This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2956699, IEEE Transactions on Pattern Analysis and Machine Intelligence

Plenty is Plague: Fine-Grained Learning for Visual Question Answering

Yiyi Zhou, Rongrong Ji, *Senior Member, IEEE*, Xiaoshuai Sun, Jinsong Su, Deyu Meng, *Senior Member, IEEE*, Yue Gao, *Senior Member, IEEE*, Chunhua Shen, *Senior Member, IEEE*

Abstract—Visual Question Answering (VQA) has attracted extensive research focus recently. Along with the ever-increasing data scale and model complexity, the enormous training cost has become an emerging challenge for VQA. In this paper, we show such a massive training cost is indeed plague. In contrast, a fine-grained design of the learning paradigm can be extremely beneficial in terms of both training efficiency and model accuracy. In particular, we argue that there exist two essential and unexplored issues in the existing VQA training paradigm that randomly samples data in each epoch, namely, the "difficulty diversity" and the "label redundancy". Concretely, "difficulty diversity" refers to the varying difficulty levels of different question types, while "label redundancy" refers to the redundant and noisy labels contained in individual question type. To tackle these two issues, in this paper we propose a fine-grained VQA learning paradigm with an actor-critic based learning agent, termed FG-A1C. Instead of using all training data from scratch, FG-A1C includes a learning agent that adaptively and intelligently schedules the most difficult question types in each training epoch. Subsequently, two curriculum learning based schemes are further designed to identify the most useful data to be learned within each inidividual question type. We conduct extensive experiments on the VQA2.0 and VQA-CP v2 datasets, which demonstrate the significant benefits of our approach. For instance, on VQA-CP v2, with less than 75% of the training data, our learning paradigms can help the model achieves better performance than using the whole dataset. Meanwhile, we also shows the effectivenesss of our method in guiding data labeling. Finally, the proposed paradigm can be seamlessly integrated with any cutting-edge VQA models, without modifying their structures.

1 Introduction

Visual Question Answering (VQA) refers to answering a natural language question by giving a reference image, which requires a holistic understanding of visual and textual contents to perform various tasks, such as counting (*how many*), telling time (*when*) and recognition (*what is*). Certain questions in VQA further require logical reasoning to get correct answers, which dramatically increases the task difficulty. To this end, most recent VQA models are built upon deep learning modules. In a typical setting [1] [2], a VQA model consists of a convolution neural network (CNN) to extract visual features, a Long Short Term Memory (LSTM) network to produce text representation, followed by a fusion module (optionally with attention components) to output the final reasoning.

To cope with various answering tasks, state-of-the-art VQA models typically need a large amount of training data and model parameters. For example, the Multimodal Compact Bilinear (MCB) model proposed in [2] has 75 million parameters, a scale almost 30 times larger than ResNet-50 [3]. Specific structures, like Attention Mechanism [4] and Compact Bilinear Pooling [5], are also widely used in VQA [2] [1] [6], which further increase the computational burden in off-line training. For instance, the HiCoAtt model in [6] needs over 100-round epochs to achieve convergence, which takes approximately a week to train using a regular server equipped with a standard Titan GPU.

Y. Zhou, R. Ji, X. Sun are with School of Information Science and Engineering, Xiamen University, Xiamen, Fujian, 361005, China.
Corresponding Author: Rongrong Ji (E-mail: rrji@xmu.edu.cn).
J. Su is with School of Software, Xiamen University, Xiamen, Fujian, 361005, China.
D. Yu is with National Engineering Laboratory for Algorithm and Analysis Technologiy on Big Data, Xian Jiaotong University, Xian, Shanxi, China.
Y. Gao is with School of Software, Tsinghua University, Beijing, China.
C. Shen is with the School of Computer Science, University of Adelaide, SA, Australia.

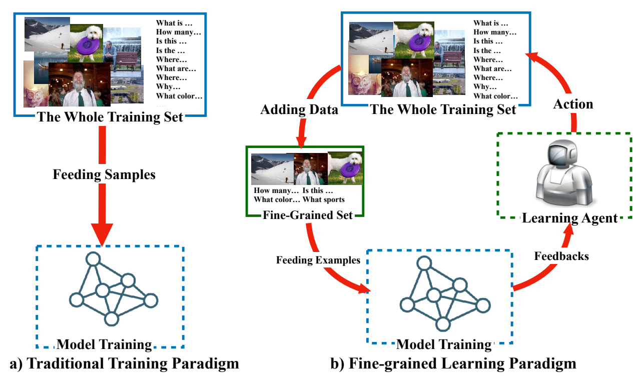


Fig. 1. A comparison between the traditional learning paradigm and our fine-grained learning paradigm.

We argue that such an expensive training cost is indeed plague. Instead, a fine-grained design of the learning paradigm can be beneficial to simultaneously boost training efficiency and model accuracy. In particular, we identify two essential and unexploited issues that widely exist in the learning paradigm of existing VQA models, *i.e.*, the "*difficulty diversity*" and the "*label redundancy*". Generally speaking, the existing VQA training paradigm typically follows a random sampling procedure to pick up training epochs, as shown in Fig.1.a. The "difficulty diversity" refers to the varying difficulty levels of different question types, while the "label redundancy" refers to the redundant and noisy label contained in each question type. The existing random sampling scheme (Fig.1.a) is contradicted with the

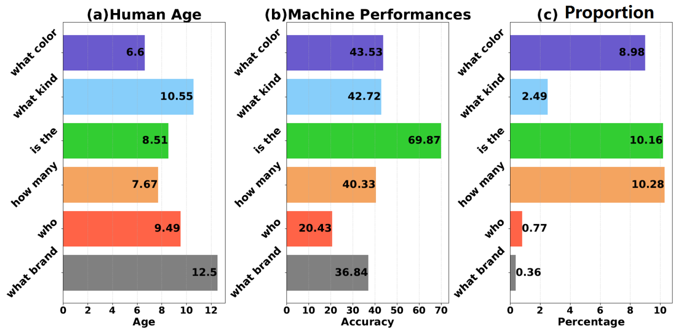


Fig. 2. Statistics of six question types from VQA1.0 [7]. Fig.a shows the ages of humans that can answer each type of question. Fig.b gives the performance of VQA models using visual and textual content on different types. These two figures serve as an indicator of the "difficulty diversity" as introduced in Sec.1. Fig.c gives the proportion of each type of questions in the dataset, which indicates the issue of "label redundancy". These statistics reflect the varying difficulties of different question types and the extremely uneven data distribution, which leads to two key issues in VQA training, *i.e.*, the "difficulty diversity" and the "label redundancy". The target of our fine-grained learning paradigm is to address these two issues by evaluating the learning progress of the VQA model on each question type and selecting the most suitable examples to improve the training efficiency and the model performance.

above two issues, as quantitatively validated latter in Fig.2. Such a learning paradigm leads to low efficiency in offline training, while the learned model is also sub-optimal. We argue, and subsequently validate, that a fine-grained control of the selecting priority and the training epoch quality affect the training quality of VQA models.

In this paper, we propose a fine-grained VQA learning paradigm with an actor-critic based learning agent, termed FG-A1C. Instead of using all training examples from the beginning, we start from a small set of training examples, and gradually augment the training data by evaluating the diversity of concept difficulties and the redundancy of supervised labels, as depicted in Fig.1.b. As the core design of FG-A1C, the learning agent consists of an *actor* network and a *critic* network. Both the actor network and the critic network receive a feedback that reflects the learning progress of the VQA model on different types of questions. Based on this feedback, the actor network first generates an action to perform data augmentation of a specific question type. Then, the critic network evaluates the action and the state, and predicts an expected reward to decide the update direction of the gradients in the actor network. After training on the augmented dataset, the model returns an actual reward for updating the critic network. Finally, the model decides which question type to be trained, upon which the model further picks a subset of examples in the selected question type. Specially, to further filter noisy examples, three data selection schemes are further proposed, which are inspired by *curriculum learning* [8] and *active learning* [9].

To validate the proposed FG-A1C approach, we conduct extensive experiments on the VQA2.0 dataset [10]. In addition to the existing random sampling paradigm, we also compare our approach against other learning paradigms like *Self-paced Learning* [11] and *Active Learning* [12]. Experiments validate the merits of the proposed paradigm. Compared to the alternative approaches and baselines, the proposed FG-A1C has achieved a significant improvement in terms of both learning efficiency and model accuracy. For instance, by using only 50% training examples, FG-A1C saves 21.4% and 25.9% training time for two recent VQA models [1] [14], introducing only 0.6% and 2.9% accuracy decreases, respectively. It is worth noting that, FG-A1C can be seamlessly integrated with almost all VQA models without modifying the model structures.

The rest of the paper is organized as: In Sec. 2, we give a brief introduction to related work. In Sec. 3, the proposed strategy is depicted in details. In Sec. 4, we describe the baselines, experimental setup, experimental results and quantitative analysis. Finally, a conclusion is given in Sec.5.

2 Related Works

2.1 Visual Question Answering

Visual Question Answering (VQA) serves as a hybrid task involving both visual content understanding and natural language processing. At present, VQA is typically regarded as a multi-modal classification problem [1] [2] [7] [13] [6]. Under this setting, the potential answers are treated as fixed categories, which are predicted based on visual and textual features extracted by deep neural networks, *e.g.*, convolutional neural networks (CNN) and recurrent neural networks (RNN). Features of two modalities are fused by concatenation [7] [14] or convolutional operation [15] before sending to the prediction layer. To precisely capture visual signals in the image, the attention mechanism [4] is further introduced, which aims to select the most relevant visual regions according to the question information.

