Incorporating additional knowledge into image captioners

Xu, Yang

2021

Xu, Y. (2021). Incorporating additional knowledge into image captioners. Doctoral thesis, Nanyang Technological University, Singapore. <https://hdl.handle.net/10356/151726>

<https://hdl.handle.net/10356/151726>

<https://doi.org/10.32657/10356/151726>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 19:46:23 SGT

Incorporating Additional Knowledge into Image Captioners

Xu Yang
School of Computer Science and Engineering
2021

Incorporating Additional Knowledge into Image Captioners

Xu Yang

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2021

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

15/01/2021

.....

Date

yangku

.....

Xu Yang

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

15/01/2021

.....

Date

.....

Prof. Hanwang Zhang

Authorship Attribution Statement

This thesis contains material from 2 papers published in or submitted to the following peer-reviewed journals and from 3 papers accepted at conferences. Among all these journal and conference papers, I am the first author.

Chapter 3 is published as [Xu Yang, Hanwang Zhang, Jianfei Cai](#). “Shuffle-Then-Assemble: Learning Object-Agnostic Visual Relationship Features.” Proceedings of the 15th European Conference on Computer Vision (**ECCV, Poster**). Munich, Germany. 2018.

The contributions of the co-authors are as follows:

- Prof Hanwang Zhang and Jianfei Cai pointed out the research direction.
- I implemented the experiments and prepared the draft of the research paper and Prof Hanwang Zhang and Jianfei Cai helped refine the draft.

Chapter 4 is published as [Xu Yang, Hanwang Zhang, Jianfei Cai](#). “Auto-encoding and Distilling Scene Graphs for Image Captioning.”, IEEE Transactions on Pattern Analysis and Machine Intelligence (**TPAMI**), Accepted. Digital Object Identifier: 10.1109/TPAMI.2020.3042192.

The contributions of the co-authors in this paper are as follows:

- Prof Hanwang Zhang and Jianfei Cai pointed out the research direction.
- I implemented the experiments and prepared the draft of the research paper and Prof Hanwang Zhang and Jianfei Cai helped refine the draft.

Chapter 5 is published as [Xu Yang, Chongyang Gao, Hanwang Zhang, Jianfei Cai](#). “Hierarchical Scene Graph Encoder-Decoder for Image Paragraph Captioning.” Association for Computing Machinery (ACM)’s annual conference on multimedia (**ACMMM, Poster**). Seattle, United States, 2020. The contributions of the co-authors in this are as follows:

- Prof Hanwang Zhang and Jianfei Cai pointed out the research direction.
- I implemented the most experiments and prepared the draft of the research paper. Chongyang Gao wrote a few part codes about the measurements. Prof Hanwang Zhang and Jianfei Cai helped refine the draft.

Chapter 6 is published as [Xu Yang, Hanwang Zhang, Jianfei Cai](#). “Learning to Collocate Neural Modules for Image Captioning.” Proceedings of the IEEE Conference on International Conference on Computer Vision (**ICCV, Poster**). Seoul, Korea, 2019.

The contributions of the co-authors are as follows:

- Prof Hanwang Zhang and Jianfei Cai pointed out the research direction.
- I implemented the experiments and prepared the draft of the research paper and Prof Hanwang Zhang and Jianfei Cai helped refine the draft.

Chapter 7 is submitted to TPAMI as [Xu Yang, Hanwang Zhang, Jianfei Cai. “Deconfounded Image Captioning: A Causal Retrospect.”](#) The contributions of the co-authors in this paper are as follows:

- Prof Hanwang Zhang and Jianfei Cai pointed out the research direction.
- I implemented the experiments and prepared the draft of the research paper and Prof Hanwang Zhang and Jianfei Cai helped refine the draft.

15/01/2021

.....

Date

yangxu

.....

Xu Yang

Acknowledgements

First and foremost, I want to express my greatest gratitude to both of my supervisors: Prof. Hanwang Zhang and Prof. Jianfei Cai, for their generous advice and continuous support. I can not forget the tough days before the deadlines of submissions that I am struggling with my research. Without their patient guidance and kind advice, I can not fulfill these research works and complete the corresponding research papers. Their encouraging academic and working attitudes have influenced my whole Ph.D. study.

I would thank my collaborators, Dr. Jiuxiang Gu, Ph.D. Xiangxi Shi, Ph.D. Kaihua Tang, Ph.D. Yujun Cai, Yuedong Chen, and Chongyang Gao for their precious suggestions and continuous encouragements on my research studies. Besides them, I would like to thank Dr. Jiasen Lu, who shares his research experience with me and discuss with me about deep learning, computer vision, and natural language processing, which largely inspires my research.

I would like to acknowledge School of Computer Science, Nanyang Technological University (NTU), for providing me NTU Research Scholarship to complete my Ph.D. study. I also want to express my gratitude to my colleagues from Multimedia and Interactive Computing Lab (MICL), for providing a complacent research and study environment.

I really enjoy and cherish the days when I study, work, entertain, and live in NTU. Also, I meet so many wonderful friends during my Ph.D. study, who support me a lot and give me invaluable suggestions and consoles.

I would like to express my thanks to them, Chi Zhang, Sheng Yang, Keyu Chen, Ruibo Li, Xiaofeng Yang, Zichuan Liu, Kaihua Tang, Xinting Hu, Jiaxin Qi, Yulei Niu, Chuanxia Zheng, Yujun Cai, Simeng Han, Hongyu Zhou, Ce Liang, Xingmeng Pan, Shuxin Cao, Ziwei He, Yuhan Ma, Boyi Shen and those who are not listed here.

I would like to thank my parents, for providing unconditional love for me throughout my life.

Abstract

Image Captioning (IC) is one of the most important visual reasoning tasks which lies at the intersection of computer vision and natural language processing that requires a machine to generate a fluent caption to correctly describe the given image. Compared with preliminary template/statistic based captioners, modern encoder-decoder framework based captioners have achieved significant improvements. Such a framework deploys a convolutional neural network (CNN) and a recurrent neural network (RNN) as the visual encoder for extracting visual features and the language decoder for generating words from the extracted features, respectively. However, in the encoder-decoder pipeline, since the visual encoder is well pre-trained by object detection task, the learned visual representations will focus more on objects' categories while neglecting other important knowledge like objects' attributes or relationships. As a result, most of the existent captioning systems prefer to recognize objects first and then infer the other words from the recognized objects. Thus, these captioning systems prefer to generate rigid and inflexible descriptions, or even worse, these systems are likely to overfit to the dataset bias.

In this thesis, we propose some novel captioners which exploit different additional knowledge to generate more descriptive and less biased captions to alleviate the abovementioned limitations. In this thesis, we have the following major contributions:

- A novel **Shuffle-Then-Assemble (STA)** pre-training strategy is proposed for learning object-agnostic visual features to construct less biased scene graphs. In this strategy, we discard all the triplet relationship annotations in an image and leave unpaired object domains without object alignments. Then we try to recover possible obj1-obj2 pairs for learning features. A cycle of residual transformations between the two domains is also designed where the identity mappings encourage the RoI features to capture shared but not object-specific visual patterns. In this way, object-agnostic visual features can be learned and less biased scene graphs can be constructed.

-
- A novel **Scene Graph Auto-Encoder (SGAE)** is proposed to incorporate the language inductive bias as the additional knowledge into the encoder-decoder image captioning framework for more human-like captions. We preserve such language inductive bias into a dictionary set through reconstructing sentences from these sentences’ scene graphs. Then during image captioning, we extract a visual scene graph from the given image and then retrieval the preserved language inductive bias from the dictionary set for more descriptive captions.
 - A **Hierarchical Scene Graph Encoder-Decoder (HSGED)** paragraph captioner is proposed to generate coherent and distinctive paragraphs. In this model, we also exploit the scene graph as the additional knowledge, which acts as the “script” to guide the paragraph generation. In particular, a sentence scene graph RNN (SSG-RNN) is designed to generate sub-graph level topics, which constrains the word scene graph RNN (WSG-RNN) to complete the corresponding sentences. Also, we propose irredundant attention in SSG-RNN and inheriting attention in WSG-RNN to respectively to improve the possibility of abstracting topics from rarely described sub-graphs and to generate more grounded sentences with the abstracted topics, both of which give rise to more distinctive paragraphs.
 - A novel captioner: **learning to Collocate Visual-Linguistic Modules (CNM)** is proposed to imitate humans who compose captions by first structuring a sentence pattern like *STH DO STH AT SOMEPLACE* and then filling in the detailed descriptions. To achieve this, we propose a neural module network for generating captions that we design specific modules for generating nouns, adjectives, and verbs. Also, soft module fusion, multi-step module execution, and a linguistic loss are deployed to make module controller be more faithful to part-of-speech collocations. In this way, our CVLNM can not only generate more correct captions, but also be more robust to fewer training samples compared with the concurrent state of the art captioners.
 - A novel **Deconfounded Image Captioning (DIC)** model is proposed to generate less biased captions. We follow the framework of causal inference to design this captioner, which can alleviate the negative effects brought by the dataset bias. We also analyze why modern image captioners are easily affected by dataset bias from the perspective of confounding effect and then

propose DICv1.0, which exploits additional commonsense knowledge as the mediator to transmit information from the image to the caption, to alleviate the negative effects brought by dataset bias. More importantly, we retrospect the major captioners in our 6-year-old IC community from the causal view and show how this causal retrospect enlightens us to develop our DICv1.0.

