Question-Guided Semantic Dual-Graph Visual Reasoning with Novel Answers

Xinzhe Zhou, Yadong Mu*
{zhouxinzhe1023,myd}@pku.edu.cn
Wangxuan Institute of Computer Technology, Peking University, Beijing, China

ABSTRACT

Visual Question Answering (VQA) has gained increasing attention as being the cross-disciplinary research of computer vision and natural language understanding. However, recent advances mostly treated it as a closed-set classification problem, by limiting the possible outputs to some fixed frequent answers available in a training set. Although effective on benchmark datasets, this paradigm is inherently defective—the VQA model would always fail on a question whose correct answer is out of the answer set, which severely hampers its generalization and flexibility. To try to close the gap, we explore an open-set VQA setting, where models are evaluated using novel samples with unseen answers given dynamic candidate answers from some candidate-generation module. For experimental purposes, two oracle candidate-sampling strategies are proposed to serve as a proxy for the candidate-generation module and generate dynamic candidate answers for testing samples. The conventional classification-based paradigm is no longer applicable in our setting. To this end, we design a matching based VQA model, in which a novel Single-Source Graph Convolutional Network (SS-GCN) module is designed to jointly leverage question guidance and dual semantic answer-graphs to produce more discriminative and relevant answer embeddings. Extensive experiments and ablation studies by re-purposing two benchmark datasets demonstrate the effectiveness of our proposed model.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision tasks**; **Natural language processing**; • **Information systems** → **Multimedia information systems**.

KEYWORDS

visual question answering, visual reasoning, graph convolutional network

ACM Reference Format:
Xinzhe Zhou, Yadong Mu. 2021. Question-Guided Semantic Dual-Graph Visual Reasoning with Novel Answers. In *Proceedings of the 2021 International Conference on Multimedia Retrieval (ICMR '21), August 21–24, 2021, Taipei, Taiwan.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3460426.3463647

*corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ICMR '21, August 21–24, 2021, Taipei, Taiwan
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8463-6/21/08…$15.00
https://doi.org/10.1145/3460426.3463647

1 INTRODUCTION

In the past years, the advancement of deep learning has greatly promoted the development of many fields, especially computer vision [13, 17] and natural language understanding [7, 29]. Based on the success of single-modality tasks, integrating multi-modality information is now seen as the next step towards general artificial intelligence. As the fusion of vision and natural language, visual question answering (VQA) [3] has gained more and more attention, and many researchers have explored a variety of approaches for it [1, 3, 5, 8, 10, 11, 14, 18, 21, 24, 25, 30, 36].

Despite the improvement on existing VQA datasets [3, 12, 16], most previous VQA models are inherently defective—they limit all possible outputs to some fixed frequent answers available in a training set, and treat VQA simply as a closed-set classification problem [1, 3, 5, 8, 10, 11, 18, 21, 24, 30, 36]. This paradigm was proven to be effective for improving accuracy on particular datasets, but its drawback is obvious—the VQA model would always fail on a question whose correct answer is out of the answer set, which severely hampers its generalization.

Several previous works paid attention to the generalization issue [2, 19, 31], and proposed different evaluation protocols. But they mostly focused on generalization to novel words [19, 31] or unseen compositions of seen concepts [2]. Few of them considered the more challenging open-set setting, where models are faced with novel samples with unseen answers. To try to close the gap and promote subsequent researches, we explore the open-set scenario in this paper. Different from conventional VQA where training and evaluation samples are i.i.d. drawn from the same distribution, we construct a testing set with all answers unseen during training to evaluate the model generalization to open-set samples[1].

We suggest that an ideal pipeline for this setting should contain two stages—first a candidate-generation stage to generate several possible answers based on pre-analysis of the image and question, and then an answering stage to choose the best answer among them. Since our focus in this work is more on the second stage, i.e., designing more generalizable answering models, we adopt an oracle approach for the first stage. Specifically, we propose two candidate-sampling strategies to simulate the answer-generation module and provide dynamic candidate answers for testing samples.

In order to perform well in the open-set setting, the VQA model is required to handle novel samples with dynamic and unseen candidates, so the conventional closed-set classification paradigm is

[1]Limited by the datasets for experiments, the testing answers are actually finite and not really open-set, but VQA models would not be aware of this and could not utilize any information about testing answers for training, unlike in some zero-shot learning setting [33], so this simulates the real open-set situation.

no longer applicable. To this end, we design a novel VQA model that selects the answer based on matching instead of classification. Our model conducts VQA by projecting image-question pairs and candidate answers to the same embedding space, and predicts the answer feasibility by computing an embedding correlation score. Previous works like [14] also adopted such a matching paradigm, but they learned the embedding for each answer separately, and purely from the answer word-embedding. Such an embedding module ignores the rich contextual information contained in the answer-relationship and the question, thus leading to significant performance degradation for novel samples, as shown in Table 5 of [14]. We propose to integrate these two aspects of information in a novel Single-Source Graph Convolutional Network (SSGCN) module, to help produce more discriminative and context-relevant answer embeddings.

To validate the effectiveness of our design, we conduct extensive experiments by re-purposing two benchmark datasets, VQA [3] and VQAv2 [1]. The results consistently demonstrate the superiority of our proposed model.

Our main contributions can be summarized below:

1) We explore a novel open-set VQA setting as a step towards a more realistic situation. Our setting extends conventional VQA which leads to sub-optimal models by treating VQA as a closed-set classification problem, and thus lacks the capability of generalizing beyond training answers.

2) We design a novel VQA model based on matching, in which a special Single-Source Graph Convolutional Network(SSGCN) module is proposed to jointly utilize question guidance and dual semantic answer-graphs to produce more discriminative and context-related answer embeddings.

3) Extensive experiments and ablation studies by re-purposing two VQA datasets demonstrate the effectiveness of our model.

2 RELATED WORK

Free-form and open-ended Visual Question Answering (VQA) was proposed in [3] as the fusion of vision and natural language. The authors [3] released the VQA dataset, together with several baseline models. However, in the original experiments [3], the authors adopted the top-K frequent-answer classification paradigm, which affected subsequent works to follow it and keep its pipeline up to now [1, 5, 8, 10, 11, 18, 21, 24, 30, 36]. The research focus of previous works on VQA was mainly about designing more effective multi-modality integration mechanisms [5, 8, 10, 11, 18, 21, 24, 30, 36], while only a few works tried to explore beyond the closed-set classification paradigm [14, 25, 34].

In [34], the authors used an LSTM to generate answers instead of selecting from a pre-defined answer set. Although more flexible compared to direct answer classification, the generation process is still a sequential closed-set classification task over a pre-defined vocabulary, so the model is still prone to fail on open-set samples where the correct answers (words) are not encountered before.

Our work is more similar to [14]. The authors proposed an answer embedding module that can generalize beyond training answers. But they only explored cross-dataset transfer learning within a closed-set framework, and their embedding module ignored the rich contextual information contained in the answer-relationship and the question, which led to significant performance degradation on novel samples, as shown in Table 5 of [14].

Recently, the authors of [25] proposed to utilize off-the-shelf visual and linguistic data to cope with novel answers. They relied on additional visual data corresponding to all possible answers / concepts to pretrain the answer classifier, which is expensive to employ since collecting visual data for every answer is barely practical. In our matching based pipeline, an answer could be handled as long as we extract its word-embedding and relationship with other candidates, which is more easier to obtain.