Due to the increasing complexity of questions in VQA, some recent works focus on investigating the revision of attention mechanism to improve the models' reasoning abilities [1] [6] [2] [16]. For instance, Yang *et al.* [1] proposed a multi-step attention operation to gradually and precisely locate potential answer regions. Lu *et al.* [6] proposed two co-attention algorithms to capture the correlation between visual and textual modalities. Fukui *et al.* [2] used a convolutional layer to produce multi-glimpse attentions. Borrowing the idea from [17], Zhu *et al.* used a grid-structured Conditional Random Field to build a structure multivariate attention to capture relations among different visual regions. Patro *et al.* [18] used negative examples to guide the learning of attentions via distinguishing obtained attention features between positive and negative examples.

Some methods further exploit information beyond the given images for VQA [19] [20] [21] [14]. For example, Wu *et al.* [20] used document embedding to encode Wiki entries as the knowledge base to help question answering. The work in [21] uses a set of off-the-shelf algorithms to obtain additional information for question answering, which includes detecting visual relationships and attributes in the image, and incorporating generated image captions in answer prediction. Tenny *et al.* [14] propose a model named *Buttom-up Top-Down attention* (BUTD) , which uses high quality regional features extracted by Fast R-CNN [22] from [23] as visual inputs, which significantly improves performance with a simple model structure. Jiang *et al.* [24] proposed a project named *Pythia* that makes subtle but important changes to BUTD and achieved significant performance improvements. Specifically, they replaced the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2956699, IEEE Transactions on Pattern Analysis and Machine Intelligence

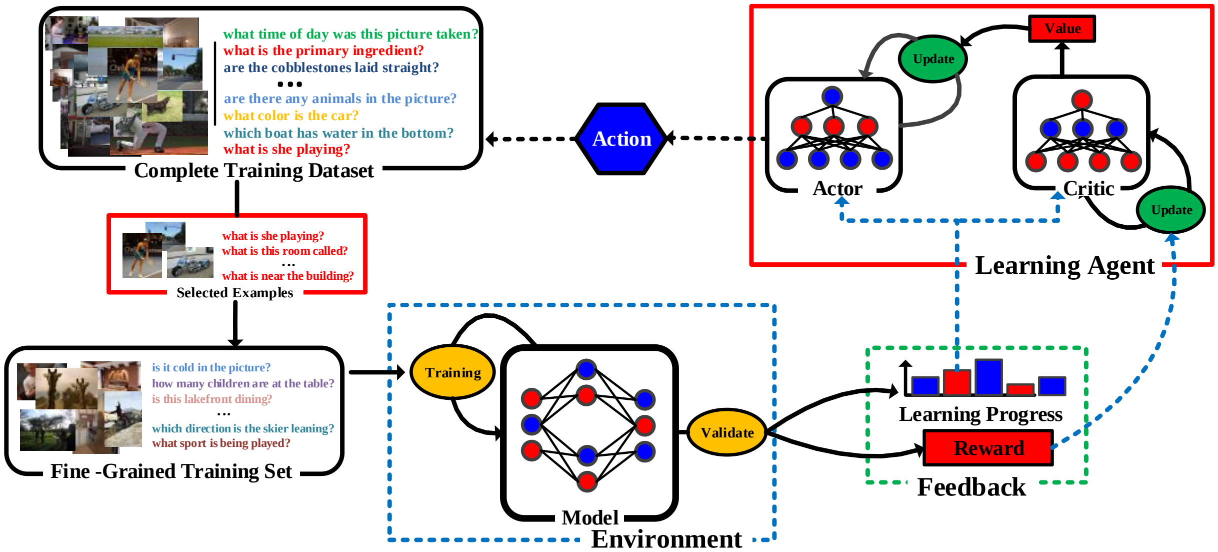


Fig. 3. Overall framework of our fine-grained learning paradigm. Our paradigm starts with a fine-grained training set, which has much fewer examples than the complete training set. A learning agent, composed of an actor network and a critic network, constantly interacts with the model training process. It evaluates the learning progress of the VQA model and generates actions of data augmentations for specific question types. The specific training data are selected via the proposed selection schemes, and integrated to augment the fine-grained training set. Afterwards, the model will be trained on the fine-grained training set and the corresponding rewards are used for updating the learning agent.

activation function and the way of feature concatenations with ReLU and element-wise product. Meanwhile, they also applied some useful training tricks to BUTD, *e.g.*, fine-tuning FRCNN features and data augmentation.

As a key step, the multi-modal fusion also receives great research focus in VQA [25] [2] [26] [27]. In [25], Kim *et al.* used a residual learning framework to obtain the deep interaction between two modalities. In [2], Fukui *et al.* first introduced the bi-linear pooling based fusion method, termed multi-modal compact bilinear pooling (MCB), to efficiently capture interactions between visual and textual features. Although MCB helps the model achieve significant performance gains, it also leads to a large increase in model parameters. Kim *et al.* [28] and Yu *et al.* [27] proposed two low-rank bi-linear pooling fusion methods, which aim to improve the model performance while reducing the number of parameters.

2.2 Learning Paradigms

Inspired by the cognitive process of humans, Bengio *et al.* [8] proposed a novel learning paradigm, termed *Curriculum Learning (CL)*, which gradually includes training examples from easy to hard. The curriculum is often derived from predetermined heuristics in particular problems, which is less adaptive to other problems [29]. Based on CL, Kumar *et al.* [11] proposed a dynamic learning paradigm termed *self-paced learning (SPL)*. SPL embeds the curriculum design into the model learning, which dynamically selects suitable examples based on the current learning progress. Jiang *et al.* [29] extended SPL by considering the diversity of training examples, which makes it more practical to different tasks. In [30], the relationship between curriculum learning and self-paced learning is explored. Another related learning paradigm is the *active earning (AL)*, which targets at achieving comparable performance with fewer training labels. AL assumes that if a model is able to select the data from which it learns, it will perform better even with fewer training examples [9]. The data selection metric of AL is very different from that of SPL. It prefers examples with more information, for instance, using the uncertainty measure to find examples with large entropies on the conditional distribution [31] [32], or examples that are closest to the classification boundary [33] [34]. A recent learning paradigm named *learning-by-asking* was proposed in [35], which also follows the spirit of active learning. The principle of [35] is similar to ours in that the paradigm requests specific training examples according to the learning state of the model. However, the main difference is that learning-by-asking heavily relies on the oracle provided by the CELEVR dataset [36] to create suitable examples, which greatly limits its application scenarios. In contrast, our scheme can accommodate most existing VQA datasets, which takes advantage of available training examples and requires no extra labels.

Reinforcement learning can be divided into three groups [37]: actor-only, critic-only and actor-critic methods, where actor and critic are synonyms for the policy and value function, respectively. The actor-only methods work with a parametrized family of polices. They merit in that the parameters are directly estimated and improved, while the shortcoming is that the gradient estimator may have a large variance. The critic-only methods aim at learning an approximation to the Bellman equation. They work well when it is possible to build a "good" approximation of the value function. However, both methods can not reliably guarantee the optimal solution of the resulting policy. Actor-critic methods aim at combing the advantages of actor-only and critic-only methods. Actor-critic learning is also investigated in deep learning [38] [39] [40].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2956699, IEEE Transactions on Pattern Analysis and Machine Intelligence

Some recent works also focus on applying reinforcement learning (RL) methods to the process of efficient data selections [41] [42] [43]. The work in [41] proposes a deep RL framework called Neural Data Filter to explore automatic and adaptive data selection in the tasks of text and image classifications. Liu *et al.* [43] followed the idea of [41] and proposed a learning scheme called imitation learning, which incorporates prior knowledge to shorten the training process of the policy network. In addition to the differences of application scenarios and the RL methods used, our scheme differs from these works in two main aspects. First, these works focus on selecting high-value examples and minimizing the amount of training examples. In practice, the process of their example evaluations typically consumes a large proportion of learning cost. In contrast, our scheme aims at boosting the training efficiency as well as reducing the amount of training examples required. Second, the learning agent in these works requires offline training, which means the RL networks need to train with at least several full training periods before being applied to the data selection. In contrast, our learning agent is set as an online learning model, which can be directly trained with any VQA models and requires few extract training costs.

3 The Proposed Fine-Grained Learning

The main target of our fine-grained learning scheme is to reduce the number of training examples as well as the cost of model training. To this end, we propose a learning agent to evaluate the learning state of the VQA model on different question types, and then augment the target data to accelerate the model training. The corresponding framework is depicted in Fig.3. In the following, we describe the design of our learning paradigm in detail.

3.1 Problem Setup

We denote the fine-grained training set as D_{train}, which is initialized with a small number of examples. After each training epoch, the VQA model, M_{vqa}, is evaluated on the validation set, (denoted as D_{val}), and the learning agent will receive a state $s \in \mathbb{R}^k$ that reflects the model performance on different question types. Based on this state, the learning agent is able to decide examples of which question type should be added to the D_{train}, such that the model can improve the overall performance.

Since the capacity of the fine-grained training set is limited, *e.g.*, 50% of the entire dataset, the learning agent should make best choices within N sampling steps to find most suitable examples for the model training. We cast this fine-grained learning into a decision process, by which reinforcement learning can be applied to maximize the performance improvements. Specifically, we design the state feature s, action space a and reward r as follows.

