Supplementary Material for Hierarchical Graph Attention Network for Few-shot Visual-Semantic Learning

Image and Text Embedding

For image embedding of an image $\mathbf{I}_i$, we build a neural network ϕ to output visual representation $\phi(\mathbf{I}_i; \boldsymbol{\theta}_\phi)$ by following the architecture used by FPAIT [11], which contains four 3×3 convolutional blocks with batch normalizations and ReLU activations. The numbers of feature channels in these four blocks are 64, 96, 128 and 256, respectively. There is a 2×2 max-pooling layer after each of the first three blocks, and a global-pooling layer after the last block. The output image embedding has a dimension of 256.

For text embedding of a question/description $\mathbf{Q}_i$, we build a neural network ψ to output the semantic representation $\psi(\mathbf{Q}_i; \boldsymbol{\theta}_\psi)$ as follows. We first transfer each word in $\mathbf{Q}_i$ into a feature vector using the pre-trained $100D$ GloVe [38] vectors, and use randomly initialized weights for all words which are out of GloVe's vocabulary, and then use the temporal convolutional network (TCN) [27] to obtain the embedding of the word sequence. Same as FPAIT [11], the output text embedding has a dimension of 512.

Meta-training/testing Paradigm

The overall meta-training/testing algorithms for HGAT are summarized in Algorithms 1 and 2.

Algorithm 1 The process of meta-training for HGAT

Input: A set of T tasks $\{\mathcal{T}_t\}_{t=1}^T$ generated from a meta-training dataset $\mathcal{D}_{\text{mtr}}$, where $\{\mathcal{T}_t\}_{t=1}^T = \{\mathcal{S}_t\}_{t=1}^T \cup \{\mathcal{Q}_t\}_{t=1}^T$
Initialize $\boldsymbol{\theta}_\phi \cup \boldsymbol{\theta}_\psi \cup \{\mathbf{W}_V^l, \mathbf{W}_S^l, \mathbf{a}_V^l, \mathbf{a}_S^l\}_{l=1}^2 \cup \{\mathbf{W}_R^l, \mathbf{a}_R^l\}_{l=1}^3$
while not done **do**
  Sample batch of tasks $< \mathcal{T}_t >$ from task set $\{\mathcal{T}_t\}_{t=1}^T$
  for all $\mathcal{T}_t$ **do**
    $\{\mathbf{V}_i^1\}, \{\mathbf{S}_i^1\} \leftarrow \text{GraphConstruct}(\{(\mathbf{I}_i, \mathbf{Q}_i, \mathbf{A}_i)\}; \{\boldsymbol{\theta}_\phi, \boldsymbol{\theta}_\psi\}), \forall i$
    for all $l = 1, 2$ **do**
      $\{\mathbf{V}_i^{l+1}\}, \{\mathbf{S}_i^{l+1}\} \leftarrow \text{NodeFeatureUpdate}(\{\mathbf{V}_i^l\}, \{\mathbf{S}_i^l\}; \{\mathbf{W}_V^l, \mathbf{W}_S^l, \mathbf{a}_V^l, \mathbf{a}_S^l\}), \forall i$
    end for
    Initialize $\mathbf{R}_i^1 \leftarrow \text{GraphConstruct}(\{(\mathbf{I}_i, \mathbf{Q}_i, \mathbf{A}_i)\}, \{\mathbf{V}_i^2\}, \{\mathbf{S}_i^2\}, \{\mathbf{V}_i^3\}, \{\mathbf{S}_i^3\}; \{\boldsymbol{\theta}_\phi, \boldsymbol{\theta}_\psi\}), \forall i$
    for all $l = 1, 2, 3$ **do**
      $\{\mathbf{R}_i^{l+1}\} \leftarrow \text{NodeFeatureUpdate}(\{\mathbf{R}_i^l\}; \{\mathbf{W}_R^l, \mathbf{a}_R^l\}), \forall i$
    end for
    Compute the loss and update $\boldsymbol{\theta}_\phi \cup \boldsymbol{\theta}_\psi \cup \{\mathbf{W}_V^l, \mathbf{W}_S^l, \mathbf{a}_V^l, \mathbf{a}_S^l\}_{l=1}^2 \cup \{\mathbf{W}_R^l, \mathbf{a}_R^l\}_{l=1}^3$
  end for
end while

