Supplementary Material for “Auto-Encoding Scene Graphs for Image Captioning”

Xu Yang, Kaihua Tang, Hanwang Zhang, Jianfei Cai
School of Computer Science and Engineering,
Nanyang Technological University,
{s170018@e,kaihua001@e}.ntu.edu.sg,{hanwangzhang@,ASJFCai@}ntu.edu.sg

This supplementary document will further detail the following aspects in the main paper: A. Network Architecture, B. Details of Scene Graphs, C. More Qualitative Examples.

A. Network Architecture

Here, we introduce the detailed network architectures of all the components in our model, which includes Graph Convolutional Network (GCN), Multi-modal Graph Convolutional Network (MGCN), Dictionary, and Decoders.

A.1. Graph Convolutional Network

In Section 4.2 of the main paper, we show how to use GCN to compute three embeddings by given a sentence scene graph, and the operations of this GCN are listed in Table A. In Table A (1) to (3), the object label l_o, relation label l_r, and attribute label l_a are all one-hot vectors. And the word embedding matrix $\boldsymbol{W}_{\Sigma_S} \in \mathbb{R}^{1,000\times 10,102}$ is used to map these one-hot vectors into continuous vector representations in Table A (4) to (6). The second dimension of $\boldsymbol{W}_{\Sigma_S}$ is the total number of object, relation, and attribute categories among all the sentence scene graphs. For g_r, g_a, g_o, and g_s in Table A (7) to (9), all of them own the same structure with independent parameters: a fully-connected layer, followed by an ReLU. The notation g_r ($D_{in} \rightarrow D_{out}$) denote that the input dimension is D_{in}, and output dimension is D_{out}.

A.2. Multi-modal Graph Convolutional Network

In Section 5.1 of the main paper, we briefly discuss the MGCN, and here we list its details in Table B. Besides the labels of objects, relations, and attributes, the input of MGCN also include object and relation RoI features, as shown in Table B (1) to (5). The RoI features are extracted from a pre-trained Faster Rcnn [3], v_r is the feature pooled from a region which cover the ‘subject’ and ‘object’. The word embedding matrix used here in Table B (6) to (8) is $\boldsymbol{W}_{\Sigma_I} \in \mathbb{R}^{1,000\times 472}$, which is different from the one used in GCN. In Table B (9) to (11), feature fusion proposed by [5] is implemented for fusing word embedding and visual feature together. Compared with Eq. (6) to Eq. (8) in the main paper, MGCN has the following modifications for computing relationship, attribute, and object embeddings: word embeddings $\boldsymbol{e}$ are substituted by fused embeddings $\boldsymbol{u}$; and g is substituted by f, which is also a function of a fully-connected layer, followed by an ReLU. With these modifications, we can formulate the computations of three embeddings in MGCN as:

Relationship Embedding $\boldsymbol{v}'_{r_{ij}}$ (Table B (12)):

$$\boldsymbol{v}'_{r_{ij}} = f_r(\boldsymbol{u}_{o_i}, \boldsymbol{u}_{r_{ij}}, \boldsymbol{u}_{o_j}). \tag{A}$$

Attribute Embedding $\boldsymbol{v}'_{a_i}$(Table B (13)):

$$\boldsymbol{v}'_{a_i} = \frac{1}{Na_i}\sum_{l=1}^{Na_i} f_a(\boldsymbol{u}_{o_i}, \boldsymbol{u}_{a_{i,l}}). \tag{B}$$

Object Embedding $\boldsymbol{v}'_{o_i}$(Table B (14)):

$$\boldsymbol{v}'_{o_i} = \frac{1}{Nr_i}[\sum_{o_j\in sbj(o_i)} f_s(\boldsymbol{u}_{o_i}, \boldsymbol{u}_{o_j}, \boldsymbol{u}_{r_{ij}}) + \sum_{o_k\in obj(o_i)} f_o(\boldsymbol{u}_{o_k}, \boldsymbol{u}_{o_i}, \boldsymbol{u}_{r_{ki}})]. \tag{C}$$

Table A. The details of GCN.

