I}}$);
- **Imitation+Gaussian-RL**: a joint imitation and reinforcement method using a Gaussian policy (Duan et al., 2016) in the latent space of the control vector $\mathbf{c}^t$ (Zhang et al., 2017). The policy is changed by modifying the control vector $\mathbf{c}^t$ the action policy depends upon.

Training Details. The training algorithm is implemented with the deep learning platform PaddlePaddle.[3] The whole network is trained from scratch in an end-to-end fashion. The network is randomly initialized without any pre-training and is trained with decayed Adagrad (Duchi et al., 2011). We use a batch size of 16, a learning rate of 1×10^{-5} and a weight decay rate of 1.6×10^{-3}. We also exploit experience replay (Wang et al., 2017; Yu et al., 2018). The reward discount factor γ is 0.99, the word embedding dimension d is 1024 and the dictionary size k is 80. The visual image size is 32×32, the maximum length of generated sentence is 6 and the memory size is 10. Word embedding vectors are initialized as random vectors and remain fixed during training. A sampling operation is used for sentence generation during training for exploration while a max operation is used during testing both for **Proposed** and for **Reinforce** baseline. The max operation is used in both training and testing for **Imitation** and **Imitation+Gaussian-RL** baselines.

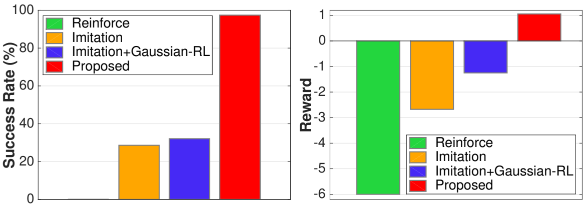

Figure 4: **Test performance** for the word-level task without image variations. Models are trained on the Animal dataset and tested on the Fruit dataset.

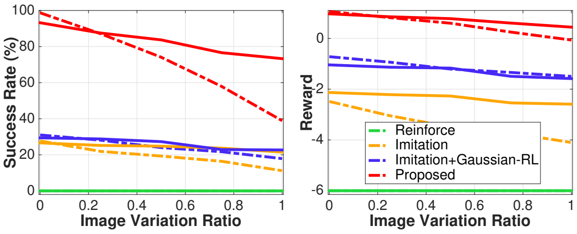

Figure 5: **Test success rate and reward** for the word-level task on the Fruit dataset under different test image variation ratios for models trained on the Animal dataset with a variation ratio of 0.5 (solid lines) and without variation (dashed lines).

5.1 Word-Level Task

In this experiment, we focus on a word-level task, which offers an opportunity to analyze and understand the underlying behavior of different algorithms while being free from distracting factors. Note that although the teacher speaks a word each time, the learner still has to learn to generate a full-sentence ended with an end-of-sentence symbol.

Figure 3 shows the evolution curves of the rewards during training for different approaches. It is observed that **Reinforce** makes very little progress, mainly due to the difficulty of exploration in the large space of sequence actions. **Imitation** obtains higher rewards than **Reinforce** during training, as it can avoid some penalty by generating sensible sentences such as questions. **Imitation+Gaussian-RL** gets higher rewards than both **Imitation** and **Reinforce**, indicating that the RL component reshapes the action policy toward higher rewards. However, as the Gaussian policy optimizes the action policy indirectly in a latent feature space, it is less efficient for exploration and learning. **Proposed** achieves the highest final reward during training.

We train the models using the Animal dataset and evaluate them on the Fruit dataset; Figure 4 sum-

[3] https://github.com/PaddlePaddle/Paddle

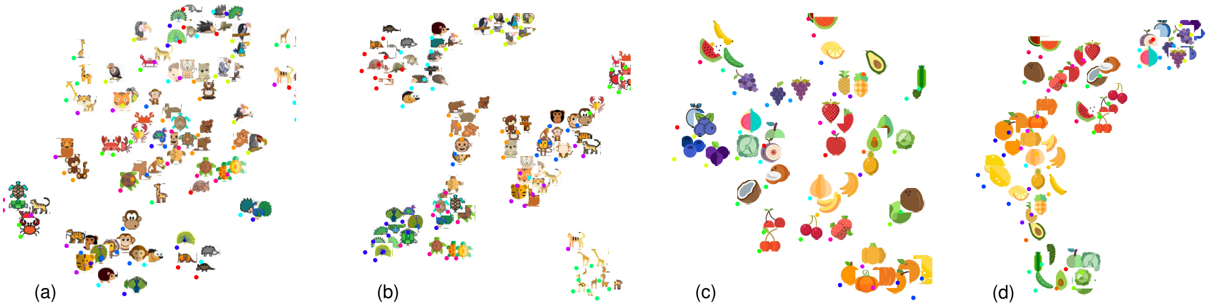

Figure 6: **Visualization of the CNN features** with t-SNE. Ten classes randomly sampled from **(a-b)** the Animal dataset and **(c-d)** the Fruit dataset, with features extracted using the visual encoder trained without (a, c) and with (b, d) image variations on the the Animal dataset.

