Supplementary Material: Learning Answer Embeddings for Visual Question Answering

Hexiang Hu*
U. of Southern California
Los Angeles, CA
hexiang.frank.hu@gmail.com

Wei-Lun Chao*
U. of Southern California
Los Angeles, CA
weilunchao760414@gmail.com

Fei Sha
U. of Southern California
Los Angeles, CA
feisha@usc.edu

In this Supplementary Material, we provide details omitted in the main paper:

- Section 1: Implementation details (Section 4.2 of the main paper).
- Section 2: uPMC Vs. Triplet-based Methods (Section 4.2 of the main paper).
- Section 3: Effects of incorporating semantic knowledge in weighted likelihood (Section 4.4 of the main paper).
- Section 4: Analysis with seen/unseen answers (Section 4.5 of the main paper).
- Section 5: Visualization of answer embeddings (Section 4.5 of the main paper).
- Section 6: Analysis on answer embeddings.

1. Implementation Details

In this section, we provide more details about the architectures of the stacked attention network (SAN) [5, 12] and the multi-layer perceptron (MLP) used for $f_{\boldsymbol{\theta}}(i, q)$ and $g_{\boldsymbol{\phi}}(a)$ in the main paper (section 4.2).

MLP as $f_{\boldsymbol{\theta}}(i, q)$ and $g_{\boldsymbol{\phi}}(a)$ As mentioned in the main paper, a one-hidden-layer MLP (with the hidden dimension of 4,096 and output dimension of 1,024) is used for both $f_{\boldsymbol{\theta}}(i, q)$ and $g_{\boldsymbol{\phi}}(a)$. The question q or answer a is represented by the average of word embeddings. Concretely, we compute the average of the pre-trained GloVe [8] vectors of words in question or answer. We then input this question vector (concatenated with the visual feature) or answer vector to the specified MLP, for obtaining output embedding. To enable better generalization on unseen answers and across datasets, we keep the average GloVe word embeddings for answer fixed in all our experiments. For the word embedding on questions, we fine-tune it as this leads to better empirical results. To represent the image feature i, we extract the activations from the last convolution layer of a 152-layer ResNet [2] pre-trained on ImageNet [9], and average them over the spatial extent to obtain a 2,048 dimensional feature vector.

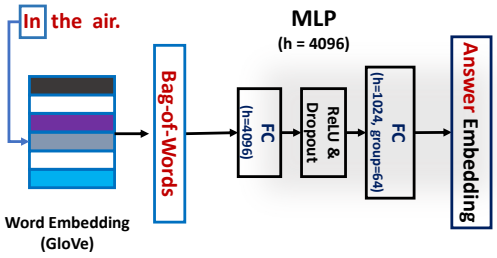


Figure 1. The multilayer perceptron (MLP) as $g_{\boldsymbol{\phi}}(a)$. The average of transformed word embeddings is first projected to the hidden space through a ReLU activation and then mapped to the embedding space. Dropout (p=0.5) is used for regularization. The same architecture is used for $f_{\boldsymbol{\theta}}(i, q)$, except the input dimension is 2,348.

The architecture of the one-hidden-layer MLP for computing answer embedding is illustrated in Fig. 1. The input is first mapped into the hidden space of 4,096 dimensions and then projected to a 1,024 dimensional embedding space. To reduce the number of parameters introduced in the MLP, we follow a similar practice suggested in [11] and apply a group-wise inner product to sparsify the weights. For both $f_{\boldsymbol{\theta}}(i, q)$ and $g_{\boldsymbol{\phi}}(a)$, the output of MLP is scaled up by a factor 10.

According to our ablation study in Section 3, we set α (cf. eq. (2) in the main text) to be multi-hot for VQA2, and use one-hot as α for all the other datasets.