Index	Input	Operation	Output	Trainable Parameters
(1)	-	object label	l_o (10,102)	-
(2)	-	relation label	l_r (10,102)	-
(3)	-	attribute label	l_a (10,102)	-
(4)	(1)	word embedding $\boldsymbol{W}_{\Sigma_S} l_o$	$\boldsymbol{e}_o$ (1,000)	$\boldsymbol{W}_{\Sigma_S}$ (1,000 × 10,102)
(5)	(2)	word embedding $\boldsymbol{W}_{\Sigma_S} l_r$	$\boldsymbol{e}_r$ (1,000)	$\boldsymbol{W}_{\Sigma_S}$ (1,000 × 10,102)
(6)	(3)	word embedding $\boldsymbol{W}_{\Sigma_S} l_a$	$\boldsymbol{e}_a$ (1,000)	$\boldsymbol{W}_{\Sigma_S}$ (1,000 × 10,102)
(7)	(4),(5)	relationship embedding (Eq.(6))	$\boldsymbol{x}_r$ (1,000)	g_r (3,000 → 1,000)
(8)	(4),(6)	attribute embedding (Eq.(7))	$\boldsymbol{x}_a$ (1,000)	g_a (2,000 → 1,000)
(9)	(4),(5)	object embedding (Eq.(8))	$\boldsymbol{x}_o$ (1,000)	g_s,g_o (3,000 → 1,000)

Table B. The details of MGCN

Index	Input	Operation	Output	Trainable Parameters
(1)	-	object RoI feature	$\boldsymbol{v}_o$ (2,048)	-
(2)	-	relation RoI feature	$\boldsymbol{v}_r$ (2,048)	-
(3)	-	object label	l_o (472)	-
(4)	-	relation label	l_r (472)	-
(5)	-	attribute label	l_a (472)	-
(6)	(3)	word embedding $\boldsymbol{W}_{\Sigma_I} l_o$	$\boldsymbol{e}_o$ (1,000)	$\boldsymbol{W}_{\Sigma_I}$ (1,000 × 472)
(7)	(4)	word embedding $\boldsymbol{W}_{\Sigma_I} l_r$	$\boldsymbol{e}_r$ (1,000)	$\boldsymbol{W}_{\Sigma_I}$ (1,000 × 472)
(8)	(5)	word embedding $\boldsymbol{W}_{\Sigma_I} l_a$	$\boldsymbol{e}_a$ (1,000)	$\boldsymbol{W}_{\Sigma_I}$ (1,000 × 472)
(9)	(1),(6)	feature fusion $\mathrm{ReLU}(\boldsymbol{W}_1^o \boldsymbol{e}_o + \boldsymbol{W}_2^o \boldsymbol{v}_o)$ $-(\boldsymbol{W}_1^o \boldsymbol{e}_o - \boldsymbol{W}_2^o \boldsymbol{v}_o)^2$	$\boldsymbol{u}_o$ (1,000)	$\boldsymbol{W}_1^o$ (1,000 × 1,000) $\boldsymbol{W}_2^o$ (1,000 × 2,048)
(10)	(2),(7)	feature fusion $\mathrm{ReLU}(\boldsymbol{W}_1^r \boldsymbol{e}_r + \boldsymbol{W}_2^r \boldsymbol{v}_r)$ $-(\boldsymbol{W}_1^r \boldsymbol{e}_r - \boldsymbol{W}_2^r \boldsymbol{v}_r)^2$	$\boldsymbol{u}_r$ (1,000)	$\boldsymbol{W}_1^r$ (1,000 × 1,000) $\boldsymbol{W}_2^r$ (1,000 × 2,048)
(11)	(1),(8)	feature fusion $\mathrm{ReLU}(\boldsymbol{W}_1^a \boldsymbol{e}_a + \boldsymbol{W}_2^a \boldsymbol{v}_o)$ $-(\boldsymbol{W}_1^a \boldsymbol{e}_a - \boldsymbol{W}_2^a \boldsymbol{v}_o)^2$	$\boldsymbol{u}_a$ (1,000)	$\boldsymbol{W}_1^a$ (1,000 × 1,000) $\boldsymbol{W}_2^a$ (1,000 × 2,048)
(12)	(9),(10)	relationship embedding (Eq. A)	$\boldsymbol{v}_r^{'}$ (1,000)	f_r (3,000 → 1,000)
(13)	(9),(11)	attribute embedding (Eq. B)	$\boldsymbol{v}_a^{'}$ (1,000)	f_a (2,000 → 1,000)
(14)	(9),(10)	object embedding (Eq. C)	$\boldsymbol{v}_o^{'}$ (1,000)	f_s,f_o (3,000 → 1,000)

Table C. The details of the re-encoder function.