State Feature. The state feature $s \in \mathbb{R}^k$ denotes the learning progress of the VQA model on each question type, where k denotes the number of question types. It can be calculated by $s_t = x_t - x_{t-1}$, where $x_t \in \mathbb{R}^k$ denotes the averaged cross-entropies of each question type in the validation set at the t-th training epoch. To explain, there is a significant gap among the difficulty of each type of question in VQA, which is difficult to measure the importance of example types by simply using the model performance to represent the learning state of the model. Instead, we adopt the learning progress as the state feature to capture the subtle changes on each tasks.

Action space. The discrete action space a is denoted as $a_i \in \{1, 2, ..., k, k+1\}$. The 1-th to the k-th actions refer to a data sampling on the corresponding question type, and the $(k+1)$-th action refers to not data augmentation. The $k+1$ action is designed to take into account that the model occasionally need certain training steps to digest the newly integrated examples.

Reward Function. The reward function is denoted as:

$$r\left(s_t, a, s_{t-1}\right) = l_{t-1} - l_t, \tag{1}$$

where l_t denotes the overall loss at the t-th step. Such an immediate reward helps the learning agent quickly adjust its parameters during the model training.

The objective of our learning scheme is to maximize the expectation of rewards in the limited sampling steps. Therefore, we set the cost-to-go function in a discounted setting as:

$$J\left(\pi\right) = E\left\{\sum_{k=0}^{\infty} \lambda^k r_{k+1} \middle| \pi\right\}. \tag{2}$$

Here, $\lambda \in [0, 1)$ is the discount factor used to trade-off the importance of immediate and future rewards. π denotes the policy that the learning agent needs to learn.

3.2 Actor-Critic based Learning Agent

In order to avoid excessive training cost, the learning agent should quickly adapt to the VQA model training. In other words, its structure should be simple. More importantly, it can be updated after each sampling step. To this end, we build the learning agent with an actor-critic setting and use a relatively shallow network structure. Specifically, it consists of two main components: the actor network (policy function) and the critic network (value function). The actor network consists of fully-connected layers and a Softmax layer with parameters ϑ, which is denoted as π_ϑ. The critic network is a one-layer network with parameter θ, denoted as V_θ. Both the actor and the critic networks receive the state vector s_t.

The actor network is to generate a data augmentation action, while the critic network evaluates the current policy by a value function approximation, which is called *policy evaluation*. Here, we use the state-value function to estimate J:

$$V_\theta\left(s_t\right) = E\left\{\sum_{i=0}^{\infty} \lambda^i r_{i+1} \middle| s_0 = s_t, \pi_\vartheta\right\}. \tag{3}$$

The Bellman equation of the state value function can be described as:

$$V_\theta = E\left\{r\left(s_t, a, s_{t+1}\right) + \gamma V_\theta\left(s_{t+1}\right)\right\}, \tag{4}$$

where $r\left(\cdot\right)$ denotes the reward function.

To find an appropriate policy, a prerequisite is that the critic should be able to accurately evaluate a given policy.

We use temporal difference (TD) [44] to update the critic. At the t-th step, the TD error δ_t can be estimated as:

$$\delta_t = r_{t+1} + \gamma V_{\theta_t}(s_{t+1}) - V_{\theta_t}(s_t). \tag{5}$$

The TD error δ_t is to decide the direction of the update gradients of the critic. The update equation is denoted as:

$$\theta_{t+1} = \theta_t + \alpha_{c,t}\delta_t \Delta_\theta V_{\theta_t}(s_t), \tag{6}$$

where $\alpha_{c,t}$ is the learning rate of the critic agent. However, Eq.6 is only a one-step estimation and does not consider the historical rewards. For model training, the rewards are often the results of a series of actions. In this case, we include the Eligibility Traces [45] to make use of past experiences. The eligibility trace gradients are denoted as z_k, and its updating equation is:

$$z_t = \lambda_\gamma z_{t-1} + \Delta_\theta V_{\theta_t}(s_t), \tag{7}$$

where λ_γ is a decay factor with $\lambda \in [0, 1)$. Then Eq.6 is modified to the following:

$$\theta_{t+1} = \theta_k + \alpha_{c,t}\delta_t z_t. \tag{8}$$

In terms of the actor, the updating equation is:

$$\vartheta_{k+1} = \vartheta_k + \alpha_{a,k}\Delta_\vartheta J_k. \tag{9}$$

According to the *policy gradient theorem*, the gradient can be denoted as:

$$\Delta_\vartheta J_k = \Delta_\vartheta \log \pi_{\vartheta_k}(s, u) V_{\theta_{k+1}}(s). \tag{10}$$

Eq.10 greatly connects both the actor network and the critic network. The value evaluation results will be used to guide the direction of the critic' gradients. When the critic can correctly predict the action reward, it helps the actor to find out the best action based on the given state vector.

3.3 Example Selection

In principle, our scheme focuses more on perceiving the model's learning progress on each question types, and performs data augmentation at the task level, which is the main difference to the previous works [41] [35] [43]. Nevertheless, we also include three example selection strategies to facilitate the model learning.

3.3.1 Active Sampling

Active sampling aims to select examples with more information, *i.e.*, more training values. Following [46], we use entropy to measure the amount of information in a sample. Given an example e_k^i from D_i, its entropy is defined as:

$$e_k^i = -\sum_{j=1}^{N} p_k^j \log p_k^j, \tag{11}$$

where N is the dimension of answer space and p_k is the prediction of M_{vqa}. However, such measurement is more likely to sample noisy examples, *e.g.*, outliers in data distribution. Therefore, we discard the first 10% of the examples during each sampling, and then selects the top H from the rest.

3.3.2 Weighted Sampling

In contrast to active sampling, weighted sampling prefers examples with low entropy during each selection, which follows the principle of *curriculum learning* [8] that manages the teaching from easy to hard. The weight of a candidate example can be calculated as:

$$w_k = \frac{e_k^{-1}}{\sum_{w_j \in D_i} e_k^{-1}}. \tag{12}$$

We then sample n examples from this weighted distribution.

3.3.3 Self-paced Sampling

Inspired by *self-paced learning* [11], [29], we further use a dynamic threshold vector, $\xi \in R^k$, to select training examples of a corresponding task. Different from the traditional SPL scheme [11], we hope to select a fixed number of examples during each sampling, which can avoid selecting too many easy examples for the model training. Specifically, given a threshold ξ^i of the i-th task, the weight of an example in this task is defined as:

$$w_k = \frac{|e_k^{-1} - \xi^i|}{\sum_{w_j \in D_i} e_k^{-1}}. \tag{13}$$

Therefore, during each augmentation, examples of which entropy values are closer to the threshold will be selected. Meanwhile, the threshold ξ^i will be increased after each action, which can be expressed as: $\xi^i \leftarrow \alpha_t \xi^i$, where $\alpha \in [1, \infty)$. The dynamic threshold guides the model to learn easy examples at the infant stage. When the model becomes more mature, more informative examples will be included.

Specifically, the motivation of the active sampling is very different from the weighted sampling and the SPL sampling. To explain, the proposed three strategies is to take account the situations of the existing VQA datasets and models. VQA datasets typically contain some questions that are too difficult to answer or have ambiguous answers. In this case, simply feeding difficult questions may be counterproductive for the model training. Meanwhile, for some simple models, simple yet informative examples might be more beneficial.

3.4 Overall Algorithm

The overall learning procedure is depicted in Alg.1. The complete dataset is dented as $D_{vqa} = \{D_1, D_2, ..., D_k\}$, where k is the number of question types. Each subset D_i contains n_i training examples. The fine-grained training set D_{train} is initialized with N randomly selected examples, and the validation set D_{val} exactly follows the data distribution of D_{vqa}. During each selection, the agent selects up to K examples from the target question type. When there is no example in the target subset D_i, the agent will make a suboptimal choice. The data selection continues until D_{train} has sufficient examples, while the model will keep training until reaching the optimal state.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2956699, IEEE Transactions on Pattern Analysis and Machine Intelligence

Algorithm 1 Training with Fine-grained A1C Learning Paradigm

Input: The complete training set D_{vqa} and the val set D_{val}. A discounting factor λ.
Output: The fine-grained training set D_{train} and the trained VQA model M_{vqa}.

```
Initialize the VQA model M^0_vqa and the learning agent M^0_A1C, and set the state vector x_0 ∈ R^n with zeros.
Initialize D_train with N random selected examples.
Evaluate M^0_vqa on D_val and obtain the model loss l_0 and the cross entropy vector x_0.
for t in M Epochs do
    Obtain an action: a_i^{t-1} by the actor network Actor(s_{t-1}).
    Select K examples in the i-th question type, and add examples to D_train.
    Evaluate M^t_vqa on D_val and obtain new overall loss l_t and cross entropy vector x_t.
    Obtain reward r_{i-1} = (l_{i-1} - l_i).
    Obtain new state s_t ← (x_t - x_{t-1})
    Update the actor and the critic with [s_{t-1}, r_{t-1}, s_t, r_t, λ] by Eq.10.
    Update weights of M^t_vqa based on D^t_train.
end for
return The trained VQA model M^t_vqa and the fine-grained training set D^t_train
```

3.5 Application of Expert Knowledge

Since the learning agent is trained simultaneously with the VQA model, it is expected to well predict the action and the reward as soon as possible. In this case, we apply some prior knowledge to the setting of model configurations. Specifically, in terms of the actor network, the values of the weights in the prediction layer are set according to the default distributions of the corresponding question types. Such a design can enable the model to tend to choose questions of most frequent types in the initial phase, such as the binary questions containing answers only "yes" or "not". These questions are usually easier to answer, which typically occupy a certain percentage in the dataset and have a great impact on the final model performance. In terms of the critic network, the values of its weight parameters are all set to non-negative. Meanwhile, before the training starts, we test the initialization of the weights to ensure the predicted reward is close to the estimated results.

