

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 inputting 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

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

$f(\cdot)$ 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}_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)

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

$V'(\cdot)$ 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:

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

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:

$$\mathbf{r} = \mathbf{M}_s \boldsymbol{\alpha}, \quad (9)$$

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 $\{w_1, w_2, \dots, w_i, \dots, w_n\}$ as (detailed in Section A.3):

$$\mathbf{c}_s = [w_1, w_2, \dots, w_i, \dots, w_n] \boldsymbol{\eta}, \quad (10)$$

where $\mathbf{w}_i \in \mathbb{R}^d$ 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 $\mathbf{w} = \{w_1, w_2, \dots, w_n\}$ and the corresponding word embedding vectors $\{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n\}$, a summary of the sentence is firstly generated using a bidirectional RNN, yielding the states $\{\overrightarrow{\mathbf{w}}_1, \overrightarrow{\mathbf{w}}_2, \dots, \overrightarrow{\mathbf{w}}_n\}$ for the forward pass and $\{\overleftarrow{\mathbf{w}}_1, \overleftarrow{\mathbf{w}}_2, \dots, \overleftarrow{\mathbf{w}}_n\}$ for the backward pass. The summary vector is the concate-

nation of the last state of forward pass and the first state of the backward pass:

$$\mathbf{s} = \text{concat}(\overrightarrow{\mathbf{w}}_n, \overleftarrow{\mathbf{w}}_1). \quad (11)$$

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

$$\bar{\mathbf{w}}_i = \text{concat}(\mathbf{w}_i, \overrightarrow{\mathbf{w}}_i, \overleftarrow{\mathbf{w}}_i). \quad (12)$$

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

$$\eta_i = \cos(f_{\text{MLP}}^{\theta_1}(\mathbf{s}), f_{\text{MLP}}^{\theta_2}(\bar{\mathbf{w}}_i)). \quad (13)$$

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:

$$\mathbf{c}_s = \mathbf{W}\boldsymbol{\eta} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n]\boldsymbol{\eta}. \quad (14)$$



















A.3.2 Importance Gate



















The content importance gate is computed as $g_{\text{mem}} = \sigma(f_{\text{MLP}}(\mathbf{s}))$, 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 | |  | T: there is orange | |  | T: i observe grape | |
| | L: | ✓ | | L: | ✗ | | L: | ✗ |
|  | T: i can see cucumber | |  | T: tell what it is | |  | T: i can observe orange | |
| | L: | ✗ | | L: | ✓ | | L: | ✗ |
|  | T: there is cucumber | |  | T: i see pineapple | |  | T: what is it | |
| | L: | ✗ | | L: | ✗ | | L: | ✗ |
|  | T: tell what you can observe | |  | T: what can you see | |  | T: i see grape | |
| | L: | ✓ | | L: | ✓ | | L: | ✗ |
|  | T: i observe cabbage | |  | T: there is orange | |  | T: tell what it is | |
| | L: | ✗ | | L: | ✗ | | L: | ✗ |
|  | T: | |  | T: what can you see | |  | T: i observe orange | |
| | L: | ✗ | | L: | ✗ | | L: | ✗ |
| | FAILURE | | | FAILURE | | | FAILURE | |

| | | | Imitation | | | | | |
|---|------------------------|---|---|--------------------------|---|---|------------------------------|---|
|  | T: apple | |  | T: what is there | |  | T: i see lemon | |
| | L: what is this | ✗ | | L: i can observe condor | ✗ | | L: what is this | ✗ |
|  | T: tell what it is | |  | T: tell what you see | |  | T: i see lemon | |
| | L: i can see condor | ✗ | | L: i can see condor | ✗ | | L: what is this | ✗ |
|  | T: this is apple | |  | T: tell what you can see | |  | T: tell what you can observe | |
| | L: what is this | ✗ | | L: i can see condor | ✗ | | L: i can see lemon | ✗ |
|  | T: tell what there is | |  | T: what is there | |  | T: tell what there is | |
| | L: i can see apple | ✗ | | L: i can see condor | ✗ | | L: what is this | ✗ |
|  | T: | |  | T: | |  | T: it is lemon | |
| | L: what is this | ✗ | | L: what is this | ✓ | | L: what is this | ✗ |
|  | T: i can observe apple | |  | T: i can see cucumber | |  | T: tell what you observe | |
| | L: what is this | ✗ | | L: what is this | ✗ | | L: what is this | ✗ |
| | FAILURE | | | FAILURE | | | FAILURE | |

| | | | Imitation+Gaussian-RL | | | | | |
|---|-------------------------|---|---|------------------------------|---|---|-----------------------------|---|
|  | T: i can see strawberry | |  | T: what do you observe | |  | T: i can observe strawberry | |
| | L: i can see snail | ✗ | | L: i can see tiger | ✗ | | L: i can see strawberry | ✓ |
|  | T: it is strawberry | |  | T: i can observe coconut | |  | T: this is pineapple | |
| | L: i can see strawberry | ✓ | | L: i can see coconut | ✓ | | L: i can see pineapple | ✓ |
|  | T: i can see strawberry | |  | T: what can you see | |  | T: this is pineapple | |
| | L: i can see strawberry | ✓ | | L: i can see tiger | ✗ | | L: i can see pineapple | ✓ |
|  | T: blueberry | |  | T: what | |  | T: what do you see | |
| | L: i can see blueberry | ✓ | | L: i can see dog | ✗ | | L: i can see pineapple | ✗ |
|  | T: what | |  | T: tell what you can observe | | | FAILURE | |
| | L: i can see blueberry | ✓ | | L: i can see tiger | ✗ | | | |
| | FAILURE | | | FAILURE | | | | |

| | | | Proposed | | | | | |
|---|--------------------------|---|---|------------------------------|---|---|--------------------------|---|
|  | T: | |  | T: tell what you can observe | |  | T: | |
| | L: tell what you observe | ✓ | | L: tell what you observe | ✓ | | L: tell what you observe | ✓ |
|  | T: this is watermelon | |  | T: i observe cherry | |  | T: i see blueberry | |
| | L: watermelon | ✓ | | L: cherry | ✓ | | L: blueberry | ✓ |
|  | T: tell what there is | |  | T: i observe orange | |  | T: there is orange | |
| | L: tell what you see | ✓ | | L: orange | ✓ | | L: orange | ✓ |
|  | T: i observe lemon | |  | T: tell what there is | |  | T: i can observe orange | |
| | L: lemon | ✓ | | L: orange | ✓ | | L: orange | ✓ |
|  | T: | | | SUCCESS | |  | T: tell what it is | |
| | L: lemon | ✓ | | | | | L: this is blueberry | ✓ |
| | SUCCESS | | | | | | SUCCESS | |

| | | | | | | | | |
|---|--------------------------|---|---|--------------------------|---|---|----------------------|---|
|  | T: i can see grape | |  | T: what | |  | T: it is coconut | |
| | L: grape | ✓ | | L: tell what you observe | ✓ | | L: i can see coconut | ✓ |
|  | T: | |  | T: i can see avocado | |  | T: this is pumpkin | |
| | L: tell what you observe | ✓ | | L: avocado | ✓ | | L: i can see pumpkin | ✓ |
|  | T: i can see pumpkin | |  | T: this is coconut | |  | T: what is it | |
| | L: pumpkin | ✓ | | L: coconut | ✓ | | L: i can see pumpkin | ✗ |
|  | T: | |  | T: what is there | |  | T: what do you see | |
| | L: grape | ✓ | | L: this is avocado | ✓ | | L: i can see pumpkin | ✓ |
| | SUCCESS | | | SUCCESS | | | FAILURE | |