Contents

Acknowledgements	vi
Abstract	viii
List of Figures	xv
List of Tables	xx
1 Introduction	1
1.1 Image Captioning: from Recognition to Reasoning	1
1.2 Neural Network Based Captioners	2
1.3 Challenges and Contributions	4
1.4 Outline of the Thesis	5
2 Literature Review	9
2.1 Overview	9
2.2 Image Captioning	9
2.2.1 Image Paragraph Captioning	11
2.2.2 Unpaired Image Captioning	12
2.3 Scene Graph	12
2.3.1 Scene Graphs Generation	12
2.3.2 Exploitation of Scene Graph.	13
2.4 Graph Convolutional Networks	14
2.5 Memory Networks	14
2.6 Knowledge Distillation	15
2.7 Neural Module Networks	15
3 Shuffle-Then-Assemble: Learning Object-Agnostic Visual Relationship Features	17
3.1 Introduction	17
3.2 Method	19
3.2.1 Visual Relationship Model	20
3.2.2 Shuffle-Then-Assemble Feature Learning	21
3.2.3 Implementation Details	24

3.2.3.1	Network Architecture	24
3.2.3.2	Training	24
3.3	Experiments	25
3.3.1	Datasets and Metrics	25
3.3.2	Settings	26
3.3.2.1	Comparing Methods	28
3.3.3	Results and Analysis	29
3.3.4	Discussion and Future Work	31
4	Auto-encoding and Distilling Scene Graphs for Image Captioning	33
4.1	Introduction	33
4.2	Encoder-Decoder Revisited	36
4.3	Auto-Encoding Scene Graphs	39
4.3.1	Scene Graphs	39
4.3.2	Attentional Graph Convolutional Network	40
4.3.3	Dictionary	42
4.3.4	Unpaired Image Captioning	43
4.3.5	Training of SGAE	44
4.4	SGAE-based Encoder-Decoder	45
4.4.1	Multi-modal AGCN	46
4.4.2	Knowledge Distillation	47
4.4.3	Training and Inference	48
4.5	Experiments	49
4.5.1	Datasets and Metrics	49
4.5.2	Ablation Studies	50
4.5.2.1	Architecture	50
4.5.2.2	Language Corpus	53
4.5.2.3	Sentence Reconstruction	56
4.5.3	Comparisons with State-of-The-Arts	57
4.5.4	Unpaired Image Captioning	58
4.5.4.1	Comparing Methods	58
4.5.4.2	Results and Analysis	59
5	Hierarchical Scene Graph Encoder-Decoder for Image Paragraph Captioning	61
5.1	Introduction	61
5.2	Hierarchical Scene Graph Encoder-Decoder	64
5.2.1	Hierarchical Scene Graph Encoder	65
5.2.1.1	Scene Graphs	65
5.2.2	Hierarchical Decoders	67
5.2.2.1	Top-Down Attention Network	67
5.2.2.2	SSG-RNN	68
5.2.2.3	Irredundant Attention	70
5.2.2.4	WSG-RNN	71

5.2.2.5	Inheriting Attention	71
5.2.3	Training Objectives	72
5.3	Experiments	74
5.3.1	Datasets, Settings, and Metrics	74
5.3.1.1	Datasets	74
5.3.1.2	Settings	74
5.3.1.3	Metrics	75
5.3.2	Ablative Studies	75
5.3.2.1	Comparing Methods	76
5.3.2.2	Results on Similarity	76
5.3.2.3	Results on Diversity	77
5.3.2.4	Results on Fine-grain Degree	77
5.3.2.5	Human Evaluation.	78
5.3.3	Comparisons with State-of-The-Arts	79
5.3.3.1	Comparing Methods.	79
5.3.3.2	Result Analysis.	80
5.3.4	Discussion.	80
6	Learning to Collocate Visual-Linguistic Modules for Image Cap-	82
	tioning	
6.1	Introduction	82
6.2	Collocation of Visual-Linguistic Neural Modules	86
6.2.1	Distinguishable Feature Extraction Modules	88
6.2.1.1	OBJECT Module	88
6.2.1.2	ATTRIBUTE Module	88
6.2.1.3	RELATION Module	89
6.2.1.4	FUNCTION Module	90
6.2.2	Module Controller	91
6.2.2.1	Module Attention	92
6.2.2.2	Soft Fusion	92
6.2.3	Syntax Loss	94
6.2.4	REASON Module	95
6.2.5	Training and Inference of CVLNM	96
6.3	Experiments	97
6.3.1	Datasets and Settings	97
6.3.2	Ablation Studies	98
6.3.2.1	Research Questions and the Corresponding Baselines	99
6.3.2.2	Evaluation Metrics for Answering Questions	100
6.3.2.3	Empirical Answers	103
6.3.3	Comparisons with the State-of-The-Art Models	106
6.3.4	Few-Shot Image Captioning	108
6.4	Conclusions	108
7	Deconfounded Image Captioning: A Causal Retrospect	110

7.1	Introduction	110
7.2	Preliminaries: Deconfounding	114
7.2.1	The Backdoor Adjustment	114
7.2.2	The Front-door Adjustment	116
7.3	Related Work: A Causal Retrospect	118
7.4	Deconfounded Image Captioning	121
7.4.1	Choosing $\mathcal{Z}$	121
7.4.2	Normalized Weighted Geometric Mean	123
7.4.3	EXPT Modules for Expectations	124
7.4.4	Network Architectures	125
7.4.5	Training Objectives	126
7.5	Experiments	126
7.5.1	Datasets	126
7.5.2	Ablation Studies	127
7.5.2.1	Deconfounding Techniques	127
7.5.2.2	Network Size	131
7.5.3	Comparisons with State of the Art	132
7.6	Epilogue	134
8	Conclusion and Future Works	136
8.1	Conclusion	136
8.2	Future Works	138
8.2.1	Auto-Parsing Network (APN)	139
8.2.2	Causal Attention	140
	List of the Publications	144
	Bibliography	146

List of Figures

3.1	(a) The triplet annotations of visual relationships in an image. (b) The key idea of the proposed Shuffle-Then-Assemble strategy is to discard the paired annotation of any relationship and leave two unpaired object domains. (c) Comparisons between the original feature maps from base CNN (top) and the desired object-agnostic ones (bottom) obtained by our pre-training (averaged over all the channels). We can see that our feature maps focus on the relationships between objects.	18
3.2	The overview of the relationship model improved by the proposed Shuffle-Then-Assemble pre-training strategy. The goal at the pre-training stage is to learn the Object-Agnostic (OA) conv-layers using Shuffle-Then-Assemble objective. Then, the traditional supervised training for the relation model can be considered as the fine-tuning stage using the desired OA feature map.	20
3.3	(a) The view of unsupervised domain transfer for Shuffle-Then-Assemble . It contains a cycle of transformations $F: A \mapsto B$ and $G: B \mapsto A$, and a pair of discriminators D_A and D_B to measure the quality of the transfer. (b) Qualitative transfer results. The directed arrow indicates the nearest-neighbor RoI in the target domain to the RoI from the source domain. (c) The residual architecture of the transformation network. (d) The architecture of the discriminator.	22
3.4	We evaluate relationship prediction task using four different experiment settings: supervised, supervised (Det), weakly-supervised and zero-shot. “?” denotes the relationship to be predicted.	26
3.5	Qualitative feature maps of two relationships on VRD. For each one, three kinds of feature maps are visualized by averaging over the 512 channels. We can see that by using the proposed Shuffle-Then-Assemble (STA), the RoI features are less likely biased to the objects and more focused on the regions of interaction of the two objects. Moreover, the observation is consistent with diverse relationship appearances.	27
3.6	Performances (R@100%) of relation classification of the four relation types using the different methods in the supervised setting.	29