Index	Input	Operation	Output	Trainable Parameters
(1)	index vector	-	$\boldsymbol{x}$ (1,000)	-
(2)	(1)	inner product $\boldsymbol{D}^T \boldsymbol{x}$	$\boldsymbol{\alpha}$ (10,000)	$\boldsymbol{D}$(1,000 × 10,000)
(3)	(2)	softmax	$\boldsymbol{\alpha}$ (10,000)	-
(4)	(3)	weighted sum $\boldsymbol{D}\boldsymbol{\alpha}$	$\hat{\boldsymbol{x}}$(1,000)	$\boldsymbol{D}$(1,000 × 10,000)

A.3. Dictionary

The re-encoder function in Section 4.3 is used to re-encode a new representation $\hat{\boldsymbol{x}}$ from an index vector $\boldsymbol{x}$ and a dictionary $\mathcal{D}$, such operation is given in Table C. As shown in Table C (2) and (3) respectively, by given an index vector $\boldsymbol{x}$, we first do inner produce between each element in $\boldsymbol{D}$ with $\boldsymbol{x}$ and then use softmax to normalize the computed results. At last, the re-encoded $\hat{\boldsymbol{x}}$ is the weighted sum of each atom in $\boldsymbol{D}$ as $\sum_{k=1}^{K} \alpha_k \boldsymbol{d}_k$, K is set as 10,000.

Table D. The details of the common structure of the two decoders.

Index	Input	Operation	Output	Trainable Parameters
(1)	-	word label	$\boldsymbol{w}_{t-1}$ (10,369)	-
(2)	-	embedding set	$\mathcal{Z}$ (1,000 $\times$ M)	-
(3)	-	output of LSTM$_2$	$\boldsymbol{h}_{t-1}^2$ (1,000)	-
(4)	(1)	word embedding $\boldsymbol{W}_\Sigma \boldsymbol{w}_{t-1}$	$\boldsymbol{e}_{t-1}$ (1,000)	$\boldsymbol{W}_\Sigma$ (1,000 $\times$ 10,369)
(5)	(2)	mean pooling	$\bar{\boldsymbol{z}}$ (1,000)	-
(6)	(3),(4),(5)	concatenate	$\boldsymbol{i}_t$ (3,000)	-
(7)	(6)	LSTM$_1$ $(\boldsymbol{i}_t; \boldsymbol{h}_{t-1}^1)$	$\boldsymbol{h}_t^1$ (1,000)	LSTM$_1$ (3,000 $\rightarrow$ 1,000)
(8)	(2),(7)	$\boldsymbol{w}_a \tanh(\boldsymbol{W}_z \boldsymbol{z}_m + \boldsymbol{W}_h \boldsymbol{h}_t^1)$	$\boldsymbol{\beta}$ (M)	$\boldsymbol{w}_a$ (512), $\boldsymbol{W}_z$ (512$\times$1,000) $\boldsymbol{W}_h$(512$\times$1,000)
(9)	(8)	softmax	$\boldsymbol{\beta}$ (M)	-
(10)	(9),(2)	weighted sum $\mathcal{Z}\boldsymbol{\beta}$	$\hat{\boldsymbol{z}}$ (1,000)	-
(11)	(7),(10)	LSTM$_2$ $([\boldsymbol{h}_t^1, \hat{\boldsymbol{z}}]; \boldsymbol{h}_{t-1}^2)$	$\boldsymbol{h}_t^2$ (1,000)	LSTM$_1$ (3,000 $\rightarrow$ 1,000)
(12)	(11)	$\boldsymbol{W}_p \boldsymbol{h}_t^2 + \boldsymbol{b}_p$	$\boldsymbol{p}_t$ (10,369)	$\boldsymbol{W}_p$ (10,369 $\times$ 1,000) $\boldsymbol{b}_p$ (10,369)
(13)	(12)	softmax	P_t (10,369)	-