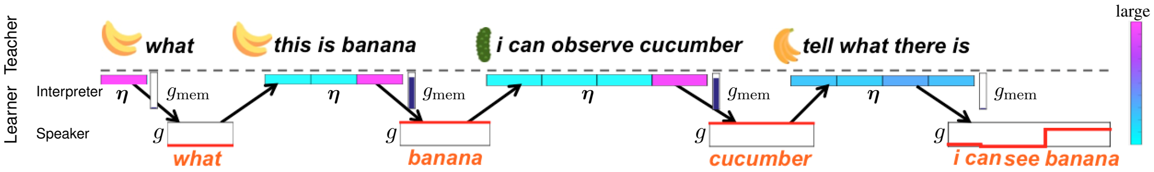

Figure 7: **Example results** of the proposed approach on novel classes. The learner can ask about the new class and use the interpreter to extract useful information from the teacher's sentence via word-level attention $\boldsymbol{\eta}$ and content importance g_{mem} jointly. The speaker uses the fusion gate g to adaptively switch between signals from RNN (small g) and external memory (large g) to generate sentence responses.

marizes the success rate and average reward over 1K testing sessions. As can be observed, **Reinforce** achieves the lowest success rate (0.0%) and reward (-6.0) due to its inherent inefficiency in learning. **Imitation** performs better than **Reinforce** in terms of both its success rate (28.6%) and reward value (-2.7). **Imitation+Gaussian-RL** achieves a higher reward (-1.2) during testing, but its success rate (32.1%) is similar to that of **Imitation**, mainly due to the rigorous criteria for success. **Proposed** reaches the highest success rate (97.4%) and average reward ($+1.1$)[4], outperforming all baseline methods by a large margin. From this experiment, it is clear that imitation with a proper usage of reinforcement is crucial for achieving adaptive behaviors (*e.g.*, asking questions about novel objects and generating answers or statements about learned objects proactively).

5.2 Learning with Image Variations

To evaluate the impact of within-class image variations on one-shot concept learning, we train models with and without image variations, and during testing compare their performance under different image variation ratios (the chance of a novel image instance being present within a session) as shown in Figure 5. It is observed that the performance of the model trained without image variations drops significantly as the variation ratio increases. We also evaluate the performance of models trained under a variation ratio of 0.5. Figure 5 clearly shows that although there is also a performance drop, which is expected, the performance degrades more gradually, indicating the importance of image variation for learning one-shot concepts. Figure 6 visualizes sampled training and testing images represented by their corresponding features extracted using the visual encoder trained without and with image variations. Clusters of visually similar concepts emerge in the feature space when trained with image variations, indicating that a more discriminative visual encoder was obtained for learning generalizable concepts.

5.3 Sentence-Level Task

We further evaluate the model on sentence-level tasks. Teacher's sentences are generated using the grammar as shown in Table 1 and have a number of variations with sentence lengths ranging from one to five. Example sentences from the teacher are presented in Appendix A.1. This task is more challenging than the word-level task in two ways: i) information processing is more difficult as the learner has to learn to extract useful information which could appear at different locations of the sentence; ii) the sentence generation is also more

[4] The testing reward is higher than the training reward mainly due to the action sampling in training for exploration.

difficult than the word-level task and the learner has to adaptively fuse information from RNN and external memory to generate a complete sentence. Comparison of different approaches in terms of their success rates and average rewards on the novel test set are shown in Figure 8. As can be observed from the figure, **Proposed** again outperforms all other compared methods in terms of both success rate (82.8%) and average reward (+0.8), demonstrating its effectiveness even for the more complex sentence-level task.