3.7	Qualitative feature maps of two zero-shot relationships on VRD. For each one, two feature maps of Base+OA with wrong prediction and STA with correct prediction are visualized by averaging over the 512 channels. We can see that by using the proposed Shuffle-Then-Assemble (STA), the RoI features are less likely biased to the objects and more focused on the regions of interaction of the two objects.	32
4.1	The sketch of our SGAE, which incorporates the auto-encoding of scene graphs (blue arrows) into the conventional encoder-decoder framework for image captioning (red arrows). The trainable shared dictionary $\mathcal{D}$ encodes the language inductive bias. In $\mathcal{D} \rightarrow \mathcal{S}$, the dark dot line from the blue arrow to the red arrow indexes the Knowledge Distillation strategy, which teaches the encoder-decoder to reason as the auto-encoder. The colors of the words in $\mathcal{S}$ correspond to the nodes in image and sentence scene graphs.	34
4.2	Top: the conventional encoder-decoder. Bottom: our proposed encoder-decoder, where the novel SGAE embeds the language inductive bias in the dictionary set, which is shared to the image captioner, and in the decoder $\mathcal{S}$ -RNN of SGAE, whose knowledge is distilled into the decoder $\mathcal{I}$ -RNN of the image captioner.	37
4.3	The sketch of the proposed Attentional Graph Convolutional Network. In particular, this is a spatial graph convolutional network, where the colored neighborhood is “convolved” into the resultant embedding. The dashed lines in (b) and (c) denote that these knowledge flows are softly filtered by the calculated attention weights. . .	40
4.4	The sketch of the re-encoder function R . The black dashed block demonstrates this function. The top part shows how “imagination” is achieved by re-encoding: the green line draws the generated phrase by re-encoding, while the red line draws the one without re-encoding. For convenience, we only sketch the attribute dictionary $\mathcal{D}_A$ here. .	42
4.5	The sketch of Knowledge Distillation, where Con denotes the concatenation operation.	47
4.6	Qualitative examples of different methods. The comparing methods are Base, the reimplemented version of Up-Down [1], MAGCN, the reimplemented version of GCN-LSTM [2], SGAE, the upgraded version of our preliminary work [3], and our SGAE-KD. 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.	53
4.7	Captions generated by using different language corpora. The id refers to the image id in MS-COCO.	53
4.8	Two examples of unpaired image captioning. The id refers to the image id in MS-COCO.	56

5.1	Illustrations of our motivation. (a) Comparisons between single sentence captioning and the informative paragraph. (b) Topics and script of the paragraph in (a). (c) A paragraph generated by flat RNN. (d) The paragraph generation process of HRNN without hierarchical constraint where the top and bottom parts denote the higher and lower RNNs, respectively. (e) The paragraph generation process of our HSGED, where the grey color means this part will not be attended.	62
5.2	The illustrations of three sub-graphs around different objects. The dash lines connect the shared nodes which facilitate the system to generate coherent sub-graph topics.	64
5.3	The sketches of three sub-graph level embeddings.	66
5.4	In HSGED, SSG-RNN and WSG-RNN are both built based upon the top-down attention network with different attention mechanisms and inputs: sub-graph level and node level embedding sets $\mathbf{U}$, $\mathbf{X}$, respectively.	69
5.5	Two qualitative examples. The bottom left and bottom right parts show the sub-graph topics and the generated paragraph, respectively. The colors highlight the alignments between the sentences and the sub-graph topics.	75
5.6	The radar charts illustrate the ratios of four part of speech (PoS) in paragraphs generated by four methods.	78
5.7	The pie charts show the results of the human evaluation on the quality of the generated paragraphs.	79
6.1	The motivation of our learning to Collocate Visual-Linguistic Neural Modules (CVLNM) for image captioning: the principle of the modular design, which regularizes the diverse training effectively. . .	83
6.2	By comparing our CVLNM with the baseline Module/O, which only uses OBJECT module in the encoder (an upgraded version of Up-Down [1]), we have three interesting findings in tackling the dataset bias: (a) more accurate grammar, where % in the bottom box for each image denotes the frequency of a certain pattern in MS-COCO, (b) more descriptive attributes, and (c) more accurate object interactions. The ratio ./ denotes the percentage of co-occurrence, <i>e.g.</i> , “sheep+field”/“sheep” = 28% means that “sheep” and “field” contributes the 28% occurrences of “sheep”. We can see that CVLNM outperforms Module/O even with highly biased training samples. . .	84
6.3	The encoder-decoder pipeline of our image captioner that Collocates Visual-Linguistic Neural Modules (CVLNM). The dash lines from RNN to FUNCTION module and the module controller mean that both of these sub-networks require the contextual knowledge of the partially observed sentences, which is accumulated by the RNN. . .	87

6.4	Three visualizations show how RELATION module generates relation specific words. The red box in each image is the attended image region when RELATION module generates a relation specific word. The thickness of lines connecting different boxes is determined by the soft attention weights computed by the self-attention technique in Eq. (6.4), <i>i.e.</i> , the larger the attention weight between two boxes, the thicker the line between these boxes.	90
6.5	The detailed structure of our module controller, which consist of three attention networks and a self-attention (SA) network to generate four soft weights to fuse the attended features of four modules (<i>i.e.</i> $\hat{v}_O$, $\hat{v}_A$, $\hat{v}_R$, $\hat{v}_F$) into a single feature $\hat{v}$	91
6.6	Each pie chart shows the comparison of two methods in human evaluation.	103
6.7	Visualizations of the caption generation processes of three models: Module/O, SA+ $\mathcal{Z}+L_s$, and CVLNM. Note that CVLNM is equal to incorporating REASON module into the baseline SA+ $\mathcal{Z}+L_s$. Different colours refer to different modules, <i>i.e.</i> , red for OBJECT module, blue for ATTRIBUTE module, purple for RELATION module, and black for FUNCTION module. Since Module/O is the single-module based model, we directly use black boxes to show the attended regions. For simplicity, we only show the module with the largest soft module collocation weight (Eq. (6.11)) and the corresponding region with the largest attention weight (Eq. (6.8)).	104
7.1	The toy example about how the confounder disturbs the learning of the true causal effect, where $\mathcal{X}$, $\mathcal{L}$, and $\mathcal{P}$ denote image features, label, and position, respectively. (a) A training dataset constructed with a bias that “0” is mostly on the left and “1” is mostly on the right. (b) If we train a digit classifier by the training set in (a), it may not capture the true causal effect $\mathcal{X} \rightarrow \mathcal{L}$, while exploiting the “position knowledge” to make predictions (<i>e.g.</i> , when recognizing pixels on the left, predicting the label as “0”) through the backdoor path $\mathcal{X} \leftarrow \mathcal{P} \rightarrow \mathcal{L}$ induced by the confounder $\mathcal{P}$. (c) As a result, if we feed this classifier with the images where “1” appears on the left and “0” on the right, the incorrect predictions will be made.	111
7.2	Three examples show how the confounder $\mathcal{D}$ induces the spurious correlation. The probability denotes the percentage of co-occurrence of two words in the training set, <i>e.g.</i> , $P(\text{person} ID_{\text{Remote}})$ means that “person” and “remote” contributes the 54% occurrences of “remote”. Here we use ID to signify the appearing visual concept. The top caption is generated by Up-Down [1] and the bottom one is generated by our UD-DICv1.0. Red and blue denote the wrong and right words, respectively, and green denotes the concept which may cause the wrong word.	112

7.3	Compared with the backdoor graph, the front-door graph requires an additional mediator $\mathcal{Z}$ to transmit the knowledge between $\mathcal{X}$ and $\mathcal{L}$. To deconfound this front-door based captioner, we should chain together the two partial causal effects $P(\mathcal{Z} do(\mathcal{X}))$ and $P(\mathcal{L} do(\mathcal{Z}))$ by blocking two backdoor paths: $\mathcal{X} \leftarrow \mathcal{D} \rightarrow \mathcal{L} \leftarrow \mathcal{Z}$ and $\mathcal{Z} \leftarrow \mathcal{X} \leftarrow \mathcal{D} \rightarrow \mathcal{L}$, respectively. The first one is already blocked due to the collider at $\mathcal{L}$ and the second one can be blocked by cutting off the link $\mathcal{X} \rightarrow \mathcal{Z}$.	116
7.4	The causal retrospect of the major captioners. The past/present/future are colored by black/blue/red, respectively. We will look ahead the recursive future in Section 7.6. In these causal graphs, the shadowed variables signify that it is hard to split them for applying the backdoor adjustment, <i>e.g.</i> , $\mathcal{D}$ in (a) and $\mathcal{S}$ in (g).	118
7.5	(a) The causal graph of our DICv1.0 captioner, where $\mathcal{Z}$ and $\mathcal{S}$ denote the commonsense structure and the caption vocabulary, respectively. There are two confounders in this model, which are $\mathcal{D}$ and $\mathcal{S}$. (b) The sketch of our DICv1.0’s decoder, where ATT/EXPT represent attention/expectation modules. (c) The sketch of EXPT($\mathcal{Z}$) (Eq. (7.20)), where $\otimes$ denotes the matrix multiplication $\bar{\mathbf{z}} = \mathbf{Z}\mathbf{p}$.	121
7.6	The accuracy of the gender, attribute, and action words when some specific visual concepts appear.	129
7.7	The pie charts each comparing two ICs.	129
7.8	Eight examples about gender, object, attribute, and action biases in captions generated by different models. “GT” denotes the ground truth caption. The visual concepts which may cause biases are colored by green. The red and blue words represent the inconsistent and consistent words, respectively. It can be found that our UD-DICv1.0 can generate the most consistent captions compared with UD and UD-BD.	130
8.1	The sub-graph (a) visualizes the segmentation of a fully connected graph containing five nodes: from s_1 to s_5 . The sub-graph (b) shows the process of constructing a hierarchical tree using APN.	140
8.2	Top: an example of image captioner with a self-attention and a top-down attention modules. Bottom: the corresponding causal graph. The reason why the prediction is “riding” but not “driving” is explained in Figure 8.4.	141
8.3	Before pre-training (<i>e.g.</i> , LXMERT [4]), attentions are correct (blue). After pre-training, attentions are wrong (red). This is because the co-occurrences of some concepts appear much more often than others, <i>e.g.</i> , “Sport+Man” appears 213 times more than “Sport+Screen” in the pre-training data.	142
8.4	This expands the causal links of the confounding path $X \leftarrow \text{-----} \rightarrow Y$ in Figure 8.2.	142