4 Experiments

We apply our approach to two VQA models, *i.e.,Stacked Attention Networks* (SAN) [1] and *Bottom-up Top-Down network* (BUTD) [14], and conduct extensive experiments on two benchmark datasets, *i.e.,* VQA2.0 [10] and VQA-CP [48].

4.1 Dataset

VQA2.0 [10] is built on top of the widely-used VQA1.0 dataset [7]. It has 204,721 images from COCO dataset [47], with about 1.1 million questions that are double of that of VQA1.0. Each question has 10 answers labeled by 10 AMT workers. The sizes of training set, the validation set and the testing set are 443,757, 214,354 and 447,739, respectively. Following the setting in [2], we select the top-3,000 most frequent answers to build the answer vocabulary, and discard training examples that are not in this vocabulary. We follow most VQA methods [1], [2], [14] that combine the training set and the validation set for model training, and separate 10,000 examples for validations. The data distribution of the validation set follows the one of the entire dataset. Therefore, we make a fair comparison between different training paradigms. For the training set, we divide its examples into seven main types, which are *Yes/No*, *Counting*, *what*, *where*, *which*, *who* and *why*. For examples that don't belong to these seven types, we classify them into the one of *others*. Detailed statistics are shown in Tab.1.

TABLE 1
Statistics of question types of VQA2.0 and VQA-CP-2.0.

Type	VQA2.0	VQA-CP2.0	Type	VQA2.0	VQA-CP2.0
Yes/No	263,186	192,958	Counting	72,058	43,216
What	270,636	169,911	Where	13,924	8,490
Which	7,830	4,308	Who	3,224	2,163
Why	6,834	4,177	Others	20,419	12,960

VQA-CP (*Visual Question Answering under Changing Priors*) datasets [48] are built upon VQA1.0 and 2.0 datasets, which aim to eliminate the effects of language priors in VQA examples. VQA-CP *v1* and *v2* are created by re-organizing the *training* and *val* splits of VQA1.0 and VQA2.0 respectively. Their distributions of answers per question type are by design different in the test split compared to the training split [48]. In this paper, we focus on the VQA-CP-v2 set, which has about 438K examples for training and 220k examples for testing. Following the above setting, we also divide the training examples into 8 main types, the number of which are also shown in Tab.1.

4.2 Experiment Setup

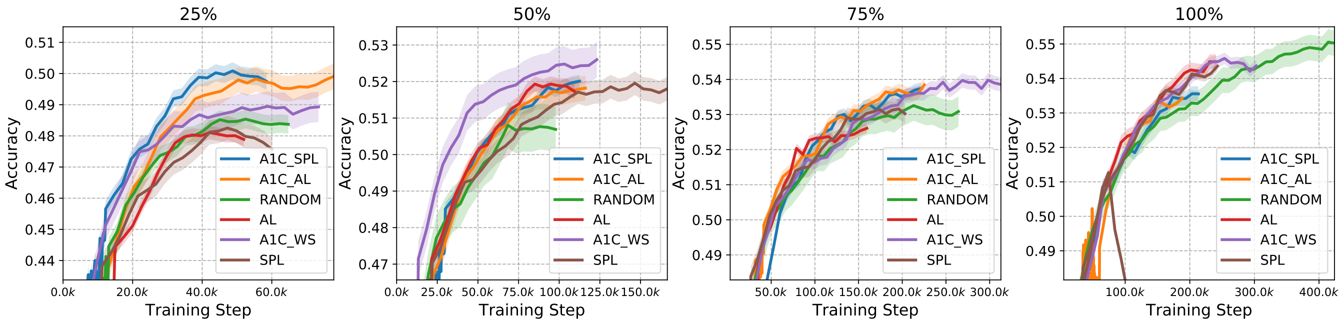
4.2.1 VQA Models

For SAN, we implement the model with L2 regularization for model variables, and use the convolutional feature maps before the last pooling of a pre-trained ResNet-152 [3] as the visual input, which has a shape of $14 \times 14 \times 2048$. We use one attention layer to attend to the visual features. The dimensions of attention embeddings and the prediction layer are set to 512 and 3,000 respectively. During training, we follow the setting in [3] that selects the most frequent answer of each example as the label, and use the *softmax cross entropy* as the model's training loss.

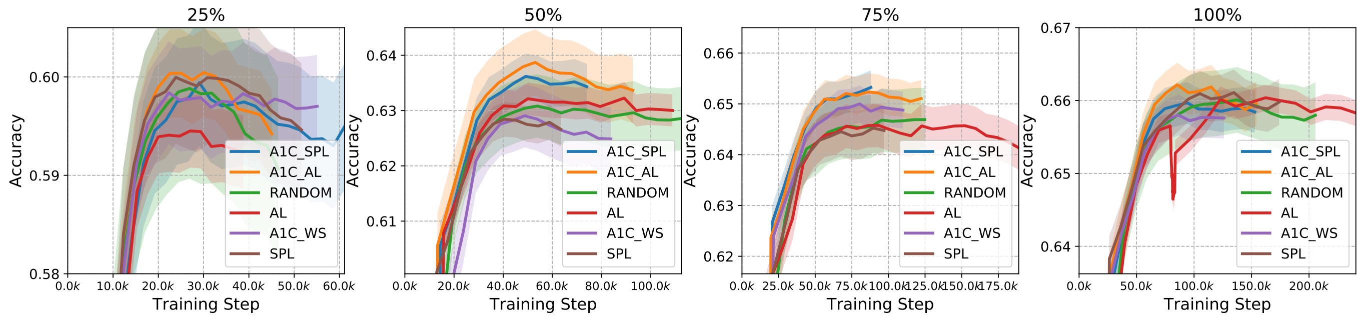
For BUTD, we abandoned the manual initializations of the textual and visual prediction layers, and the rest of the model structure is the same to the original one in [14]. The dimensions of attention embedding and the prediction layer are set to 512 and 3,000 respectively. Following the setting in [14], we use the regional features extracted by Faster RCNN as the visual input [23]. Meanwhile, we convert the given answer list of each example into a soft label vector [14] and use the *binary cross entropy* as the model's training loss.

For both models, we use Adam [49] as the optimizer, and the learning rate and batch size are set to 1e-5 and 64, respectively.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2956699, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) SAN



(b) BUTD

Fig. 4. Learning curves of different learning paradigms with different proportions of training examples on VQA2.0 dataset.

4.2.2 Learning Paradigms

We compare our paradigms with three baselines, which are *Random Sampling*, *Self-paced Learning* [11] and *Active Learning* [9], respectively. For simplicity, we denote them as *Random*, *SPL* and *AL*. For SPL, we augment the examples of entropy values below the threshold to the training set. For AL, we add a fixed number of examples based on the sorting of entropy values. Meanwhile, we denote our learning paradigm with three sampling strategies, *i.e., Active Sampling*, *Weighted Sampling*, and *Self-paced Sampling*, as *FG-A1C-AL*, *FG-A1C-WS* and *FG-A1C-SPL*, respectively. These paradigms all selects a fixed number of training examples during each sampling. For all paradigms, we test their performance on 25%, 50% and 75% proportions of training examples, respectively.

In terms of our RL learning agent, the Actor is a shallow network consisting of a fully-connected layer with dimensions of 7×14, and a Softmax Layer with a dimension of 14×8, while the Critic network has two fully-connected layers with dimensions of 7×14 an 7×1. The activation function used is *tanh*.

On the VQA2.0 dataset, the settings of all learning paradigms are as follows. For all paradigms except *Random*, the numbers of initial training examples for all four proportions are 80K, 160K, 240K and 320K, respectively. The numbers of examples of each sampling are 3K, 6K, 8K and 8K. For SAN, the training interval steps for validations are 1K, 2K, 3K and 4K for proportions of 25%, 50% and 75% and 100%, while the ones for BUTD are 100, 200, 300 and 400, respectively. The different settings of training interval are due to the different performance of the two models. Due to the advantages of network architectures and FRCNN visual features, BUTD can digest sampled examples faster than SAN. For *Random*, we train the model with all available examples from scratch. On VQA-CP dataset, the sizes of initial training sets under different proportions are all set to 30K, while the settings of samplings and the training intervals are the same with the ones of VQA2.0. For all paradigms, the early stop is applied when the performance is not improved after 5 validations.

In terms of the evaluation metric, we use *VQA Accuracy* [7] for both two datasets, which can be denoted as:

$$Acc\,(ans) = \min\left\{\frac{\#humans\ that\ said\ ans}{3}, 1\right\}. \tag{14}$$

This metric means that if the prediction is consistent with three or more manually labeled answers, the accuracy is 1.

4.3 Experimental Results

4.3.1 VQA2.0

We first present the learning curves and evaluation results of two VQA models under different proportions of training examples in Fig.4 and Tab.2. From Fi.g4, we can first observe that the proposed fine-grained learning paradigms can successfully train two VQA models and achieve clear improvements in terms of both the training efficiency and the model accuracy, especially when fewer training examples are available. For instance, with the setting of 25% training examples, FG-A1C-SPL helps SAN achieves above 5% performance gains and about 20% training cost to the *random* paradigm. For BUTD, FG-A1C-AL achieves about 3% and 15% improvements in terms of both the model accuracy and training efficiency under the setting of 50%.