There are also several works exploring changing the VQA setting to evaluate the model generalization ability [2, 19, 31]. In [31], the authors proposed a zero-shot setting for multi-choice VQA, to evaluate the model generalization to samples with unseen words. In [2], the authors focused on generalization to unseen compositions of seen concepts, and proposed a compositional split of the VQA dataset [3] for evaluating the compositionality of models. The authors of [19] drew inspiration from the human's ability to transfer knowledge from the input (i.e., reading and listening) to the output (i.e., writing and speaking), and proposed the Zero-Shot Transfer VQA(ZST-VQA) dataset to explicitly evaluate the knowledge-transfer performance from question (answer) to answer (question). Different from our work, the focuses of these works are either on handling novel words or handling novel compositions, while the open-set setting of our concern is more challenging in the sense that both novel factors exist in the unseen answers.

3 METHODS

We first give details of the proposed open-set setting in Section 3.1, and then elaborate on our model and the training strategy in Sections 3.2 and 3.3.

3.1 Open-Set VQA

Formally, for a VQA dataset $\mathcal{D}_{all} = \{(I_1, Q_1, A_1), (I_2, Q_2, A_2), \ldots, (I_N, Q_N, A_N)\}$, where I, Q, and A stand for image, question, and ground-truth answer, respectively; N is the total number of all samples. We collect all answers as $\mathcal{A}_{all}$, and divide $\mathcal{A}_{all}$ into three disjoint sets $\mathcal{A}_{tr}$, $\mathcal{A}_{val}$, and $\mathcal{A}_{te}$; samples are also partitioned according to their ground-truth answers as $\mathcal{D}_{tr}$, $\mathcal{D}_{val}$, and $\mathcal{D}_{te}$, i.e., we constrain them to have disjoint answers explicitly. Models are trained on $\mathcal{D}_{tr}$ and validated on $\mathcal{D}_{val}$, with no knowledge about $\mathcal{D}_{te}$ or $\mathcal{A}_{te}$. Evaluation on $\mathcal{D}_{te}$ is the same as conventional VQA—providing each (I_i, Q_i) pair and comparing the model output to the ground-truth answer A_i. We call this setting open-set because the answer set during evaluation, $\mathcal{A}_{te}$, is totally unknown during training, and is beyond the closed set $\mathcal{A}_{tr} \cup \mathcal{A}_{val}$.

To handle the open-set evaluation on $\mathcal{D}_{te}$, we suggest a two-stage pipeline. Ideally, when the model is given (I_i, Q_i), another special module should be run first to generate a set of candidate answers $A_{i,s}^{cand}$, utilizing some internal knowledge bases and auxiliary tools (e.g., image-analysis model); here s denotes the size of the candidate set, i.e., $|A_{i,s}^{cand}| = s$, which could be affected by user-preference for the answer granularity, domain, etc. Then at the second stage, the VQA model / the answering module chooses from $A_{i,s}^{cand}$ a best prediction as output.

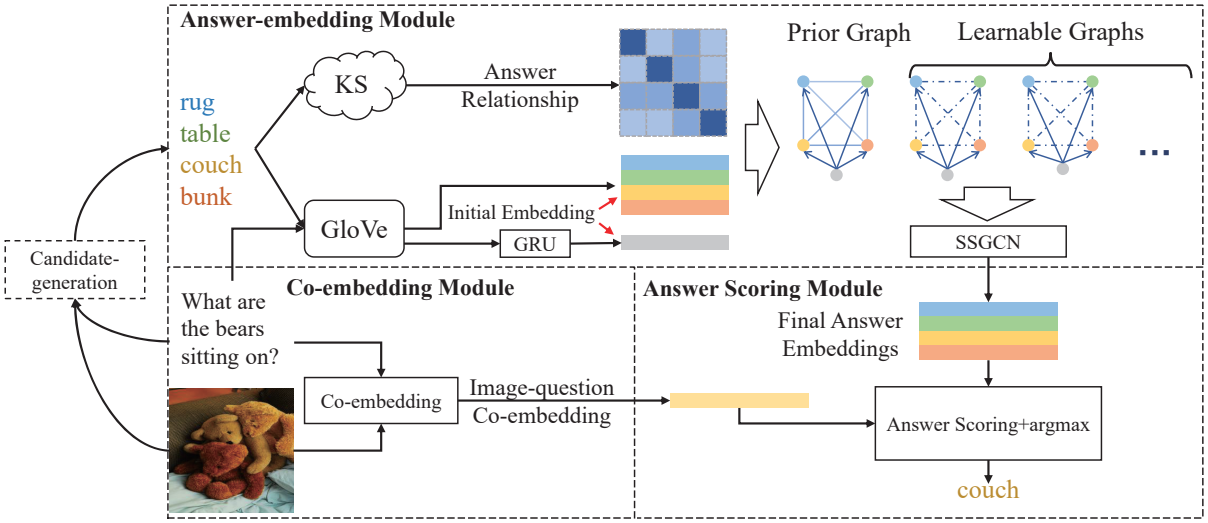


Figure 1: The structure of the whole model. Given an input image, a question, and a candidate answer set, first the co-embedding module integrates the image and question to produce a joint embedding feature, and then the novel SSGCN module generates embeddings for all answers, utilizing both the dual semantic answer-graphs and the question guidance. Finally, the answer scoring function computes the feasibility score of each answer based on the correlation of embeddings, and a simple argmax gives the final output answer.

As stated above, the focus of this work is on designing more generalizable answering models to handle novel samples given dynamic and unseen candidate answers, while developing an effective and efficient answer-generation module is out of our scope and would be left for future work. Therefore, in all our experiments, we simplify the first stage and design two oracle strategies to generate candidate answers for each question.

The first strategy is to simply randomly select some answers from $\mathcal{A}_{te}$ and group them with the correct answer (known to the oracle but not to the model) as a candidate set. This strategy is easy to implement, but the resultant answer sets may contain candidates with irrelevant semantics, so are not challenging enough. To attempt to address this, we propose another semantic candidate-sampling strategy. Specifically, we extract the pre-trained *GloVe* word-embeddings [27] for each answer (average for multi-words answers), and construct an answer set of size s for a particular question by fetching the top-$2s$ nearest neighbors of its ground-truth answer in $\mathcal{A}_{te}$, using cosine distance of the embedding, and then randomly select $s-1$ wrong answers from the neighbors to form the candidate set together with the ground-truth. Note that we keep some randomness in the semantic strategy to simulate the unpredictability in practice. We measure the performance using both sampling strategies with different candidate sizes s in Section 4 .

3.2 Matching-based VQA Model

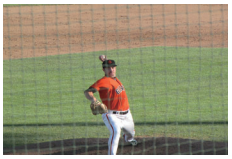
To handle open-set samples given candidate answers, we design a matching-based VQA model as the answering module. Our model mainly consists of three modules: an attention-based image-question co-embedding module, a question-guided semantic dual-graph answer-embedding module, and an answer scoring function. The structure is shown in Figure 1.

Attention-based Image-question Co-embedding Module. As the common practice in VQA, the input from two modalities should be integrated together first in order to align their semantics and fuse the features. The integration mechanisms have been extensively studied by previous works [5, 10, 11, 18, 21, 24, 30, 36]. Since we mainly concern about generating a better answer embedding for the open-set condition in this work, basically any co-embedding module could be used here. We choose the structure of [30] for our model as well as all baselines due to its simplicity and effectiveness. We refer readers to [30] for the details. The output of this module is a single feature $\tilde{\boldsymbol{h}}$, which encodes the image and question jointly.