SAN as $f_{\boldsymbol{\theta}}(i, q)$ Details about the stacked attention network (SAN) is shown in Fig. 2. To represent a question, a single layer bidirectional LSTM (bi-LSTM) with the hid-

* Equal contributions

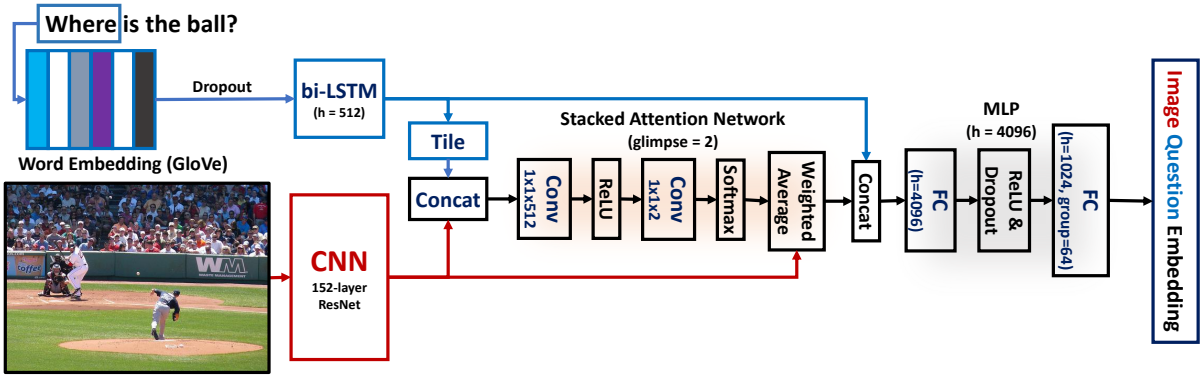


Figure 2. The stacked attention network (SAN) as $f_{\boldsymbol{\theta}}(i, q)$. We follow the similar architecture as in [5] to obtain the visual semantic embedding of images and questions.

den dimension of 512 is used on top of the question GloVe word embeddings. Similarly to MLP setting, we fine-tune the question word embedding. At the same time, for image feature i, we extract the output of the last convolution layer from a 152-layer ResNet and obtain a feature tensor of dimensionality $14 \times 14 \times 2048$, as suggested in [5]. A stacked attention module [12] with two glimpses is then used to obtain the question attended visual features, using both the outputs of question LSTM and ResNet-152 spatial visual feature. Next, a one-hidden-layer MLP (same architecture as previously mentioned) is used to embed the concatenated feature of questions and attended images into a 1,024-dimensional embedding space. Again, the output of the MLP is scaled up by a factor 10.

For our best performing model fPMC(SAN$\star$), we used the SAN as $f_{\boldsymbol{\theta}}(i, q)$ and a two-layer bi-LSTM as answer embedding function $g_{\boldsymbol{\phi}}(a)$, with dimensionality of 512. For this bi-LSTM, we set the drop out rate to be 0.5 between the first and second LSTMs. We perform max fusion on the hidden states to obtain the holistic answer feature over the answer sentence. Both the output of $f_{\boldsymbol{\theta}}(i, q)$ and $g_{\boldsymbol{\phi}}(a)$ are then scaled up by a factor of 10 and next used to produce the score through inner product for the $(i_n, q_n, \mathcal{C}_n)$ triplet.

Configuration for competing methods For our classification model baseline (CLS), we use the same LSTM+SAN+MLP architecture as above, except that the output dimension is the total number of top-frequency answers. For the un-factorized PMC (uPMC), we concatenate the answer feature together with image and question features from SAN+LSTM and then input into a one-layer MLP with hidden dimensionality of 4096. It is then used to produce a singleton score for the input triplet.

Optimization Details For all above methods, we train for 50 epochs on each dataset using Adam [6] optimization with initial learning rate of 0.001. We follows the same learning rate decay strategy suggested in [5], which gives as follows:

$$l_t = 0.5^{\frac{t}{t_{decay}}} \cdot l_0 \tag{1}$$

Here, l_t denotes the learning rate at epoch t, l_0 is the initial learning rate. t_{decay} represents the preset decay schedule, which is 15 in all our experiments. For fPMC we set the A_o to be 3000 across all experiments; for uPMC, due to its large consumption of memory and computation inefficiency during training, we set the A_o to be 300 for all settings (this is the largest feasible size of A_o for uPMC(SAN) with reasonable computation and memory consumption).