List of Tables

3.1	The performances (Recall@K%) of compared methods on two datasets under Supervised setting and Supervised (Det) setting.	29
3.2	The performances (Recall@K%) of compared methods on two datasets under Weakly Supervised setting (WS) and Zero-Shot setting(ZS).	29
3.3	Computed ratios of two kinds of feature maps	30
4.1	The performances of various methods on MS-COCO Karpathy split. The metrics: B@N, M, R, C and S denote BLEU@N, METEOR, ROUGE-L, CIDEr-D and SPICE. The best results are marked in boldface.	51
4.2	The performances of using different language corpora.	52
4.3	Results of human evaluation, where each row compares two methods. In each column, “1st”/“2nd” lists the percentage of votes which prefer the fist/second method in that row and “Similar” denotes the votes for “two captions are similar”.	55
4.4	The performances of the sentence reconstruction using different scene graphs.	56
4.5	The performances of various methods on MS-COCO Karpathy split trained by cross-entropy loss only.	56
4.6	The performances of various methods on the online MS-COCO test server.	57
4.7	The performances of unpaired image captioning.	59
5.1	The performances of various baselines on Stanford image paragraph dataset. The metrics: B@N, M, and C, denote BLEU@N, METEOR, and CIDEr-D, respectively.	76
5.2	The diversity scores (the larger, the more diverse) of various baselines. GT denotes ground-truth.	77
5.3	The object and relation SPICE score of various baselines on Stanford image paragraph dataset.	78
5.4	The performances of various state-of-the-art methods. The top and middle sections show the results sampled by beam search and tri-gram repetition penalty, respectively. The model in the bottom section uses a different RL reward.	80
6.1	The sizes of different trainable parameters.	98

6.2	The performances of various baselines on Karpathy split. The metrics: B@4, M, R, C, S, CHs, and CHi denote BLEU@4, METEOR, ROUGE-L, CIDEr-D, SPICE, CHAIRs, and CHAIRi, respectively. The symbols $\uparrow$ and $\downarrow$ mean the higher the better and the lower the better, respectively.	102
6.3	The recalls (%) of five part-of-speech words, where nouns correspond to OBJECT module; adjectives correspond to ATTRIBUTE module; and verbs, prepositions, and quantifiers correspond to RELATION module.	102
6.4	The accuracy (%) of the predicted module layout by the module controller. The column “Average” means the average accuracy of all four modules.	102
6.5	The performances of various methods on MS-COCO Karpathy split trained by cross-entropy loss only (left part) and the mixture of cross-entropy loss and self-critical reward (right part).	106
6.6	The performances of various compared image captioners on the on-line MS-COCO test server.	106
6.7	The CIDEr-D deterioration of using fewer training sentences. The values in the bracket are the real CIDEr-D scores.	107
7.1	The performances of various ablation studies on MS COCO Karpathy split. The metrics: B@4, M, R, C, S, CHs, CHi, A@Gen, A@Attr, and A@Act denote BLEU@4, METEOR, ROUGE-L, CIDEr-D, SPICE, CHAIRs, CHAIRi, the accuracy of gender, attribute, and action words. The symbols $\uparrow$ and $\downarrow$ mean the higher the better and the lower the better, respectively.	128
7.2	The performances of various ablation studies about the network size on MS COCO Karpathy split. For the convenience of the comparisons, we add an index before each model for better reference. . . .	132
7.3	The performances of the state of the art ICs on Karpathy split. The top and bottom parts report the performances trained by CIDEr-D computed from 5 captions and the whole training set, respectively. “Group” specifies each IC’s category according to Fig. 7.4. The symbol $\dagger$ means the re-implemented model.	133
7.4	The performances of single methods on the online MS COCO test server. The top and bottom parts report the performances trained by CIDEr-D computed from 5 captions and the whole training set, respectively.	134

Chapter 1

Introduction

1.1 Image Captioning: from Recognition to Reasoning

As humans first learn single words from recognizing the category or attribute of a single object and then learn the whole sentences from composing these single elements, nowadays intelligent systems are also hoped to evolve as such learning process: from recognition to reasoning. Since AlexNet [5] demonstrated the power of the deep neural network in image classification, which is one of the most fundamental computer vision task, low-level recognition tasks like detection and segmentation have also made great progress by more and more advanced deep architectures like Faster RCNN [6], YOLO [7], FCN [8], and U-Net [9]. The success of these low-level recognition tasks builds solid cornerstones which permit the machine to reason as humans, *e.g.*, to infer the relationships among objects, to summarize the complex scenes, and to incorporate emotions.

To empower machines with the ability of reasoning, researchers have proposed various visual reasoning tasks, *e.g.*, image captioning(IC) [10], visual question and answering (VQA) [11], and visual grounding (VG) [12]. Among these tasks, the efforts of collecting IC data are usually cheaper than VQA or VG since human annotators can quickly produce a caption given an image, while VQA collectors need to design suitable questions about each image and then answer these questions or VG collectors need to draw bounding boxes for specific regions of the images

based on the descriptions. Furthermore, since image-caption pairs contain abundant implicit visual-language alignments, such datasets [10, 13] are usually used in pre-training large-scale visual reasoning architectures [4, 14] for solving sequential visual reasoning tasks. Based on these observations, we can treat IC as one most fundamental visual reasoning task and the focus of this thesis is about Image Captioning. Specifically, we retrospect the famous neural image captioner, analysis the major limitations of them, and then propose certain potential solutions, which are the contributions of this thesis, to ameliorate these limitations.

1.2 Neural Network Based Captioners

Most of the earlier non-deep network based captioners are template/statistic based methods [15–18] which first recognize the objects, relationships, and attributes from the image and then fill these words into the templates parsed from language statistics. However, two impediments limit the performances of such template/statistic models. Firstly, the accuracies of these earlier recognition algorithms are not favorable that many important visual patterns may be missed or falsely recognized and the resultant captions will also not ground to the image. Secondly, the visual recognition and template generation modules are separately optimized and thus the two parts may be inconsistent and thus incorrect captions may be generated, *e.g.*, the forms of words like the third-person-singulars of verbs or the plurals of nouns may be wrong.

More or less inspired by the achievements in neural machine translation [19], deep neural network based encoder-decoder frameworks are also designed and developed to solve image captioning [20], which achieve much better performances than those early template/statistic based captioners. A classic encoder-decoder captioner contains a visual encoder to extract high-level representations from images and a language decoder to generate the captions from the extracted features. The visual encoder is usually a well-trained CNN on image classification or object detection [6, 21]. In this way, the visual patterns of an image will be more accurately and exhaustively recognized and thus the generated captions may contain fewer errors or miss fewer objects. The language decoder is built upon various recurrent networks like Long Short-Term Memory (LSTM) [22] or Gated Recurrent Neural (GRU) [23]. Compared with those template/statistic captioners, these recurrent

network based decoders have stronger representation power that they can memorize more language statistic and thus are able to generate more diverse descriptions instead of rigid ones by template/statistic captioners.

After applying the encoder-decoder framework to solve captioning, researchers discovered that only a small part of knowledge will be transferred from the encoder to the decoder, and thus the abundant visual patterns are not sufficiently exploited. To enlarge the “information bottleneck”, attention mechanism [24] is incorporated into these frameworks which selects a few regions from the image to generate each word. Specifically, attention mechanism computes a series of soft attention weights for soft selection. After SA&T, more and more advanced attention mechanisms are developed and used in the encoder-decoder captioners, which further improve the qualities of the generated captions. For example, semantic attention [25, 26] not only selects the regions from the image, but also from some discrete semantic labels like the categories or the attributes of the objects. KWT [27] introduces a sentinel gate to decide when to attend to the image or just exploit the previously generated caption, which can be treated as a binary vision and language attention. Some researchers also notice that in human visual system, there are two different attentions which are top-down attention affected by the current task and bottom-up attention generated in an automatic way [1]. Based on such observation, they propose to extract salient regions which more likely contain objects as the bottom-up attention and design a top-down attention decoder for selecting those salient regions.

In addition to the progress of the network architectures in image captioning, some other methods are proposed to address the “exposure bias” problem [28, 29] for ameliorating the optimization. Specifically, when we train an image captioner, the words are generated conditioned on the ground-truth words, while during inference, the words are generated conditioned on the sampled words. Thus, there is a mismatch between the training and the inference, which causes performance degradation. Scheduled sampling strategy [30] is proposed to relieve the problem, which generates new words also conditioned on the sampled words during training. Furthermore, Reinforcement Learning based rewards [29, 31] are used to train the captioners by measuring the similarity between the sampled sentences and the ground-truth sentences.

1.3 Challenges and Contributions

Equipping with these advanced network architectures, attention mechanisms, and training objects, better captions can be generated. However, these captions are far from the ones generated by humans that they are **less descriptive** and **more biased** than humans’ descriptions. One major reason for these critical limitations is that the deployed visual encoders are usually well trained by image classification or object detection. As a result, the visual encoder tends to embed abundant knowledge about objects’ categories while neglecting the attributes and relationships into the high-level representations, while that neglected knowledge is equivalent important for generating descriptive and unbiased captions.