Question-guided Semantic Dual-graph Answer-embed Module. After encoding the image-question pair, we then proceed to embed all candidate answers to the same space in order to conduct correlation-based matching to select answers. As stated above, we propose to utilize the semantic dual answer-graphs and the question guidance in this embedding process to help generate more discriminative and semantics-relevant embeddings. The details are given below.

Given a candidate set $A^{cand}_{i,s}$, we first construct a graph over all answers in it. An answer-graph connects every two candidate answers with a weighted edge, which reflects the semantic relevance between them. Such relevance information could help embed all candidates jointly instead of separately as in [14], so as to properly adjust the whole embedding distribution to better preserve semantics and underline their differences. To fully utilize the rich semantic relationship between answers, we propose to integrate two types of graphs.

Q: What is the cat putting his face in?
A: Pitcher
(Meaning: A large clay container with a small opening and one or two handles)



Q: What position is this man playing?
A: Pitcher
(Meaning: The player who throws ball to batter)

Figure 2: Examples showing the answer ambiguity. The same answer *"pitcher"* means quite differently for the two questions. Samples are chosen from the VQAv2 [1] dataset.

First, a prior graph with fixed connection weights. Formally, let $\mathcal{G}_{A,p} = \{\mathcal{N}, \mathcal{E}\}$ denotes the prior graph, where each node $n_i \in \mathcal{N}$ denotes one answer and an edge between two nodes $(n_i, n_j) \in \mathcal{E}$ has a weight $e(n_i, n_j)$ representing the semantic similarity score of them. We retrieve the similarity scores of answers from some external Knowledge Source (KS). The implementation details are deferred to Section 4.2 for clarity.

Then, complementary to the external knowledge, we build another group of learnable graphs $\mathcal{G}_{A,l_m} = \{\mathcal{N}, \mathcal{E}\}(l_m \in \{1, 2, 3, \ldots, head\_num\})$ to discover more latent relationships contained in the data. We use the multi-head attention mechanism [32] to project all answers into multiple semantic spaces, and then compute correlations in each space as edge weights to construct different graphs. The detailed implementations are elaborated later.

Besides the answer-graphs, we also propose to integrate question information as guidance for answer embedding. Most previous works either consider question together with the image [5, 10, 11, 14, 18, 21, 24, 30, 36], or directly fuse question, image, and answer to score the feasibility [8, 15], while the role of question as the semantic context to guide the answer embedding is largely unexplored. It is intuitive that the question context could help filter and disambiguate answers. For example, for the question *"What fruits are pictured on the wall?"*, answers like *"dog"* or *"electricity"* would rarely be correct, so their embeddings would tend to be far from the input embedding no matter what images are given. In addition, as shown in Figure 2, for the same word *"pitcher"*, it is the correct answer for both questions *"What position is this man playing?"* and *"What is the cat putting his face in?"*, but their meanings are totally different. Such semantic ambiguity is ubiquitous in natural language, and could be alleviated considering question as the context. Even for answers with less ambiguity, the question could also help produce more context-relevant embeddings.

To jointly utilize answer relationship and question guidance, we incorporate question Q into the above answer-graphs in a unified way. To this end, we create another node n_q to represent Q. In order that the question could guide the answer embedding globally, we connect n_q to all answer nodes n_i with directed edges from n_q to n_i. Following the terminology of graph theory, n_q is called a source, and the corresponding new graph $\mathcal{G}_{A+Q,p}$ / $\mathcal{G}_{A+Q,l_m}$ is a single-source graph.

Based on the above construction, we design a Single-Source Graph Convolutional Network (SSGCN) module to evolve all the nodes in a unified framework. First, we initialize all the nodes. The initial feature $\boldsymbol{f}_i^0$ for each answer node n_i is extracted from the pre-trained word-embedding model *GloVe* [27], with simple average for multi-word answers. For the question node n_q, we use a learnable GRU to aggregate the *GloVe* embeddings [27] of all tokens into one feature $\boldsymbol{f}_q^0$. After initialization, our SSGCN proceeds as Eqns. (1)-(8):

$$\boldsymbol{a}_{i,1}^{t+1} = \sum_{j \in \mathcal{N}_e(i)} \frac{e(n_i, n_j)}{|\mathcal{N}_e(i)|} W_a^{t+1} \boldsymbol{f}_j^t, \tag{1}$$

$$\boldsymbol{a}_{i,2}^{t+1} = \text{MultiHeadAttentionalAggregation}_a(\boldsymbol{f}_i^t, \boldsymbol{f}_{\{j\}}^t), \tag{2}$$

$$\boldsymbol{a}_{i,4}^{t+1} = \text{MultiHeadAttentionalAggregation}_q(\boldsymbol{f}_q^t, \boldsymbol{f}_{\{j\}}^t), \tag{3}$$

$$\boldsymbol{a}_{i,3}^{t+1} = W_q^{t+1} \boldsymbol{f}_q^t, \tag{4}$$

$$\boldsymbol{a}_{i,5}^{t+1} = W_s^{t+1} \boldsymbol{f}_i^t, \tag{5}$$

$$\tilde{\boldsymbol{a}}_i^{t+1} = \text{LayerNormalization}(\sum_{k=1}^{5} \boldsymbol{a}_{i,k}^{t+1}), \tag{6}$$

$$\boldsymbol{a}_i^{t+1} = \text{LayerNormalization}(\text{ResidualUnit}(\tilde{\boldsymbol{a}}_i^{t+1})), \tag{7}$$

$$\boldsymbol{f}_q^{t+1} = \text{ResidualUnit}(\boldsymbol{f}_q^t). \tag{8}$$

Eqns. (1)-(7) evolve answer node n_i by aggregating features from the dual-graph and question. Eqn. (1) aggregates neighboring node features on the prior graph $\mathcal{G}_{A,p}$. $\mathcal{N}_e(i)$ represents all neighboring answer nodes of n_i with positive weights $e(n_i, n_j)$. W_a^{t+1} is a shared feature transformation matrix for all answer nodes. Eqn. (2) corresponds to the multi-head attention mechanism to generate multiple learned graphs $\mathcal{G}_{A,l_m}$ and aggregate features on them. The detailed process is shown in Eqns. (9)-(12)—we project every answer node embedding in time t, $\boldsymbol{f}_i^t$, to multiple key / value spaces (Eqns. (9)(10)), and aggregate values for each node with different key-induced connection weights (Eqn. (11)). All key-induced weights from the l_m-th key-value space forms the learned graph $\mathcal{G}_{A,l_m}$ implicitly. All aggregated values for each node on different learned graphs are finally concatenated and further transformed to one feature as Eqn. (12). *key_dim* is the dimension of all keys k_{i,l_m}^{t+1}. $\boldsymbol{x} \cdot \boldsymbol{y}$ means the dot-product between two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. After that, Eqn. (3) utilizes the question node to generate a global answer context by projecting it to the above key-value spaces and aggregating values similar to the above. Eqn. (4) further transforms the question embedding with matrix W_q^{t+1} and directly feeds it to all answer nodes as question context. Eqn. (5) serves as an approximate skip-connection to supply information from each node itself in the last step. For $t = 0$, W_s^{t+1} is a normal linear transformation matrix to account for the change of dimensions. For $t > 0$, W_s^{t+1} is fixed to the identity matrix and thus becomes the standard skip-connection [13]. Eqns. (6)(7) finally aggregate all above information with further transformation. LayerNormalization$(\cdot)$ is proposed in [4] and we use it to accelerate and stabilize training. ResidualUnit$(\cdot)$ [13] is implemented as $\text{ResidualUnit}(x) = W_2\text{ReLU}(W_1 x + b_1) + b_2 + x$. Eqn. (8) is a self-evolving operation for the question node, since we think that the question is the source input and should not depend

on others.