2. uPMC Vs. Triplet-based Methods

We follow the exact multiple-choice (MC) setting of [1, 4] to train MLP (with the (i, q, a) triplet as input) on Visual7W. While getting good results on Visual7W (65.7%), its transfer performance suffers (13.6% to VQA2 and 30.2% to qaVG). This is because in training, [1, 4] only differentiates between the correct answer and a few negative answers, not the entire universe of possible answers. Meanwhile, training the binary scoring function in [1, 4] requires to carefully control the calibration between positive and negatives, which made it challenging when the number of negative answers scales up.

Therefore, we adapt their model to also utilize our PMC framework for training (i.e., uPMC(MLP)), which optimize stochastic multi-class cross-entropy with negative answers sampling. The transfer performance improves by a large margin. (Visual7W→qaVG: improving from 30.2% to 48.4%.)

Table 1. Detailed analysis of different $\alpha(a, d)$ for weighted likelihood. The reported number is the accuracy on VQA2 (validation).

Method	Weighting Criterion	Acc.
fPMC(SAN)	one-hot	58.0
	multi-hot	**60.0**
	WUPS	57.8

Table 2. Analysis of cross dataset performance over Seen/Unseen answers using either CLS or PMC for Visual QA

	Visual7W											
	CLS(SAN)			uPMC(SAN)			fPMC(SAN)			fPMC(SAN$\star$)		
	S	U	All	S	U	All	S	U	All	S	U	All
VQA2	59.8	25.0	45.8	57.4	54.6	56.8	60.7	58.5	60.2	61.7	59.4	62.5
qaVG	63.4	25.0	58.9	66.7	45.3	66.0	69.1	47.7	68.4	70.2	46.9	69.5

3. Semantic Knowledge in Weighted Likelihood

As mentioned in section 4.4 of the main paper, we report in Table 1 the ablation study on using different weight function $\alpha(a, d)$ in the weighted likelihood formulation (cf. Eq. (2) of the main paper). We compare three different types of $\alpha(a, d)$ on VQA2:

- **one-hot**: Denote t_n as the dominant answer in $\mathcal{C}_n$. We set $\mathcal{C}_n \leftarrow \{t_n\}$ (*i.e.*, now $\mathcal{C}_n$ becomes a singleton) and apply

  $$\alpha(a, d) = \mathbb{I}[a = d] \text{ (cf. Eq. (3) of the main paper).}$$

  In this case, only one answer is considered positive to a (i, q) pair. No extra semantic relationship is encoded.

- **multi-hot**: We keep the given $\mathcal{C}_n$ (the ten user annotations collected by VQA2; *i.e.* $|\mathcal{C}_n| = 10$) and apply

  $$\alpha(a, d) = \mathbb{I}[a = d] \text{ (cf. Eq. (3) of the main paper)}$$

  to obtain a multi-hot vector $\sum_{a \in \mathcal{C}_n} \alpha(a, d)$ for soft weighting, leading to a loss similar to [5, 3].

- **WUPS**: We again consider $\mathcal{C}_n \leftarrow \{t_n\}$, but utilize the WUPS score [10, 7] (the range is [0, 1]) together with Eq. (6) of the main paper to define $\alpha(a, d)$. We set $\lambda = 0.9$ and give d which has $\textbf{WUPS}(a, d) = 1$ a larger weight (*i.e.*, 8).

The results suggest that the multi-hot vector computed from multiple user annotations provides the best semantic knowledge among answers for learning the model.

4. Analysis with Seen/Unseen Answers

Next, we present an analysis on transfer learning results, comparing the performance of methods over seen and unseen answer sets. Specifically, we study the transfer learning result from VQA2 and qaVG to Visual7W. Here, **seen (S)** refers to those multiple choices where at least one candidate answer is seen in the training vocabulary, and **unseen (U)** refers to those multiple choices where all the candidate answers are not observed in the training vocabulary. As shown in Table 2, we see that our fPMC model performs better than the CLS model on both seen and unseen answer set. While CLS model obtains random performance (the random chance is 25 %) on the unseen answer set, our fPMC model achieved at least 20% (in absolute value) better performance. In general. uPMC is also working well comparing to CLS. This performance improvement is gain mostly by taking answer semantics from the word vectors into account.