A.4. Decoders

We followed the language decoder proposed by [2] to set our two decoders of Eq. (4) and Eq. (5) in the main paper. Both decoders have the same architecture, as shown in Table D, except for the different embedding sets used as their inputs. For convenience, we introduce the decoders' common architecture without differentiating them between Eq. (4) and Eq. (5), and then detail the difference between them at the end of this section.

The implemented decoder contains two LSTM layers and one attention module. The input of the first LSTM contains the concatenation of three terms: word embedding vector $\boldsymbol{W}_\Sigma \boldsymbol{w}_{t-1}$, mean pooling of embedding set $\bar{\boldsymbol{z}}$, and the output of the second LSTM $\boldsymbol{h}_{t-1}^2$. We use them as input since they can provide abundant accumulated context information. Then, an index vector $\boldsymbol{h}_{t-1}^1$ is created by LSTM$_1$ in Table D (7), which will be used to instruct the decoder to put attention on suitable embedding of $\mathcal{Z}$ by an attention module. Given $\mathcal{Z}$ and $\boldsymbol{h}_{t-1}^1$, the formulations in Table D (8) and (9) can be applied for computing a M-dimension attention distribution $\boldsymbol{\beta}$, and then we can create the attended embedding $\hat{\boldsymbol{z}}$ by weighted sum as in (10). By inputting $\hat{\boldsymbol{z}}$ and $\boldsymbol{h}_{t-1}^1$ into LSTM$_2$ and implementing (11) to (13), the word distribution P_t can be got for sampling a word at time t.

For two decoders in Eq. (4) and Eq. (5), they only differ in using different embedding sets $\mathcal{Z}$ as inputs. In SGAE (Eq. (5)), $\mathcal{Z}$ is set as $\hat{\mathcal{X}}$. While in SGAE-based encoder-decoder (Eq. (4)), we have a small modification that the vector $\boldsymbol{z} \in \mathcal{Z}$ is set as follows: $\boldsymbol{z} = [\boldsymbol{v}', \hat{\boldsymbol{v}}]$, where $\boldsymbol{v}' \in \mathcal{V}'$ ($\mathcal{V}'$ is the scene graph-modulated feature set in Section 5.1), and $\hat{\boldsymbol{v}} \in \hat{\mathcal{V}}$ ($\hat{\mathcal{V}}$ is the re-encoded feature set in Section 5.1).

B. Details of Scene Graph

B.1. Sentence Scene Graph

For each sentence, we directly implemented the software provided by [1] to parse its scene graph. And we filtered them by removing objects, relationships, and attributes which appear less than 10 among all the parsed scene graphs. After filtering, there are 5,364 objects, 1,308 relationships, and 3,430 attributes remaining. We grouped them together and used word embedding matrix $\boldsymbol{W}_{\Sigma_S}$ in Table A to transform nodes' labels to continuous vector representations.