$$\boldsymbol{k}_{i,l_m}^{t+1} = W_{k,l_m}^{t+1} \boldsymbol{f}_i^t, \tag{9}$$

$$\boldsymbol{v}_{i,l_m}^{t+1} = W_{v,l_m}^{t+1} \boldsymbol{f}_i^t, \tag{10}$$

$$\tilde{\boldsymbol{v}}_{i,l_m}^{t+1} = \sum_{j=1}^{|\mathcal{N}|} \text{Softmax}_j\left(\frac{\boldsymbol{k}_{i,l_m}^{t+1} \cdot \boldsymbol{k}_{j,l_m}^{t+1}}{\sqrt{key\_dim}}\right) \boldsymbol{v}_{j,l_m}^{t+1}, \tag{11}$$

$$\boldsymbol{a}_{i,2}^{t+1} = W_c^{t+1}[\tilde{\boldsymbol{v}}_{i,1}^{t+1}, \tilde{\boldsymbol{v}}_{i,2}^{t+1}, \ldots, \tilde{\boldsymbol{v}}_{i,head\_num}^{t+1}]. \tag{12}$$

After T times of evolution, we output the answer embedding $\boldsymbol{f}_{a,i} = W_o \boldsymbol{f}_i^T$.

Answer Scoring Function. After the above two modules, we acquire the image-question co-embedding $\tilde{\boldsymbol{h}}$ and all answer embeddings $\boldsymbol{f}_{a,i}$. Then this function predicts the feasibility score for each answer with Eqn. (13), where $\sigma(\cdot)$ stands for a sigmoid function. The final answer is simply obtained by $\arg\max_i \hat{s}_i$:

$$\hat{s}_i = \sigma(\tilde{\boldsymbol{h}}^T \boldsymbol{f}_{a,i}). \tag{13}$$

3.3 Loss

During training, to encourage the model not to be dependent on the closed-set $\mathcal{A}_{tr}$, instead of computing a classification loss over the whole $\mathcal{A}_{tr}$, we simulate the evaluation condition where answer sets are not fixed. To this end, for each iteration, we sample a random mini-batch $\mathcal{B} = \{I_1, Q_1, A_1), (I_2, Q_2, A_2), \ldots, (I_{|\mathcal{B}|}, Q_{|\mathcal{B}|}, A_{|\mathcal{B}|})\}$, and collect all the answers as $\mathcal{A}_{\mathcal{B}} = \bigcup_{i=1}^{|\mathcal{B}|} \{A_i\}$. The size of $\mathcal{A}_{\mathcal{B}}$ is not fixed due to answer overlapping and multi-answer samples, which partially simulates the dynamic condition. Based on this paradigm, following the suggestions of [30], we use a multi-class binary cross-entropy loss with soft target scores $s_{b,i}$ for the b-th sample and the i-th answer. The loss function is thus:

$$\mathcal{L} = -\sum_{b=1}^{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{A}_{\mathcal{B}}|} (s_{b,i} \log \hat{s}_{b,i} + (1 - s_{b,i}) \log(1 - \hat{s}_{b,i})). \tag{14}$$

The soft score is defined up to different datasets. For example, in the VQA [3] and VQAv2 [12] datasets, 10 persons have provided answers for one question, and the soft score of an answer is defined as $\min(1, \frac{\#\text{ humans that provided that answer}}{3})$.

4 EXPERIMENTS

4.1 Datasets

We re-purpose the VQA [3] and VQAv2 [12] datasets for our open-set experiments.

VQA [3] is based on the MSCOCO [20] images with manually collected questions and answers. The training, validation and evaluation splits contain 248,349, 121,512 and 244,302 samples, respectively. 10 answers from 10 human annotators are collected for each question, but annotations for only the former two sets are provided, so we conduct experiments only based on these two sets.

VQAv2 [12] expands the VQA [3] dataset by collecting complementary samples, which makes the dataset more balanced. The training, validation and evaluation splits contain 443,757, 214,354 and 447,793 samples, respectively. Same as above, only the training and validation annotations are used in our experiments.

Both datasets are re-organized as Section 3.1 described. Some example processed data together with the sampled candidates using both sampling strategies are shown in Figure 3. As can be seen that the semantic strategy in general provides many reasonable distractors when contrasted with the random strategy. For example, for the similar questions about "color", the random strategy provides "crane" as the distractor, which is not even valid for the question, while the semantic strategy provides a more feasible "yellow and red". This is mainly because feasible distractors are likely to be close to the correct answer in semantics, thus would be more probable to be sampled by the semantic strategy since it takes semantic similarity into consideration, while the random strategy totally ignores this.

Compared with the similar multi-choice VQA setting (MC-VQA) [3] which also provides candidates for each question, their candidates are partially annotated by humans, which is labor-intensive, while our strategies are fully automatic so are more scalable. Another difference from the MC-VQA is that all our candidates are unseen during training, and the candidate size is dynamic instead of fixed as in MC-VQA, which makes our setting more close to reality.

4.2 Implementation Details

We describe details for building the prior graph utilizing external KS here.

For constructing the prior graph, we need to extract the semantic similarity between answers. To this end, we explore two different approaches: the first one is based on answer word-embeddings extracted from *GloVe* [27]; specifically, we compute cosine similarity of every two answer word-embeddings, and set the weight of the corresponding edge to it. *GloVe* embedding has been proven to be semantic, and performs well in the word analogy and word similarity tasks [27], so building graph based on it should be reasonable for our purpose.

An alternative approach is to utilize the lexical database WordNet [23], since it describes the word structure based on semantic meaning. We compute the similarity score of every two answer phrases using the method of [22], on basis of *path* [28] and *wup* [35] word similarity metrics. We represent models with these three graphs as SSGCN-G, SSGCN-Wp, and SSGCN-Ww, respectively.

4.3 Competing Methods

Since our open-set VQA evaluates models with unseen answers, most previous VQA models are not suitable since they follow the closed-set classification paradigm and would be always wrong. One exception is [14], as introduced in Section 1; they explored a matching-based model similar to ours, so we use their model, AnsEmbed, as one important competing method. Besides, considering the similarity of our setting to zero-shot learning (ZSL), we also choose two widely-used ZSL methods, DeViSE [9] and ConSE [26], on basis of the conventional VQA model [30] for comparison. There is another type of model that could be used here—the (image, question, answer) triplet-scoring based multi-choice VQA model, and we also compare with one representative model, TriScore [15].