Algorithm 2 The process of meta-testing for HGAT

Input: A task $\mathcal{T}$ sampled from $\{\mathcal{T}_{T+t}\}_{t=1}^{T'}$, where $\mathcal{T} = \mathcal{S} \cup \mathcal{Q}$, $\mathcal{S} = \{(\mathbf{I}_i, \mathbf{Q}_i, \mathbf{A}_i)\}_{i=1}^{N \times K}$, $\mathcal{Q} = \{(\mathbf{I}_i, \mathbf{Q}_i)\}_{i=N \times K+1}^{N \times K+M}$
Parameters: $\boldsymbol{\theta}_\phi \cup \boldsymbol{\theta}_\psi \cup \{\mathbf{W}_V^l, \mathbf{W}_S^l, \mathbf{a}_V^l, \mathbf{a}_S^l\}_{l=1}^2 \cup \{\mathbf{W}_R^l, \mathbf{a}_R^l\}_{l=1}^3$
Output: $\{\hat{\mathbf{A}}_i\}_{i=N \times K+1}^{N \times K+M}$
$\{\mathbf{V}_i^1\}, \{\mathbf{S}_i^1\} \leftarrow \text{GraphConstruct}(\{(\mathbf{I}_i, \mathbf{Q}_i, \mathbf{A}_i)\}; \{\boldsymbol{\theta}_\phi, \boldsymbol{\theta}_\psi\}), \forall i$
for all $l = 1, 2$ **do**
  $\{\mathbf{V}_i^{l+1}\}, \{\mathbf{S}_i^{l+1}\} \leftarrow \text{NodeFeatureUpdate}(\{\mathbf{V}_i^l\}, \{\mathbf{S}_i^l\}; \{\mathbf{W}_V^l, \mathbf{W}_S^l, \mathbf{a}_V^l, \mathbf{a}_S^l\}), \forall i$
end for
Initialize $\mathbf{R}_i^1 \leftarrow \text{GraphConstruct}(\{(\mathbf{I}_i, \mathbf{Q}_i, \mathbf{A}_i)\}, \{\mathbf{V}_i^2\}, \{\mathbf{S}_i^2\}, \{\mathbf{V}_i^3\}, \{\mathbf{S}_i^3\}; \{\boldsymbol{\theta}_\phi, \boldsymbol{\theta}_\psi\}), \forall i$
for all $l = 1, 2, 3$ **do**
  $\{\mathbf{R}_i^{l+1}\} \leftarrow \text{NodeFeatureUpdate}(\{\mathbf{R}_i^l\}; \{\mathbf{W}_R^l, \mathbf{a}_R^l\}), \forall i$
end for
$\{\hat{\mathbf{A}}_i\}_{i=N \times K+1}^{N \times K+M} \leftarrow \text{QueryLabelPredict}(\{\mathbf{R}_i^4\}_{i=N \times K+1}^{N \times K+M})$

Benchmark Datasets

Toronto COCO-QA consists of $78,736$ question-answer (QA) pairs for training and $38,948$ QA pairs for testing. Each QA pair associates with one image from MSCOCO [30] and is labeled with one of the four QA types (i.e., object, number, color and location). Following the same pre-processing steps of FPAIT [11], the Toronto COCO-QA is transformed into the format which can be used for the few-shot VQA. Consequently, $57,834$ QA pairs with a set of 256 unique answers are used in the meta-training phase, and $13,965$ QA pairs with a set of 65 unique answers are used in the meta-testing phase. The two answer sets are mutually exclusive.

Visual Genome-QA, the largest dataset for VQA, contains over 1.7 million QA pairs with more than $100,000$ images sampled from MSCOCO [30]. Compared with Toronto COCO-QA, more categories of questions, which may start with the "who", "what", "where", "when", "why", "how" and "which", are provided. The Visual Genome-QA is transformed into the format for few-shot VQA with similar pre-processing steps for Toronto COCO-QA. Finally, there are $554,795$ QA pairs with a set of 244 unique answers for meta-training, and $136,473$ QA pairs for meta-testing with a set of 82 unique answers. The two answer sets are mutually exclusive.

COCO-FITB, proposed and used by FPAIT [11], is transformed from MSCOCO [30] by processing MSCOCO Captions [8] to generate image-caption pairs in the fill-in-the-blank format. $181,844$ image-caption pairs with a set of 159 unique blank words are used in meta-training, and $34,919$ image-caption pairs with a set of 43 unique blank words are used in meta-testing. The two sets of blank words are mutually exclusive.

Baseline Implementation

We re-implemented the Prototypical Net [42], Relation Net [44], R2D2 [6], DN4 [28], GNN [14], and EGNN [22] and extended these algorithms from few-shot classification to few-shot visual-semantic learning. For Prototypical Net, Relation Net, R2D2 and DN4, we adopted the concatenation of the corresponding visual and semantic representations as input feature for each sample in the support/training sets and query/test sets. For GNN, each sample from the support or query sets is represented as a node in the first layer of GNNs, and each node is initialized as the concatenation of its visual and semantic representations as well as the one-hot encoding of its label. Note that for the unknown labels (e.g., query samples), unlike Garcia *et al*. [14], the one-hot encoding is initialized to a zero vector instead of a uniform vector. For EGNN, both of the node features and the edge features need to be initialized. The node features are initialized as the concatenation of its visual and semantic representations. Following the definition in Kim *et al*. [22], each edge feature is a 2-dimensional vector with a value representing the relationship between its two connected nodes. If the two connected nodes belong to one class, the edge feature is set to $[1\ 0]$; otherwise, it is set to $[0\ 1]$. In addition, all the edges connected to the samples with unknown labels are set to $[0.5\ 0.5]$. It is nothing that for both of the GNN and EGNN, only one query node exists in each layer of GNNs.