In this way, the captioner will easily build connections between the extracted visual representations between certain objects. Consequently, given an image, these captioners will only generate simple and trivial captions from the recognized objects, which are no better than just a list of object detection [32]. Moreover, even sometimes the attribute or relation words are generated, these words are not from visual reasoning but more likely from the dataset bias. Specifically, after recognizing some frequently appeared objects, these captioners will infer attributes or relationships from high co-occurrence between these words with the recognized objects, and thus generate “there is a red apple” when only green apples appearing due to the high co-occurrence between “red” and “apple” in the dataset [33]. More seriously, social bias like gender discrimination may be enlarged during this process, *e.g.*, when observing a snowboard, these captioners may directly infer there is a man snowboarding [34].

In this thesis, we propose three novel neural network based captioners to ameliorate these two critical issues for generating more descriptive and less biased captions. We build these captioners by applying different strategies to incorporate three distinctive additional knowledge into modern advanced captioners. Generally, additional knowledge not only provides the captioners with more semantic attributes or relations, but also some “rules” which humans usually use to compose captions, *e.g.*, summarizing, rephrasing, or even commonsense reasoning. More interestingly, as we will see in a captioner inspired from the causal inference framework [35], such knowledge can be treated as the mediator which permits us to deploy the front-door adjustment for deconfounding the dataset bias.

We consider using three kinds of knowledge which are the language inductive bias preserved in language corpora, the sentence patterns which humans use to compose sentences, and the commonsense knowledge. Moreover, quite different strategies are applied to incorporate this knowledge into the captioners. In particular, we use scene graph to represent the language inductive bias and transfer it from pure language domain to vision language domain for descriptive and human-like captions. For sentence patterns, we design diverse modules corresponding to the patterns, *e.g.*, OBJECT module for nouns, and first collocate these modules and then generate different kinds of word from these modules, *e.g.*, generating nouns from OBJECT module. For commonsense knowledge, it acts as the mediator transmitting knowledge from the image to the caption in a front-door path. We address this commonsense knowledge by the “subject-predicate-object” form and transform it into an embedding by a simple relation embedding operation for further retrieving to generate less biased captions.

1.4 Outline of the Thesis

The whole thesis is organized as follows.

- Chapter 1 first introduces the background of image captioning and then briefly discusses the history of developments of the captioners, from classic template/statistic based systems to modern neural network based encoder-decoder frameworks. This chapter also points out two major limitations of these modern captioners that they usually generate less descriptive and more biased captions than humans. At last, we propose to incorporate additional knowledge into the modern captioners for alleviating these two limitations, which is also the main contribution of this thesis.
- Chapter 2 reviews all the related research about the motivations of our research, the techniques which inspire ours, and the compared state of the art models. We also compare the models which are similar to ours and pinpoint the differences between these models and ours.
- Chapter 3 proposes a novel **Shuffle-Then-Assemble (STA)** pre-training strategy for learning object-agnostic visual features to construct scene graphs,

which will be used to form the additional knowledge in the following chapters. Here “agnostic” means that the feature is less likely biased to the classes of paired objects. To achieve this, we first discard all the triplet relationship annotations in an image, leaving two unpaired object domains without obj1-obj2 alignment. Then, our feature learning is to recover possible obj1-obj2 pairs. In particular, we design a cycle of residual transformations between the two domains, where the identity mappings encourage the RoI features to capture shared but not object-specific visual patterns. Extensive experiments on two visual relationship benchmarks show that by using our pre-trained features, naive relationship models can be consistently improved and even outperform other concurrent state-of-the-art relationship models.

- Chapter 4 proposes **Scene Graph Auto-Encoder (SGAE)** to incorporate the language inductive bias as the additional knowledge into the encoder-decoder image captioning framework for more human-like captions. Specifically, we use the scene graph, which represents the complex structural layout of both image ($\mathcal{I}$) and sentence ($\mathcal{S}$), as the bridge to transfer such inductive bias. In the language domain, we use SGAE to learn a dictionary set ($\mathcal{D}$) that helps reconstruct sentences in the $\mathcal{S} \rightarrow \mathcal{G}_\mathcal{S} \rightarrow \mathcal{D} \rightarrow \mathcal{S}$ auto-encoding pipeline, where $\mathcal{D}$ encodes the desired language prior and the decoder learns to caption from such a prior; in the vision-language domain, we share $\mathcal{D}$ in the $\mathcal{I} \rightarrow \mathcal{G}_\mathcal{I} \rightarrow \mathcal{D} \rightarrow \mathcal{S}$ pipeline and distill the knowledge of the language decoder of the auto-encoder to that of the encoder-decoder based image captioner to transfer the language inductive bias. We validate the effectiveness of SGAE on various captioning settings including standard MS-COCO image captioning benchmark and two more challenging settings which are: transferring inductive bias from other language corpora and unpaired image captioning. The experiments validate the superiority of SGAE.
- Chapter 5 proposes **Hierarchical Scene Graph Encoder-Decoder (HSGED)** for generating coherent and distinctive paragraphs, which mimics humans’ manner to compose a long paragraph about an image: by implicitly composing a mental “script” and then comply with it to generate the paragraph. We use the image scene graph as the additional knowledge “script” to incorporate rich semantic knowledge and, more importantly, the hierarchical constraints into the model. Specifically, we design a sentence scene graph RNN (SSG-RNN) to generate sub-graph level topics, which constrains

the word scene graph RNN (WSG-RNN) to generate the corresponding sentences. We propose irredundant attention in SSG-RNN to improve the possibility of abstracting topics from rarely described sub-graphs and inheriting attention in WSG-RNN to generate more grounded sentences with the abstracted topics, both of which give rise to more distinctive paragraphs. We validate HSGED on Stanford image paragraph dataset and the experiment results demonstrate that it can generate more coherent and distinctive paragraphs.

- Chapter 6 proposes a novel framework: **learning to Collocate Visual-Linguistic Modules (CVLNM)** to imitate how humans compose captions that they do not speak word by word from scratch, while quickly structure a pattern like STH DO STH AT SOMEPLACE and then fills in the detailed descriptions. To achieve this, we incorporate the sentence patterns as additional knowledge into the encoder-decoder framework. In particular, we make the following technical contributions for CVLNM training: 1) compact modules are designed corresponding to sentence patterns — one for function words and three for visual content words (*e.g.*, noun, adjective, and verb), 2) soft module fusion and multi-step module execution, robustifying the visual reasoning in partial observation, 3) a linguistic loss for module controller being faithful to part-of-speech collocations (*e.g.*, an adjective is before a noun). Experiment results demonstrate that our CVLNM can not only generate better captions but be more robust to fewer training samples compared with the concurrent state of the art captioners.
- Chapter 7 proposes **Deconfounded Image Captioning (DIC)** based on the causal framework to alleviate the negative effects brought by the dataset bias. In this chapter, we first analyze why modern image captioners are easily affected by dataset bias from the perspective of confounding effect and then propose DICv1.0, which exploits additional commonsense knowledge as the mediator to transmit information from the image to the caption, to alleviate the negative effects brought by dataset bias. DIC is based on causal inference, whose two principles: the backdoor and front-door adjustments, help us review previous works and design new effective models. In particular, we showcase that DICv1.0 can strengthen two prevailing captioning models and can achieve better performances than the concurrent state of the art captioners. More importantly, we retrospect the major captioners in our

6-year-old IC community from the causal view and show how this causal retrospect enlightens us to develop our DICv1.0.

Chapter 2

Literature Review

2.1 Overview

In this chapter, we review the literatures related to the research discussed in this thesis.

2.2 Image Captioning

There is a long history for researchers to develop automatic image captioning methods. Most early image captioners are templates/statistic based models that they first generate sentence patterns and then fill the content words like objects' categories, attributes, and their relations detected from the image into the fixed patterns [15–18]. Though the language inductive bias or the sentence patterns, which are additional knowledge, are already incorporated into these templates, the performances of these captioners are still limited. There are two major reasons limiting these captioner: 1) the performances of the visual detectors are not satisfactory and 2) the processes for composing templates and for detecting words are not jointly trained.

Compared with these early models, the modern captioners have achieved striking advances due to the progress of the deep learning techniques in both computer vision, *e.g.*, more accurate visual pattern recognition systems [6, 36], and natural

language processing fields, *e.g.*, encoder-decoder pipeline [19] and attention mechanism [37]. Inspired by them, researchers have developed attention based encoder-decoder image captioners. Particularly, S&T [20] builds an encoder-decoder captioner with end-to-end training and SA&T [24] exploits the attention mechanism to generate words by adaptively focusing on different regions of an image at different time steps. After that, more advanced visual attention mechanisms have been proposed for generating better captions. For example, KWTL [27] can decide whether to attend to the image for generating vision-related words or just function words, bottom-up attention [1] selects regions only from saliency regions which are more likely to contain objects, and CAVP [38] and LBPF [39] stack more attention sub-networks for multiple-step reasoning. Interestingly, researchers have also designed semantic attention mechanisms [25, 26] which incorporate semantic labels recognized from the image as additional knowledge into the captioners. For example, some methods exploit object [32], attribute [40], and relationship [2] knowledge into their captioning models.