4.4 Results and Analyses

Evaluation metric. Following the convention of VQA [3] and VQAv2 [12], 10 human annotators have provided answers for every question, and the feasibility score of any answer is defined as

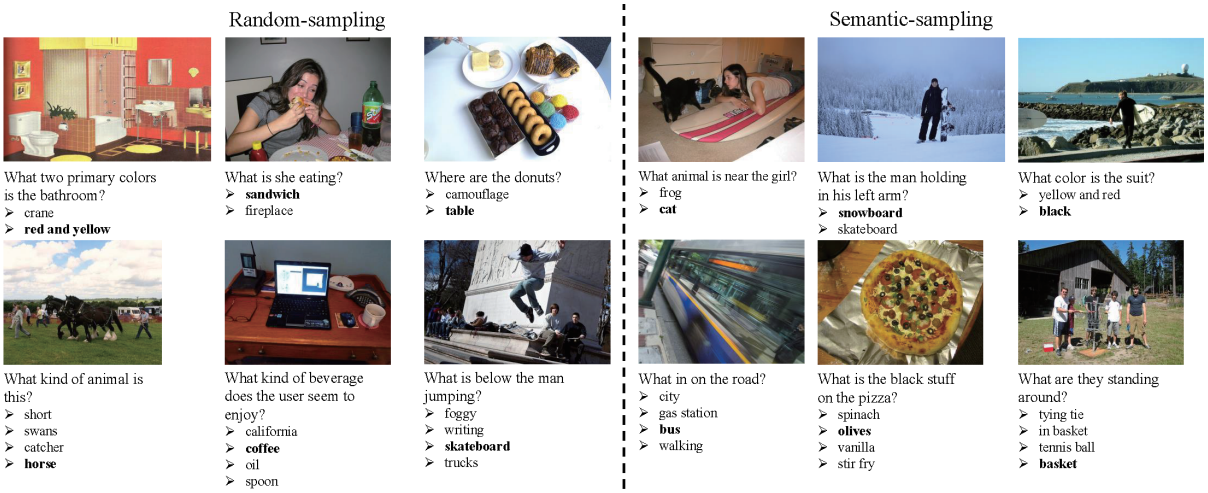


Figure 3: Some examples of the evaluation data together with the generated candidates. Data with different candidate sizes 2 and 4 under two sampling strategies are shown. The most feasible answers are marked in bold.

Table 1: Evaluation results under the OS-VQA setting on the VQA dataset, with pre-defined candidate sizes $\{2, 4, 8, 16, 32, 64, 128, 170\}$. Random candidate-sampling is used for the table above and semantic candidate-sampling for the table below. Numbers in bold are the best results, and numbers in blue are the best single-model results.

Model	2	4	8	16	32	64	128	170
ConSE [26]	0.9027	0.8251	0.7498	0.6828	0.5936	0.4976	0.3881	0.3403
DeViSE [9]	0.8662	0.7676	0.6903	0.6220	0.5558	0.4976	0.4404	0.4183
TriScore [15]	0.9345	0.8796	0.8133	0.7417	0.6607	0.5662	0.4773	0.4451
AnsEmbed [14]	0.9384	0.8902	0.8350	0.7746	0.7070	0.6308	0.5541	0.5216
SSGCN-G	0.9444	0.9074	0.8601	0.8114	0.7569	0.6935	0.6297	0.6007
SSGCN-Wp	0.9460	0.9092	0.8665	0.8166	0.7547	0.6879	0.6134	0.5828
SSGCN-Ww	0.9412	0.9083	0.8639	0.8135	0.7530	0.6883	0.6189	0.5897
SSGCN-3E	**0.9536**	**0.9237**	**0.8881**	**0.8429**	**0.7919**	**0.7323**	**0.6652**	**0.6380**
Model	2	4	8	16	32	64	128	170
ConSE [26]	0.6874	0.5614	0.4977	0.4637	0.4593	0.4531	0.3864	0.3408
DeViSE [9]	0.7800	0.6650	0.5977	0.5444	0.5068	0.4778	0.4410	0.4189
TriScore [15]	0.8237	0.7237	0.6482	0.5892	0.5609	0.5321	0.4786	0.4462
AnsEmbed [14]	0.8639	0.7745	0.7131	0.6624	0.6282	0.6049	0.5555	0.5219
SSGCN-G	0.8798	0.7957	0.7530	0.7161	0.6853	0.6710	0.6258	0.6011
SSGCN-Wp	0.8828	0.8133	0.7613	0.7080	0.6823	0.6626	0.6133	0.5828
SSGCN-Ww	0.8829	0.8137	0.7630	0.7177	0.6844	0.6646	0.6180	0.5895
SSGCN-3E	**0.8972**	**0.8384**	**0.7949**	**0.7551**	**0.7229**	**0.7059**	**0.6653**	**0.6374**
Human	0.9392	0.9325	0.9063	0.9038	-	-	-	-

Table 2: Evaluation results under the OS-VQA setting on the VQAv2 dataset with pre-defined candidate sizes $\{2, 4, 8, 16, 32, 64, 128, 250\}$. Random candidate-sampling is used for the table above and semantic candidate-sampling for the table below. Numbers in bold are the best results, and numbers in blue are the best single-model results.

Model	2	4	8	16	32	64	128	250
ConSE [26]	0.8564	0.7598	0.6513	0.5416	0.4403	0.3539	0.2855	0.2283
DeViSE [9]	0.8005	0.6890	0.5918	0.5056	0.4319	0.3548	0.2916	0.2366
TriScore [15]	0.9295	0.8776	0.8093	0.7268	0.6215	0.5097	0.3954	0.2910
AnsEmbed [14]	0.9330	0.8876	0.8332	0.7688	0.6917	0.6055	0.5220	0.4373
SSGCN-G	0.9412	0.9080	0.8627	0.8065	0.7404	0.6695	0.5896	0.5116
SSGCN-Wp	0.9424	0.9081	0.8606	0.8043	0.7357	0.6614	0.5760	0.4985
SSGCN-Ww	0.9328	0.8987	0.8494	0.7930	0.7211	0.6457	0.5629	0.4886
SSGCN-3E	**0.9482**	**0.9205**	**0.8786**	**0.8295**	**0.7657**	**0.6945**	**0.6178**	**0.5421**
Model	2	4	8	16	32	64	128	250
ConSE [26]	0.5862	0.4100	0.3501	0.3130	0.2970	0.2891	0.2849	0.2270
DeViSE [9]	0.7156	0.5619	0.4504	0.3894	0.3456	0.3070	0.2924	0.2366
TriScore [15]	0.8155	0.6927	0.6055	0.5326	0.4776	0.4297	0.3963	0.2910
AnsEmbed [14]	0.8475	0.7499	0.6875	0.6305	0.5804	0.5349	0.5236	0.4371
SSGCN-G	0.8705	0.7807	0.7262	0.6697	0.6309	0.6028	0.5915	0.5112
SSGCN-Wp	0.8738	0.7766	0.7137	0.6650	0.6246	0.5863	0.5801	0.4986
SSGCN-Ww	0.8621	0.7693	0.7048	0.6501	0.6115	0.5758	0.5644	0.4881
SSGCN-3E	**0.8858**	**0.8008**	**0.7440**	**0.6931**	**0.6605**	**0.6285**	**0.6171**	**0.5419**
Human	0.9067	0.8733	0.8721	0.8508	-	-	-	-

$\min(1, \frac{\#\text{ humans that provided that answer}}{3})$. We evaluate all models by measuring their average feasibility scores over all samples.