We also visualize the information extraction and the adaptive sentence composing process of the proposed approach when applied to a test set. As shown in Figure 7, the agent learns to extract useful information from the teacher's sentence and use the content importance gate to control what content is written into the external memory. Concretely, sentences containing object names have a larger $g_{\rm mem}$ value, and the word corresponding to object name has a larger value in the attention vector $\boldsymbol{\eta}$ compared to other words in the sentence. The combined effect of $\boldsymbol{\eta}$ and $g_{\rm mem}$ suggests that words corresponding to object names have higher likelihoods of being written into the external memory. The agent also successfully learns to use the external memory for storing the information extracted from the teacher's sentence, to fuse it adaptively with the signal from the RNN (capturing the syntactic structure) and to generate a complete sentence with the new concept included. The value of the fusion gate g is small when generating words like "*what*,", "*i*," "*can*," and "*see*," meaning it mainly relies on the signal from the RNN for generation (*c.f.*, Eqn.(2) and Figure 7). In contrast, when generating object names (*e.g.*, "*banana*," and "*cucumber*"), the fusion gate g has a large value, meaning that there is more emphasis on the signal from the external memory. This experiment showed that the proposed approach is applicable to the more complex sentence-level task for language learning and one-shot learning. More interestingly, it learns an interpretable operational process, which can be easily understood. More results including example dialogues from different approaches are presented in Appendix A.4.

6 Discussion

We have presented an approach for grounded language acquisition with one-shot visual concept learning in this work. This is achieved by purely interacting with a teacher and learning from feedback arising naturally during interaction through joint imitation and reinforcement learning, with a memory augmented neural network. Experimental results show that the proposed approach is effective for language acquisition with one-shot visual concept learning across several different settings compared with several baseline approaches.

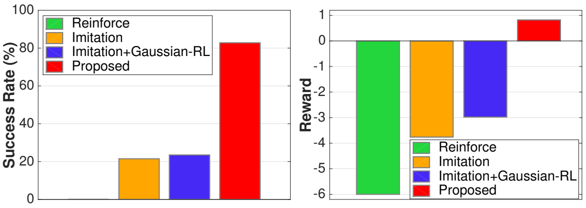

Figure 8: **Test performance** for sentence-level task with image variations (variation ratio=0.5).

In the current work, we have designed and used a computer game (synthetic task with synthetic language) for training the agent. This is mainly due to the fact that there is no existing dataset to the best of our knowledge that is adequate for developing our addressed interactive language learning and one-shot learning problem. For our current design, although it is an artificial game, there is a reasonable amount of variations both within and across sessions, e.g., the object classes to be learned within a session, the presentation order of the selected classes, the sentence patterns and image instances to be used etc. All these factors contribute to the increased complexity of the learning task, making it non-trivial and already very challenging to existing approaches as shown by the experimental results. While offering flexibility in training, one downside of using a synthetic task is its limited amount of variation compared with real-world scenarios with natural languages. Although it might be non-trivial to extend the proposed approach to real natural language directly, we regard this work as an initial step towards this ultimate ambitious goal and our game might shed some light on designing more advanced games or performing real-world data collection. We plan to investigate the generalization and application of the proposed approach to more realistic environments with more diverse tasks in future work.

Acknowledgments

We thank the reviewers and PC members for theirs efforts in helping improving the paper. We thank Xiaochen Lian and Xiao Chu for their discussions.

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*.

Alan Baddeley. 1992. Working memory. *Science*, 255(5044):556–559.

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. 2017. An actor-critic algorithm for sequence prediction. In *International Conference on Learning Representations (ICLR)*.

Ann C. Baker, Patricia J. Jensen, and David A. Kolb. 2002. *Conversational Learning: An Experiential Approach to Knowledge Creation*. Copley Publishing Group.

Arielle Borovsky, Marta Kutas, and Jeff Elman. 2003. Learning to use words: Event related potentials index single-shot contextual word learning. *Cognzition*, 116(2):289–296.

K. Cho, B. Merrienboer, C. Glehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. 2014a. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Kyunghyun Cho, Bart van Merriënboer, Çalar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

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

Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. 2016. Benchmarking deep reinforcement learning for continuous control. In *International Conference on International Conference on Machine Learning (ICML)*.

J. Duchi, E. Hazan, and Y. Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159.

Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. 2016. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS)*.

Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines. *CoRR*, abs/1410.5401.

Ji He, Jianshu Chen, Xiaodong He, Jianfeng Gao, Lihong Li, Li Deng, and Mari Ostendorf. 2016. Deep reinforcement learning with a natural language action space. In *Association for Computational Linguistics (ACL)*.