Compare to Standard VQA/IC Methods

Method	COCO-QA		COCO-FITB	
	5-way accuracy		5-way accuracy	
	1-shot	5-shot	1-shot	5-shot
HCA	55.40	66.78	54.33	62.91
SAAA	56.72	67.23	55.67	64.16
CNN+TCN	57.19	71.82	59.95	70.32
HGAT	**63.13**	**75.41**	**63.36**	**74.14**

Table 5. Comparison with standard VQA and IC methods.

Since Dong *et al*. [11] compared on COCO-QA and COCO-FITB, we followed its setting and strategies for a fair comparison. Results in Table 5 show that the HGAT is more suitable and outperforms standard methods by a significant margin.

Ablation Studies

Based on the model only with the three-layer relation-aware GNNs in Stage-2 (Case-1), we gradually add visual-specific GNNs (Case-6), semantic-specific GNNs (Case-7), both visual-specific GNNs and semantic-specific GNNs (Case-8) and attention-based co-learning framework (Case-9) in Stage-1 for ablation studies. Corresponding cases (i.e., Case-2 to Case-5) without the 3-layer relation-aware GNNs in Stage-2 are also exploited.

Case	Stage-1	Stage-2	Visual Relations	Semantic Relations	Attention-based Co-learning	5-way accuracy 1-shot	5-way accuracy 5-shot	10-way accuracy 1-shot	10-way accuracy 5-shot
1		✓				76.10	82.14	63.99	66.30
2	✓		✓			75.33	81.67	62.08	66.42
3	✓			✓		75.84	80.25	62.64	65.76
4	✓		✓	✓		76.78	82.32	64.16	68.22
5	✓		✓	✓	✓	77.47	83.26	64.90	70.03
6	✓	✓	✓			77.55	84.01	63.77	69.26
7	✓	✓		✓		78.14	83.88	64.23	68.61
8	✓	✓	✓	✓		78.86	84.55	65.21	70.06
9	✓	✓	✓	✓	✓	**79.56**	**86.10**	**66.62**	**72.13**

Table 6. Full table of ablation studies on Visual Genome-QA for few-shot visual question answering.

Besides the $\mathbf{V}_i$ and $\mathbf{S}_i$ terms (which contains the intra- and inter-relationship from both visual and semantic features) in Eq. 9, whose effects have been justified in the ablation studies of Section 4.4, we also studied the efficacy of $\phi(\mathbf{I}_i)$, $\psi(\mathbf{Q}_i)$, and $h(\mathbf{A}_i)$ using 5-way 1-shot task on Visual-Genome-QA. As shown in Table 7, the absence of any term leads to a performance degradation.

	w/o h	w/o ϕ	w/o ψ	Full HGAT
5-way 1-shot	77.88	78.01	78.53	**79.56**

Table 7. Study on the efficacy of $\phi(\mathbf{I}_i)$, $\psi(\mathbf{Q}_i)$, and $h(\mathbf{A}_i)$ in Equation 9.

Experiments have been conducted on Visual Genome-QA to study the number of GNNs layers in Stage-1 (i.e., modal-specific GNNs) and Stage-2 (i.e., relation-aware GNNs). Results on 5-way 1-shot and 5-way 5-shot classifications in terms of the accuracy are shown in Table 8. Each row represents the results of the same number of model-specific GNN layers (Stage-1), and each column represents the results of the same number of relation-aware GNN layers (Stage-2). Specifically, when the numbers of GNN layers in Stage-1 and Stage-2 are set to 2 and 3, respectively, the optimal performance is achieved.

# of Layers	2	3	4
2	78.33/84.69	**79.56/86.10**	79.16/85.51
3	77.85/83.97	78.90/85.01	78.77/85.63

Table 8. Study on GNNs layers in Stage-1 (row-wise) and Stage-2 (column-wise).

Time Complexity Analysis

Given a N-way K-shot task with M query samples, each GNN layer has $n = N \times K + M$ nodes. The time complexity is $O(n^3)$ for the attention (Eqs. (5), (6), and (11)) and $O(n^2)$ for the feature updates (Eqs. (7), (8), and (12)). Thus, the overall time complexity is only $O(Ln^3)$ for HGAT with L GNN layers.