Results under the Open-set VQA setting. Table 1 and 2 show the evaluation results on re-purposed VQA [3] and VQAv2 [12] datasets, under open-set VQA (OS-VQA) setting. Both random and semantic candidate-sampling strategies are used for evaluation. We compare all three versions of our model, as well as a simple ensemble model, SSGCN-3E, with all baselines.

It is clear that our models surpass all baselines with large margins, no matter which prior graph used. Particularly, comparing to the best competing method, AnsEmbed [14], the superiority of our models demonstrates the effectiveness of utilizing answer relationship and question guidance. Meanwhile, we can observe that the relative performance promotion of our models to AnsEmbed becomes larger as candidate size increases, which indicates that these two aspects of information are of greater importance for more difficult conditions. Ablation studies in Section 4.5 further validate this.

When comparing among our models, we can see that their performance is roughly close to each other, while SSGCN-G tends to be slightly superior generally, particularly for large candidate sizes. We conjecture that the prior graph built with *GloVe* [27] contains more useful information for our VQA task. The model performance should be further enhanced if better KS used. Also, with a simple ensemble of all our three models, the model beats others absolutely, illustrating the great potential of combining different KSs.

Finally, comparing between random-sampling and semantic-sampling, we can observe that the results under semantic-sampling

Table 3: Results under gOS-VQA setting on VQA dataset with pre-defined candidate sizes {2, 4, 8, 16, 32, 64, 128, 256, 340}. Random candidate-sampling is used for the table above and semantic candidate-sampling for the below. More details and analysis of the evaluation please refer to the text. Numbers in bold represent the best results, and numbers in blue show the best single-model results.

Model / candidate size	2	4	8	16	32	64	128	256	340
ConSE [26]	0.9150	0.8616	0.8002	0.7426	0.6793	0.6024	0.5326	0.4686	0.4462
DeViSE [9]	0.8837	0.8092	0.7485	0.6943	0.6442	0.5960	0.5461	0.5001	0.4795
TriScore [15]	0.9432	0.9040	0.8533	0.7931	0.7212	0.6391	0.5527	0.4632	0.4251
AnsEmbed [14]	0.9506	0.9189	0.8823	0.8416	0.7912	0.7349	0.6748	0.6057	0.5786
SSGCN-G	0.9559	0.9294	0.8984	0.8626	0.8167	0.7652	0.7053	0.6437	0.6178
SSGCN-Wp	0.9566	0.9301	0.9002	0.8621	0.8165	0.7605	0.6975	0.6370	0.6106
SSGCN-Ww	0.9539	0.9305	0.8986	0.8595	0.8112	0.7563	0.6928	0.6276	0.6011
SSGCN-3E	**0.9613**	**0.9409**	**0.9150**	**0.8798**	**0.8390**	**0.7881**	**0.7310**	**0.6691**	**0.6436**
Model / candidate size	2	4	8	16	32	64	128	256	340
ConSE [26]	0.6820	0.5212	0.4471	0.4069	0.3909	0.3898	0.3872	0.3839	0.3860
DeViSE [9]	0.7991	0.6628	0.5791	0.5143	0.4817	0.4495	0.4366	0.4248	0.4171
TriScore [15]	0.8276	0.6863	0.5904	0.4777	0.4214	0.3642	0.3461	0.3279	0.3157
AnsEmbed [14]	0.8601	0.7600	0.6845	0.6202	0.5815	0.5510	0.5360	0.5187	0.5081
SSGCN-G	0.8640	0.7603	0.6890	0.6243	0.5838	0.5476	0.5377	0.5203	0.5161
SSGCN-Wp	0.8652	0.7715	0.7068	0.6418	0.6043	0.5689	0.5504	0.5300	0.5219
SSGCN-Ww	0.8660	0.7776	0.7099	0.6390	0.5975	0.5550	0.5328	0.5168	0.5112
SSGCN-3E	**0.8765**	**0.7940**	**0.7300**	**0.6643**	**0.6258**	**0.5863**	**0.5697**	**0.5525**	**0.5478**

Table 4: Results under gOS-VQA setting on VQAv2 dataset with pre-defined candidate sizes {2, 4, 8, 16, 32, 64, 128, 256, 500}. Random candidate-sampling is used for the table above and semantic candidate-sampling for the below. More details and analysis of the evaluation please refer to the text. Numbers in bold represent the best results, and numbers in blue show the best single-model results.

Model / candidate size	2	4	8	16	32	64	128	256	500
ConSE [26]	0.9025	0.8344	0.7675	0.7023	0.6349	0.5769	0.5232	0.4811	0.4442
DeViSE [9]	0.8336	0.7393	0.6687	0.6110	0.5578	0.5126	0.4703	0.4288	0.3935
TriScore [15]	0.9441	0.9083	0.8635	0.8081	0.7352	0.6528	0.5623	0.4657	0.3726
AnsEmbed [14]	0.9482	0.9233	0.8883	0.8505	0.8020	0.7464	0.6829	0.6150	0.5449
SSGCN-G	0.9534	0.9329	0.9047	0.8706	0.8247	0.7774	0.7191	0.6569	0.5956
SSGCN-Wp	0.9524	0.9338	0.9042	0.8663	0.8205	0.7696	0.7085	0.6456	0.5827
SSGCN-Ww	0.9498	0.9287	0.8991	0.8637	0.8167	0.7649	0.7059	0.6451	0.5853
SSGCN-3E	**0.9567**	**0.9409**	**0.9159**	**0.8840**	**0.8413**	**0.7957**	**0.7402**	**0.6805**	**0.6209**
Model / candidate size	2	4	8	16	32	64	128	256	500
ConSE [26]	0.6934	0.5385	0.4623	0.4289	0.4169	0.4133	0.4119	0.4133	0.4097
DeViSE [9]	0.7609	0.6161	0.5349	0.4725	0.4325	0.3982	0.3810	0.3737	0.3640
TriScore [15]	0.8407	0.7224	0.6255	0.5241	0.4655	0.4032	0.3735	0.3524	0.3139
AnsEmbed [14]	0.8673	0.7755	0.7006	0.6354	0.5915	0.5446	0.5208	0.5028	0.4808
SSGCN-G	0.8713	0.7794	0.7026	0.6464	0.6032	0.5609	0.5439	0.5299	0.5180
SSGCN-Wp	0.8714	0.7789	0.6998	0.6466	0.6055	0.5598	0.5396	0.5241	0.5087
SSGCN-Ww	0.8641	0.7842	0.7063	0.6525	0.6081	0.5670	0.5429	0.5268	0.5135
SSGCN-3E	**0.8796**	**0.7988**	**0.7240**	**0.6730**	**0.6329**	**0.5893**	**0.5696**	**0.5536**	**0.5410**

are typically worse than in random-sampling, which validates that our semantic-sampling strategy does provide more challenging candidates, as empirically seen from the examples in Figure 3. One may notice that as the candidate size becomes larger, the performance gap between these two strategies tends to grow larger first, then shrinks instead. We believe that the beginning trend of gap-enlarging (typically when candidate size grows from 2 to 16) is as expected since adding more semantically-similar candidates would cause more confusion, while the impact of adding more random candidates would be much smaller; as for the gap-shrinking trend after candidate size grows larger (typically when candidate size is larger than 16), the reason is mainly because of experiment limit—we have only a finite set of all answers to sample from—the $\mathcal{A}_{te}$ ($|\mathcal{A}_{te}| = 173$ for VQA [3] and $|\mathcal{A}_{te}| = 251$ for VQAv2 [12]), so when the candidate size grows larger, the two sampling strategies would sample more overlapping candidates (since the whole set is finite), which causes the models to perform similarly.