Derek M. Houston and Richard T. Miyamoto. 2011. Effects of early auditory experience on word learning and speech perception in deaf children with cochlear implants: Implications for sensitive periods of language development. *Otol Neurotol*, 31(8):1248–1253.

Patricia K. Kuhl. 2004. Early language acquisition: cracking the speech code. *Nat Rev Neurosci*, 5(2):831–843.

Brenden M. Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua B. Tenenbaum. 2011. One shot learning of simple visual concepts. In *Proceedings of the 33th Annual Meeting of the Cognitive Science Society*.

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. 2017. Multi-agent cooperation and the emergence of (natural) language. In *International Conference on Learning Representations (ICLR)*.

Jiwei Li, Alexander H. Miller, Sumit Chopra, MarcAurelio Ranzato, and Jason Weston. 2017. Learning through dialogue interactions by asking questions. In *International Conference on Learning Representations (ICLR)*.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016. Deep reinforcement learning for dialogue generation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. 2013. Playing Atari with deep reinforcement learning. In *NIPS Deep Learning Workshop*.

Igor Mordatch and Pieter Abbeel. 2018. Emergence of grounded compositional language in multi-agent populations. In *Association for the Advancement of Artificial Intelligence (AAAI)*.

Anna Ingeborg Petursdottir and James R. Mellor. 2016. Reinforcement contingencies in language acquisition. *Policy Insights from the Behavioral and Brain Sciences*, 4(1):25–32.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *International Conference on Learning Representations (ICLR)*.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. 2016. Meta-learning with memory-augmented neural networks. In *International Conference on Machine Learning (ICML)*.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Association for the Advancement of Artificial Intelligence (AAAI)*.

B. F. Skinner. 1957. *Verbal Behavior*. Copley Publishing Group.

Bradly C. Stadie, Pieter Abbeel, and Ilya Sutskever. 2017. Third-person imitation learning. In *International Conference on Learning Representations (ICLR)*.

Amanda Stent and Srinivas Bangalore. 2014. *Natural Language Generation in Interactive Systems*. Cambridge University Press.

Florian Strub, Harm de Vries, Jérémie Mary, Bilal Piot, Aaron C. Courville, and Olivier Pietquin. 2017. End-to-end optimization of goal-driven and visually grounded dialogue systems. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. 2016. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems (NIPS)*.

Richard S. Sutton and Andrew G. Barto. 1998. *Reinforcement Learning: An Introduction*. MIT Press.

S. I. Wang, P. Liang, and C. Manning. 2016. Learning language games through interaction. In *Association for Computational Linguistics (ACL)*.

Z. Wang, V. Bapst, N. Heess, V. Mnih, R. Munos, K. Kavukcuoglu, and N. Freitas. 2017. Sample efficient actor-critic with experience replay. In *International Conference on Learning Representations (ICLR)*.

Sandra R. Waxman. 2004. *Everything had a name, and each name gave birth to a new thought: links between early word learning and conceptual organization*. Cambridge, MA: The MIT Press.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Peihao Su, David Vandyke, and Steve J. Young. 2015. Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Jason Weston. 2016. Dialog-based language learning. In *Advances in Neural Information Processing Systems (NIPS)*.

Mark Woodward and Chelsea Finn. 2016. Active one-shot learning. In *NIPS Deep Reinforcement Learning Workshop*.

Haonan Yu, Haichao Zhang, and Wei Xu. 2018. Interactive grounded language acquisition and generalization in a 2D world. In *International Conference on Learning Representations (ICLR)*.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. SeqGAN: Sequence generative adversarial nets with policy gradient. In *Association for the Advancement of Artificial Intelligence (AAAI)*.

Haichao Zhang, Haonan Yu, and Wei Xu. 2017. Listen, interact and talk: Learning to speak via interaction. In *NIPS Workshop on Visually-Grounded Interaction and Language*.

A Appendix

A.1 Datasets and Example Sentences

The Animal dataset contains 40 animal classes with 408 images in total, with about 10 images per class on average. The Fruit dataset contains 16 classes and 48 images in total with 3 images per class. The object classes and images are summarized in Table 2 and Figure 9. Example sentences from the teacher in different cases (questioning, answering, and saying nothing) are presented in Table 3.

Table 2: Object classes for two datasets.

