Learning by Asking Questions

Ishan Misra[1] * Ross Girshick[2] Rob Fergus[2]
Martial Hebert[1] Abhinav Gupta[1] Laurens van der Maaten[2]
[1]Carnegie Mellon University [2]Facebook AI Research

Abstract

We introduce an interactive learning framework for the development and testing of intelligent visual systems, called learning-by-asking (LBA). We explore LBA in context of the Visual Question Answering (VQA) task. LBA differs from standard VQA training in that most questions are not observed during training time, and the learner must ask questions it wants answers to. Thus, LBA more closely mimics natural learning and has the potential to be more data-efficient than the traditional VQA setting. We present a model that performs LBA on the CLEVR dataset, and show that it automatically discovers an easy-to-hard curriculum when learning interactively from an oracle. Our LBA generated data consistently matches or outperforms the CLEVR train data and is more sample efficient. We also show that our model asks questions that generalize to state-of-the-art VQA models and to novel test time distributions.

1. Introduction

Machine learning models have led to remarkable progress in visual recognition. However, while the training data that is fed into these models is crucially important, it is typically treated as predetermined, static information. Our current models are *passive* in nature: they rely on training data curated by humans and have no control over this supervision. This is in stark contrast to the way we humans learn — by *interacting* with our environment to gain information. The interactive nature of human learning makes it sample efficient (there is less redundancy during training) and also yields a learning curriculum (we ask for more complex knowledge as we learn).

In this paper, we argue that next-generation recognition systems need to have *agency* — the ability to decide what information they need and how to get it. We explore this in the context of visual question answering (VQA; [4, 23, 58]). Instead of training on a fixed, large-scale dataset, we propose an alternative *interactive* VQA setup called *learning-by-asking* (LBA): at training time, the learner receives only images and decides *what questions to ask*. Questions asked by the learner are answered by an oracle (human supervision). At test-time, LBA is evaluated exactly like VQA using well understood metrics.

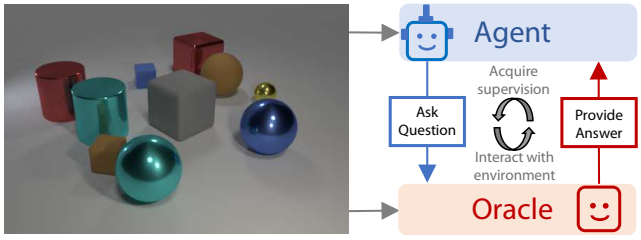


Figure 1: **The Learning-by-Asking (LBA) paradigm.** We present an open-world Visual Question Answering (VQA) setting in which an agent interactively learns by asking questions to an oracle. Unlike standard VQA training, which assumes a fixed dataset of questions, in LBA the agent has the potential to learn more quickly by asking "good" questions, much like a bright student in a class. LBA does not alter the test-time setup of VQA.

The interactive nature of LBA requires the learner to construct meta-knowledge about what it knows and to select the supervision it needs. If successful, this facilitates more sample efficient learning than using a fixed dataset, because the learner will not ask redundant questions.

We explore the proposed LBA paradigm in the context of the CLEVR dataset [23], which is an artificial universe in which the number of unique objects, attributes, and relations are limited. We opt for this synthetic setting because there is little prior work on asking questions about images: CLEVR allows us to perform a controlled study of the algorithms needed for asking questions. We hope to transfer the insights obtained from our study to a real-world setting.

Building an interactive learner that can ask questions is a challenging task. First, the learner needs to have a "language" model to form questions. Second, it needs to understand the input image to ensure the question is relevant and coherent. Finally (and most importantly), in order to be sample efficient, the learner should be able to evaluate its own knowledge (self-evaluate) and ask questions which

*Work done during internship at Facebook AI Research.

will help it to learn new information about the world. The only supervision the learner receives from the interaction is the answer to the questions it poses. Interestingly, recent work [43] shows that even humans are not good at asking informative questions.

We present and study a model for LBA that combines ideas from visually grounded language generation [38], curriculum learning [6], and VQA. Specifically, we develop an epsilon-greedy [51] learner that asks questions and uses the corresponding answers to train a standard VQA model. The learner focuses on mastering concepts that it can rapidly improve upon, before moving to new types of questions. We demonstrate that our LBA model not only asks meaningful questions, but also *matches the performance* of human-curated data. Our model is also *sample efficient* and by interactively asking questions it reduces the number of training samples needed to obtain the baseline question-answering accuracy by 40%.

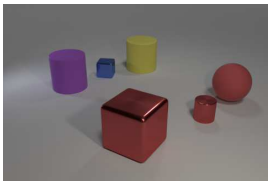
2. Related Work

Visual question answering (VQA) is a surrogate task designed to assess a system's ability to thoroughly understand images. It has gained popularity in recent years due to the release of several benchmark datasets [4, 35, 58]. Motivated by the well-studied difficulty of analyzing results on real-world VQA datasets [22, 41, 57], Johnson *et al*. [23] recently proposed a more controlled, synthetic VQA dataset that we adopt in this work.

Current VQA approaches follow a traditional supervised learning paradigm. A large number of image-question-answer triples are collected and a subset of this data is randomly selected for training. Learning-by-asking (LBA) uses an alternative and more challenging setting: training images are drawn from a distribution, but the learner decides what question it needs to ask to learn the most. The learner receives only answer level supervision from these interactions. It must learn to formulate questions as well as model its own knowledge to remove redundancy in question-asking. LBA also has the potential to generalize to open-world scenarios.

There is also significant progress on building models for VQA using LSTMs with convolutional networks [19, 31], stacked attention networks [55], module networks [3, 21, 24], relational networks [46], and others [40]. LBA is independent of the backbone VQA model and can be used with any existing architecture.

Visual question generation (VQG) was recently proposed as an alternative to image captioning [34, 38, 42]. Our work is related to VQG in the sense that we require the learner to generate questions about images, however, our objective in doing so is different. Whereas VQG focuses on asking questions that are relevant to the image content, LBA requires the learner to ask questions that are both relevant and informative to the learner when answered. A positive side effect is that LBA circumvents the difficulty of evaluating the quality of generated questions (which also hampers image captioning [2]), because the question-answering accuracy of our final model directly correlates with the quality of the questions asked. Such evaluation has also been used in recent works in the language community [54, 56].



✗ What size is the purple cube?
✗ What size is the red thing in front of the yellow cylinder?

✗ What color is the shiny sphere?
✗ What is the color of the cube to the right of the brown thing?

Figure 2: Examples of **invalid** questions for images in the CLEVR universe. Even syntactically correct questions can be invalid for a variety of reasons such as referring to absent objects, incorrect object properties, invalid relationships in the scene or being ambiguous, *etc*.

Active learning (AL) involves a collection of unlabeled examples and a learner that selects which samples will be labeled by an oracle [26, 33, 48, 53]. Common selection criteria include entropy [25], boosting the margin for classifiers [1, 12] and expected informativeness [20]. Our setting is different from traditional AL settings in multiple ways. First, unlike AL where an agent selects the image to be labeled, in LBA the agent selects an image and *generates a question*. Second, instead of asking for a single image level label, our setting allows for richer questions about objects, relationships *etc*. for a single image. While [11, 49] did use simple predefined template questions for AL, templates offer limited expressiveness and a rigid query structure. In our approach, questions are generated by a learned language model. Expressive language models, like those used in our work, are likely necessary for generalizing to real-world settings. However, they also introduce a new challenge: there are many ways to generate invalid questions, which the learner must learn to discard (see Figure 2).

Exploratory learning centers on settings in which an agent explores the environment to acquire supervision [37, 50]; it has been studied in the context of, among others, computer games and navigation [28, 39], multi-user games [36], inverse kinematics [5], and motion planning for humanoids [14]. Exploratory learning problems are generally framed with reinforcement learning in which the agent receives (delayed) rewards, which are used to learn a policy that maximizes the expected rewards. A key difference in the LBA setting is that it does *not* have sparse delayed rewards. Contextual multi-armed bandits [9, 30, 32] are another class of reinforcement learning algorithms that more closely resemble LBA. However, unlike bandits, online performance is irrelevant in LBA: our aim is not to minimize regret, but to minimize the error of the final VQA model.

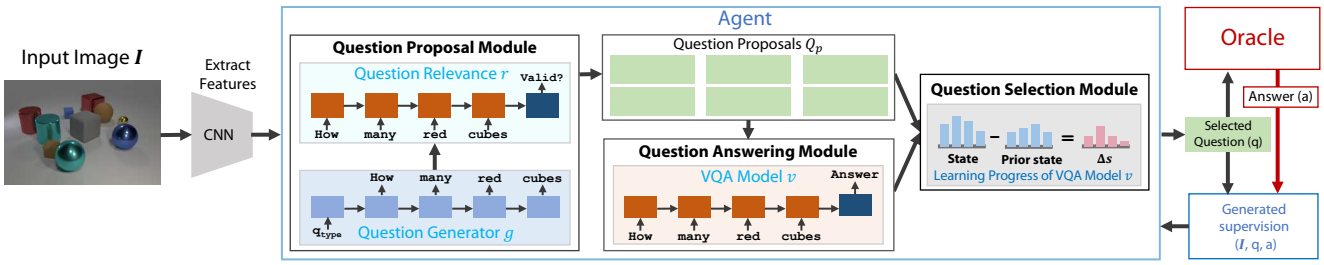


Figure 3: **Our approach to the learning-by-asking setting for VQA.** Given an image $\mathbf{I}$, the agent generates a diverse set of questions using a question generator g. It then filters out "irrelevant" questions using a relevance model r to produce a list of question proposals. The agent then answers its own questions using the VQA model v. With these predicted answers and its self-knowledge of past performance, it selects one question from the proposals to be answered by the oracle. The oracle provides answer-level supervision from which the agent learns to ask informative questions in subsequent iterations.

3. Learning by Asking

We now formally introduce the learning-by-asking (LBA) setting. We denote an image by $\mathbf{I}$, and assume there exists a set of all possible questions $\mathcal{Q}$ and a set of all possible answers $\mathcal{A}$. At training time, the learner receives as input: (1) a training set of N images, $\mathcal{D}_{\text{train}} = \{\mathbf{I}_1, \ldots, \mathbf{I}_N\}$, sampled from some distribution $p_{\text{train}}(\mathbf{I})$; (2) access to an oracle $o(\mathbf{I}, q)$ that outputs an answer $a \in \mathcal{A}$ given a question $q \in \mathcal{Q}$ about image $\mathbf{I}$; and (3) a small bootstrap set of $(\mathbf{I}, q, a)$ tuples, denoted $\mathcal{B}_{\text{init}}$.