Results under generalized OS-VQA setting. Taking inspiration from the generalized zero-shot learning (gZSL) research [6], we also explore to generalize our OS-VQA to a generalized setting, i.e., generalized OS-VQA (gOS-VQA). Like gZSL [6], we abandon the constraint that the testing answer set should be disjoint with the training answer set, and allowing the ground truth answer for a testing sample to be either seen or unseen in the training phase, which is more flexible and realistic.

To this end, we pre-reserve a subset, $res_{gOS\text{-}VQA}$, from the original $\mathcal{D}_{tr}$, which is not used during training. Then we merge the testing set of OS-VQA, $test_{OS\text{-}VQA}$, with this pre-reserved set, and use the union set for gOS-VQA evaluation. The basic evaluation protocol is the same as OS-VQA, but some care should be taken here during candidate-sampling to prevent the candidates from biasing towards either seen answers (from $res_{gOS\text{-}VQA}$) or unseen answers (from $test_{OS\text{-}VQA}$). For this purpose, we modify the random and semantic strategies with the constraint that the candidate set should contain equal numbers of seen and unseen answers in expectation, by sampling with weights instead of uniformly.

Table 5: Ablation study results on the VQA datasets under the OS-VQA setting with pre-defined candidate sizes {2, 4, 8, 16, 32, 64, 128, 170}. Random candidate-sampling is used for the table above and semantic candidate-sampling for the table below.

Model	2	4	8	16	32	64	128	170
SSGCN-G	0.9444	0.9074	0.8601	0.8114	0.7569	0.6935	0.6297	0.6007
w/o Q	0.9474	0.9114	0.8674	0.8171	0.7592	0.6905	0.6095	0.5777
w/o prior graph	0.9454	0.9076	0.8640	0.8145	0.7563	0.6830	0.6082	0.5775
w/o learnable graph	0.9437	0.9000	0.8561	0.8016	0.7342	0.6616	0.5907	0.5614
SSGCN-Wp	0.9460	0.9092	0.8665	0.8166	0.7547	0.6879	0.6134	0.5828
w/o Q	0.9457	0.9081	0.8618	0.8109	0.7459	0.6785	0.6077	0.5778
w/o prior graph	0.9454	0.9076	0.8640	0.8145	0.7563	0.6830	0.6082	0.5775
w/o learnable graph	0.9451	0.9031	0.8510	0.7972	0.7342	0.6648	0.5869	0.5555
SSGCN-Ww	0.9412	0.9083	0.8639	0.8135	0.7530	0.6883	0.6189	0.5897
w/o Q	0.9428	0.9061	0.8602	0.8115	0.7510	0.6788	0.6061	0.5727
w/o prior graph	0.9454	0.9076	0.8640	0.8145	0.7563	0.6830	0.6082	0.5775
w/o learnable graph	0.9406	0.9019	0.8534	0.7974	0.7320	0.6669	0.5961	0.5696
Model	2	4	8	16	32	64	128	170
SSGCN-G	0.8798	0.7957	0.7530	0.7161	0.6853	0.6710	0.6258	0.6011
w/o Q	0.8840	0.7985	0.7479	0.7048	0.6787	0.6630	0.6112	0.5782
w/o prior graph	0.8771	0.8012	0.7440	0.7111	0.6742	0.6600	0.6104	0.5795
w/o learnable graph	0.8674	0.7906	0.7377	0.6902	0.6634	0.6438	0.5922	0.5611
SSGCN-Wp	0.8828	0.8133	0.7613	0.7080	0.6823	0.6626	0.6133	0.5828
w/o Q	0.8725	0.7929	0.7447	0.7062	0.6736	0.6563	0.6068	0.5773
w/o prior graph	0.8771	0.8012	0.7440	0.7111	0.6742	0.6600	0.6104	0.5795
w/o learnable graph	0.8731	0.7907	0.7333	0.6916	0.6567	0.6385	0.5892	0.5562
SSGCN-Ww	0.8829	0.8137	0.7630	0.7177	0.6844	0.6646	0.6180	0.5895
w/o Q	0.8686	0.7918	0.7452	0.7013	0.6705	0.6595	0.6043	0.5723
w/o prior graph	0.8771	0.8012	0.7440	0.7111	0.6742	0.6600	0.6104	0.5795
w/o learnable graph	0.8645	0.7888	0.7304	0.6959	0.6646	0.6425	0.5953	0.5696

The results are shown in Table 3 and 4. All our models still perform better than baseline methods, with a similar tendency that the performance promotion compared to baselines grows larger as the candidate size increases. We also note that the performance gap between the best baseline method, AnsEmbed [14], and our model is smaller, compared to the OS-VQA setting. This is reasonable since the reserved samples from $res_{gOS\text{-}VQA}$ are more close to the training phase (the answers are seen), and models like AnsEmbed [14] are good at these in-distribution samples thanks to the powerful fitting ability of deep network. These samples relieve the total difficulty compared to OS-VQA setting therefore narrows the gap. But we

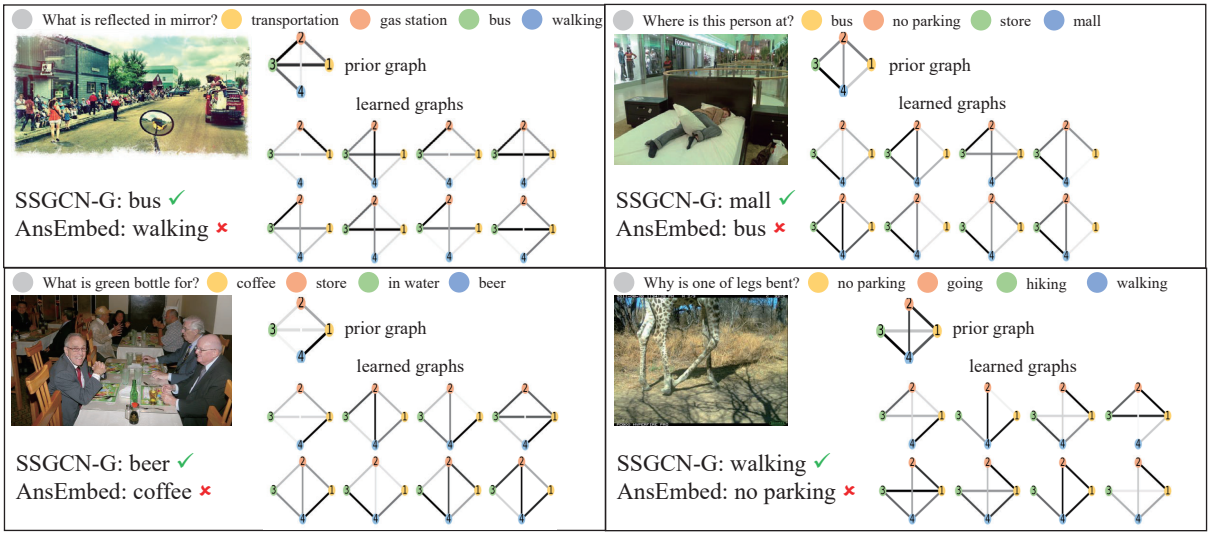


Figure 4: Qualitative results of some examples from the evaluation set on which AnsEmbed [14] fails while our model answers correctly. We also visualize the structure of the prior graph and all learned graphs to highlight their differences and complementarity.

point out that in practice, the unpredictable questions should tend to involve more unseen candidates than seen, and relying on seen data too much could hardly perform well.