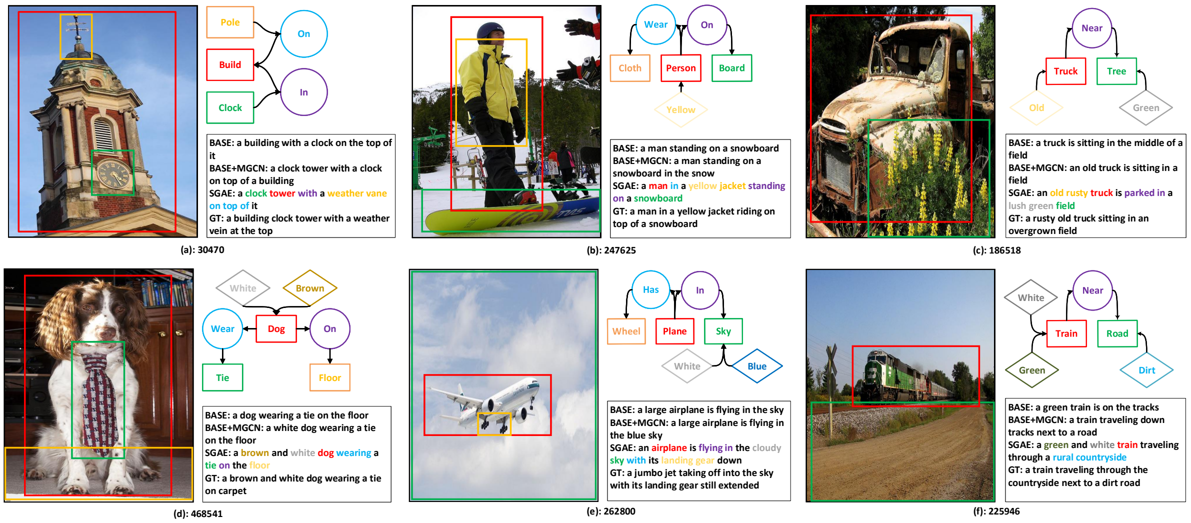
B.2. Image Scene Graph

Compared with sentence scene graphs, the parsing of image scene graphs is more complicated that we used Faster-RCNN as the object detector [3] to detect and classify objects, MOTIFS relationship detector [4] to classify relationships between objects, and one simple attribute classifier to predict attributes. The details of them are given as follows.

Object Detector: For detecting objects and extracting their RoI features, we followed [2] to train Faster-RCNN. After training, we used 0.7 as the IoU threshold for proposal NMS, and 0.3 as threshold for object NMS. Also, we selected at least 10 objects and at most 100 objects for each image. RoI pooling was used to extract these objects' features, which will be

Table E. The details of attribute classifier.

Index	Input	Operation	Output	Trainable Parameters
(1)	object RoI feature	-	$\boldsymbol{v}$ (2,048)	-
(2)	(1)	fc	$\boldsymbol{f}_1$ (1,000)	fc(2,048 → 1,000)
(3)	(2)	ReLU	$\boldsymbol{f}_1$ (1,000)	-
(4)	(3)	fc	$\boldsymbol{f}_2$ (103)	fc(1,000 → 103)
(5)	(4)	softmax	P_a (103)	-



BASE: a building with a clock on the top of it
BASE+MGCN: a clock tower with a clock on top of a building
SGAE: a clock tower with a weather vane on top of it
GT: a building clock tower with a weather vein at the top

(a): 30470

BASE: a man standing on a snowboard
BASE+MGCN: a man standing on a snowboard in the snow
SGAE: a man in a yellow jacket standing on a snowboard
GT: a man in a yellow jacket riding on top of a snowboard

(b): 247625

BASE: a truck is sitting in the middle of a field
BASE+MGCN: an old truck is sitting in a field
SGAE: an old rusty truck is parked in a lush green field
GT: a rusty old truck sitting in an overgrown field

(c): 186518

BASE: a dog wearing a tie on the floor
BASE+MGCN: a white dog wearing a tie on the floor
SGAE: a brown and white dog wearing a tie on the floor
GT: a brown and white dog wearing a tie on carpet

(d): 468541

BASE: a large airplane is flying in the sky
BASE+MGCN: a large airplane is flying in the blue sky
SGAE: an airplane is flying in the cloudy sky with its landing gear down
GT: a jumbo jet taking off into the sky with its landing gear still extended

(e): 262800

BASE: a green train is on the tracks
BASE+MGCN: a train traveling down tracks next to a road
SGAE: a green and white train traveling through a rural countryside
GT: a train traveling through the countryside next to a dirt road

(f): 225946

Figure A. Qualitative examples of different baselines. For each figure, the image scene graph is pruned to avoid clutter. The id refers to the image id in MS-COCO. Word colors correspond to nodes in the detected scene graphs.

used as the input to the relationship classifier, attribute classifier, and MGCN.