The learner receives a budget of B answers that it can request from the oracle. Using these B oracle consultations, the learner aims to construct a function $v(a|\mathbf{I}, q)$ that predicts a score for answer a to question q about image $\mathbf{I}$. The small bootstrap set is provided for the learner to initialize various model components; as we show in our experiments, training on $\mathcal{B}_{\text{init}}$ alone yields poor results.

The challenge of the LBA setting implies that, at training time, *the learner must decide which question to ask about an image* and the only supervision the oracle provides are the answers. As the number of oracle requests is constrained by a budget B, the learner must ask questions that maximize (in expectation) the learning signal from each image-question pair sent to the oracle.

At test time, we assume a standard VQA setting and evaluate models by their question-answering accuracy. The agent receives as input M pairs of images and questions, $\mathcal{D}_{\text{test}} = \{(\mathbf{I}_{N+1}, q_{N+1}), \ldots, (\mathbf{I}_{N+M}, q_{N+M})\}$, sampled from a distribution $p_{\text{test}}(\mathbf{I}, q)$. The images in the test set are sampled from the same distribution as those in the training set: $\sum_{q \in \mathcal{Q}} p_{\text{test}}(\mathbf{I}, q) = p_{\text{train}}(\mathbf{I})$. The agent's goal is to maximize the proportion of test questions that it answers correctly, that is, to maximize:

$$\frac{1}{M} \sum_{m=1}^{M} \mathbb{I}[\operatorname*{argmax}_a v(a|\mathbf{I}_{N+m}, q_{N+m}) = o(\mathbf{I}_{N+m}, q_{N+m})].$$

We make no assumptions on the marginal distribution over test questions, $p_{\text{test}}(q)$.

4. Approach

We propose an LBA agent built from three modules: (1) a **question proposal module** that generates a set of question proposals for an input image; (2) a **question answering module** (or VQA model) that predicts answers from $(\mathbf{I}, q)$ pairs; and (3) a **question selection module** that looks at both the answering module's state and the proposal module's questions to pick a single question to ask the oracle. After receiving the oracle's answer, the agent creates a tuple $(\mathbf{I}, q, a)$ that is used as the online learning signal for all three modules. Each of the modules is described in a separate subsection below; the interactions between them are illustrated in Figure 3.

For the CLEVR universe, the **oracle** is a program interpreter that uses the ground-truth scene information to produce answers. As this oracle only understands questions in the form of programs (as opposed to natural language), our question proposal and answering modules both represent questions as programs. However, unlike [21, 24], we do *not* exploit prior knowledge of the CLEVR programming language in any of the modules; instead, it is treated as a simple means that is required to communicate with the oracle. See supplementary material for examples of programs and details on the oracle.

When the LBA model asks an invalid question, the oracle returns a special answer indicating (1) that the question was invalid and (2) whether or not all the objects that appear in the question are present in the image.

4.1. Question Proposal Module

The question proposal module aims to generate a diverse set of questions (programs) that are relevant to a given image. We found that training a single model to meet both these requirements resulted in limited diversity of questions. Thus, we employ two subcomponents: (1) a **question generation model** g that produces questions $q_g \sim g(q|\mathbf{I})$; and (2) a **question relevance model** $r(\mathbf{I}, q_g)$ that predicts whether a generated question q_g is *relevant* to an image $\mathbf{I}$. Figure 2 shows examples of irrelevant questions that need to be filtered by r. The question generation and relevance

models are used repeatedly to produce a set of question proposals, $\mathcal{Q}_p \subseteq \mathcal{Q}$.

Our **question generation model**, $g(q|\mathbf{I})$, is an image-captioning model that uses an LSTM conditioned on image features (first hidden input) to generate a question. To increase the diversity of generated questions, we also condition the LSTM on the "question type" while training [13] (we use the predefined question types or families from CLEVR). Specifically, we first sample a question type $\mathrm{q}_{\mathrm{type}}$ uniformly at random and then sample a question from the LSTM using a beam size of 1 and a sampling temperature of 1.3. For each image, we filter out all the questions that have been previously answered by the oracle.

Our **question relevance model**, $r(\mathbf{I}, q)$, takes the questions from the generator g as input and filters out irrelevant questions to construct a set of question proposals, $\mathcal{Q}_p$. The special answer provided by the oracle whenever an invalid question is asked (as described above) serves as the online learning signal for the relevance model. Specifically, the model is trained to predict (1) whether or not an image-question pair is valid and (2) whether or not all objects that are mentioned in the question are all present in the image. Questions for which both predictions are positive (*i.e.*, that are deemed by the relevance model to be valid and to contain only objects that appear in the image) are put in the question proposal set, $\mathcal{Q}_p$. We sample from the generator until we have 50 question proposals per image that are predicted to be valid by $r(\mathbf{I}, q)$.

4.2. Question Answering Module (VQA Model)

Our question answering module is a standard VQA model, $v(a|\mathbf{I}, q)$, that learns to predict the answer a given an image-question pair $(\mathbf{I}, q)$. The answering module is trained online using the supervision signal from the oracle.

A key requirement for selecting good questions to ask the oracle is the VQA model's capability to self-evaluate its current state. We capture the state of the VQA model at LBA round t by keeping track of the model's question-answering accuracy $s_t(a)$ per answer a on the training data obtained so far. The state captures information on *what the answering module already knows*; it is used by the question selection module.

4.3. Question Selection Module

The question selection module defines a policy, $\pi(\mathcal{Q}_p; \mathbf{I}, s_{1,\ldots,t})$, that selects the most informative question to ask the oracle from the set of question proposals $\mathcal{Q}_p$. To select an informative question, the question selection module uses the current state of the answering module (how well it is learning various concepts) and the difficulty of each of the question proposals. These quantities are obtained from the state $s_t(a)$ and the beliefs of the current VQA model, $v(a|\mathbf{I}, q)$ for an image-question pair, respectively.

The state $s_t(a)$ contains information about the current knowledge of the answering module. The difference in the state values at the current round, t, and a past round, $t-\Delta$, measures how fast the answering module is improving for each answer. Inspired by curriculum learning [5, 6, 29, 45], we use this difference to select questions on which the answering module can improve the fastest. Specifically, we compute the expected accuracy improvement under the answer distribution for each question $q_p \in \mathcal{Q}_p$:

$$h(q_p; \mathbf{I}, s_{1,\ldots,t}) = \sum_{a \in \mathcal{A}} v(a|\mathbf{I}, q_p) \left(\frac{s_t(a) - s_{t-\Delta}(a)}{s_t(a)} \right). \tag{1}$$

We use the expected accuracy improvement as an informativeness value that the learner uses to pick a question that helps it improve rapidly (thereby enforcing a curriculum). In particular, our selection policy, $\pi(\mathcal{Q}_p; \mathbf{I}, s_{1,\ldots,t})$, uses the informativeness scores to select the question to ask the oracle using an epsilon-greedy policy [51]. The greedy part of the selection policy is implemented via $\mathrm{argmax}_{q_p \in \mathcal{Q}_p} h(q_p; \mathbf{I}, s_{1,\ldots,t})$, and we set $\epsilon = 0.1$ to encourage exploration. Empirically, we find that our policy automatically discovers an easy-to-hard curriculum (see Figures 6 and 8). In all experiments, we set $\Delta = 20$; whenever $t < \Delta$, we set $s_{t-\Delta}(a) = 0$.

4.4. Training Phases

Our model is trained in three phases: (1) an initialization phase in which the generation, relevance, and VQA models (g, r and v) are pre-trained on a small bootstrap set, $\mathcal{B}_{\mathrm{init}}$, of $(\mathbf{I}, q, a)$ tuples; (2) an online learning-by-asking (LBA) phase in which the model learns by interactively asking questions and updates r and v; and (3) an offline phase in which a new VQA model v_{offline} is trained from scratch on the union of the bootstrap set and all of the $(\mathbf{I}, q, a)$ tuples obtained by querying the oracle in the online LBA phase.

Online LBA training phase. At each step in the LBA phase (see Figure 3), the proposal module picks an image $\mathbf{I}$ from the training set $\mathcal{D}_{\mathrm{train}}$ uniformly at random.[1] It then generates a set of relevant question proposals, $\mathcal{Q}_p$ for the image. The answering module tries to answer each question proposal. The selection module uses the state of the answering module along with the answer distributions obtained from evaluating the answering module to pick an informative question, q, from the question proposal set. This question is asked to the oracle o, which provides just the answer $a = o(\mathbf{I}, q)$ to generate a training example $(\mathbf{I}, q, a)$. This training example is used to perform a single gradient step on the parameters of the answering module v and the relevance model r. The language generation model g remains fixed because the oracle does not provide a direct learning signal for it. This process is repeated until the training budget of B oracle answer requests is exhausted.

Offline VQA training phase. We evaluate the quality of

[1] A more sophisticated image selection policy may accelerate learning. We did not explore this in our study.

the asked questions by training a VQA model v_{offline} from scratch on the union of the bootstrap set, $\mathcal{B}_{\text{init}}$, and the $(\mathbf{I}, q, a)$ tuples generated in the LBA phase. We find that offline training of the VQA model leads to slightly improved question-answering accuracy and reduces variance.

4.5. Implementation Details

The LSTM in g has 512 hidden units. After a linear projection, the image features are fed as its first hidden state. We input a discrete variable representing the question type as the first token into the LSTM before starting generation. Following [24], we use a prefix-tree program representation for the questions.

We implement the relevance model, r, and the VQA model, v, using the stacked attention network architecture [55] using the implementation of [24]. The only modification we make is to concatenate the spatial coordinates to the image features before computing attention as in [46]. We do not share weights between r and v.

To generate the invalid pairs $(\mathbf{I}, q)$ for bootstrapping the relevance model, we permute the pairs from the bootstrap set $\mathcal{B}_{\text{init}}$ and assume that all such permuted pairs are invalid. Note that the bootstrap set does not have the special answer indicating whether invalid questions ask about objects not present in the image, and these answers are obtained only in the online LBA phase.

Our models use image features from a ResNet-101 [17] pre-trained on ImageNet [44], in particular, from the `conv4_23` layer of that network. We use ADAM [27] with a fixed learning rate of $5e-4$ to optimize all models. Additional implementation details are presented in the supplementary material.

5. Experiments

Datasets. We evaluate our LBA approach in the CLEVR universe [23], which provides a training set (`train`) with 70k images and 700k $(\mathbf{I}, q, a)$ tuples. We use 70k of these tuples as our bootstrap set, $\mathcal{B}_{\text{init}}$. We evaluate the quality of the data collected by LBA by measuring the question-answering accuracy of the final VQA model, v_{offline}, on the CLEVR validation (`val`) [23] set. As CLEVR `train` and `val` have identical answer and question-type distributions, this gives models trained on CLEVR `train` an inherent advantage. Thus, we also measure question-answering accuracy on the CLEVR-Humans [24] dataset, which has a different distribution; see Figure 9.[2]

Models. Unless stated otherwise, we use the stacked attention model as the answering module v and evaluate three different choices for the final offline VQA model v_{offline}:

CNN+LSTM encodes the image using a CNN, the question using an LSTM, and predicts answers using an MLP.

