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.

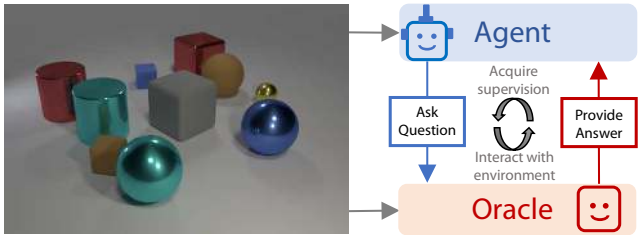


Figure 1: **The Learning-by-Asking (LBA) paradigm.** We present an open-world Visual Question Answering (VQA) setting in which an agent interactively learns by asking questions to an oracle. Unlike standard VQA training, which assumes a fixed dataset of questions, in LBA the agent has the potential to 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%.

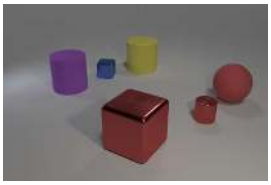
2. Related Work

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.

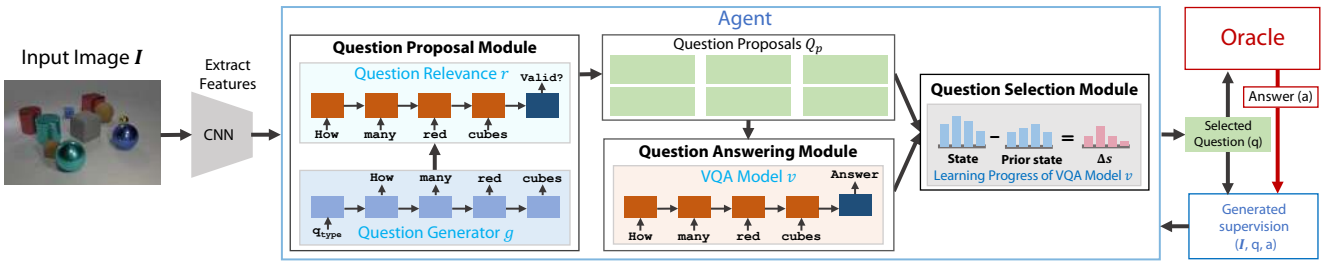


Figure 3: **Our approach to the learning-by-asking setting for VQA.** Given an image $\mathbf{I}$, the agent generates a diverse set of questions using a question generator g. It then filters out "irrelevant" questions using a relevance model r to produce a list of question proposals. The agent then 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.

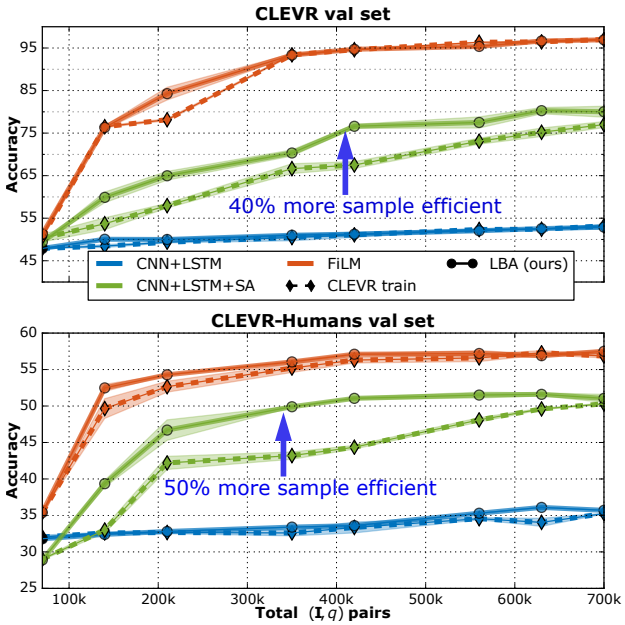


Figure 4: **Top:** CLEVR `val` accuracy for VQA models trained on CLEVR `train` (diamonds) *vs*. LBA-generated data (circles). **Bottom:** Accuracy on CLEVR-Humans for the same set of models. Shaded regions denote one standard deviation in accuracy. On CLEVR-Humans, LBA is 50% more sample efficient than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