5. Visualization on Answer Embeddings

As promised in the main text, we provide the t-SNE visualization of the answer embedding. To better demonstrate the effectiveness of learning answer embedding, we re-train the answer embedding model with randomly initialized answer vectors. We provide visualization on both the initial answer embedding and learned answer embedding, to reflect the preservation of semantics and syntactics in the learned embedding.

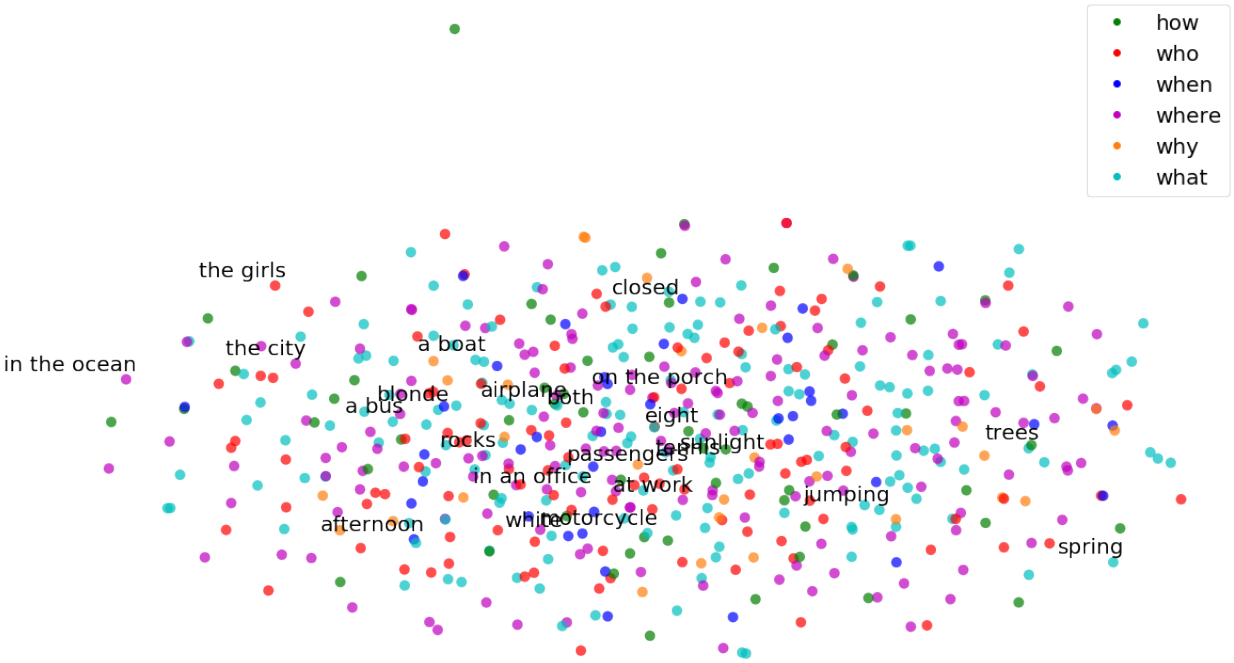
According to Fig. 3, we can observe that a clear structure in the answer embedding are obtain in our learned embedding. While the random initialization of the embedding remains chaos, our learned embedding successfully provide both semantic and syntactic similarities between answers. For example, semantically similar answers such as **"airplane"** and **"motorcycle"** are close to each other, and syntactically similar answers like **"in an office"** and **"on the porch"** are close. Besides, we also observe that answers are clustered according to its majority question type, which meets our expectation for the answer embedding's structure. Here we take majority because one answer can be used for multiple questions of different types.