[2]To apply our VQA models to CLEVR-Humans we translate English to CLEVR-programming language using [24]; see supplementary material for details.

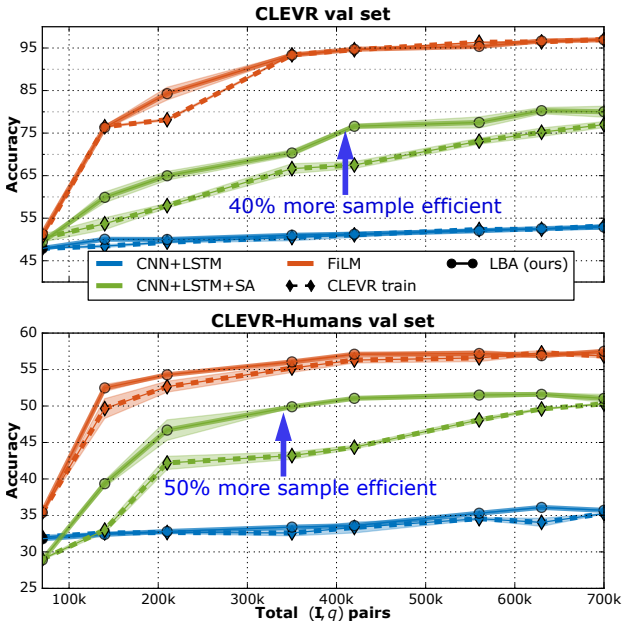


Figure 4: **Top:** CLEVR `val` accuracy for VQA models trained on CLEVR `train` (diamonds) *vs*. LBA-generated data (circles). **Bottom:** Accuracy on CLEVR-Humans for the same set of models. Shaded regions denote one standard deviation in accuracy. On CLEVR-Humans, LBA is 50% more sample efficient than CLEVR `train`.

CNN+LSTM+SA extends CNN+LSTM with the stacked attention (SA) model [55] described in Section 4.2. This is the same as our default answering module v.

FiLM [40] uses question features from a GRU [10] to modulate the image features in each CNN layer.

Unless stated otherwise, we use CNN+LSTM+SA models in all ablation analysis experiments, even though it has lower VQA performance than FiLM, because it trains much faster (6 hours *vs*. 3 days). For all v_{offline} models, we use the training hyperparameters from their respective papers.

5.1. Quality of LBA-Generated Questions

In Figure 4, we compare the quality of the LBA-generated questions to CLEVR `train` by measuring the question-answering accuracy of VQA models trained on both datasets. The figure shows (top) CLEVR `val` accuracy and (bottom) CLEVR-Humans accuracy. From these plots, we draw four observations.

(1) Using the bootstrap set alone (leftmost point) yields poor accuracy and LBA provides a significant learning signal.

(2) The quality of the LBA-generated training data is at least as good as that of the CLEVR `train`. This is an impressive result given that CLEVR `train` has the dual advantage of matching the distribution of CLEVR `val` and being human curated for training VQA models. Despite these ad-

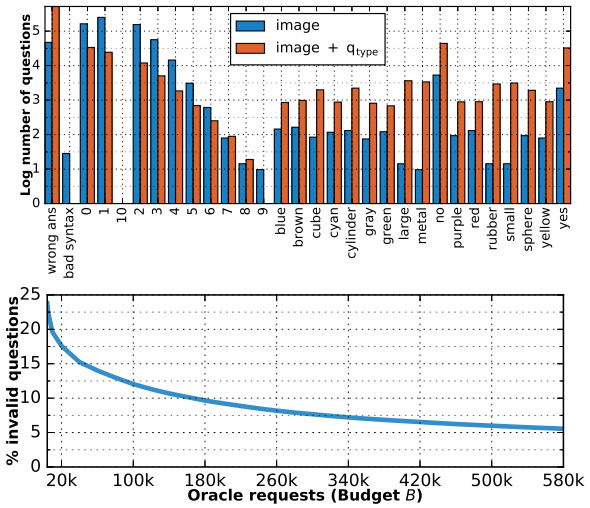


Figure 5: **Top:** Histogram of answers to questions generated by g with and without question-type conditioning. **Bottom:** Percentage of invalid questions sent to the oracle.

Generator g	Relevance r	Budget B 0k	70k	210k	350k	560k	630k
$\mathbf{I}$	None	49.4	43.2	45.4	49.8	52.9	54.7
$\mathbf{I} + \mathrm{q}_{\mathrm{type}}$	None	49.4	46.3	49.5	58.7	60.5	63.4
$\mathbf{I} + \mathrm{q}_{\mathrm{type}}, \tau = 0.3$	Ours	49.4	60.6	67.4	70.2	70.8	70.1
$\mathbf{I} + \mathrm{q}_{\mathrm{type}}, \tau = 0.7$	Ours	49.4	60.2	70.5	76.7	77.5	77.6
$\mathbf{I} + \mathrm{q}_{\mathrm{type}}, \tau = 1.3$	Ours	49.4	60.3	71.4	76.9	79.8	78.2
$\mathbf{I} + \mathrm{q}_{\mathrm{type}}$	Perfect	49.4	67.7	75.7	80.0	81.2	81.1

Table 1: CLEVR `val` accuracy for six budgets B. We condition the generator on the image ($\mathbf{I}$) or on the image and the question type ($\mathbf{I}$ + $\mathrm{q}_{\mathrm{type}}$), vary the generator sampling temperatures τ, and use three different relevance models. We re-run the LBA pipeline for each of these settings.

v_{offline} Model	Budget B 0k	70k	210k	350k	560k	630k
CNN+LSTM	47.1	48.0	49.2	49.1	52.3	52.7
CNN+LSTM+SA	49.4	63.9	68.1	76.1	78.4	82.3
FiLM	51.2	76.2	92.9	94.8	95.2	97.3

Table 2: CLEVR `val` accuracy for three v_{offline} models when FiLM is used as the online answering module v.

vantages, LBA matches and sometimes surpasses its performance. More importantly, LBA shows better generalization on CLEVR-Humans which has a different answer distribution (see Figure 9).

(3) LBA data is sometimes more sample efficient than CLEVR `train`: for instance, on both CLEVR `val` and CLEVR-Humans. The CNN+LSTM+SA model only requires 60% of $(\mathbf{I}, q, a)$ LBA tuples to achieve the accuracy of the same model trained on all of CLEVR `train`.

(4) Finally, we also observe that our LBA agents have low variance at each sampled point during training. The shaded error bars show one standard deviation computed from 5 independent runs using different random seeds. This is an important property for drawing meaningful conclusions from interactive training environments (*c.f.*, [18]).

Qualitative results. Figure 6 shows five samples from the LBA-generated data at various iterations t. They provide insight into the curriculum discovered by our LBA agent. Initially, the model asks simple questions about colors (row 1) and shapes (row 2). It also makes basic mistakes (rightmost column of rows 1 and 2). As the answering module v improves, the selection policy π asks more complex questions about spatial relationships and counts (rows 3 and 4).

5.2. Analysis: Question Proposal Module

Analyzing the generator g. We evaluate the diversity of the generated questions by looking at the distribution of corresponding answers. In Figure 5 (top) we use the final LBA model to generate 10 questions for each image in the training set. We plot the histogram of the answers to these questions for generators with and without "question type" conditioning. The histogram shows that conditioning the generator g on question type leads to better coverage of the answer space. We also note that about 4% of the generated questions have invalid programming language syntax.

We observe in the top two rows of Table 1 that the increased question diversity translates into improved question-answering accuracy. Diversity is also controlled by the sampling temperature, τ, used in g. Rows 3-5 show that a lower temperature, which gives less diverse question proposals, negatively impacts final accuracy.

Analyzing the relevance model r. Figure 5 (bottom) displays the percentage of invalid questions sent to the oracle at different time steps during online LBA training. The invalid question rate decreases during training from 25% to 5%, even though question complexity appears to be increasing (Figure 6). This result indicates that the relevance model r improves significantly during training.

We can also decouple the effect of the relevance model r from the rest of our setup by replacing it with a "perfect" relevance model (the oracle) that flawlessly filters all invalid questions. Table 1 (row 6) shows that the accuracy and sample efficiency differences between the "perfect" relevance model and our relevance model are small, which suggests our model performs well.

5.3. Analysis: Question Answering Module

Thus far we have tested our policy π with only one type of answering module v, CNN+LSTM+SA. Now, we verify that π works with other choices by implementing v as the FiLM model and rerunning LBA. As in Section 5.1, we evaluate the LBA-generated questions by training the three v_{offline} models. The results in Table 2 suggest that our selection policy generalizes to a new choice of v.

5.4. Analysis: Question Selection Module

To investigate the role of the selection policy in LBA, we compare four alternatives: (1) random selection from the