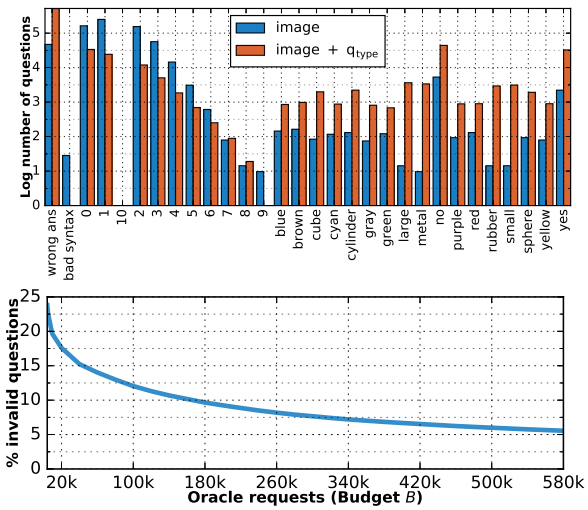


Figure 5: **Top:** Histogram of answers to questions generated by g with and without question-type conditioning. **Bottom:** Percentage of invalid questions sent to the oracle.

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

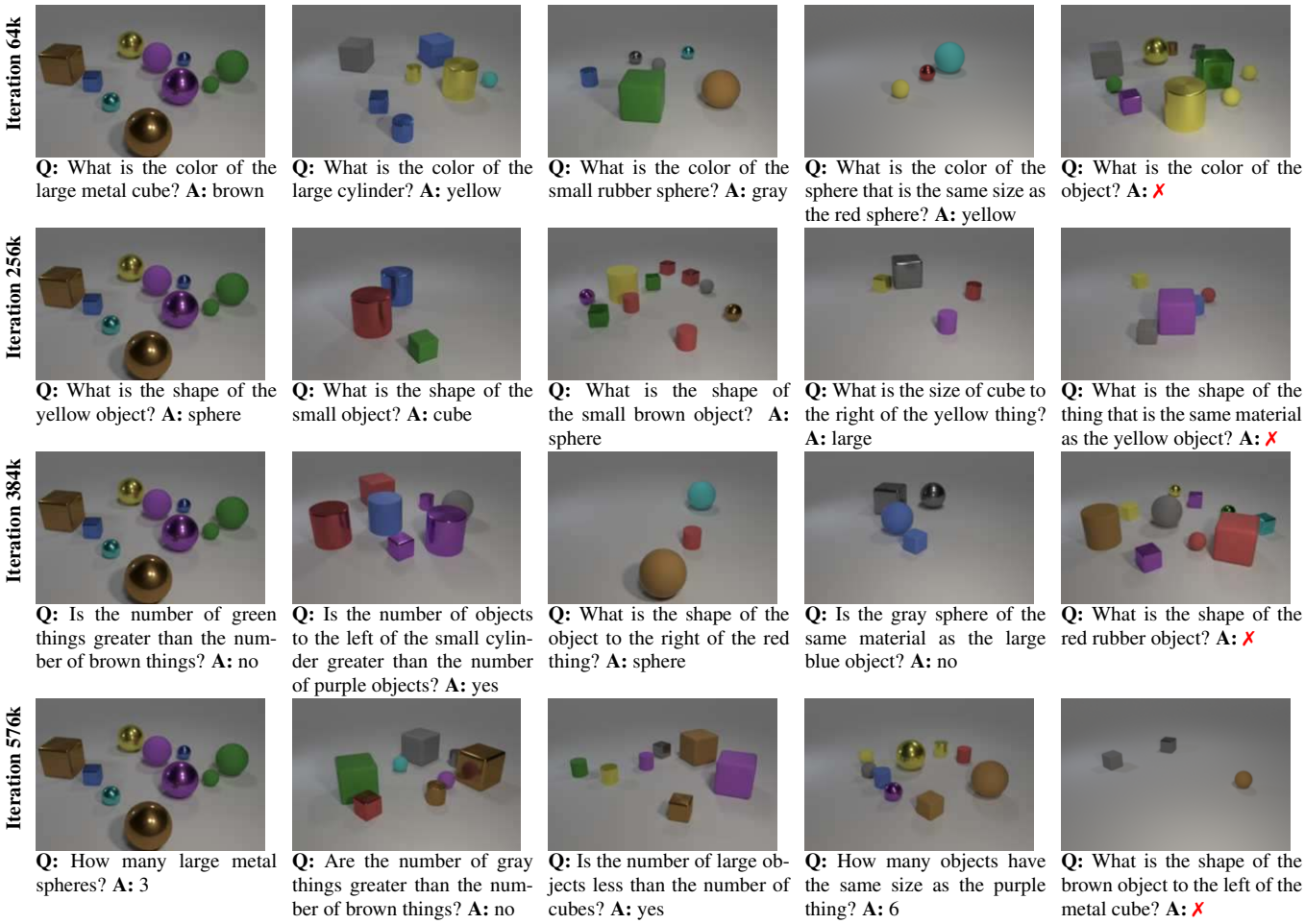


Figure 6: Example questions asked by our LBA agent at different iterations (manually translated from programs to English). Our agent asks increasingly sophisticated questions as training progresses — starting with simple color questions and moving on to shape and count questions. We also see that the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

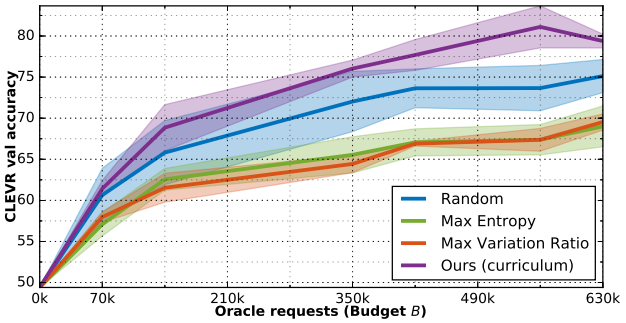


Figure 7: Accuracy of CNN+LSTM+SA trained using LBA with four different policies for selecting question proposals (Sec 4.3). Our selection policy is more sample efficient.