We also notice that the advantages of our learning paradigms become less significant when the proportion of training examples used increases after a certain value. For

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2956699, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 2
Evaluation results of SAN and BUTD with different learning paradigms on the VQA2.0-Test-dev.

SAN	25%				50%				75%				100%			
Method	All	Y/N	Num.	Other	All	Y/N	Num.	Other	All	Y/N	Num.	Other	All	Y/N	Num.	Other
Random	48.5	67.4	31.7	39.2	51.2	68.9	32.1	40.2	54.3	72.7	34.6	43.1	55.2	73.0	34.2	44.6
SPL	49.1	68.5	32.7	36.4	51.8	71.0	32.6	41.2	53.8	71.8	35.4	42.5	54.2	71.7	35.5	43.5
AL	48.5	66.5	31.7	40.4	51.9	70.0	32.4	41.6	53.4	70.8	34.7	42.7	54.2	71.2	35.8	43.7
A1C-SPL	50.1	66.9	31.4	40.1	52.1	68.4	32.1	42.5	54.4	70.7	34.6	45.1	54.8	72.9	35.1	43.8
A1C-AL	49.9	66.0	29.6	40.8	52.1	68.8	31.4	42.6	54.2	71.0	34.2	44.5	53.6	70.8	35.5	42.9
A1C-WS	50.1	68.0	31.2	38.9	52.6	69.0	30.8	43.6	54.6	71.5	37.0	44.1	55.0	73.4	35.0	43.8
BUTD	25%				50%				75%				100%			
Method	All	Y/N	Num.	Other	All	Y/N	Num.	Other	All	Y/N	Num.	Other	All	Y/N	Num.	Other
Random	60.0	77.0	39.3	49.8	64.1	80.6	44.9	54.4	65.0	82.4	43.2	55.1	66.2	83.0	46.8	56.2
SPL	60.3	77.1	40.0	50.5	63.7	81.2	42.5	62.8	65.5	81.9	45.7	56.0	66.2	83.1	46.1	56.0
AL	59.5	74.3	38.7	52.4	64.3	81.0	44.7	54.5	65.2	81.4	46.4	55.7	66.0	82.8	47.1	57.0
A1C-SPL	60.0	76.0	39.7	51.2	65.0	81.9	43.1	54.8	65.7	82.1	46.3	56.0	66.8	83.3	48.2	57.0
A1C-AL	60.9	76.6	40.0	52.4	64.6	80.8	42.7	55.8	65.8	81.4	44.5	57.2	67.0	83.6	47.7	57.2
A1C-WS	60.3	75.6	40.0	51.9	64.2	82.0	45.8	53.4	65.2	80.1	47.3	56.7	66.5	83.2	47.4	56.5

instance, when trained with the full dataset, the BUTD performance by FG-A1C-SPL is slightly better than that by Random, *i.e.,* 66.8 *v.s.* 66.2. To explain, when trained with the full data, the final performance is mostly determined by the quality of the entire dataset, rather than the schedule of each training epoch. But we still can see that our learning paradigm can help the model to converge to optimal more quickly, *e.g.,* above 20% training saving on SAN as shown in Fig.4.

Another observation is that the proposed AL and WS sampling strategies have different effects on two VQA models. Specifically, WS can help SAN achieve better model performance than AL, while AL is more suitable for BUTD. To analysis, as a classical VQA model, the learning ability of SAN is largely limited by its network design and the visual features used. For instances, its *softmax cross entropy* based objective function is much less efficient than that based on *multi-label binary cross entropy* [14]. Thus, WS can collect questions with more certain content and less noisy label information to help SAN achieve the best performance. In contrast, BUTD, as an up-to-date VQA model, shows a better question answering ability than SAN, which requires more informative examples to reach the optimal state. Compared to FG-A1C-WS and FG-A1C-AL, we find that FG-A1C-SPL is more general, which shows good efficiency in both SAN and BUTD, as shown in Fig.4 and Tab.2. To explain, FG-A1C-SPL can adjust the thresholds of different question types according to the learning pace of models, so either easy or informative examples of each question type can both be included to the training set. Meanwhile, compared to SPL [11], we fixed the number of sampled examples to avoid collecting too many easy examples. Its main shortage lies in the selections of the pace and the initial thresholds, which requires both prior experiences and cross-validations.

We further compared our learning paradigms with 25%, 50% and 75% of training data used to the *Random* paradigm trained with the whole dataset in Fig.5. Since the time for each training step are different on different hardwares, we define a notation called "learning step" to access the training efficiency. For our learning paradigms, its leanring steps includes *training steps*, *validation steps* and the *example evaluation steps*, while *the learning steps* of *Random* consists of *training steps* and *validation steps*. Since the learning agent in FG-A1C are two shallow networks, the time required for its policy generation and gradient updates are very short and neglectable to the whole training process. Therefore, we do not include the training cost of the A1C agent.

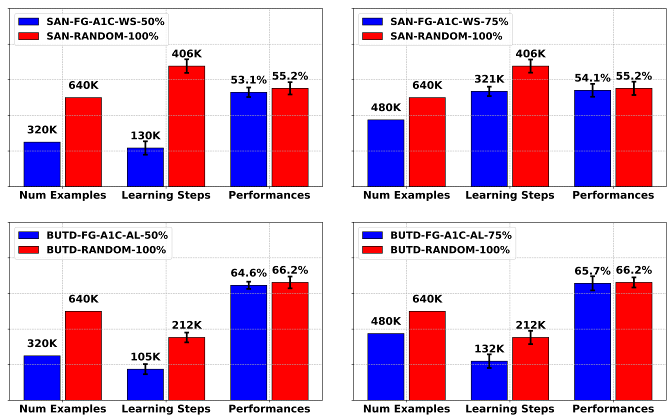
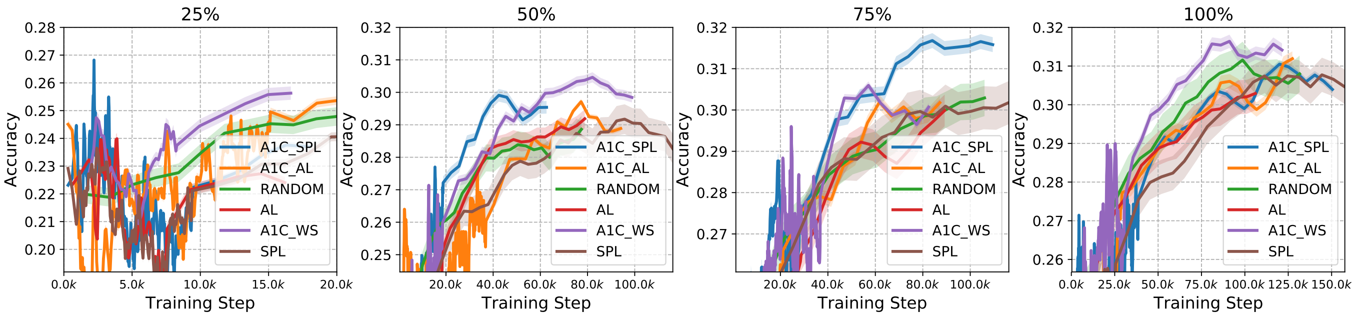


Fig. 5. Comparisons of the training expenditures and the model performance between FG-A1C paradigms and the random sampling scheme on the VQA2.0 dataset.

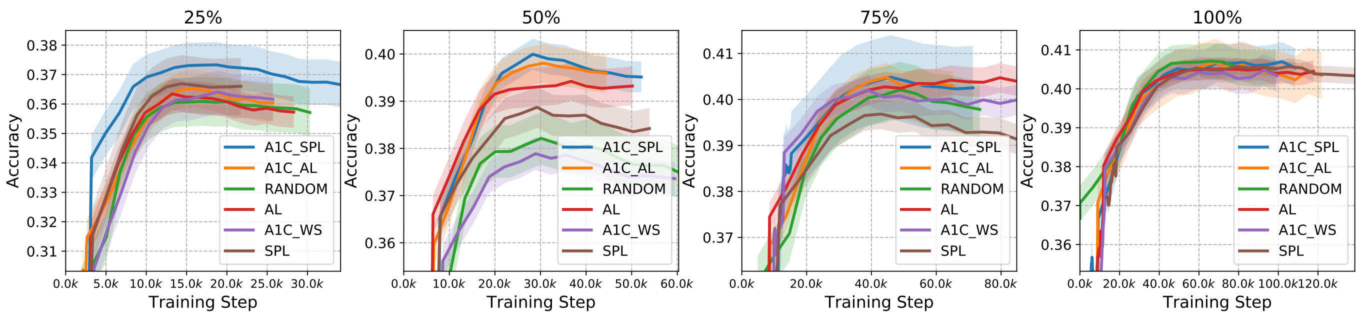
From Fig.5, we draw the following observations. In terms of SAN, FG-A1C-WS can help the model saves 20% on training cost with 75% of training examples, while the performance is reduced by only about 0.9%. With only 50% of training data, the training cost saved by FG-A1C-WL is more significant, *i.e.,* 60%, while the accuracy is still within an acceptable range, *i.e.,* 2.1%. For BUTD, the improvement of training efficiency is still prominent. With 50% and 75% of training data, FG-A1C-SPL achieves a training savings of 50% and 38%, respectively, while the accuracy losses are still small, *i.e.,* 1.6% and 0.5%, respectively. Considering that BUTD is an up-to-date model with a strong performance, these achievements are indeed outstanding.