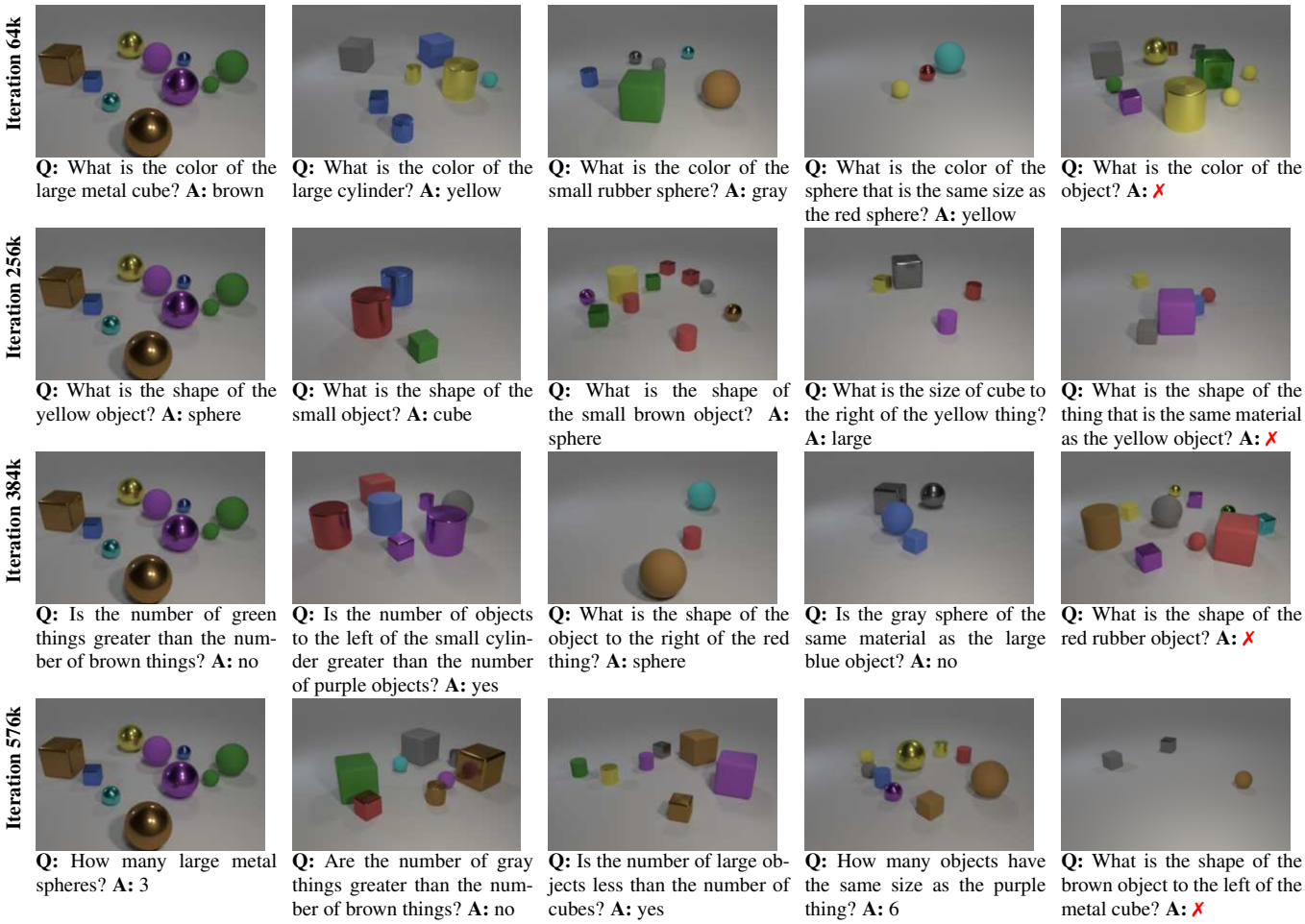


Figure 6: Example questions asked by our LBA agent at different iterations (manually translated from programs to English). Our agent asks increasingly sophisticated questions as training progresses — starting with simple color questions and moving on to shape and count questions. We also see that the invalid questions (right column) become increasingly complex.

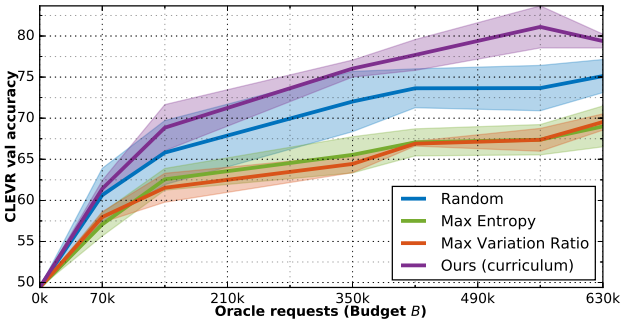


Figure 7: Accuracy of CNN+LSTM+SA trained using LBA with four different policies for selecting question proposals (Sec 4.3). Our selection policy is more sample efficient.

question proposals; (2) using the prediction entropy of the answering module v for each proposal after four forward passes with dropout (like in [47]); (3) using the variation ratio [15] of the prediction; and (4) our curriculum policy from Section 4.3. We run LBA training with five different random seeds and report the mean accuracy and stdev of a CNN+LSTM+SA model for each selection policy in Figure 7. In line with results from prior work [47], the entropy-based policies perform worse than random selection. By contrast, our curriculum policy substantially outperforms random selection of questions. Figure 8 plots the normalized informativeness score h (Equation 1) and the training question-answering accuracy ($s(a)$ grouped by per answer type). These plots provide insight into the behavior of the curriculum selection policy, π. Specifically, we observe a delayed pattern: a peak in the the informativeness score (blue arrow) for an answer type is followed by an uptick in the accuracy (blue arrow) on that answer type. We also observe that the policy's informativeness score suggests an easy-to-hard ordering of questions: initially (after 64k requests), the selection policy prefers asking the easier `color` questions, but it gradually moves on to `size` and `shape` questions and, eventually, to the difficult `count`

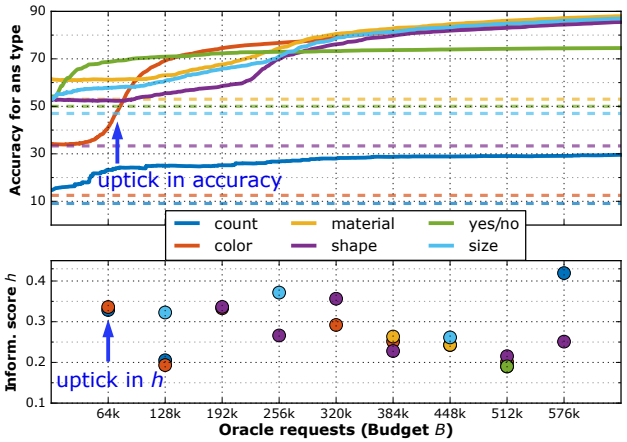


Figure 8: **Top:** Accuracy during training (solid lines) and chance level (dashed lines) per answer type. **Bottom:** Normalized informative scores per answer type, averaged over 10k questions. See Section 5.4 for details.

questions. We emphasize that this easy-to-hard curriculum is learned automatically without any extra supervision.

5.5. Varying the Size of the Bootstrap Data

We vary the size of the bootstrap set $\mathcal{B}_{\text{init}}$ used for initializing the g, r, v models and analyze its effect on the LBA generated data. In Table 3 we show the accuracy of the final v_{offline} model on CLEVR `val`. A smaller bootstrap set results in reduced performance. We also see that with less than 5% (rows 1 and 2) of the CLEVR training dataset as our bootstrap set, LBA asks questions that can match the performance using the entire CLEVR training set. Empirically, we observed that the generator g performs well on smaller bootstrap sets. However, the relevance model r needs enough valid and invalid (permuted) $(\mathbf{I}, q, a)$ tuples in the bootstrap set to filter irrelevant question proposals. As a result, a smaller bootstrap set affects the sample efficiency of LBA.

$\lvert\mathcal{B}_{\text{init}}\rvert$	Budget B						
	0k	70k	140k	210k	350k	560k	630k
20k	48.2	56.4	63.5	66.9	72.6	75.8	76.2
35k	48.8	58.6	64.3	68.7	74.9	76.1	76.3
70k	49.4	61.1	67.6	72.8	78.0	78.2	79.1

Table 3: Accuracy on CLEVR validation data at different budgets B as a function of the bootstrap set size, $\lvert\mathcal{B}_{\text{init}}\rvert$.

6. Discussion and Future Work

This paper introduces the learning-by-asking (LBA) paradigm and proposes a model in this setting. LBA moves away from traditional *passively* supervised settings where human annotators provide the training data in an *interactive* setting where the learner seeks out the supervision it needs. While passive supervision has driven progress in visual recognition [16, 17], it does not appear well suited for general AI tasks such as visual question answering (VQA). Curating large amounts of diverse data which generalizes to a wide variety of questions is a difficult task. Our results suggest that interactive settings such as LBA may facilitate learning with higher sample efficiency. Such high sample efficiency is crucial as we move to increasingly complex visual understanding tasks.

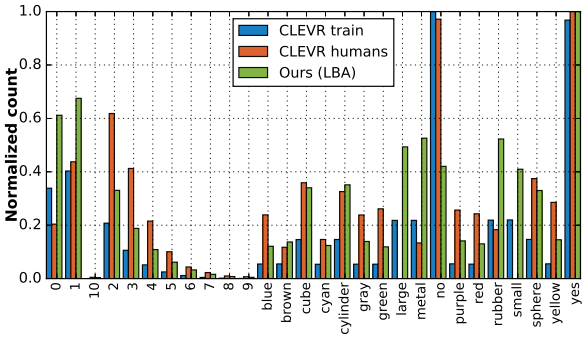


Figure 9: Answer distribution of CLEVR `train`, LBA-generated data, and the CLEVR-Humans dataset.

An important property of LBA is that it does not tie the distribution of questions and answers seen at training time to the distribution at test time. This more closely resembles the real-world deployment of VQA systems where the distribution of user-posed questions to the system is unknown and difficult to characterize beforehand [8]. The CLEVR-Humans distribution in Figure 9 is an example of this. This issue poses clear directions for future work [7]: we need to develop VQA models that are less sensitive to distributional variations at test time; and not evaluate them under a single test distribution (as in current VQA benchmarks).

A second major direction for future work is to develop a "real-world" version of a LBA system in which (1) CLEVR images are replaced by natural images and (2) the oracle is replaced by a human annotator. Relative to our current approach, several innovations are required to achieve this goal. Most importantly, it requires the design of an effective mode of communication between the learner and the human "oracle". In our current approach, the learner uses a simple programming language to query the oracle. A real-world LBA system needs to communicate with humans using diverse natural language. The efficiency of LBA learners may be further improved by letting the oracle return privileged information that does not just answer an image-question pair, but that also explains *why* this is the right or wrong answer [52]. We leave the structural design of this privileged information to future work.

Acknowledgments: The authors would like to thank Arthur Szlam, Jason Weston, Saloni Potdar and Abhinav Shrivastava for helpful discussions and feedback on the manuscript; Soumith Chintala and Adam Paszke for their help with PyTorch.