Relationship Classifier: We used the LSTM structure proposed in [4] as our relationship classifier. After training, we predicted a relationship for each two objects whose IoU is larger than 0.2.

Attribute Classifier: The detail structure of our attribute classifier is given in Table E. After training, we predicted top-3 attributes for each object.

For each image, by using predicted objects, relationships and attributes, an image scene graph can be built. As detailed in Section 6.1 of the main paper, the total number of used objects, relationships, and attributes here is 472, thus we used a 472 × 1,000 word embedding matrix to transform the nodes' labels into the continuous vectors as in Table B (6) to (8).

The codes and all these parsed scene graphs will be published for further research upon paper acceptance.

C. More Qualitative Examples

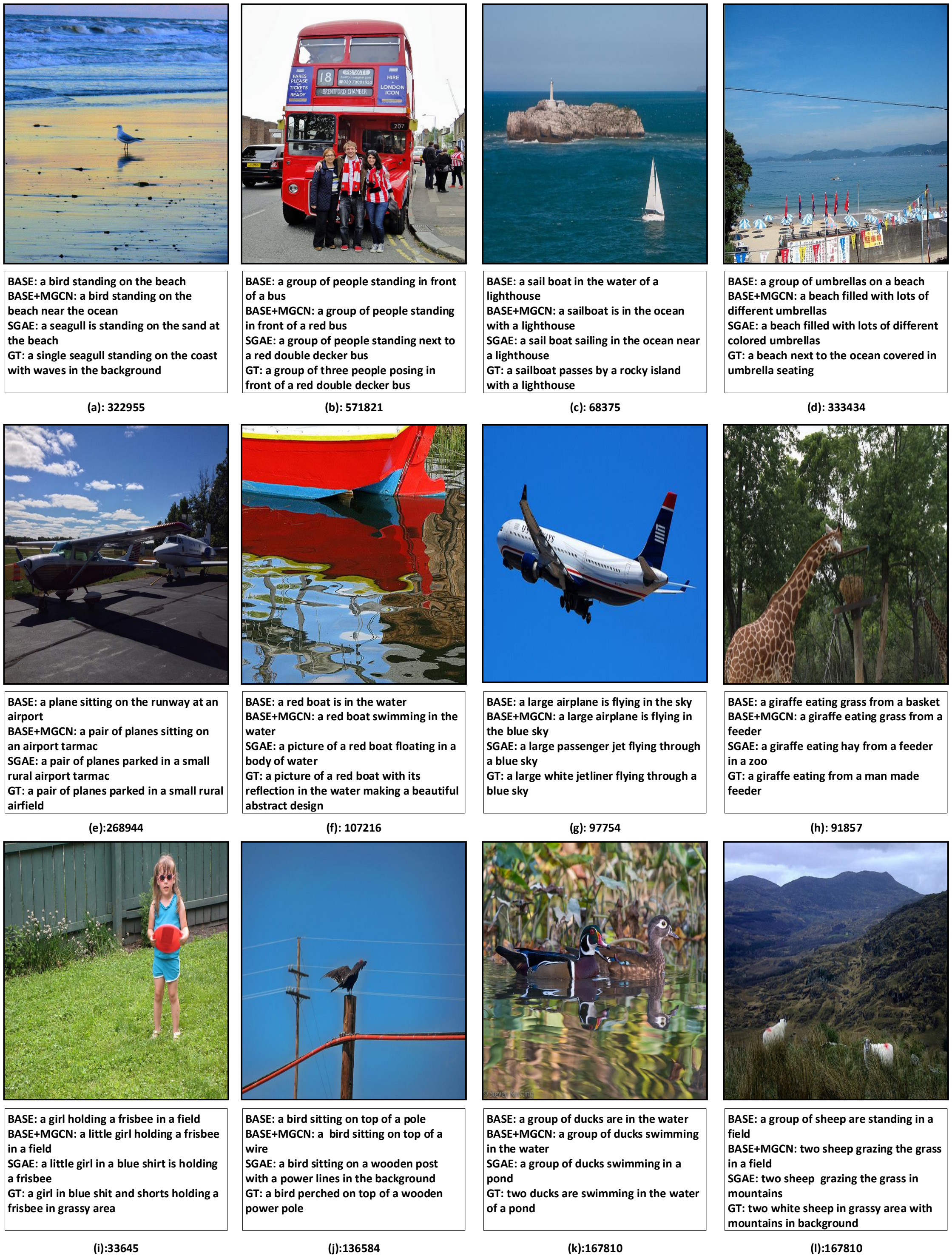
Figure A and B show more examples of generated captions of our methods and some baselines. We can find that the captions generated by SGAE prefer to use some more accurate words to describe the appeared objects, attributes, relationships or scenes. For instance, in Figure A (a), the object 'weather vane' is used while this object is not accurately recognized by the object detector; in Figure A (c), SGAE prefers the attribute 'old rusty'; in Figure B SGAE describes the relationship between boat with water as 'floating' instead of 'swimming'; and in Figure B, the scene 'mountains' is inferred by using SGAE.