4.3.2 VQA-CP v2

We further evaluate our learning paradigms on *VQA-CP v2* dataset, which has a different label distribution of training and testing sets. The learning curves and experimental results of all paradigms are shown in Fig.6 and Tab.3. From



(a) SAN



(b) BUTD

Fig. 6. Learning curves of different learning paradigms with different proportions of training examples on VQA-CP v2 dataset.

TABLE 3
Evaluation results of SAN and BUTD with different learning paradigms on the VQA-CP-v2 test split.

SAN	25%				50%				75%				100%			
Method	All	Y/N	Num.	Other	All	Y/N	Num.	Other	All	Y/N	Num.	Other	All	Y/N	Num.	Other
Random	25.6	37.2	10.1	24.0	27.9	38.8	10.5	26.9	29.8	39.4	11.0	29.4	30.2	39.8	11.6	30.3
SPL	25.0	38.2	12.3	20.1	28.6	39.0	9.27	28.4	28.9	39.2	8.0	29.3	30.3	39.1	11.4	30.3
AL	24.3	40.0	14.8	15.0	28.2	38.8	10.3	27.5	29.2	38.5	10.9	28.9	29.5	38.7	11.0	29.1
A1C-SPL	26.5	38.8	10.1	24.6	28.9	37.3	11.5	29.2	30.3	39.2	12.0	30.6	30.4	39.0	11.8	31.0
A1C-AL	25.5	37.4	11.0	22.3	28.6	38.2	11.3	28.3	30.6	39.4	11.0	31.3	30.5	39.1	11.0	31.4
A1C-WS	26.3	38.6	10.2	24.0	29.4	39.2	11.2	28.8	29.6	39.7	10.8	29.1	30.2	39.7	10.8	29.1
BUTD	25%				50%				75%				100%			
Method	All	Y/N	Num.	Other	All	Y/N	Num.	Other	All	Y/N	Num.	Other	All	Y/N	Num.	Other
Random	34.2	40.4	11.5	38.0	37.8	41.1	12.5	43.1	38.5	41.5	12.6	44.1	38.5	41.7	12.7	44.0
SPL	35.2	40.4	11.0	39.2	37.3	41.0	12.1	32.3	39.0	41.9	11.9	45.0	39.2	42.3	12.9	44.7
AL	35.1	40.3	11.5	39.5	37.3	41.1	12.5	42.0	39.0	42.0	12.6	44.7	39.1	42.3	12.9	44.5
A1C-SPL	35.8	40.7	11.5	39.9	38.4	42.2	12.7	42.9	39.7	42.2	12.8	45.1	39.6	41.9	13.2	45.7
A1C-AL	35.4	41.5	12.0	38.7	38.7	41.6	12.8	43.7	40.2	41.9	13.2	45.9	39.4	42.0	12.6	45.2
A1C-WS	35.0	40.1	11.8	38.4	36.8	41.0	12.2	41.3	38.8	42.2	12.5	44.2	39.6	42.7	12.9	45.3

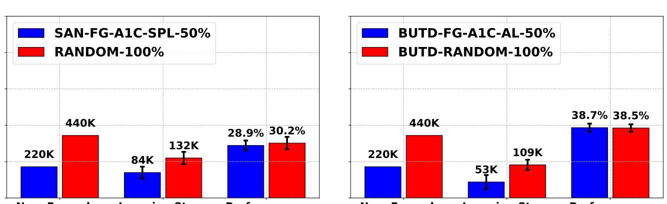
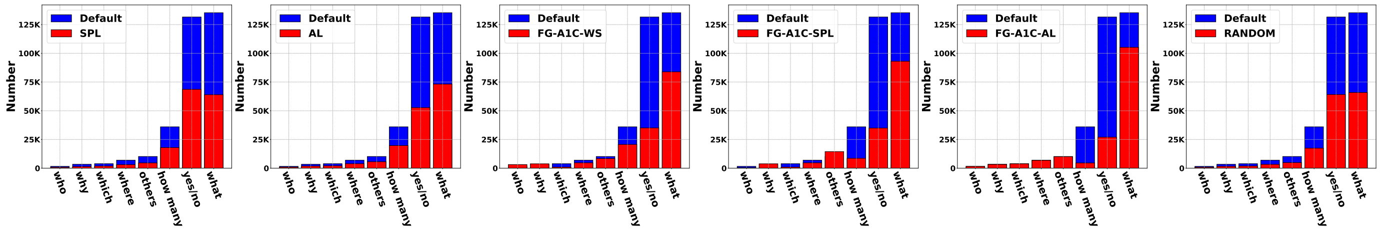


Fig. 7. Comparisons of the training expenditures and the model performance between FG-A1C paradigms and the random sampling scheme on the VQA-CP dataset.

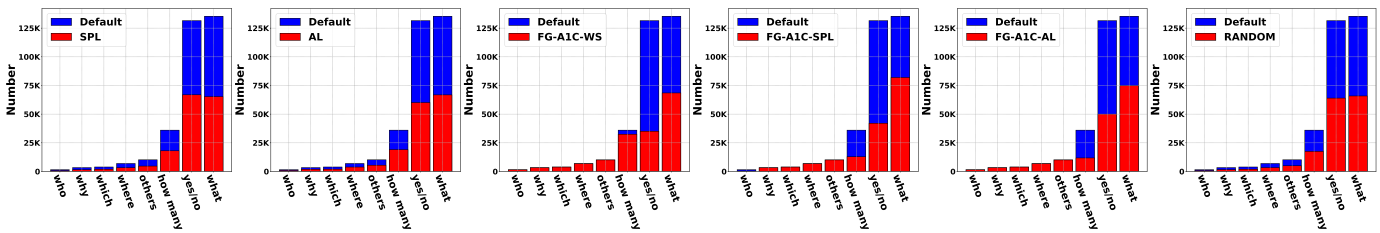
these results, the same conclusion can be drawn that our learning paradigms still shows better ability to improve the model performance and training efficiency than baselines on VQA-CP dataset. Particularly, the performance gains are more significant than those on VQA2.0. For instance, with 25% OF the training data, FG-A1C-SPL achieves about 5% increase in BUTD performance to the *Random* paradigm. Meanwhile, an important observation is that with only 75% of training data, our learning paradigms can help both SAN and BUTD achieve the best performance rather than using all training examples. Considering the different data distributions for training and testing of VQA-CP, these results greatly confirm that our learning paradigms can perceive the learning state of VQA models and select most efficient examples of specific question types to help the model reach the optimal state.

Fig.7 gives the comparisons of training cost and model performance between our learning paradigms and the Random with the full dataset. From this figure, we can still witness the improvements of training efficiency by our paradigms. For instance, FG-A1C-WL can help SAN achieves a 36% training saving with 50% of training ex-

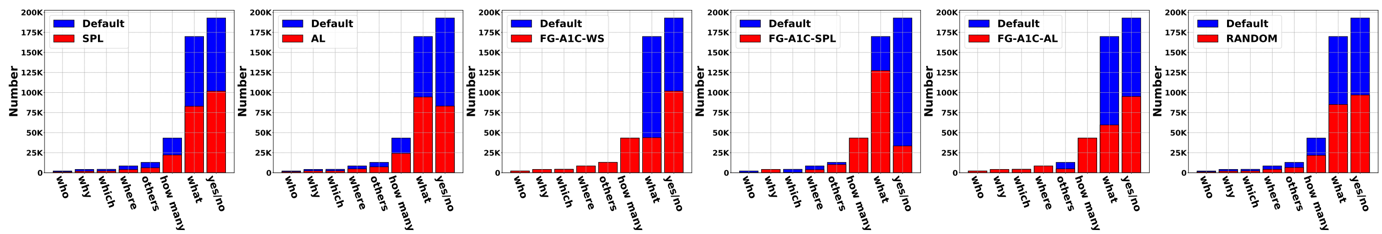
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2956699, IEEE Transactions on Pattern Analysis and Machine Intelligence



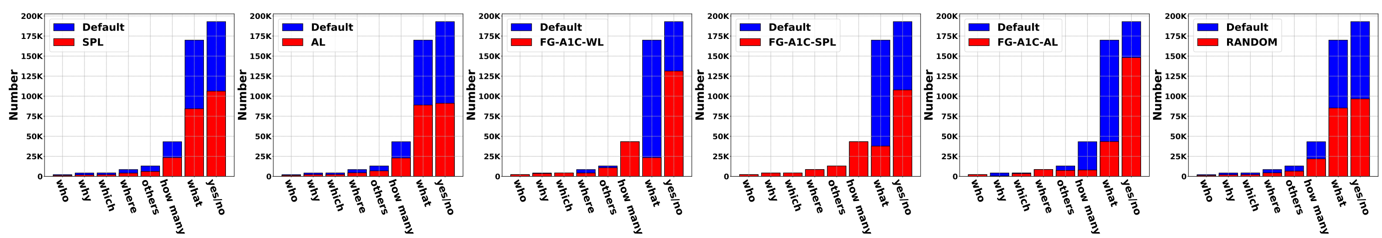
(a) VQA2.0-SAN



(b) VQA2.0-BUTD



(c) VQA-CP-SAN



(d) VQA-CP-BUTD

Fig. 8. Sample distributions of different learning paradigms on the VQA2.0 and VQA-CP v2 datasets.These distributions reflect preferences of different sampling scheme.

amples, while the performance loss is only 1.3 point. For BUTD, FG-A1C-AL saves 52% atraining costs with 50% of the training data, while the model performance is better. A notable difference to VQA2.0 is that both SAN and BUTD reaches the optimal performance by our paradigms with only 75% of training data.