References

[1] Y. Abramson and Y. Freund. Active learning for visual object recognition. *Technical report, UCSD*, 2004. 2

[2] P. Anderson, B. Fernando, M. Johnson, and S. Gould. SPICE: Semantic propositional image caption evaluation. In *ECCV*, 2016. 2

[3] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Learning to compose neural networks for question answering. In *NAACL*, 2016. 2

[4] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. VQA: Visual question answering. In *CVPR*, 2015. 1, 2

[5] A. Baranes and P.-Y. Oudeyer. Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems*, 2013. 2, 4

[6] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *ICML*. ACM, 2009. 2, 4

[7] L. Bottou. Two Big Challenges in Machine Learning. http://leon.bottou.org/talks/2challenges. Accessed: Nov 15, 2017. 8

[8] L. Bottou, J. Peters, J. Quiñonero-Candela, D. X. Charles, D. M. Chickering, E. Portugaly, D. Ray, P. Simard, and E. Snelson. Counterfactual reasoning and learning systems: The example of computational advertising. *JMLR*, 2013. 8

[9] S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 2012. 2

[10] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014. 5

[11] J. Choi, S. J. Hwang, L. Sigal, and L. S. Davis. Knowledge transfer with interactive learning of semantic relationships. In *AAAI*, pages 1505–1511, 2016. 2

[12] B. Collins, J. Deng, K. Li, and L. Fei-Fei. Towards scalable dataset construction: An active learning approach. *ECCV*, 2008. 2

[13] F. Ferraro, N. Mostafazadeh, I. Misra, A. Agrawal, J. Devlin, R. Girshick, X. He, P. Kohli, D. Batra, C. L. Zitnick, et al. Visual storytelling. In *NAACL*, 2016. 4

[14] M. Frank, J. Leitner, M. Stollenga, A. Förster, and J. Schmidhuber. Curiosity driven reinforcement learning for motion planning on humanoids. *Frontiers in neurorobotics*, 2014. 2

[15] L. C. Freeman. *Elementary applied statistics: for students in behavioral science*. John Wiley & Sons, 1965. 7

[16] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *ICCV*, 2017. 8

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5, 8

[18] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger. Deep reinforcement learning that matters. *arXiv:1709.06560*, 2017. 6

[19] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 1997. 2

[20] N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel. Bayesian active learning for classification and preference learning. *arXiv:1112.5745*, 2011. 2

[21] R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko. Learning to reason: End-to-end module networks for visual question answering. *ICCV*, 2017. 2, 3

[22] A. Jabri, A. Joulin, and L. van der Maaten. Revisiting visual question answering baselines. In *ECCV*. Springer, 2016. 2

[23] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. *CVPR*, 2016. 1, 2, 5

[24] J. Johnson, B. Hariharan, L. van der Maaten, J. Hoffman, L. Fei-Fei, C. L. Zitnick, and R. Girshick. Inferring and executing programs for visual reasoning. *ICCV*, 2017. 2, 3, 5

[25] A. J. Joshi, F. Porikli, and N. Papanikolopoulos. Multi-class active learning for image classification. In *CVPR*, 2009. 2

[26] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Active learning with gaussian processes for object categorization. In *ICCV*, 2007. 2

[27] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[28] T. D. Kulkarni, K. Narasimhan, A. Saeedi, and J. Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *NIPS*, 2016. 2

[29] M. P. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. In *NIPS*, 2010. 4

[30] J. Langford and T. Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *NIPS*, 2008. 2

[31] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1, 1989. 2

[32] L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *WWW*, 2010. 2

[33] X. Li and Y. Guo. Adaptive active learning for image classification. In *CVPR*, 2013. 2

[34] F. Liu, T. Xiang, T. M. Hospedales, W. Yang, and C. Sun. iVQA: Inverse visual question answering. *arXiv:1710.03370*, 2017. 2

[35] M. Malinowski and M. Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *NIPS*, 2014. 2

[36] K. E. Merrick and M. L. Maher. *Motivated reinforcement learning: curious characters for multiuser games*. Springer Science & Business Media, 2009. 2

[37] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell. *Machine learning: An artificial intelligence approach*. Springer Science & Business Media, 2013. 2

[38] N. Mostafazadeh, I. Misra, J. Devlin, M. Mitchell, X. He, and L. Vanderwende. Generating natural questions about an image. In *ACL*, 2016. 2

[39] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction. In *ICML*, 2017. 2

[40] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville. Film: Visual reasoning with a general conditioning layer. *arXiv:1709.07871*, 2017. 2, 5

[41] A. Ray, G. Christie, M. Bansal, D. Batra, and D. Parikh. Question relevance in vqa: identifying non-visual and false-premise questions. *arXiv:1606.06622*, 2016. 2

[42] A. Rothe, B. M. Lake, and T. Gureckis. Question asking as program generation. In *Advances in Neural Information Processing Systems*, pages 1046–1055, 2017. 2

[43] A. Rothe, B. M. Lake, and T. Gureckis. Do people ask good questions? 2018. 2

[44] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115, 2015. 5

[45] M. Sachan and E. P. Xing. Easy questions first? a case study on curriculum learning for question answering. In *ACL*, 2016. 4

[46] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. *arXiv:1706.01427*, 2017. 2, 5

[47] O. Sener and S. Savarese. A geometric approach to active learning for convolutional neural networks. *arXiv:1708.00489*, 2017. 7

[48] B. Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010. 2

[49] B. Siddiquie and A. Gupta. Beyond active noun tagging: Modeling contextual interactions for multi-class active learning. In *CVPR*, 2010. 2

[50] J. Storck, S. Hochreiter, and J. Schmidhuber. Reinforcement driven information acquisition in non-deterministic environments. In *Proceedings of the international conference on artificial neural networks, Paris*, 1995. 2

[51] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998. 2, 4

[52] V. Vapnik and R. Izmailov. Learning using privileged information: similarity control and knowledge transfer. *JMLR*, 2015. 8

[53] S. Vijayanarasimhan and K. Grauman. Large-scale live active learning: Training object detectors with crawled data and crowds. *IJCV*, 2014. 2

[54] T. Wang, X. Yuan, and A. Trischler. A joint model for question answering and question generation. *arXiv preprint arXiv:1706.01450*, 2017. 2

[55] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In *CVPR*, 2016. 2, 5

[56] Z. Yang, J. Hu, R. Salakhutdinov, and W. W. Cohen. Semi-supervised qa with generative domain-adaptive nets. *arXiv preprint arXiv:1702.02206*, 2017. 2

[57] P. Zhang, Y. Goyal, D. Summers-Stay, D. Batra, and D. Parikh. Yin and yang: Balancing and answering binary visual questions. In *CVPR*, 2016. 2

[58] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei. Visual7w: Grounded question answering in images. In *CVPR*, 2016. 1, 2

通过提问来学习

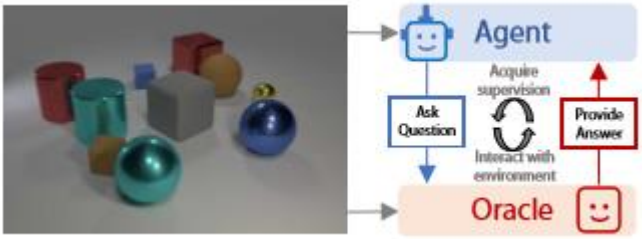
作者来自卡内基梅隆大学、Facebook

摘要

本文为智能视觉系统的开发和测试引入了一个交互式学习框架，称为 learnig-by-asking（LBA）。 我们在 Visual Question Answering（VQA）任务的上下文中探索 LBA。 LBA 与标准 VQA 训练的不同之处在于，在训练期间没有观察到大多数问题，学习者必须提出它想要回答的问题。因此，LBA 更接近于模仿自然学习，并且有可能比传统的 VQA 设置更加有效。本文提出了一个在 CLEVR 数据集上执行 LBA 的模型，并发现它在从 oracle 交互式学习时会自动发现一个易于理解的类别。 我们的 LBA 生成的数据始终匹配或优于 CLEVR 训练数据，并且更具样本效率。 本文还发现该模型提出的问题可以推广到最先进的 VQA 模型和新的测试时间分布。

1、介绍

机器学习模型在视觉识别方面取得了显著的进步。然而，虽然喂入这些模型的训练数据至关重要，但通常将其视为预定的静态信息。我们目前的模型本质上是被动的：它们依赖于由人类策划的训练数据，并且无法控制这种监督。这与我们人类学习的方式形成鲜明对比 - 通过与我们的环境互动来获取信息。 人类学习的互动性使其具有高效的样本（训练期间冗余更少），并产生学习类别（我们学习时需要更复杂的知识）。



图一：LBA 范例。 我们提出了一个开放世界的视觉问题回答（VQA）设置，其中代理通过向 oracle 提问来交互式学习。 与标准的 VQA 培训（假定固定的问题数据集）不同，在 LBA 中，代理人有可能通过询问“好”问题来更快地学习，就像课堂上聪明的学生一样。 LBA 不会改变 VQA 的测试时间设置。

在这篇论文中，本文认为下一代的识别系统需要有代理——能够决定他们需要什么信息以及如何获取信息。本文在视觉问题回答的背景下探讨这个问题（VQA; [4,23,58]）。本文没有在固定的大规模数据集上进行训练，而是提出了一种交互式 VQA 设置，称为 learning-by-ask (LBA)。在训练时，学习者只接收图像并决定要问什么问题。学习者提出的问题由 oracle（人工监督）回答。 在测试时，使用易于理解的指标对 LBA 进行与 VQA 完全相同的评估。

LBA 的互动性要求学习者构建关于其知识的元知识，并选择所需的监督。 如果成功，这有助于比使用固定数据集更多的样本有效学习，因为学习者不会提出多余的问题。

本文在 CLEVR 数据集[23]的背景下探索了提出的 LBA 范式，这是一个人造空间，在其中唯一对象，属性和关系的数量是有限的。 我们选择这种合成设置是因为之前几乎没有提出有关图像问题的工作：CLEVR 允许我们对提出问题所需的算法进行对照研究。 我们希望将从我们的研究中获得的见解转移到现实世界。

建立一个可以提问的互动学习者是一项具有挑战性的任务。 首先，学习者需要有一种“语言”模型来形成问题。 其次，它需要了解输入图像，以确保问题的相关性和连贯性。 最后（也是最重要的），为了获得样本效率，学习者应该能够评估自己的知识（自我评估）并提出有助于学习有关世界的新信息的问题。 学习者从交互中获得的唯一监督是对其提出的

问题的回答。 有趣的是，最近的工作[43]表明即使是人类也不擅长提出信息丰富的问题。

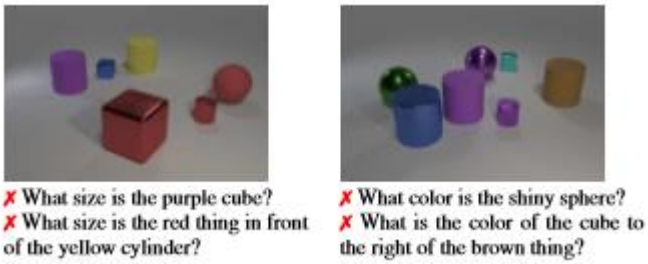
我们提出并研究了 LBA 的模型，该模型结合了视觉基础语言生成的创意[38]，课程学习[6]和 VQA 的思想。 具体而言，我们开发了一个 epsilon-greedy [51]学习者，它会询问问题并使用相应的答案来训练标准的 VQA 模型。 在转向新类型的问题之前，学习者专注于掌握可以快速改进的概念。 我们证明了我们的 LBA 模型不仅提出了有意义的问题，而且还匹配了人工策划数据的性能。 我们的模型也是样本有效的，并通过交互式提问，它将获得基线问答精度所需的训练样本数量减少了 40％。

2、相关工作

视觉问答（VQA）是一项代理任务，旨在评估系统彻底理解图像的能力。 由于几个基准数据集的发布，它近年来越来越受欢迎[4,35,58]。 受到充分研究的分析现实世界 VQA 数据集[22,41,57]结果的困难的启发，Johnson 等人。 [23]最近提出了一个我们在这项工作中采用的更受控制的合成 VQA 数据集。