6. Analysis on Answer Embeddings

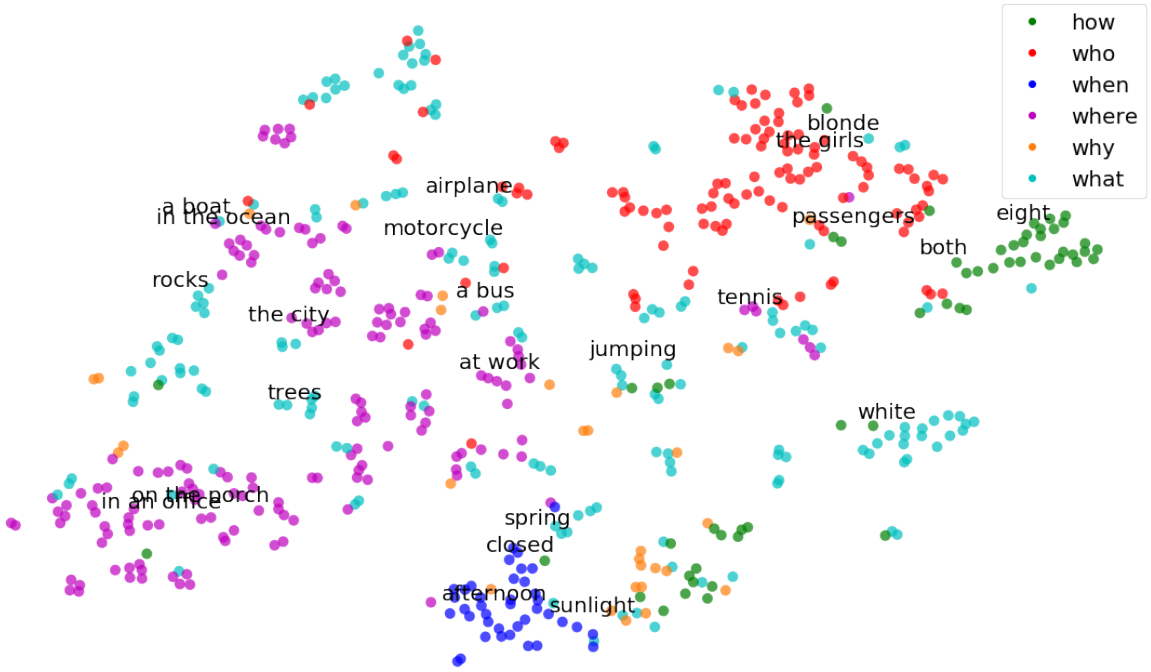
Table 3. Results for the baseline method that fix answer embedding as GloVe. (We show results with SAN as $f_{\boldsymbol{\theta}}(i, q)$).

Target	VQA2		Visual7W		qaVG	
Source	Fixed	Learning	Fixed	Learning	Fixed	Learning
VQA2	57.5	**60.0**	47.5	**60.2**	37.6	**54.8**

Finally, we provide results for an additional baseline algorithm where $f_{\boldsymbol{\theta}}(i, q)$ directly maps to the fixed space of average GloVe answer representations. Here we need to keep the GloVe embedding fixed to enable transferability. Table 3 shows the results on the VQA2 dataset. We compare its performance to our approach of learning answer embedding with MLP as $g_{\boldsymbol{\phi}}(a)$ in terms of both in-domain and transfer learning performance—learning an-



(a) Random initialized answer embedding



(b) Learned answer embedding

Figure 3. **t-SNE visualization.** We randomly select 1000 answers from Visual7W and visualize them in the initial answer embedding and learned answer embeddings. Each answer is marked with different colors according to their question types. (*e.g.* when, how, who, where, why, what). To make the figure clear for reading, we randomly sub-sampled the text among those 1000 answers to visualize.

swer embeddings outperforms this simple baseline in all cases. Associated with the previous visualization results, we can conclude that learning answer embedding can effectively capture the semantic relationship between answers and image-question pairs while obtaining superior performance on both within-domain performance and transfer learning performance.

References

[1] W.-L. Chao, H. Hu, and F. Sha. Being negative but constructively: Lessons learnt from creating better visual question answering datasets. In *NAACL*, 2018. 2

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[3] I. Ilievski and J. Feng. A simple loss function for improving the convergence and accuracy of visual question answering models. In *CVPR Workshop*, 2017. 3

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

[5] V. Kazemi and A. Elqursh. Show, ask, attend, and answer: A strong baseline for visual question answering. *arXiv preprint arXiv:1704.03162*, 2017. 1, 2, 3

[6] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 2

[7] M. Malinowski and M. Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *NIPS*, 2014. 3

[8] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014. 1

[9] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 1

[10] Z. Wu and M. Palmer. Verbs semantics and lexical selection. In *ACL*, 1994. 3

[11] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. 1

[12] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In *CVPR*, 2016. 1, 2