Set	#cls/img	Object Names
Animal	40/408	armadillo, bear, bull, butterfly, camel, cat, chicken, cobra, condor, cow, crab, crocodile, deer, dog, donkey, duck, elephant, fish, frog, giraffe, goat, hedgehog, kangaroo, koala, lion, monkey, octopus, ostrich, panda, peacock, penguin, pig, rhinoceros, rooster, seahorse, snail, spider, squirrel, tiger, turtle
Fruit	16/48	apple, avocado, banana, blueberry, cabbage, cherry, coconut, cucumber, fig, grape, lemon, orange, pineapple, pumpkin, strawberry, watermelon

Table 3: Example sentences from the teacher.

Category	Example Sentences
Empty	*“”*
Question	*“what”* *“what is it”* *“what is this”* *“what is there”* *“what do you see”* *“what can you see”* *“what do you observe”* *“what can you observe”* *“tell what it is”* *“tell what this is”* *“tell what there is”* *“tell what you see”* *“tell what you can see”* *“tell what you observe”* *“tell what you can observe”*
Answer / Statement	*“apple”* *“it is apple”* *“this is apple”* *“there is apple”* *“i see apple”* *“i observe apple”* *“i can see apple”* *“i can observe apple”*

A.2 Network Details

A.2.1 Visual Encoder

The visual encoder takes an input image and outputs a visual feature vector. It is implemented as a convolutional neural network (CNN) followed by fully connected (FC) layers. The CNN has four layers. Each layer has 32, 64, 128, 256 filters of size 3×3, followed by max-poolings with a pooling size of 3 and a stride of 2. The ReLU activation is used for all layers. Two FC layers with output dimensions of 512 and 1024 are used after the CNN, with ReLU and a linear activations respectively.

A.2.2 Interpreter and Speaker

Interpreter and *speaker* are implemented with interpreter-RNN and speaker-RNN respectively and they share parameters. The RNN is implemented using the Gated Recurrent Unit (Cho et al., 2014a) with a state dimension of 1024. Before inputing to the RNN, word ids are first projected to a word embedding vector of dimension 1024 followed with two FC layers with ReLU activations and a third FC layer with linear activation, all having output dimensions of 1024.

A.2.3 Fusion Gate

The fusion gate g is implemented as two FC layers with ReLU activations a third FC layer with a sigmoid activation. The output dimensions are 50, 10 and 1 for each layer respectively.

A.2.4 Controller

The controller $f(\cdot)$ together with the identity mapping forms a residue-structured network as

$$\mathbf{c} = \mathbf{h} + f(\mathbf{h}). \tag{7}$$

$f(\cdot)$ is implemented as two FC layers with ReLU activations and a third FC layer with a linear activation, all having an output dimensions of 1024.

A.2.5 Value Network

The value network is introduced to estimate the expected accumulated future reward. It takes the state vector of interpreter-RNN $\mathbf{h}_{\rm I}$ and the confidence c as input. It is implemented as two FC layers with ReLU activations and output dimensions of 512 and 204 respectively. The third layer is another FC layer with a linear activation and an output dimension of 1. It is trained by minimizing a cost as (Sutton and Barto, 1998)

$$\mathcal{L}^{\rm V}{=}\mathbb{E}_{p_\theta^{\rm S}}\big(V(\mathbf{h}_{\rm I}^t, c^t) - r^{t+1}{-}\lambda V'(\mathbf{h}_{\rm I}^{t+1}, c^{t+1})\big)^2.$$

$V'(\cdot)$ denotes a target version of the value network, whose parameters remain fixed until copied from $V(\cdot)$ periodically (Mnih et al., 2013).