当前的 VQA 方法遵循传统的监督学习范式。 收集大量（图片、问题、答案）三元组，随机选择该数据的子集进行训练。 逐个学习（LBA）使用另一种更具挑战性的设置：培训图像是从分布中提取的，但学习者决定了要求学习最多的问题。 学习者仅从这些交互中接收答案级别监督。 它必须学会制定问题并建立自己的知识模型，以消除提问中的冗余。 LBA 还有可能推广到开放世界的场景。

使用具有卷积网络的 LSTM [19,31]，堆叠注意网络[55]，模块网络[3,21,24]，关系网络[46]以及其他[40]，在构建 VQA 模型方面也取得了重大进展。 LBA 独立于骨干 VQA 模型，可以与任何现有架构一起使用。



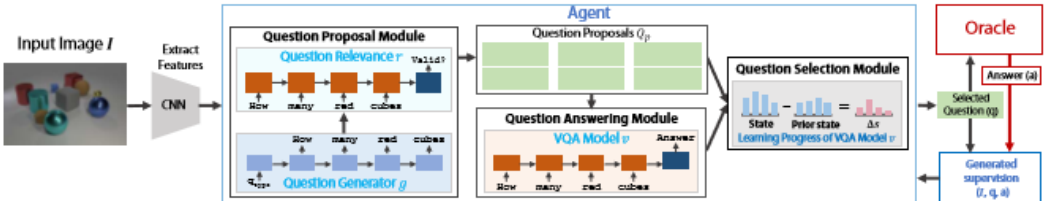
图二：CLEVR Universe 中图像的无效问题示例。 即使语法上正确的问题也可能因各种原因而无效，例如引用缺席对象，不正确的对象属性，场景中的无效关系或模糊等。

最近提出视觉问题生成（VQG）作为图像字幕的替代[34,38,42]。 我们的工作与 VQG 有关，因为我们要求学习者产生关于图像的问题，但是，我们这样做的目的是不同的。 尽管 VQG 专注于提出与图像内容相关的问题，但 LBA 要求学习者在回答时向学习者提出既相关又有信息的问题。 一个积极的副作用是 LBA 避免了评估生成问题质量的困难（这也妨碍了图像标题[2]），因为我们最终模型的问答精确度直接与所提问题的质量相关。 这种评估也被用于语言社区的近期工作[54,56]。

主动学习（AL）涉及一系列未标记的例子和一个学习者，它选择哪些样本将被 oracle 标记[26,33,48,53]。常见的选择标准包括熵[25]，提高分类的余量[1,12]和预期的信息量[20]。我们的设置与传统的 AL 设置有多种不同。首先，与代理选择要标记的图像的 AL 不同，在 LBA 中，代理选择图像并生成问题。其次，我们的设置不是要求单个图像级别标签，而是为单个图像提供有关对象，关系等的更丰富的问题。虽然[11,49]确实使用简单的预定义模板问题用于 AL，但模板提供有限的表达能力和严格的查询结构。在我们的方法中，问题是由学习的语言模型生成的。表达式语言模型，如我们工作中使用的那些，可能是推广到真实世界设置所必需的。然而，他们也引入了一个新的挑战：有很多方法可以产生无效的问题，学习者必

须学会放弃这些问题（见图 2）。

探索性学习集中在一个代理人探索环境以获得监督的环境中[37,50]；它已经在计算机游戏和导航[28,39]，多用户游戏[36]，反向运动学[5]和人形机器人运动规划[14]的背景下进行了研究。 探究性学习问题通常以强化学习为框架，其中代理人接收（延迟）奖励，用于学习最大化预期奖励的政策。 LBA 设置的一个关键区别是它没有稀疏的延迟奖励。 Contextual multi-armed bandits [9,30,32]是另一类强化学习算法，更类似于 LBA。 然而，与 bandits 不同，在线表现与 LBA 无关：我们的目标不是尽量减少遗憾，而是尽量减少最终 VQA 模型的错误。



图三：针对 VQA 的通过询问学习的设置方法。喂入一张图片 I，代理使用问题生成器 g 生成一组不同的问题。然后，它使用相关性模型 r 来过滤掉“不相关的”问题，以生成问题提议列表。 然后，代理使用 VQA 模型回答自己的问题。利用这些预测的答案及其对过去表现的自我知识，它从提议中选择一个问题，由 oracle 回答。 oracle 提供答案级别的监督，代理从中学习在后续迭代中询问信息性问题。

3、通过提问学习

我们现在正式介绍 LBA 设置。 我们用 I 表示图像，并假设存在一组所有可能的问题 Q 和一组所有可能的答案 A.在训练时，学习者接收输入：（1）N 个图像的训练集，$\mathrm{D}_{\text{train}}$ = { I_1，…，I_N }，从某些分布 $P_{train}(I)$ 中采样；（2）访问 oracle $o(I,q)$，其输出答案 a∈A 给出关于图像 I 的问题 q∈Q；（3）一个小的引导程序集 (I,q,a) 元组，表示为 B_{init} 。

学习者收到可以从 oracle 请求的 B 个答案的预算。 使用这些 B oracle 咨询，学习者的目的是构建一个函数 $v(\alpha \mid I,q)$，它预测答案 a 的分数，对问题 q 关于图像 I。提供小的自举集，供学习者初始化各种模型组件；正如我们在实验中所展示的那样，单独对 B_{init} 进行培训会产生不良结果。

LBA 设置的挑战意味着，在训练时，学习者必须决定询问关于图像的哪个问题，并且 oracle 提供的唯一监督是答案。 由于 oracle 请求的数量受到预算 B 的约束，因此学习者必须提出最大化来自发送到 oracle 的每个图像--问题对的学习信号的问题。

在测试时，我们假设一个标准的 VQA 设置，并通过他们的问答精度来评估模型。 代理接收 M 对图像和问题作为输入，$D_{test}=\{(I_{N+1},q_{N+1})$，…，$(I_{N+M},q_{N+M})\}$，从分布 $P_{test}(I,q)$ 中采样。 测试集中的图像从与训练集中的图像相同的分布中采样：$\sum_{q\in Q} p_{test}(I,q)=p_{train}(I)$ 代理的目标是最大化它正确回答的测试问题的比例，即最大化：

$$\frac{1}{M}\sum_{m=1}^{M}\mathbb{I}[\operatorname*{argmax}_{a} v(a|\mathbf{I}_{N+m}, q_{N+m}) = o(\mathbf{I}_{N+m}, q_{N+m})].$$

我们对测试问题的边际分布 $p_{test}(q)$ 没有做出任何假设。

4、途径

我们提出了一个由三个模块构建的 LBA 代理：（1）问题提议模块，为输入图像生成一组问题提议；（2）预测来自（I，q）对的答案的问答模块（或 VQA 模型）；（3）一个问题选择模块，它查看应答模块的状态和提议模块的问题，以选择一个问题来询问 oracle。 在收到 oracle 的答案后，代理会创建一个元组 (I, q, a)，用作所有三个模块的在线学习信号。每个模块在下面的单独小节中描述；它们之间的相互作用如图 3 所示。

对于 CLEVR 空间，oracle 是一个程序解释器，它使用地面真实场景信息来产生答案。由于这个 oracle 只能以程序的形式（而不是自然语言）理解问题，我们的问题提议和回答模块都将问题表示为程序。 但是，与[21,24]不同，我们不会在任何模块中利用 CLEVR 编程语言的先验知识；相反，它被视为与 oracle 通信所需的简单方法。 有关程序的示例和有关 oracle 的详细信息，请参阅补充材料。

当 LBA 模型询问无效问题时，oracle 返回一个特殊答案，表示（1）问题无效，（2）问题中出现的所有对象是否都出现在图像中

4.1 问题提案模块

问题提案模块旨在生成与给定图像相关的各种问题（程序）。 我们发现，训练单一模型以满足这两个要求导致问题的多样性有限。 因此，我们使用两个子组件：（1）问题产生模型 g 用来产生问题 $q_g \sim g(q \mid I)$；（2）问题相关性模型 r（I，q_g），其预测生成的问题 q_g 是否与图像 I 相关。图 2 示出了需要由 r 过滤的无关问题的示例。 问题生成和相关性模型被重复使用以产生一组问题提议，$Q_p \subseteq Q$

我们的问题生成模型 $g(q \mid I)$ 是一个图像捕获模型，它使用以图像特征（第一个隐藏输入）为条件的 LSTM 来生成问题。 为了增加生成问题的多样性，我们还在培训[13]时使用“问题类型”来调整 LSTM（我们使用来自 CLEVR 的预定义问题类型或系列）。 具体而言，我们首先在随机时均匀地对问题类型 q_{type} 进行采样，然后使用光束大小为 1 和采样温度为 1.3 从 LSTM 中采样问题。 对于每个图像，我们都会过滤掉 oracle 之前已经回答过的所有问题

我们的问题相关性模型 r（I，q）将来自生成器 g 的问题作为输入并过滤掉不相关的问题来构建一组问题提议 Q_p。 每当提出无效问题时（如上所述），oracle 提供的特殊答案用作相关性模型的在线学习信号。 具体而言，训练该模型以预测（1）图像问题对是否有效以及（2）问题中提到的所有对象是否都存在于图像中。 两个预测都是正面的问题（即，相关性模型认为有效并且仅包含出现在图像中的对象）的问题被放入问题提议集 Q_p 中。 我们从生成器中进行采样，直到每个图像有 50 个问题提案，预计 r（I，q）有效。

4.2 问答模块

我们的问答模块是标准 VQA 模型 $v(a \mid I, q)$，它学习如何预测给定图像--问题对（I，

q）的答案。使用来自 oracle 的监督信号在线训练应答模块。

选择好问题来询问 oracle 的一个关键要求是 VQA 模型能够自我评估其当前状态。 我们通过跟踪到目前为止获得的训练数据的每个答案 a 的模型的问题转移准确度$s_t(a)$来捕获 LBA 回合的 VQA 模型的状态。 状态捕获有关应答模块已知的内容的信息；它由问题选择模块使用。

4.3 问题选择模块

问题选择模块定义一个策略$\pi(Q_p; I, s_1, ..., t)$，它选择最具信息量的问题，从问题提议集Q_p中询问 oracle。为了选择提供信息的问题，问题选择模块使用应答模块的当前状态（学习各种概念的程度）以及每个问题提议的困难。 这些量分别从状态$s_t(a)$和当前 VQA 模型的信念$v(a \mid I, q)$获得图像 - 问题对。

状态$s_t(a)$包含有关应答模块的当前知识的信息。 当前轮次，t 和过去轮次的状态值的差异 t-Δ 测量应答模块对每个答案的改进速度。 受课程学习的启发[5,6,29,45]，我们使用这种差异来选择答案模块可以最快地提高的问题。 具体而言，我们计算每个问题$q_p \in Q_p$的答案分布下的预期准确度改进：

$$h(q_p; \mathbf{I}, s_{1,...,t}) = \sum_{a \in \mathcal{A}} v(a|\mathbf{I}, q_p) \left(\frac{s_t(a) - s_{t-\Delta}(a)}{s_t(a)} \right). \quad (1)$$