4.4 Sample Distributions

To further analyze the learning paradigms, we visualize their sample distributions in Fig.8. We find out that different paradigms present very distinct sample preferences, some of which are different from our prior knowledge. The sample distributions of *random paradigm* are consistent with the default data distribution of the whole training set. In the case of SAN, SPL presents a favor towards question types with a smaller number of potential answers, like *yes/no*. Its sample distributions also uncover its shortcoming. Concretely, hard questions like "*why*" and "*where*" are barely selected, which fails to obtain sustained growth during SAN training. Under the case of BUTD, the distribution of SPL will be more balanced, and better performance is achieved accordingly. The reason is that, in BUTD, the entropy values of different question types are closer than that in SAN. In contrast to SPL, AL prefers questions that are hard to predict, like "*what*" or "*where*". Such a preference also leads to a problem that the *yes/no* questions are less selected, which occupies a large proportion in the dataset. Compared with the baselines, the sample distribution of FG-A1C-WS is more balanced. Overall, FG-A1C-WS presents a favor towards hard questions, like "*others*" and "*what is*", which are difficult to learn but also beneficial to enhance the accuracy. Meanwhile, it also takes *yes/no* questions into account since they have a high proportion. In sum, FG-A1C paradigms can use the learning agent to perform targeted data augmentations and make a good trade-off between different types of questions, which achieves the best performance by using fewer examples.

Fig.9 displays sampled questions by different learning paradigms. From this figure we can observe that examples

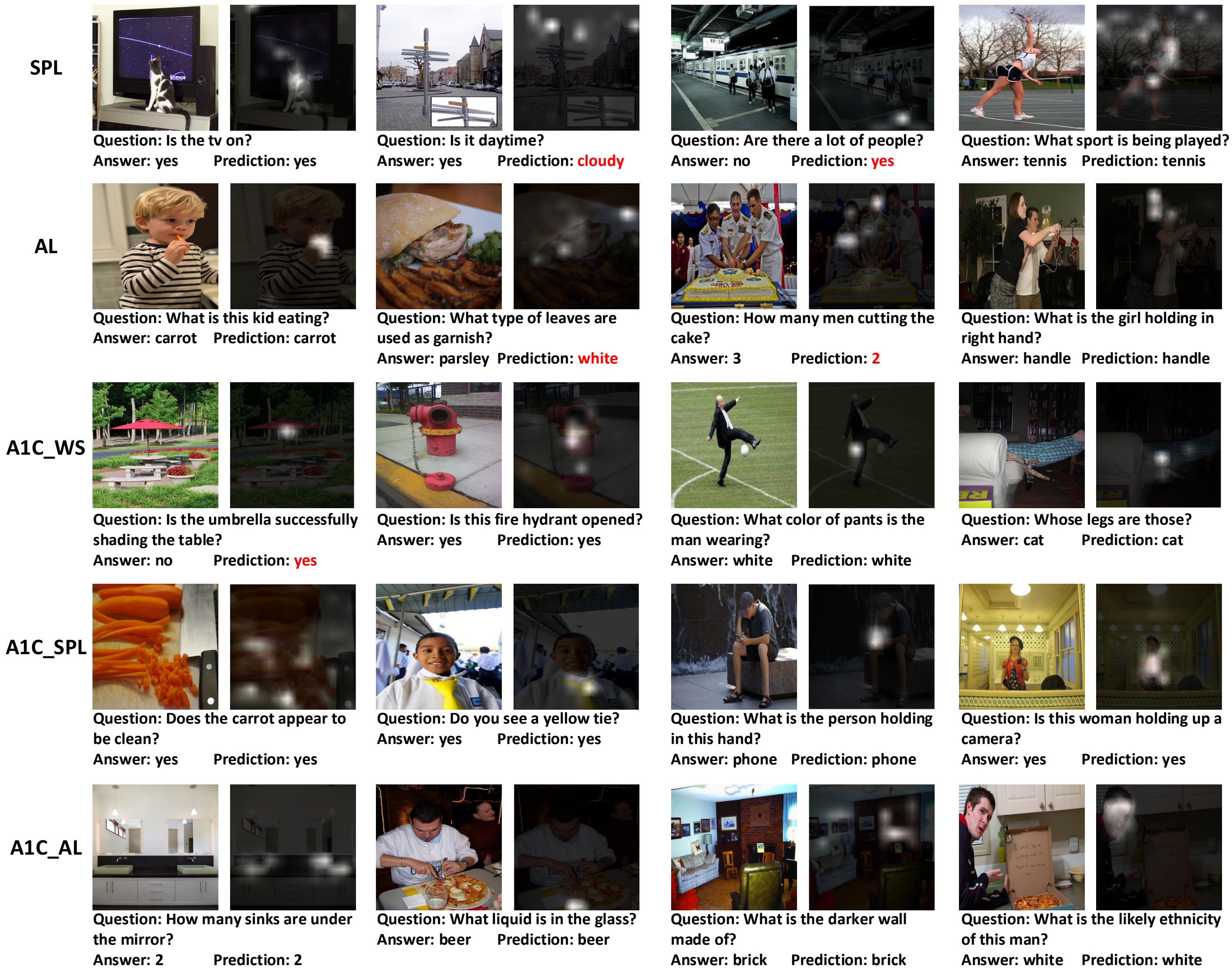


Fig. 9. The sampled questions of different learning paradigms.

sampled by active-learning based methods, *e.g.*, AL and A1C-AL, are relatively more difficult than those sampled by curriculum-learning based methods, *e.g.*, SPL, A1C-SPL and A1C-WS. In easy examples, the structure of the question content is simpler, and the involved tasks are typically identifying objects or recognizing scenes *et al.*, which require less reasoning ability. In terms of hard examples, the question content is more complex, and the corresponding answer entities in images are more difficult to find out. Another observation is that under questions with the same difficulties, models trained by our learning schemes show a better ability to answer predictions, which suggests that our fine-grained learning can help the model improve the ability of question answering more efficiently with limited training examples.

4.5 Guiding Data Labeling

Our learning paradigms can further guide data labeling, since the sampling strategies proposed are all label-free. To validate this argument, we regard the VisualGenome (VG) [50] as an un-labeled VQA dataset, and use the proposed learning paradigms, *i.e.*, FG-A1C-AL, to guide data labeling to improve the performance of BUTD on VQA2.0.

TABLE 4
Evaluations of BUTD on VQA2.0 test-dev with Visual Genome dataset. "VG" denotes the number of Visual Genome examples used. "STEP" denotes the number of the training steps.

Paradigm	VG	STEP	All	Yes/No	Num.	Others
Random* [14]	512K	-	65.3	81.8	44.2	57.3
Random	512K	412K	66.9	83.4	48.6	57.1
FG-A1C-AL	250K	341K	67.0	83.7	47.6	57.2
FG-A1C-AL	150K	227K	67.0	83.3	47.6	57.1
FG-A1C-SPL	250K	240K	67.2	83.9	48.5	57.2
FG-A1C-SPL	150K	227K	67.2	84.0	48.5	57.0

*is the result reported in [14]

Specifically, we follow the setting in [14] to select about half a million examples from visual genome as candidates. These examples are also categorized into eight question types defined in Sec.4.1. For the Random paradigm, we directly augment these VG examples to the training set of VQA2.0. For FG-A1C-AL, we first train BUTD with the training set of VQA2.0 for several epochs, and then perform data sampling after each training interval.

Tab.4 gives the evaluation results of BUTD with different number of VG examples used on VQA2.0 test-dev split. From this table, we can first observe that with less augmented VG examples, FG-A1C-AL can help BUTD achieve a superior performance. Meanwhile, the training expenditures by our paradigm are sill much cheaper than that of traditional training scheme. These results confirms the functionality of guiding labeling of the proposed learning paradigm.

5 Conclusion

In this paper, we have proposed a fine-grained learning paradigm with *actor-critic* learning, termed FG-A1C, towards efficient training of Visual Question Answering. This paradigm aims at solving two practical yet largely unexploited issues in VQA, *i.e.*, *difficulty diversity* and *label redundancy*. Compared to the traditional training paradigm, FG-A1C starts with a few examples, and uses a learning agent to perform targted data augmentations. This learning agent can evaluate the training state of VQA models, and decide which question types should be added to the subsequent training epochs to tackle the difficulty diversity issue. Such target data augmentation can alleviate the "difficulty diversity" issue to a large extent. Meanwhile, we also propose three data selection approaches to decide which samples should be selected from individual question types, which well handles the label redundancy issue. To validate the merits of FG-A1C, we apply it to two most recent VQA models, *i.e.*, SAN [1] and BUTD [14], and conduct extensive experiments on VQA2.0 dataset. Experimental results show that our approach can outperform baselines with different groups of training examples. FG-A1C can help VQA achieve comparable performance with much fewer examples and less training time. Most importantly, it can be seamlessly embedded to the existing VQA models, as well as other learning-related computer vision tasks.

Acknowledgments

This work is supported by the National Key R&D Program (No.2017YFC0113000, and No.2016YFB1001503), Nature Science Foundation of China (No.U1705262, No.61772443, and No.61572410), Post Doctoral Innovative Talent Support Program under Grant BX201600094, China Post-Doctoral Science Foundation under Grant 2017M612134, Scientific Research Project of National Language Committee of China (Grant No. YB135-49), and Nature Science Foundation of Fujian Province, China (No. 2017J01125 and No. 2018J01106).