Compared with these models which only simply add the embeddings of these detected semantic labels to the feature set which will be attended to, we form diverse additional knowledge in suitable structures and exploit distinctive strategies to exploit them for generating better descriptions. Specifically, in SGAE (Chapter 4) and HSGED (Chapter 5), we use scene graph to represent the additional knowledge and design different neural graph operations to embed these scene graphs. In CNM (Chapter 6), the sentence pattern, which is another kind of additional knowledge, is used to collocate the designed neural modules for encouraging each module to learn the corresponding knowledge, *e.g.*, OBJECT module should learn more about nouns. Moreover, from this module network view, several recent are the special cases of our CNM, *e.g.*, semantic attention based captioners [25, 26] exploit OBJECT, ATTRIBUTE, and RELATION modules to predict object categories, attributes, and actions, respectively, and then directly input these semantic words into the language decoder for captioning. In DIC (Chapter 7), we treat the commonsense knowledge as additional knowledge and use it as the mediator in a front-door causal graph for achieving the front-door adjustment to generated less biased captions. Importantly, we also provide a causal retrospect to compare our DIC with these major captioners proposed in recent years, where we can find that most of them are weaker approximations of the backdoor and front-door adjustments, while our DIC is designed based on both backdoor and front-door adjustments.

In addition to the progress of the network architectures in image captioning, some other methods are proposed to address the “exposure bias” problem [28, 29] for ameliorating the optimization. Specifically, when we train an image captioner, the words are generated conditioned on the ground-truth words, while during inference, the words are generated conditioned on the sampled words. Thus, there is a mismatch between the training and the inference, which causes performance degradation. Scheduled sampling strategy [30] is proposed to relieve the problem, which generates new words also conditioned on the sampled words during training. Furthermore, Reinforcement Learning based rewards [29, 31] are used to train the captioners by measuring the similarity between the sampled sentences and the ground-truth sentences. Considering these two techniques can alleviate the “exposure bias” problem and improve the performances, we also deploy them in our captioners including SGAE, HSGED, CNM, and DIC.

2.2.1 Image Paragraph Captioning

The captioners discussed in Section 2.2 mainly focus on summarizing the whole image into one single sentence, while the generated sentence is usually too short to detail the rich semantic contents in an image. To overcome this limitation, researchers propose another task: Dense Captioning to generate more descriptions for all the salient regions [41, 42]. However, since the detected regions are usually heavily overlapped and disordered, the corresponding sentences are redundant and incoherent. As a result, their usability is still damaged [43].

Image paragraph captioning addresses the shortcomings of both single sentence captioning and dense captioning which tries to generate coherent and distinctive paragraphs [43]. Since each sentence of a paragraph is controlled by a topic, researchers propose Hierarchical RNNs (HRNNs) [43–47] that higher-level and lower-level RNNs respectively abstract topics and generate sentences based on the abstracted topics. Researchers also propose advanced techniques to refine the prototypical HRNN, *e.g.*, generative models like GAN [45] or VAE [44] for stronger consistency; the trigram repetition penalty based sampling method for diversity [48]. Besides, dense sentence-level rewards [49] and curiosity-driven reinforcement learning [50] are used for more robust training, all of which could also be applied in our proposed framework, HSGED. However, most of them are built without enough

hierarchical constraints, so the qualities of the generated paragraphs are unsatisfactory. In contrast, our HSGED exploits the scene graph as additional knowledge and the script to transfer its hierarchical and semantic knowledge from the vision domain to the text domain for more coherent and distinctive paragraphs.

2.2.2 Unpaired Image Captioning

Another line of image captioning research is on a more challenging task: Unpaired Image Captioning [51], which only has an image set and a caption corpus during training but without any image-caption pairs. A few preliminary works, including Pivoting Language [51], adversarial training [52, 53] and multi-modal embedding alignment [54], have been proposed to tackle this problem. In this thesis, we extend SGAE to such unpaired scenarios, which exploits the scene graph as the bridge to narrow the gap between the vision and language domains.

2.3 Scene Graph

2.3.1 Scene Graphs Generation

Visual scene graphs [55, 56] contain the structured semantic contents of an image, which typically include the knowledge of the present objects, their attributes, and their pairwise relationships. As a mid-level vision task which is more complex than the low-level vision tasks and underpins the high-level vision language tasks, researchers have realized the importance of this task and proposed a variety of approaches to generate scene graphs from the images [57–63]. These methods usually first exploit some advanced object detectors, *e.g.*, Faster RCNN [6], YOLO [64], or Mask RCNN [65], to detect the discrete objects and then recognize the pairwise object relationships to connect the discrete object nodes to construct a graph.

Modeling the object interactions such as verbs [66, 67], actions [68–70], and visual phrases [71–74] has been extensively studied in literature. However, a key challenge is that it is very expensive to label relationship triplets due to the combinatorial complexity, leading to the resultant relationship model brittle to rare or unseen compositions. Unfortunately, few efforts have been explored to resolve

the small-data and incomplete-annotation issue. Though weakly-supervised visual relationship model [75, 76] can reduce the labelling cost, its performance is still far from practical use compared to supervised models. To address with this challenge, in Chapter 3, we investigate a principled visual solution named as STA towards the challenge, while not relying on external knowledge such as dataset-specific statistics [77–79] and language priors [80, 81].

Besides extracting scene graphs from visual scenes, scene graphs can also be extracted from textual data, *e.g.*, applying a network to parse image captions for scene graph generation [82], or designing certain rules to transfer the object and relationship classification results [83] or the dependency parser tree of a text [84] into the resultant scene graph. In SGAE (Chapter 4), we exploit Motif [58] and Spice [84] to extract scene graphs from images and captions, respectively.

2.3.2 Exploitation of Scene Graph.

The scene graph is formed by connecting discrete objects with their attributes and with other objects through pairwise relationships, so it contains rich semantic and topological knowledge [61]. Observing such advantages, researchers exploit Graph Neural Networks (GNN) [85–88] to embed scene graphs in various computer vision tasks, *e.g.*, image retrieval [55], image generation [89], image captioning [3], and visual reasoning [90].

In this thesis, we propose to use scene graphs as the additional knowledge to help transfer more abundant information from the images to the captions or the paragraphs. In particular, in SGAE (Chapter 4), we preserve sentence scene graphs into a dictionary and then retrieval knowledge from this dictionary based on the image scene graphs for generating more descriptive captions. In HSGED (Chapter 5), we exploit both explicit semantic knowledge of a scene graph for more fine-grained paragraphs and implicit topological knowledge for more coherent paragraphs. Moreover, different from the method [91] using visual relationship which directly fuses the object and relation features into a flat RNN without any topic guidance, HSGED treats the scene graph as a script to facilitate the topic guidance and to regularize the training.

2.4 Graph Convolutional Networks

Graph Convolutional Network (GCN) generalizes CNN from regular grid data, *e.g.*, image, speech, or text to any irregular graph data, *e.g.*, social networks, citation networks, or knowledge graphs. There are two types of GCNs: spectral GCN [92–94] and spatial GCN [95–97]. Considering spectral GCN is usually used for the undirected graph and our scene graphs are directed, we apply spatial GCN in our thesis to transform a scene graph into a series of continuous embeddings, which can be seamlessly integrated into the deep network. Specifically, in SGAE, we deploy an Attentional GCN [60] to selectively incorporate the attribute and relationship nodes into an embedding to avoid redundancy brought by the noisy scene graphs. In HSGED, we deploy two different embeddings which are sub-graph level embedding and node-level embedding for generating topic vectors and captions, respectively. Importantly, we follow the suggestions in [88] to design both GCNs to achieve stronger representation power.

2.5 Memory Networks

Recently, many works try to augment a working memory into a network for preserving a dynamic knowledge base to facilitate the subsequent inference, *e.g.*, for preserving training samples to achieve one-shot learning [98], for preserving the context of a document to solve language modeling [99], question answering [100, 101], and reading comprehension [102], for preserving visual knowledge to deal with visual question answering [103], or even for preserving a series of reasoning decisions to address long term machine reasoning [104]. Among these methods, differentiable attention mechanisms [100, 101] are usually applied to preserve (extract) useful knowledge in (from) the memory for the tasks at hand. The extraction of the knowledge is achieved by query-key attention computation where the query is some kind of context knowledge (*e.g.*, the questions in question answering task) and the key comes from the memory. Inspired by these methods, in SGAE, we also implement a memory architecture to preserve language inductive bias, which is represented by the sentence scene graph embedding during the training of the auto-encoder while being extracted during image captioning by using the visual scene graph embedding as the query.