我们使用预期的准确度提高作为信息量值，学习者使用该值来选择有助于其快速改进的问题（从而强制执行课程）。 特别是，我们的选择策略$\pi(Q_p; I, s_1, ..., t)$使用信息性分数来选择问题，以便使用 epsilon-greedy 策略来询问 oracle [51]。 选择策略的贪婪部分是通过$\arg\max_{q_p \in Q_p} h(q_p; I, s_1, ..., t)$实现的，我们设置 ϱ= 0.1 以鼓励探索。 根据经验，我们发现我们的政策会自动发现一个易于理解的课程（见图 6 和图 8）。 在所有实验中，我们设定 Δ= 20；每当 t <Δ 时，我们设定s_t -Δ（a）= 0。

4.4 训练阶段

我们的模型分三个阶段进行训练：（1）初始化阶段，其中生成，相关性和 VQA 模型（g，r 和 v）在小引导集 Binit 上预先训练（I，q，a ）元组；（2）在线学习（LBA）阶段，模型通过交互式提问和更新 r 和 v 来学习；（3）一个新的 VQA 阶段，其中新的 VQA 模型从头开始训练自举集和通过在线 LBA 阶段查询 oracle 获得的所有（I，q，a）元组。在线 LBA 培训阶段。在 LBA 阶段的每个步骤（参见图 3），提议模块随机均匀地从训练集D_{train}中选择图像 I.然后，它生成一组相关的问题提议，即图像的Q_p。应答模块尝试回答每个问题提议。选择模块使用应答模块的状态以及从评估应答模块获得的应答分布来从问题提议集中挑选信

息性问题 q。这个问题被问到 oracle o，它提供了答案 a = o（I，q）来生成训练样例（I，q，a）。该训练示例用于对应答模块 v 和相关性模型 r 的参数执行单个梯度步骤。语言生成模型 g 仍然固定，因为 oracle 没有为它提供直接的学习信号。重复该过程直到 B oracle 应答请求的训练预算用尽

离线 VQA 培训阶段。我们通过在引导集，B_{init} 和 LBA 阶段生成的（I，q，a）元组的并集上从头开始训练 VQA 模型来评估所提问题的质量。我们发现 VQA 模型的培训可以略微提高问答准确度并减少差异。

4.5 实现细节

GST 中的 LSTM 具有 512 个隐藏单元。 在线性投影之后，图像特征作为其第一隐藏状态被喂入。 在开始生成之前，我们将表示问题类型的离散变量作为第一个标记输入到 LSTM 中。 在[24]之后，我们为问题使用一个 prefix-tree 程序表达。

我们使用堆叠注意网络架构[55]使用[24]的实现来实现相关性模型 r 和 VQA 模型 v。 我们所做的唯一修改是在计算注意力之前将空间坐标与图像特征连接起来，如[46]中所述。 我们不在 r 和 v 之间共享权重。

为了生成用于引导相关性模型的无效对（I，q），我们对来自引导集 B_{init} 的对进行置换，并假设所有这些置换对都是无效的。 请注意，引导程序集没有特殊答案，表明无效问题是否询问图像中不存在的对象，这些答案仅在在线 LBA 阶段获得

我们的模型使用来自 ImageNet [44]预训练的 ResNet-101 [17]的图像特征，特别是来自该网络的 conv4_23 层的图像特征。 我们使用 ADAM [27]，固定学习率为 5e-4 来优化所有模型。 其他实施细节见补充材料。

5.实验

数据集：我们在 CLEVR 宇宙中评估我们的 LBA 方法[23]，它提供了一个训练集（训练），其中有 70k 图像和 700k（I，q，a）元组。 我们使用 70k 这些元组作为我们的引导程序集 B_{init}。 我们通过测量 CLEVR 验证（val）[23]集合中最终 VQA 模型的问题准确性来评估 LBA 收集的数据的质量。 由于 CLEVR 训练和 val 具有相同的答案和问题类型分布，这使得在 CLEVR 上训练的模型具有固有的优势。 因此，我们还测量了 CLEVR-Humans [24]数据集的问答精度，该数据集具有不同的分布；见图 9.2

模型：除非另有说明，否则我们使用堆叠注意模型作为应答模块 v，并评估三种不同的选择，用于最终的 VQA 模型。

CNN+LSTM 使用 CNN 编码图像，使用 LSTM 的问题，并使用 MLP 预测答案。

CNN + LSTM + SA 使用第 4.2 节中描述的堆叠注意（SA）模型[55]扩展 CNN + LSTM。这与我们的默认应答模块 v 相同。

FiLM [40]使用来自 GRU [10]的问题特征来调制每个 CNN 层中的图像特征。

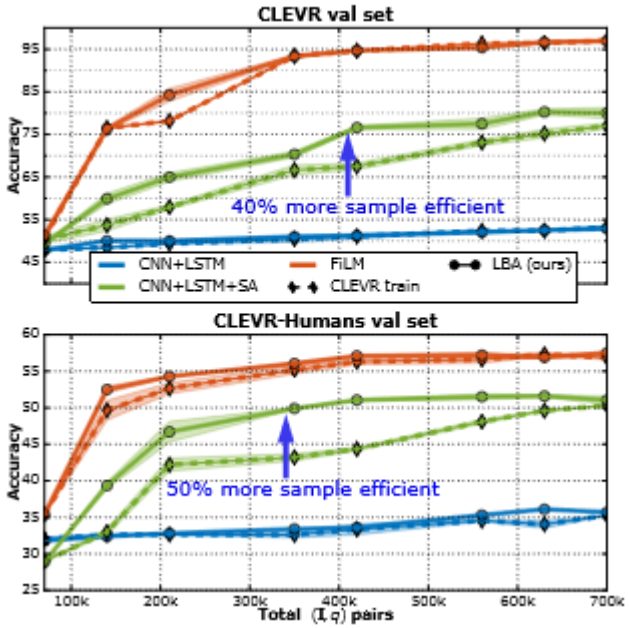


图 4：上图：对于使用 CLEVR 列车（钻石）和 LBA 生成的数据（圆圈）进行训练的 VQA 模型的 CLEVR val 准确度。

下图：CLEVR-Humans 对同一组模型的准确性。 阴影区域表示准确度的一个标准偏差。 在 CLEVR-Humans 上，LBA 比 CLEVR 列车的样本效率高 50％。

除非另有说明，否则我们在所有消融分析实验中使用 CNN + LSTM + SA 模型，即使它具有比 FiLM 更低的 VQA 性能，因为它训练得更快（6 小时对 3 天）。 对于所有 $v_{offline}$ 模型，我们使用各自论文中的训练超参数。

5.1 LBA 生成问题的质量

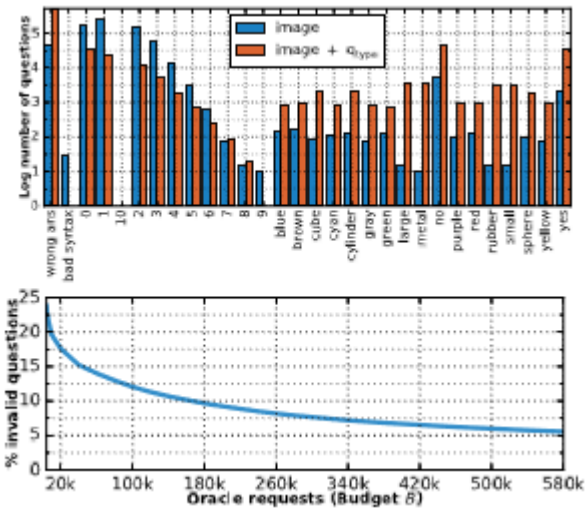


图 5：上图：有问题和无问题调节的 g 生成的问题答案的直方图。

下图：发送到 oracle 的无效问题的百分比

在图 4 中，我们通过测量在两个数据集上训练的 VQA 模型的问答精度，将 LBA 生成问题的质量与 CLEVR 训练进行比较。 图示显示（顶部）CLEVR val 准确度和（底部）CLEVR-Humans 准确度。 从这些图中，我们得出了四个观察结果。

（1）单独使用自举套件（最左边的点）会产生较差的精度，LBA 会提供一个重要的学习信号。

（2）LBA 生成的训练数据的质量至少与 CLEVR 训练数据的质量一样好。 这是一个令人印象深刻的结果，因为 CLEVR 训练具有匹配 CLEVR val 的分布和人类策划用于训练 VQA 模型的双重优势。 尽管有这些优势，LBA 仍然可以匹配并且有时甚至超过其性能。 更重要的是，LBA 显示出对 CLEVR-Humans 的更好的概括，其具有不同的答案分布（参见图 9）。

（3）LBA 数据有时比 CLEVR 训练更有效：例如，在 CLEVR val 和 CLEVR-Humans 上。CNN + LSTM + SA 模型仅需要 60％的（I，q，a）LBA 元组来实现在所有 CLEVR 列车上训练的相同模型的准确性。

4）最后，我们还观察到我们的 LBA 代理在训练期间的每个采样点具有低方差。 阴影误差条显示使用不同随机种子从 5 次独立运行计算的一个标准偏差。 这是从交互式培训环境中得出有意义结论的重要特性（参见[18]）。

定性结果。 图 6 显示了在不同迭代 t 下来自 LBA 生成的数据的五个样本。 它们提供了我们的 LBA 代理发现的课程的洞察力。 最初，模型询问有关颜色（第 1 行）和形状（第 2 行）的简单问题。 它也会产生基本错误（第 1 行和第 2 行的最右边一列）。 随着应答模块 v 的改进，选择策略 π 询问关于空间关系和计数的更复杂的问题（第 3 行和第 4 行）。

5.2 分析：问题提出模块

分析发生器 g 我们通过查看相应答案的分布来评估所生成问题的多样性。 在图 5（上图）中，我们使用最终的 LBA 模型为训练集中的每个图像生成 10 个问题。 对于有和没有“问题类型”条件的生成器，我们绘制了这些问题的答案的直方图。 直方图显示，在问题类型上调节发生器 g 可以更好地覆盖答案空间。 我们还注意到，大约 4％的生成问题具有无效的编程语言语法。

我们在表 1 的前两行中观察到，问题多样性的增加转化为提高的问答精度。 多样性也受采样温度 τ 控制，以 g 为单位。 第 3-5 行显示较低的温度，提供较少的问题提案，对最终的准确性产生负面影响。

分析相关性模型 r。图 5（底部）显示了在线 LBA 培训期间在不同时间步骤发送给 oracle 的无效问题的百分比。 尽管问题的复杂性似乎在增加，但无效的问题率在培训期间从 25％降低到 5％（图 6）。 该结果表明相关性模型 r 在训练期间显着改善