Visualization Samples

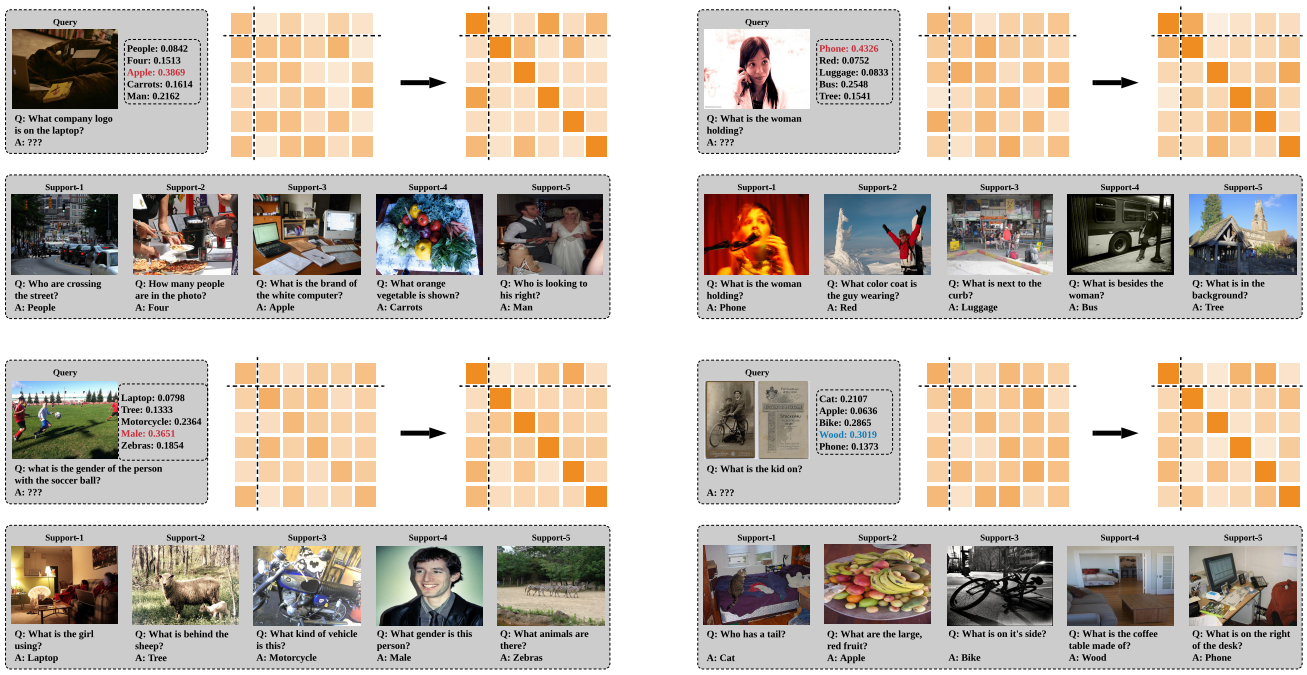

Figure 5. Attention visualizations of the 3rd layer in the relation-aware GNNs for 5-way 1-shot VQA on Visual Genome-QA. Dark/light color denotes higher/lower values. Red/blue color denotes correct/wrong predictions on query samples.

References

Dong, X.; Zhu, L.; Zhang, D.; Yang, Y.; and Wu, F. 2018. Fast parameter adaptation for few-shot image captioning and visual question answering. In Proceedings of the ACMMM, 54–62.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In Proceedings of EMNLP, 1532–1543.

Lea, C.; Flynn, M. D.; Vidal, R.; Reiter, A.; and Hager, G. D. 2017. Temporal convolutional networks for action segmentation and detection. In Proceedings of IEEE CVPR, 156–165.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In ECCV.

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

Snell, J.; Swersky, K.; and Zemel, R. S. 2017. Prototypical networks for few-shot learning. In NIPS.

Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P. H.; Hospedales; and T. M. 2018. Learning to compare: Relation network for few-shot learning. In Proceedings of IEEE CVPR, 1199-1208.

Bertinetto, L.; Henriques, J. F.; Torr, P. H.; and Vedaldi, A. 2019. Meta-learning with differentiable closed-form solvers. In ICLR.

Li, W.; Wang, L.; Xu, J.; Huo, J.; Gao, Y.; and Luo, J. 2019. Revisiting local descriptor based image-to-class measure for few-shot learning. In Proceedings of IEEE CVPR, 7260-7268.

Garcia, V.; and Bruna, J. 2017. Few-shot learning with graph neural networks. In ICLR.

Kim, J.; Kim, T.; Kim, S.; and Yoo, C. D. 2019. Edge-labeling graph neural network for few-shot learning. In Proceedings of IEEE CVPR, 11-20.