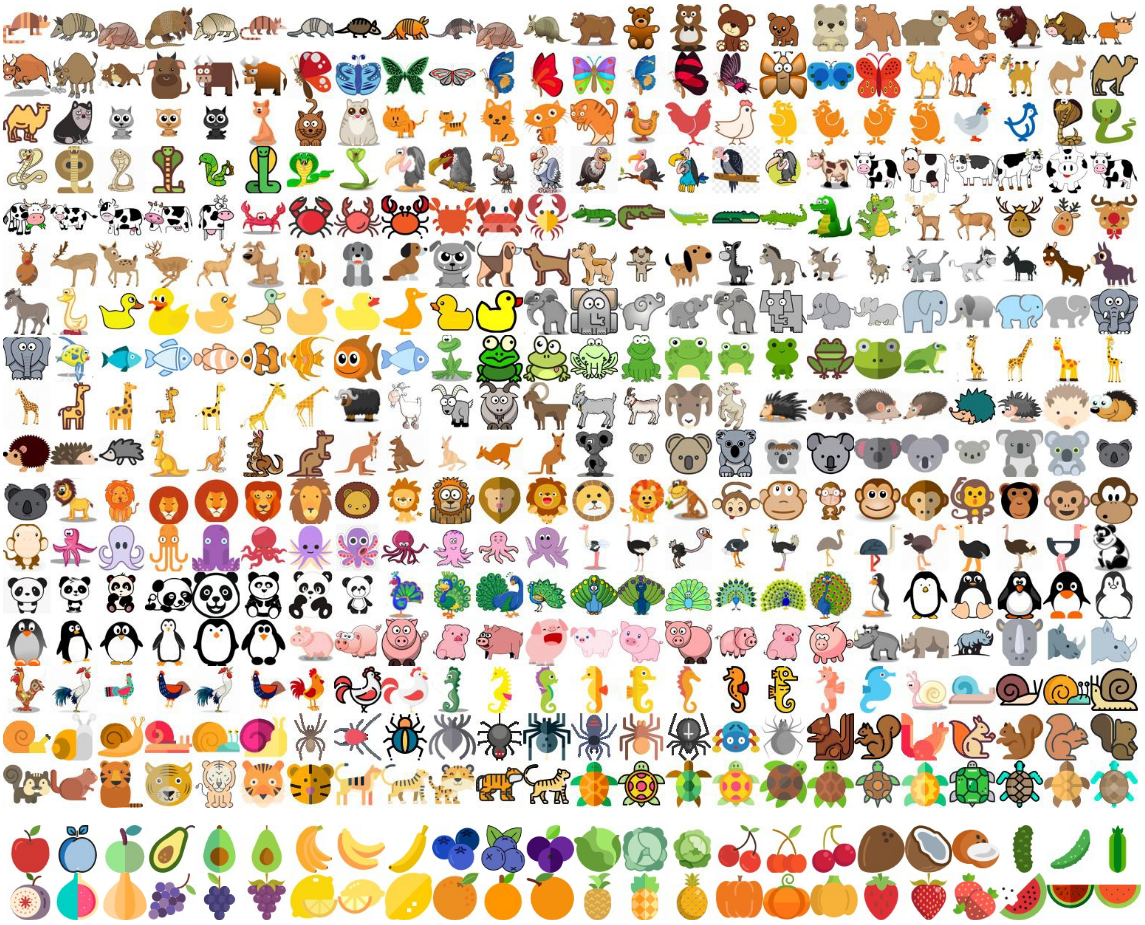

Figure 9: **Dataset images. Top**: Animal dataset. **Bottom**: Fruit dataset.

A.2.6 Confidence Score

The confidence score c is defined as follows:

$$c{=}\max(\mathbf{E}^{\mathsf{T}}\mathbf{r}), \tag{8}$$

where $\mathbf{E}{\in}\mathbb{R}^{d\times k}$ is the word embedding table, with d the embedding dimension and k the vocabulary size. $\mathbf{r}{\in}\mathbb{R}^{d}$ is the vector read out from the sentence modality of the external memory as:

$$\mathbf{r} = \mathbf{M}_s\boldsymbol{\alpha}, \tag{9}$$

where $\boldsymbol{\alpha}$ a soft reading weight obtained through the visual modality by calculating the cosine similarities between $\mathbf{k}_v$ and the slots of $\mathbf{M}_v$. The content stored in the memory is extracted from teacher's sentence $\{w_1, w_2, \cdots, w_i, \cdots, w_n\}$ as (detailed in Section A.3):

$$\mathbf{c}_s = [\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_i \cdots, \boldsymbol{w}_n]\boldsymbol{\eta}, \tag{10}$$

where $\boldsymbol{w}_i{\in}\mathbb{R}^{d}$ denotes the embedding vector extracted from the word embedding table $\mathbf{E}$ for the word w_i. Therefore, for a well-learned concept with effective $\boldsymbol{\eta}$ for information extraction and effective $\boldsymbol{\alpha}$ for information retrieval, $\mathbf{r}$ should be an embedding vector mainly corresponding to the label word associated with the visual image. Therefore, the value of c should be large and the maximum is reached at the location where that label word resides in the embedding table. For a completely novel concept, as the memory contains no information about it, the reading attention α will not be focused and thus $\mathbf{r}$ would be an averaging of a set of existing word embedding vectors in the external memory, leading to a small c value.