References

[1] P. Anderson, B. Fernando, M. Johnson, and S. Gould. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, pages 382–398. Springer, 2016. 3

[2] 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. 3

[3] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1, 3



BASE: a bird standing on the beach
BASE+MGCN: a bird standing on the beach near the ocean
SGAE: a seagull is standing on the sand at the beach
GT: a single seagull standing on the coast with waves in the background

(a): 322955

BASE: a group of people standing in front of a bus
BASE+MGCN: a group of people standing in front of a red bus
SGAE: a group of people standing next to a red double decker bus
GT: a group of three people posing in front of a red double decker bus

(b): 571821

BASE: a sail boat in the water of a lighthouse
BASE+MGCN: a sailboat is in the ocean with a lighthouse
SGAE: a sail boat sailing in the ocean near a lighthouse
GT: a sailboat passes by a rocky island with a lighthouse

(c): 68375

BASE: a group of umbrellas on a beach
BASE+MGCN: a beach filled with lots of different umbrellas
SGAE: a beach filled with lots of different colored umbrellas
GT: a beach next to the ocean covered in umbrella seating

(d): 333434

BASE: a plane sitting on the runway at an airport
BASE+MGCN: a pair of planes sitting on an airport tarmac
SGAE: a pair of planes parked in a small rural airport tarmac
GT: a pair of planes parked in a small rural airfield

(e):268944

BASE: a red boat is in the water
BASE+MGCN: a red boat swimming in the water
SGAE: a picture of a red boat floating in a body of water
GT: a picture of a red boat with its reflection in the water making a beautiful abstract design

(f): 107216

BASE: a large airplane is flying in the sky
BASE+MGCN: a large airplane is flying in the blue sky
SGAE: a large passenger jet flying through a blue sky
GT: a large white jetliner flying through a blue sky

(g): 97754

BASE: a giraffe eating grass from a basket
BASE+MGCN: a giraffe eating grass from a feeder
SGAE: a giraffe eating hay from a feeder in a zoo
GT: a giraffe eating from a man made feeder

(h): 91857

BASE: a girl holding a frisbee in a field
BASE+MGCN: a little girl holding a frisbee in a field
SGAE: a little girl in a blue shirt is holding a frisbee
GT: a girl in blue shit and shorts holding a frisbee in grassy area

(i):33645

BASE: a bird sitting on top of a pole
BASE+MGCN: a bird sitting on top of a wire
SGAE: a bird sitting on a wooden post with a power lines in the background
GT: a bird perched on top of a wooden power pole

(j):136584

BASE: a group of ducks are in the water
BASE+MGCN: a group of ducks swimming in the water
SGAE: a group of ducks swimming in a pond
GT: two ducks are swimming in the water of a pond

(k):167810

BASE: a group of sheep are standing in a field
BASE+MGCN: two sheep grazing the grass in a field
SGAE: two sheep grazing the grass in mountains
GT: two white sheep in grassy area with mountains in background

(l):167810

Figure B. 12 qualitative examples of different baselines.

[4] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5831–5840, 2018. 3, 4

[5] Y. Zhang, J. Hare, and A. Prügel-Bennett. Learning to count objects in natural images for visual question answering. In *ICLR*, 2018. 1