2.6 Knowledge Distillation

Knowledge Distillation (KD) abstracts the response to input of a teacher learner as the knowledge and distills such knowledge to another student learner for achieving knowledge transfer [105]. Since KD does not require that the teacher and student learners own the same architecture or even the same task [106], it can be applied to various machine learning applications. For example, it can achieve model compression by teaching a small network to learn from a huge network trained by a large-scale dataset [107, 108]. Lifelong Learning [109, 110] also exploits KD to preserve the knowledge of the previous datasets to avoid catastrophic forgetting. KD has also been exploited for protecting modern learning systems from the attack of adversarial examples [111] by reducing the amplitude of network gradients. In SGAE, we treat the language scene graph auto-encoder and the image captioning encoder-decoder as the teacher and the student, respectively, by which we hope the encoder-decoder learns to compose captions as the auto-encoder. By distilling knowledge, the inductive bias is better transferred between the pure-language and the vision-language domains.

2.7 Neural Module Networks

Psychophysics evidence [112] has strongly suggested that our brain follows the principle of modular and thus it can disentangle the compositional task into solvable sub-tasks. Motivated by this, researchers in the field of representation learning design modular network which can disentangle different variation factors from the complex data for more effectively and robustly learning and inference [113, 114].

For visual reasoning tasks (*e.g.*, image captioning [24] or VQA [11]), they are usually composed by a series of low-level vision tasks (*e.g.*, image classification [115], object detection [36], or relation recognition [80]) and nature language processing tasks (*e.g.*, language generation [19, 116] or language understanding [117, 118]). Then, an intuitive way to solve visual reasoning tasks is to decompose them into many simple but diverse sub-tasks and design distinguishable modules to address each sub-task.

More specifically, VQA [11] requires an AI agent to understand different aspects of a visual scene and the given question for correctly answering, which can be solved by a series of modular networks [119–122], *e.g.*, COLOR module for “what is the color of an object”, LOCATE module “where is one object”, or COUNT module for “how many is one object”. Similarly, visual reasoning [123–125] and visual grounding [126–129] can also be decomposed into different sub-tasks to be addressed by specific modules. In these tasks, high-quality module structures can be parsed from the provided sentences like the questions in VQA or the descriptions in visual grounding.

In CNM (Chapter 6), we incorporate sentence patterns as additional knowledge into the encoder-decoder captioner for designing distinguishable modules. However, unlike the other tasks where the module layouts can be perfectly parsed, only partially observed sentences are available in image captioning. To address such a challenge, we design a module controller with the previously generated module layout as input to dynamically collocate neural modules. This controller is also regularized by a part-of-speech based syntax loss for more human-like module layouts.

Chapter 3

Shuffle-Then-Assemble: Learning Object-Agnostic Visual Relationship Features¹

3.1 Introduction

As the maturity of low-level vision solutions such as object classification and detection [36, 130], we are more ambitious to pursue higher-level vision-language tasks such as image captioning [131], visual Q&A [132, 133], and visual chatbot [134]. Unfortunately, we gradually realize that many of the state-of-the-art systems prefer to capture training set bias while not the underlying reasoning [131–133]. Recently, a promising way is to use visual compositions such as scene graph [55, 61] and relationship context [135, 136] for explainable visual reasoning. Therefore, visual relationship detection (VRD) [57, 75, 77] — the task of predicting elementary triplets such as “person ride bike” and “car park on road” in an image — is becoming an indispensable building block connecting vision with language.

Despite the relatively preliminary stage of VRD compared to object detection, a major challenge of VRD is the high cost of annotating the (obj1, rel, obj2) triplets

¹This work is chapter as: Xu Yang, Hanwang Zhang, Jianfei Cai. “Shuffle-Then-Assemble: Learning Object-Agnostic Visual Relationship Features.” Proceedings of the 15th European Conference on Computer Vision (**ECCV**, **Poster**). Munich, Germany. 2018.

FIGURE 3.1: (a) The triplet annotations of visual relationships in an image. (b) The key idea of the proposed **Shuffle-Then-Assemble** strategy is to discard the paired annotation of any relationship and leave two unpaired object domains. (c) Comparisons between the original feature maps from base CNN (top) and the desired object-agnostic ones (bottom) obtained by our pre-training (averaged over all the channels). We can see that our feature maps focus on the relationships between objects.

as shown in FIGURE 3.1 (a). Unlike labeling objects in images, labeling visual relationships is prohibitively expensive as it requires combinatorial checks of the three entries. Therefore, the relationships in existing VRD datasets [56, 80] are long-tailed, and the resultant relationship models are inevitably biased to the dominant obj1-obj2 combinations. For example, as reported in pioneering works [57, 80], the recognition rate of unseen triplet compositions is significantly lower than the seen ones. This deficiency clearly prevents the wide application of VRD in compositional reasoning. Though it can be alleviated by exploiting external knowledge such as language priors [80] and large-scale weak supervision [75], we still lack a principled solution in the visual modeling perspective.

Unsupervised feature learning (or pre-training) is arguably the most popular remedy for training deep models with small data [137–141]. Therefore, we are inspired to learn object-agnostic convolutional feature maps that are less likely biased to certain obj1-obj2 combinations. Such features should be highly responsive to object parts¹ involved in a relationship. A plausible way is to append additional conv-layers to the original base CNN (*e.g.*, VGG16 [21] or ResNet-150 [36]) to remove the object-sensitive responses inherited from image classification pre-training dataset (*e.g.*, ImageNet [142]). For example, as shown in FIGURE 3.1 (c), compared with the base CNN’s feature map, the object-agnostic one ignores object patterns but focuses on the shared patterns of interacted objects. Therefore, we

¹The parts can be at the pixel-level as well as the receptive field-level.

raise a question: how to learn the object-agnostic feature maps without additional relationship labeling cost?

In this research, we propose a novel **Shuffle-Then-Assemble** feature learning strategy. As shown in FIGURE 3.1 (b), “shuffle” is to discard the original one-to-one paired object alignments, and thus no explicit obj1-obj2 class information is used; “assemble” is to pose the relationship modeling into an unsupervised pair recover problem by transferring Region-of-Interest (ROI) features between the two unpaired domains. Our intuitive motivation is two-fold: 1) if the ROI features extracted from the resultant feature maps still encode object-specific information, features are not likely to be transferred between the two domains of heterogeneous objects; 2) the unsupervised fashion encourages the exploration of many more possible relationships which are usually missing in annotation. As shown in FIGURE 3.1 (a), some simple spatial relationships such as “chair beside bag” are missing and correspondent relationships are usually ignored, *i.e.*, “chair under person” is missing as “person sit on chair” is labeled. Inspired by the recent advances in unsupervised domain transfer methods [143–146], we design a cycle of transformations to transfer between the two domains: either way maps an ROI from domain A (or B) to B (or A), and then an adversarial network is used to confuse the mapping with RoIs in B (or A). In particular, we use a residual structure for the transformation network, where the identity mapping encourages the feature map to capture shared but not object-specific visual patterns.

We demonstrate the effectiveness of the proposed **Shuffle-Then-Assemble** strategy on two benchmarks: VRD [80] and VG [56]. We observe consistent improvement of using our pre-trained features against various ablative baselines and other state-of-the-art methods. For example, compared to feature maps without pre-training, we can boost the Recall@100 of supervised, weakly supervised, and zero-shot relationship prediction by absolute 4.74%, 4.42%, 4.04%, respectively on VRD, and 4.41%, 4.2%, 5.81%, respectively on VG.

3.2 Method

FIGURE 3.2 illustrates the overview of **Shuffle-Then-Assemble** pre-training and fine-tuning strategies. The goal of the feature learning process is to pre-train the

FIGURE 3.2: The overview of the relationship model improved by the proposed **Shuffle-Then-Assemble** pre-training strategy. The goal at the pre-training stage is to learn the Object-Agnostic (OA) conv-layers using **Shuffle-Then-Assemble** objective. Then, the traditional supervised training for the relation model can be considered as the fine-tuning stage using the desired OA feature map.

Object-Agnostic (OA) conv-layers, which result in the desired OA feature map for better relationship modeling. We will first introduce the widely-used relationship modeling framework and its limitations, and then detail how to use the proposed feature learning method to overcome them.

3.2.1 Visual Relationship Model

The input of the visual relationship model is an image with a pair of object bounding boxes, and the output is an “obj1-rel-obj2” triplet, where “obj1” and “obj2” are the object classes of the two bounding boxes, and “rel” is the relationship class. In this research, we adopt the common practice as in [57, 80] that we do not directly model the triplet composition as a whole [73, 147], which requires $\mathcal{O}(C^2R)$ complexity for C object and R relationship classes; instead, we model objects and relationships separately to reduce the complexity down to $\mathcal{O}(C + R)$, with the help of much more matured object detection models [6, 130, 148]. Therefore, without loss of generality, we refer to a relationship model as an R -way classifier.

Suppose $\mathbf{x}_i$ and $\mathbf{x}_j$ are the RoI features of any pair of object bounding boxes (i, j) (e.g., the red and blue cubes in FIGURE 3.2 by RoI pooling [149]), the r -th relationship score is obtained by a softmax classifier whose input is a simple

concatenation of the two features:

$$S(i, j, r; \theta) = \frac{\exp(\mathbf{w}_r^T \text{MLP}([\mathbf{x}_i, \mathbf{x}_j]))}{\sum_{t=1}^R \exp(\mathbf{w}_t^T \text{MLP}([\mathbf{x}_i, \mathbf{x}_j]))}, \quad (3.1)$$

where $\mathbf{w}_t \in \theta$ is the parameter of the classifier and the configuration of $\text{MLP}(\cdot)$ is detailed in FIGURE 3.2. Note that although Eq. (3.1) is a naive model and there are fruitful ways of combining $\mathbf{x}_i$ and $\mathbf{x}_j$ in literature, such as appending independent MLPs for each RoI [57], the union RoI [77], and even the fusion with textual features [135], our feature learning can be seamlessly incorporated into any of them. We will leave the evaluations of applying these tweaks for future work.