A.3 Sentence Content Extraction and Importance Gate

A.3.1 Content Extraction

We use an attention scheme to extract useful information from a sentence to be written into memory. Given a sentence $\mathbf{w} = \{w_1, w_2, \cdots, w_n\}$ and the corresponding word embedding vectors $\{\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_n\}$, a summary of the sentence is firstly generated using a bidirectional RNN, yielding the states $\{\overrightarrow{\boldsymbol{w}_1}, \overrightarrow{\boldsymbol{w}_2}, \cdots, \overrightarrow{\boldsymbol{w}_n}\}$ for the forward pass and $\{\overleftarrow{\boldsymbol{w}_1}, \overleftarrow{\boldsymbol{w}_2}, \cdots, \overleftarrow{\boldsymbol{w}_n}\}$ for the backward pass. The summary vector is the concatenation of the last state of forward pass and the first state of the backward pass:

$$\boldsymbol{s} = \mathrm{concat}(\overrightarrow{\boldsymbol{w}_n}, \overleftarrow{\boldsymbol{w}_1}). \tag{11}$$

The context vector is the concatenation of the word embedding vector and the state vectors of both forward and backward passes:

$$\bar{\boldsymbol{w}}_i = \mathrm{concat}(\boldsymbol{w}_\mathrm{i}, \overrightarrow{\boldsymbol{w}_i}, \overleftarrow{\boldsymbol{w}_i}). \tag{12}$$

The word level attention $\boldsymbol{\eta} = [\eta_1, \eta_2, \cdots, \eta_i, \cdots]$ is computed as the cosine similarity between transformed sentence summary vector $\boldsymbol{s}$ and each context vector $\bar{\boldsymbol{w}}_i$:

$$\eta_i = \cos\big(f^{\theta_1}_{\mathrm{MLP}}(\boldsymbol{s}), f^{\theta_2}_{\mathrm{MLP}}(\bar{\boldsymbol{w}}_i)\big). \tag{13}$$

Both MLPs contain two FC layers with output dimensions of 1024 and a linear and a Tanh activation for each layer respectively. The content $\mathbf{c}_s$ to be written into the memory is computed as:

$$\mathbf{c}_s = \mathbf{W}\boldsymbol{\eta} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_n]\boldsymbol{\eta}. \tag{14}$$

A.3.2 Importance Gate

The content importance gate is computed as $g_{\mathrm{mem}}{=}\sigma(f_{\mathrm{MLP}}(\mathbf{s}))$, meaning that the importance of the content to be written into the memory depends on the sentence from the teacher. The MLP contains two FC layers with ReLU activation and output dimensions of 50 and 30 respectively. Another FC layer with a linear activation, and an output dimension of 20 is used. The output layer is an FC layer with an output dimension of 1 and a sigmoid activation σ .

A.4 Example Dialogues on Novel Data

We train models on the Animal dataset and perform the evaluation on the Fruit dataset. Example dialogues of different approaches are shown in Table 4. It is observed that **Reinforce** arrives at a policy that the learner keeps silent. Both **Imitation** and **Imitation+Gaussian-RL** can generate sensible sentences, but cannot speak adaptively according to context. **Proposed** can speak according to context adaptively, asking information about novel classes, and answering questions after being taught only once by the teacher.

Table 4: Example dialogues from different approaches.

Reinforce

🥒	T: *what*	
	L:	✔
🥒	T: *i can see cucumber*	
	L:	✘
🥒	T: *there is cucumber*	
	L:	✘
🥬	T: *tell what you can observe*	
	L:	✔
🥬	T: *i observe cabbage*	
	L:	✘
🫛	T:	
	L:	✘
	FAILURE	

🍊	T: *there is orange*	
	L:	✘
🍍	T: *tell what it is*	
	L:	✔
🍍	T: *i see pineapple*	
	L:	✘
🍊	T: *what can you see*	
	L:	✔
🍊	T: *there is orange*	
	L:	✘
🍍	T: *what can you see*	
	L:	✘
	FAILURE	

🍇	T: *i observe grape*	
	L:	✘
🍊	T: *i can observe orange*	
	L:	✘
🍇	T: *what is it*	
	L:	✘
🍇	T: *i see grape*	
	L:	✘
🍊	T: *tell what it is*	
	L:	✘
🍊	T: *i observe orange*	
	L:	✘
	FAILURE	