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`

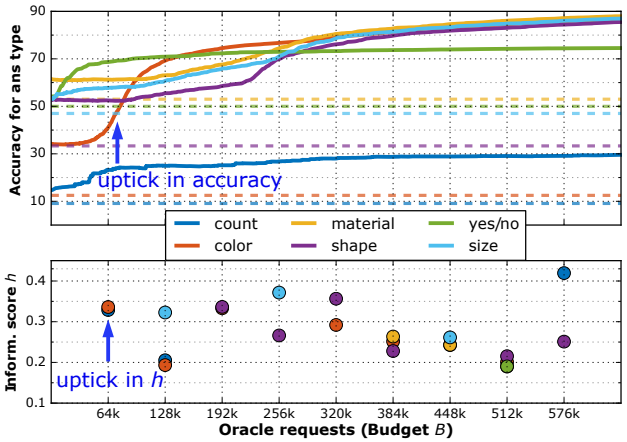


Figure 8: **Top:** Accuracy during training (solid lines) and chance level (dashed lines) per answer type. **Bottom:** Normalized informative scores per answer type, averaged over 10k questions. See Section 5.4 for details.

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.

	Budget B						
$\lvert\mathcal{B}_{\text{init}}\rvert$	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, $|\mathcal{B}_{\text{init}}|$.

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.

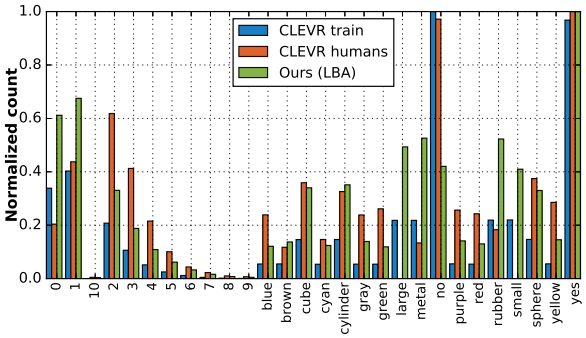


Figure 9: Answer distribution of CLEVR `train`, LBA-generated data, and the CLEVR-Humans dataset.

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