References

[1] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 21–29, 2016.

[2] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847*, 2016.

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[4] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, volume 14, pages 77–81, 2015.

[5] Y. Gao, O. Beijbom, N. Zhang, and T. Darrell. Compact bilinear pooling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 317–326, 2016.

[6] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In *Advances In Neural Information Processing Systems*, pages 289–297, 2016.

[7] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433, 2015.

[8] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009.

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

[10] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. *arXiv preprint arXiv:1612.00837*, 2016.

[11] M. P. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, pages 1189–1197, 2010.

[12] I. Grondman, L. Busoniu, G. A. Lopes, and R. Babuska. A survey of actor-critic reinforcement learning: Standard and natural policy gradients. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):1291–1307, 2012.

[13] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Learning to compose neural networks for question answering. *arXiv preprint arXiv:1601.01705*, 2016.

[14] D. Teney, P. Anderson, X. He, and A. V. Den Hengel. Tips and tricks for visual question answering: Learnings from the 2017 challenge. *computer vision and pattern recognition*, 2018.

[15] L. Ma, Z. Lu, and H. Li. Learning to answer questions from image using convolutional neural network. *arXiv preprint arXiv:1506.00333*, 2015.

[16] C. Zhu, Y. Zhao, S. Huang, K. Tu, and Y. Ma. Structured attentions for visual question answering. In *Proc. IEEE Int. Conf. Comp. Vis*, volume 3, 2017.

[17] Y. Kim, C. Denton, L. Hoang, and A. M. Rush. Structured attention networks. *international conference on learning representations*, 2017.

[18] B. Patro and V. P. Namboodiri. Differential attention for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7680–7688, 2018.

[19] H. Xu and K. Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. *european conference on computer vision*, pages 451–466, 2016.

[20] Q. Wu, P. Wang, C. Shen, A. Dick, and A. van den Hengel. Ask me anything: Free-form visual question answering based on knowledge from external sources. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4622–4630, 2016.

[21] P. Wang, Q. Wu, C. Shen, and A. v. d. Hengel. The vqa-machine: Learning how to use existing vision algorithms to answer new questions. *arXiv preprint arXiv:1612.05386*, 2016.

[22] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2017.

[23] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, volume 3, page 6, 2018.

[24] Y. Jiang, V. Natarajan, X. Chen, M. Rohrbach, D. Batra, and D. Parikh. Pythia v0. 1: the winning entry to the vqa challenge 2018. *arXiv preprint arXiv:1807.09956*, 2018.

[25] J.-H. Kim, S.-W. Lee, D. Kwak, M.-O. Heo, J. Kim, J.-W. Ha, and B.-T. Zhang. Multimodal residual learning for visual qa. In *Advances in Neural Information Processing Systems*, pages 361–369, 2016.

[26] J.-H. Kim, K.-W. On, J. Kim, J.-W. Ha, and B.-T. Zhang. Hadamard product for low-rank bilinear pooling. *arXiv preprint arXiv:1610.04325*, 2016.

[27] Z. Yu, J. Yu, J. Fan, and D. Tao. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering.

In *Proc. IEEE Int. Conf. Comp. Vis*, volume 3, 2017.

[28] J. Kim, K. W. On, W. Lim, J. Kim, J. Ha, and B. Zhang. Hadamard product for low-rank bilinear pooling. *international conference on learning representations*, 2017.

[29] L. Jiang, D. Meng, S.-I. Yu, Z. Lan, S. Shan, and A. Hauptmann. Self-paced learning with diversity. In *Advances in Neural Information Processing Systems*, pages 2078–2086, 2014.

[30] L. Jiang, D. Meng, Q. Zhao, S. Shan, and A. G. Hauptmann. Self-paced curriculum learning. In *AAAI*, volume 2, page 6, 2015.

[31] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12. Springer-Verlag New York, Inc., 1994.

[32] B. Settles and M. Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1070–1079. Association for Computational Linguistics, 2008.

[33] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.

[34] A. Vlachos. Active learning with support vector machines. *Master of Science School of Informatics University of Edinburgh*, 2004.

[35] I. Misra, R. B. Girshick, R. Fergus, M. Hebert, A. Gupta, and L. V. Der Maaten. Learning by asking questions. *computer vision and pattern recognition*, pages 11–20, 2018.

[36] J. Johnson, B. Hariharan, L. V. Der Maaten, L. Feifei, C. L. Zitnick, and R. B. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. *computer vision and pattern recognition*, pages 1988–1997, 2017.

[37] V. R. Konda and J. N. Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014, 2000.

[38] D. Bahdanau, P. Brakel, K. Xu, A. Goyal, R. Lowe, J. Pineau, A. Courville, and Y. Bengio. An actor-critic algorithm for sequence prediction. *arXiv preprint arXiv:1607.07086*, 2016.

[39] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937, 2016.

[40] Z. Wang, V. Bapst, N. Heess, V. Mnih, R. Munos, K. Kavukcuoglu, and N. de Freitas. Sample efficient actor-critic with experience replay. *arXiv preprint arXiv:1611.01224*, 2016.

[41] Y. Fan, F. Tian, T. Qin, J. Bian, and T. Liu. Learning what data to learn. *arXiv: Learning*, 2017.

[42] P. Bachman, A. Sordoni, and A. Trischler. Learning algorithms for active learning. *international conference on machine learning*, pages 301–310, 2017.

[43] M. Liu, W. L. Buntine, and G. Haffari. Learning how to actively learn: A deep imitation learning approach. pages 1874–1883, 2018.

[44] R. S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, 1988.

[45] D. Precup, R. S. Sutton, and S. P. Singh. Eligibility traces for off-policy policy evaluation. In *Seventeenth International Conference on Machine Learning*, pages 759–766, 2000.

[46] Z. Zhou, J. Shin, L. Zhang, S. Gurudu, M. Gotway, and J. Liang. Fine-tuning convolutional neural networks for biomedical image analysis: Actively and incrementally *. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[47] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[48] A. Agrawal, D. Batra, D. Parikh, and A. Kembhavi. Don't just assume; look and answer: Overcoming priors for visual question answering. *computer vision and pattern recognition*, pages 4971–4980, 2018.

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

[50] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *arXiv preprint arXiv:1602.07332*, 2016.

Yiyi Zhou received the BS and MS degrees from Dalian Jiaotong University of China and Durham University of UK in 2012 and 2014, respectively. He is currently pursuing the Ph.D. degree with Xiamen University. His research interests include visual question answering, multi-modal learning and social network embedding.

Rongrong Ji is currently a Professor, Director of the Intelligent Multimedia Technology Laboratory, and Dean Assistant of the School of Information Science and Engineering, Xiamen University, Xiamen, China. His work mainly focuses on innovative technologies for multimedia signal processing, computer vision, and pattern recognition, with over 100 papers published in international journals and conferences. Prof. Ji is a Member of ACM. He was the recipient of the ACM Multimedia Best Paper Award and Best Thesis Award of Harbin Institute of Technology. He serves as Associate/Guest Editor for international journals and magazines like Neurocomputing, Signal Processing, Multimedia Tools and Applications, the IEEE MultiMedia Magazine, and Multimedia Systems. He also serves as program committee member for several tier-1 international conferences.

Xiaoshuai Sun Xiaoshuai Sun is an assistant professor of School of Computer Science and Technology, Harbin Institute of Technology, China. From Sep. 2015 to Dec. 2016, he has been working as a post-doc research fellow with Prof. Heng Tao Shen at School of Information Technology and Electrical Engineering, the University of Queensland, Australia. He received his doctoral degree from Harbin Institute of Technology in January 2015 under the supervision of Prof. Hongxun Yao. From September 2012 to June 2013, he worked as a research intern in Microsoft Research Asia (MSRA) mentored by Dr. Xin-Jing Wang. His current research interests include deep learning, computer vision and pattern recognition, multimedia content analysis and retrieval.

Jinsong Su Jinsong Su was born in 1982. He received the Ph.D. degree from the Chinese Academy of Sciences, Beijing, China. He is currently an Associate Professor with Xiamen University, Xiamen, China. His research interests include natural language processing and machine translation.

Deyu Meng received his BA and MS degree in Applied Mathematics, and Ph.D degree in Computer Science, all from Xi'an Jiaotong University. Currently he works for Xi'an Jiaotong University as an associate professor. From August 2012 to July 2014, he took my two-year sabbatical leave in Carnegie Mellon University. In August 2012 he joined the Robotics Institute of School of Computer Science, working with Dr. Fernando De la Torre, and in August 2012, he joined Language Technologies Institute of School of Computer Science at Carnegie Mellon University, working with Dr. Alex Hauptmann. His research interests include machine learning and its applications to multimedia content analysis and computer vision.

Yue Gao Yue Gao (SM'14) received the B.S. degree from the Harbin Institute of Technology, Harbin, China, and the M.E. and Ph.D. degrees from Tsinghua University, Beijing, China.

Chunhua Shen is a Professor at School of Computer Science, University of Adelaide. He is a Project Leader and Chief Investigator at the Australian Research Council Centre of Excellence for Robotic Vision (ACRV), for which he leads the project on machine learning for robotic vision. Before he moved to Adelaide as a Senior Lecturer, he was with the computer vision program at NICTA (National ICT Australia), Canberra Research Laboratory for about six years. His research interests are in the intersection of computer vision and statistical machine learning. He studied at Nanjing University, at Australian National University, and received his PhD degree from the University of Adelaide. From 2012 to 2016, he holds an Australian Research Council Future Fellowship.