Imitation

🍏	T: *apple*	
	L: *what is this*	✘
🫐	T: *tell what it is*	
	L: *i can see condor*	✘
🍎	T: *this is apple*	
	L: *what is this*	✘
🫐	T: *tell what there is*	
	L: *i can see apple*	✘
🫐	T:	
	L: *what is this*	✘
🫐	T: *i can observe apple*	
	L: *what is this*	✘
	FAILURE	

🍎	T: *what is there*	
	L: *i can observe condor*	✘
🫛	T: *tell what you see*	
	L: *i can see condor*	✘
🫐	T: *tell what you can see*	
	L: *i can see condor*	✘
🍏	T: *what is there*	
	L: *i can see condor*	✘
🥒	T:	
	L: *what is this*	✔
🥒	T: *i can see cucumber*	
	L: *what is this*	✘
	FAILURE	

🍋	T: *i see lemon*	
	L: *what is this*	✘
🍋	T: *i see lemon*	
	L: *what is this*	✘
🍍	T: *tell what you can observe*	
	L: *i can see lemon*	✘
🍋	T: *tell what there is*	
	L: *what is this*	✘
🍋	T: *it is lemon*	
	L: *what is this*	✘
🍍	T: *tell what you observe*	
	L: *what is this*	✘
	FAILURE	

Imitation+Gaussian-RL

🍓	T: *i can see strawberry*	
	L: *i can see snail*	✘
🍓	T: *it is strawberry*	
	L: *i can see strawberry*	✔
🍓	T: *i can see strawberry*	
	L: *i can see strawberry*	✔
🫐	T: *blueberry*	
	L: *i can see blueberry*	✔
🫐	T: *what*	
	L: *i can see blueberry*	✔
	FAILURE	

🍋	T: *what do you observe*	
	L: *i can see tiger*	✘
🥥	T: *i can observe coconut*	
	L: *i can see coconut*	✔
🍋	T: *what can you see*	
	L: *i can see tiger*	✘
🥥	T: *what*	
	L: *i can see dog*	✘
🍋	T: *tell what you can observe*	
	L: *i can see tiger*	✘
	FAILURE	

🍓	T: *i can observe strawberry*	
	L: *i can see strawberry*	✔
🍍	T: *this is pineapple*	
	L: *i can see pineapple*	✔
🍍	T: *this is pineapple*	
	L: *i can see pineapple*	✔
🍓	T: *what do you see*	
	L: *i can see pineapple*	✘
	FAILURE	

Proposed

🍉	T:	
	L: *tell what you observe*	✔
🍉	T: *this is watermelon*	
	L: *watermelon*	✔
🍋	T: *tell what there is*	
	L: *tell what you see*	✔
🍋	T: *i observe lemon*	
	L: *lemon*	✔
🍋	T:	
	L: *lemon*	✔
	SUCCESS	

🍒	T: *tell what you can observe*	
	L: *tell what you observe*	✔
🍒	T: *i observe cherry*	
	L: *cherry*	✔
🍊	T: *i observe orange*	
	L: *orange*	✔
🍊	T: *tell what there is*	
	L: *orange*	✔
	SUCCESS	

🫐	T:	
	L: *tell what you observe*	✔
🫐	T: *i see blueberry*	
	L: *blueberry*	✔
🍊	T: *there is orange*	
	L: *orange*	✔
🍊	T: *i can observe orange*	
	L: *orange*	✔
🫐	T: *tell what it is*	
	L: *this is blueberry*	✔
	SUCCESS	

🍇	T: *i can see grape*	
	L: *grape*	✔
🎃	T:	
	L: *tell what you observe*	✔
🎃	T: *i can see pumpkin*	
	L: *pumpkin*	✔
🍇	T:	
	L: *grape*	✔
	SUCCESS	

🥑	T: *what*	
	L: *tell what you observe*	✔
🥑	T: *i can see avocado*	
	L: *avocado*	✔
🥥	T: *this is coconut*	
	L: *coconut*	✔
🥑	T: *what is there*	
	L: *this is avocado*	✔
	SUCCESS	

🥥	T: *it is coconut*	
	L: *i can see coconut*	✔
🎃	T: *this is pumpkin*	
	L: *i can see pumpkin*	✔
🥥	T: *what is it*	
	L: *i can see pumpkin*	✘
🎃	T: *what do you see*	
	L: *i can see pumpkin*	✔
	FAILURE	