Human evaluation results. To provide a context of the model results and an intuitive understanding of our evaluation data, we also conduct user study and ask humans to answer the questions. We restrict the user study to only adopt the semantic-sampling strategy since the semantic candidates are more reasonable and challenging. We also limit the candidate size to be no greater than 16, because in practice too many candidates tend to quickly exhaust the participants and lead to inferior performance than expected. The final results are shown at the bottom rows of Tables 1 and 2, respectively.

Our models outperform the baseline methods, yet are still clearly inferior to human performance. Another empirical insight is that as the candidate size grows, the performance of all methods (including human) has declined, but humans are relatively more robust to the variation of candidate size. This suggest that the effectiveness and robustness of human are still far better than current models despite the advancement in the VQA fields.

4.5 Ablation Study

To further validate the effectiveness of our design, we conduct several ablation studies. We identify three core components in our model—the question guidance, the prior graph, and the learnable graphs. We test their usefulness by removing each one of them separately, and compare the results to the full model.

For ablation study, we remove the question node and skip the above two steps of operation. The resultant answer embedding module becomes a normal GCN without source node.

We conduct ablation studies for all three versions of our model under the OS-VQA setting. Table 5 shows the results on VQA [3] dataset. It can be seen that removing any component would tend to harm the performance, which validates the effectiveness of our design. Meanwhile, the performance degradation is more obvious for larger candidate sizes, which is consistent with the above conclusion that all these components play more important roles under harder conditions.

4.6 Qualitative Results

To give a straightforward illustration of the effect of SSGCN, we conduct some qualitative analyses. Specifically, we show some samples from the evaluation set for which AnsEmbed [14] fails while our SSGCN answers correctly in Figure 4. To provide detailed illustration of the effect of our dual answer-graphs, we also plot the structure of the prior graphs and all learnable graphs used in the SSGCN module. It is clear that the weights distribution on different graphs are quite distinct, which demonstrates that they capture different semantic relationships.

5 CONCLUSION

In this work, we explored changing the conventional closed-set classification paradigm in VQA and proposed an open-set VQA setting.We developed a matching based VQA model for the setting, in which a novel SSGCN module was proposed to jointly employ question guidance and semantic dual answer-graphs to produce more discriminative and relevant answer embeddings. Extensive experiments by re-purposing two benchmark datasets, VQA [3] and VQAv2 [12], demonstrate the superiority of our proposed model compared to several baseline models. Additional ablation studies further validate the effectiveness.

ACKNOWLEDGMENTS

This work is supported by National Key R&D Program of China (2020AAA0104400), National Natural Science Foundation of China (61772037) and Beijing Natural Science Foundation (Z190001).

REFERENCES

[1] Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. Don't Just Assume; Look and Answer: Overcoming Priors for Visual Question Answering. In *CVPR*.

[2] Aishwarya Agrawal, Aniruddha Kembhavi, Dhruv Batra, and Devi Parikh. 2017. C-VQA: A Compositional Split of the Visual Question Answering (VQA) v1.0 Dataset. *CoRR* abs/1704.08243 (2017).

[3] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual Question Answering. In *ICCV*.

[4] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer Normalization. *CoRR* abs/1607.06450 (2016).

[5] Hedi Ben-younes, Rémi Cadène, Matthieu Cord, and Nicolas Thome. 2017. MUTAN: Multimodal Tucker Fusion for Visual Question Answering. In *ICCV*.

[6] Wei-Lun Chao, Soravit Changpinyo, Boqing Gong, and Fei Sha. 2016. An Empirical Study and Analysis of Generalized Zero-Shot Learning for Object Recognition in the Wild. In *ECCV*.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*.

[8] Tuong Do, Thanh-Toan Do, Huy Tran, Erman Tjiputra, and Quang D. Tran. 2019. Compact Trilinear Interaction for Visual Question Answering. In *ICCV*.

[9] Andrea Frome, Gregory S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. 2013. DeViSE: A Deep Visual-Semantic Embedding Model. In *NeurIPS*.

[10] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. 2016. Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding. In *EMNLP*.

[11] Peng Gao, Haoxuan You, Zhanpeng Zhang, Xiaogang Wang, and Hongsheng Li. 2019. Multi-modality Latent Interaction Network for Visual Question Answering. In *ICCV*.

[12] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. In *CVPR*.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR*.

[14] Hexiang Hu, Wei-Lun Chao, and Fei Sha. 2018. Learning Answer Embeddings for Visual Question Answering. In *CVPR*.

[15] Allan Jabri, Armand Joulin, and Laurens van der Maaten. 2016. Revisiting Visual Question Answering Baselines. In *ECCV*.

[16] Kushal Kafle and Christopher Kanan. 2017. An Analysis of Visual Question Answering Algorithms. In *ICCV*.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *NeurIPS*.

[18] Linjie Li, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019. Relation-Aware Graph Attention Network for Visual Question Answering. In *ICCV*.

[19] Yuanpeng Li, Yi Yang, Jianyu Wang, and Wei Xu. 2018. Zero-Shot Transfer VQA Dataset. *CoRR* abs/1811.00692 (2018).

[20] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *ECCV*.

[21] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical Question-Image Co-Attention for Visual Question Answering. In *NeurIPS*.

[22] Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and Knowledge-based Measures of Text Semantic Similarity. In *AAAI*.

[23] George A Miller. 1998. *WordNet: An electronic lexical database*. MIT press.

[24] Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. 2017. Dual Attention Networks for Multimodal Reasoning and Matching. In *CVPR*.

[25] Hyeonwoo Noh, Taehoon Kim, Jonghwan Mun, and Bohyung Han. 2019. Transfer Learning via Unsupervised Task Discovery for Visual Question Answering. In *CVPR*.

[26] Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg Corrado, and Jeffrey Dean. 2014. Zero-Shot Learning by Convex Combination of Semantic Embeddings. In *ICLR*.

[27] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *EMNLP*.

[28] Roy Rada, Hafedh Mili, Ellen Bicknell, and Maria Blettner. 1989. Development and application of a metric on semantic nets. *IEEE Trans. Systems, Man, and Cybernetics* 19 (1989).

[29] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *CoRR* abs/1910.10683 (2019).

[30] Damien Teney, Peter Anderson, Xiaodong He, and Anton van den Hengel. 2018. Tips and Tricks for Visual Question Answering: Learnings From the 2017 Challenge. In *CVPR*.

[31] Damien Teney and Anton van den Hengel. 2016. Zero-Shot Visual Question Answering. *CoRR* abs/1611.05546 (2016).

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NeurIPS*.

[33] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. 2018. Zero-Shot Recognition via Semantic Embeddings and Knowledge Graphs. In *CVPR*.

[34] Qi Wu, Peng Wang, Chunhua Shen, Anthony R. Dick, and Anton van den Hengel. 2016. Ask Me Anything: Free-Form Visual Question Answering Based on Knowledge from External Sources. In *CVPR*.

[35] Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*.

[36] Yiyi Zhou, Rongrong Ji, Jinsong Su, Xiaoshuai Sun, and Weiqiu Chen. 2019. Dynamic Capsule Attention for Visual Question Answering. In *AAAI*.