我们还可以通过将“完美”相关模型（oracle）替换为无意中填充所有无效问题的相关模型来解除相关模型 r 与其他设置的影响。 表 1（第 6 行）显示“完美”相关性模型与我们的相关性模型之间的准确度和样本效率差异很小，这表明我们的模型表现良好。

5.3 分析：问题回答模块

到目前为止，我们仅使用一种类型的应答模块 v，CNN + LSTM + SA 测试了我们的策略 π。 现在，我们通过将 v 作为 FiLM 模型并重新运行 LBA 来验证 π 是否适用于其他选择。如第 5.1 节所述，我们通过训练三个 vo ffl ine 模型来评估 LBA 生成的问题。 表 2 中的结果表明我们的选择政策推广到 v 的新选择。

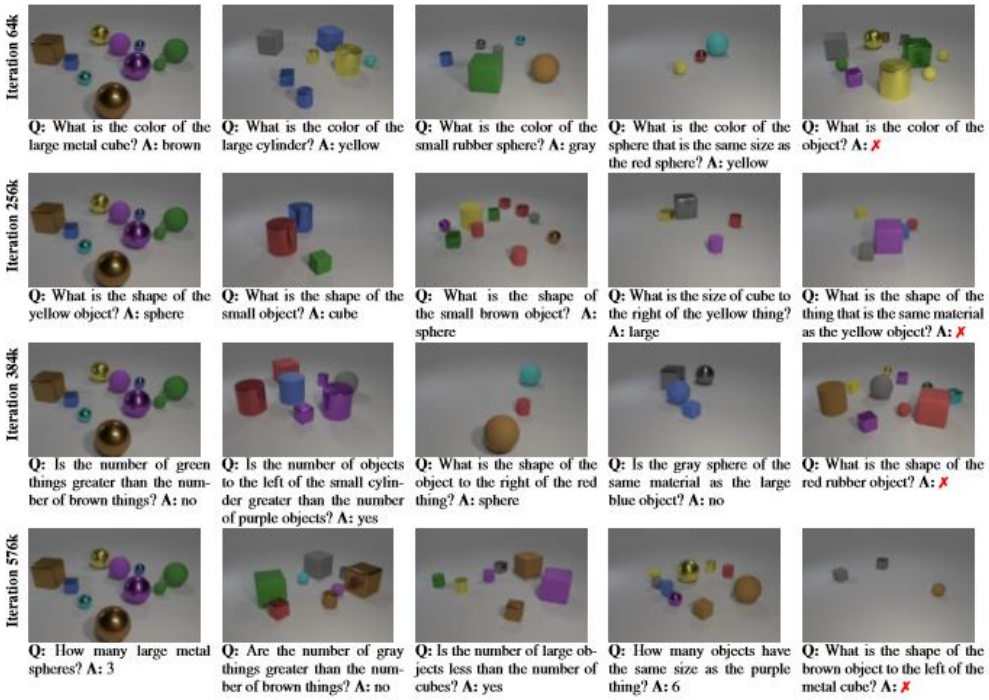


图 6: 我们的 LBA 代理在不同的迭代中提出的示例问题(从程序手动翻译成英语)。随着培训的进展，我们的代理人会提出越来越复杂的问题 - 从简单的颜色问题开始，然后继续塑造和计算问题。 我们还看到无效问题（右栏）变得越来越复杂

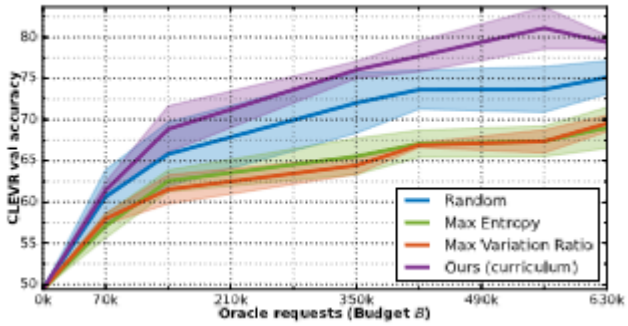


图 7：使用 LBA 训练 CNN + LSTM + SA 的准确性，使用四种不同的策略来选择问题提案（第 4.3 节）。 我们的选择政策更具样本效率

5.4 分析：问题选择模块

为了研究选择政策在 LBA 中的作用，我们比较了四种选择：（1）从问题提案中随机选择；（2）在四次前向丢失后，使用应答模块 v 的预测熵为每个提议（如[47]中所述）；（3）使用预测的变化率[15]；（4）我们的课程政策来自第 4.3 节。我们使用五种不同的随机种子进行 LBA 培训，并报告图 7 中每种选择策略的 CNN + LSTM + SA 模型的平均准确度和 stdev。与先前工作的结果一致[47]，基于熵的策略表现更差比随机选择。相比之下，我们的课程政策大大优于随机选择的问题。图 8 绘制了归一化信息量分数 h（等式 1）和训练问题答案准确度（s（a）按每个答案类型分组)。这些图提供了对课程选择政策 π 的行为的深入了解。具体而言，我们观察到延迟模式：答案类型的信息度分数（蓝色箭头）中的峰值之后是该答案类型的准确度（蓝色箭头）的上升。我们还观察到政策的信息量分数表明了一个易于难以排序的问题：最初（在 64k 请求之后），选择策略更喜欢询问更容易的颜色问题，但它逐渐

转向大小和形状问题，最终，对难以计数的问题。我们强调，这种易于学习的课程是自动学习的，无需任何额外的监督。

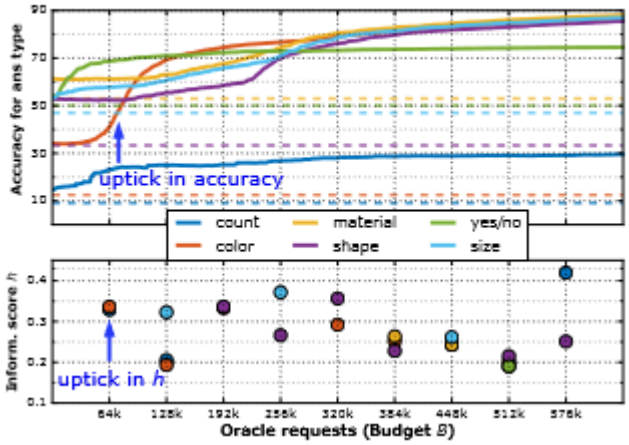


图 8：上图：每种答案类型的训练准确性（实线）和机会级别（虚线）。
下图：每种答案类型的归一化信息分数，平均超过 10k 的问题。 详细信息请参见第 5.4 节。

5.5 改变 Bootstrap 数据的大小

我们改变用于初始化 g，r，v 模型的引导程序集 Binit 的大小，并分析其对 LBA 生成数据的影响。 在表 3 中，我们显示了最终的模型在 CLEVR val 上的准确性。 较小的引导程序集会导致性能降低。 我们还看到，只有不到 5%（第 1 行和第 2 行）的 CLEVR 训练数据集作为我们的引导程序集，LBA 会使用整个 CLEVR 训练集提出可以匹配性能的问题。 根据经验，我们观察到发生器 g 在较小的引导程序集上表现良好。 但是，相关性模型 r 在引导程序集中需要足够的有效和无效（置换）(I，q，a）元组来过滤不相关的问题提议。 因此，较小的引导程序集会影响 LBA 的样本效率。

$\|\mathcal{B}_{\text{init}}\|$	Budget B						
	0k	70k	140k	210k	350k	560k	630k
20k	48.2	56.4	63.5	66.9	72.6	75.8	76.2
35k	48.8	58.6	64.3	68.7	74.9	76.1	76.3
70k	49.4	61.1	67.6	72.8	78.0	78.2	79.1

Table 3: Accuracy on CLEVR validation data at different budgets B as a function of the bootstrap set size, |Binit|.

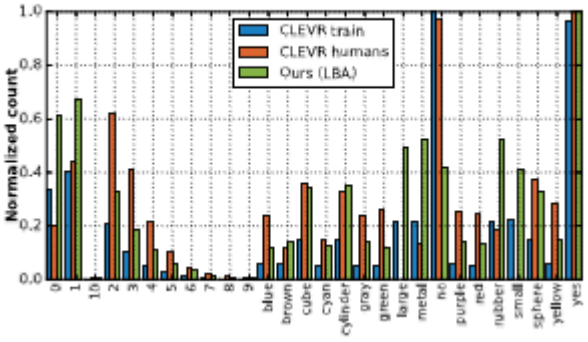


图 9： CLEVR train 的答案分配，LBA 生成数据和 CLEVR-Humans 数据集的分

布。

6.讨论和拓展

本文介绍了学习问题（LBA）范例，并在此设置中提出了一个模型。 LBA 远离传统的被动监督设置，其中人类注释器在交互式环境中提供训练数据，其中学习者寻求所需的监督。虽然被动监督推动了视觉识别的进步[16,17]，但它似乎不适合一般人工智能任务，如视觉问答（VQA）。 策划大量不同的数据，这些数据可以推广到各种各样的问题，这是一项艰巨的任务。 我们的结果表明，诸如 LBA 之类的交互式设置可以促进更高样本效率的学习。 当我们转向日益复杂的视觉理解任务时，如此高的样本效率至关重要。

LBA 的一个重要特性是它不会将在训练时看到的问题和答案的分布与测试时的分布联系起来。 这更类似于 VQA 系统的实际部署，其中用户提出的问题分配给系统是未知的，并且难以预先表征[8]。 图 9 中的 CLEVRHumans 分布就是一个例子。 这个问题为未来的工作提供了明确的方向[7]：我们需要开发对测试时分布变化不太敏感的 VQA 模型；而不是在单个测试分布下评估它们（如当前的 VQA 基准）。

未来工作的第二个主要方向是开发 LBA 系统的“真实世界”版本，其中（1）CLEVR 图像被自然图像替换，以及（2）oracle 被人类注释器替换。 相对于我们目前的方法，需要一些创新来实现这一目标。 最重要的是，它需要在学习者和人类“神谕”之间设计有效的交流方式。 在我们当前的方法中，学习者使用简单的编程语言来查询 oracle。 真实世界的 LBA 系统需要使用多种自然语言与人类进行通信。 通过让 oracle 返回不仅仅回答图像 - 问题对的特权信息，可以进一步提高 LBA 学习者的效率，但这也解释了为什么这是正确或错误的答案[52]。 我们将这种特权信息的结构设计留给未来的工作。

致谢：作者要感谢 Arthur Szlam，Jason Weston，Saloni Potdar 和 Abhinav Shrivastava 对手稿的有益讨论和反馈; Soumith Chintala 和 Adam Paszke 对 PyTorch 的帮助。