The relationship model can be trained by minimizing the cross-entropy loss of Eq. (3.1), summing over all the ground-truth pairs. However, due to the limited annotation of the relationship triplets, relationship models trained on these extremely long-tailed annotations are inevitably biased to the dominant object classes. One may wonder why it is object-biased as Eq. (3.1) does not use any object class information at all? The reason resides in the base CNN feature map. Almost all state-of-the-art visual recognition systems deploy the base CNN [21, 36, 150] pre-trained on ImageNet [142] or ImageNet+MSCOCO [151], where the training task is object recognition. Therefore, the resultant feature map for extracting RoI will naturally favor the sensitivity to object classes — each RoI feature encodes the discriminative information of the object (cf. the original feature map of FIGURE 3.2), and leads the parameters in Eq. (3.1) over-fitted to specific object patterns. For example, if most of the triplets of “stand on” is “person stand on street”, then the “stand on” classifier will mistakenly consider the joint pattern “person” and “street” into “stand on”, and fails in cases of “person stand on chair” or “dog stand on street”.

3.2.2 Shuffle-Then-Assemble Feature Learning

To alleviate the bias, we propose **Shuffle-then-Assemble** strategy to pre-train the Object-Agnostic (OA) conv-layers for obtaining the OA feature map. As discussed above, the bias is mainly due to the dominant object pairs in training data, therefore, our key idea is to discard the original one-to-one pairwise annotations, *i.e.*, “shuffle”, leaving two unaligned domains of RoIs for “obj1” and “obj2”, and

FIGURE 3.3: (a) The view of unsupervised domain transfer for **Shuffle-Then-Assemble**. It contains a cycle of transformations $F: A \mapsto B$ and $G: B \mapsto A$, and a pair of discriminators D_A and D_B to measure the quality of the transfer. (b) Qualitative transfer results. The directed arrow indicates the nearest-neighbor RoI in the target domain to the RoI from the source domain. (c) The residual architecture of the transformation network. (d) The architecture of the discriminator.

then we attempt to recover the one-to-one alignment, *i.e.*, “assemble”, by unsupervised domain transfer. Note that this pre-training strategy does not require additional cost of supervision. As shown in FIGURE 3.3 (b), we manage to align potential relationships without any one-to-one supervision, *e.g.*, **obj1** may relate to **obj6** with respect to “sit” and **obj3** may relate to **obj2** with respect to “hold”.

The unsupervised domain transfer method used in **Shuffle-Then-Assemble** follows recent progress on adversarial domain transfer [143–146]. Noteworthy, the motivation of using adversarial domain transfer emphasizes more on the unsupervised alignments but NOT the feature transfer as in traditional domain transfer applications such as [152], where the domain transfer is used to close the gap between conveniently available synthetic data and real data. Here, our idea is more similar to [140] which discovers alignments between images that are very visually different such as “spotted bags” and “spotted shoes”, or “frontal faces” and “frontal cars”.

As illustrated in FIGURE 3.3 (a), we want to guide the pre-training of the OA

conv-layers by learning mapping functions between domain A and B , where each of them consists of RoI features, $\mathbf{a} \in A$ and $\mathbf{b} \in B$, extracted from the tentative OA feature map. For the purpose of domain transfer, we have a cycle of two mappings: $F: A \mapsto B$ and $G: B \mapsto A$, to discover the underlying relationships between A and B . Recall that there is one-to-one supervision between the two domains, we adopt the adversarial objective $\mathcal{L}_{adv}$ such that the mapped features $\{F(\mathbf{a})\}$ and $\{G(\mathbf{b})\}$ are indistinguishable from B and A , respectively; in particular, the indistinguishability is measured by two discriminators D_A and D_B :

$$\begin{aligned} \mathcal{L}_{adv}(A, B; \phi, F, G, D_A, D_B) = & \underbrace{\mathbb{E}_{\mathbf{a}}[\log D_A(\mathbf{a})] + \mathbb{E}_{\mathbf{b}}[\log D_B(\mathbf{b})]}_{\text{maximize by } D_A \text{ and } D_B} + \underbrace{\mathbb{E}_{\mathbf{b}}[\log(1 - D_A(G(\mathbf{b})))] + \mathbb{E}_{\mathbf{a}}[\log(1 - D_B(F(\mathbf{a})))]}_{\text{minimize by } F \text{ and } G}, \end{aligned} \quad (3.2)$$

where ϕ is the OA conv-layers that generate A and B , D_A is a binary classifier that tries to classify $D_A(\mathbf{a}) \mapsto 1$ and $D_A(F(\mathbf{b})) \mapsto 0$, and D_B is defined similarly. In this adversarial way, we will eventually obtain F and G which discover the hidden alignments between the two domains, *i.e.*, indistinguishable by the discriminators.

To encourage more explorations of the potential relationship alignments between the RoIs in two domains, *e.g.*, avoid from mapping many RoIs in A to only one RoI in B with respect to a trivial spatial relationship such as “on” and “by”, we further impose the “cycle-consistent” loss to be minimized by G and F :

$$\mathcal{L}_{cycle}(A, B; \phi, F, G) = \mathbb{E}_{\mathbf{a}}[\|\mathbf{a} - G(\mathbf{b})\|_1] + \mathbb{E}_{\mathbf{b}}[\|\mathbf{b} - F(\mathbf{a})\|_1]. \quad (3.3)$$

The loss penalizes two different RoIs, *e.g.*, $\mathbf{a}$ and $\mathbf{a}'$, mapped to the same RoI $\mathbf{b}$ as it is hard to satisfy $\mathbf{a} \approx G(\mathbf{b})$ and $\mathbf{a}' \approx G(\mathbf{b})$ simultaneously.

By putting Eq. (3.2) and Eq. (3.3) together, the full objective for pre-training the OA conv-layers is:

$$\phi^* = \arg \min_{\phi} \min_{F, G} \max_{D_A, D_B} \mathcal{L}_{adv}(A, B; \phi, F, G, D_A, D_B) + \lambda \mathcal{L}_{cycle}(A, B; \phi, F, G), \quad (3.4)$$

where $\lambda > 0$ is a trade-off hyper-parameter. Then, we can use ϕ^* to obtain $\mathbf{x}_i$ and $\mathbf{x}_j$, and fine-tune a better relationship model θ using existing triplet supervision as in Eq. (3.1). Next, we will introduce the proposed implementation of F and G .

3.2.3 Implementation Details

3.2.3.1 Network Architecture

For base CNN, we adopt Faster RCNN (VGG16) [6], which takes short width to be 600 and outputs the original $1/16 \times 1/16 \times 512$ feature map. As shown in FIGURE 3.2, our OA conv-layer has 1 filter of the size 1×1 , stride 1, followed by a Leaky Relu [153]. The transformation network is detailed in FIGURE 3.3 (c). Each transformation contains two blocks of residual network. The motivation of applying the residual structure is two-fold. 1) The shortcut encourages to find shared regions of two RoIs, since the shared RoI features will pass directly via the shortcut. This makes the optimization not only more light-weighted, but also easier to find the intrinsic inter-related visual patterns as relationships. 2) If any object-specific information is still encoded in the RoI feature, the shortcut will make it harder to achieve the final domain transfer as domain A and B usually contain diverse objects. The discriminator network is detailed in FIGURE 3.3 (d), which is composed by two fully-connected layers followed by a Leaky Relu. It takes a 50,176-d (two $7 \times 7 \times 512$ RoI feature) vectorized RoI feature as input and outputs a sigmoidal scalar between 0 and 1.

3.2.3.2 Training

At the feature pre-training stage, to collect sufficient RoIs in each domain, we augment the number of original bounding boxes by additional ones with IoU larger than 0.7, extracted by using the Region Proposal Network [6]. For each original bounding boxes, 10 RoIs are sampled. To stabilize the adversarial training in Eq. (3.4), we adopt three practices: 1) We apply least-square GAN [154] to replace the negative log likelihood by a least square loss. 2) The optimizer for training D_A and D_B is set to SGD and the optimizer for G , F and ϕ is set to Adam [155]. The initial learning rate is set to $1e-4$ for both optimizers. 2) D_A and D_B are trained three times more compared with G , F and ϕ . The trade-off λ in Eq. (3.4) is set to 10. Every mini-batch is one image with randomly selected 128 triplets. The epochs for training these networks are set to 20 on VRD dataset and set to 5 on VG dataset.

At the fine-tune stage for training relationship classifier, the short width of image is still set to 600. Every mini-batch is one image with 128 randomly selected triplets. The optimizer is Adam with initial learning rate set to 1e-5 in all the experiments. The epochs are set to 50 and 30 on VRD dataset and VG dataset, respectively.

3.3 Experiments

We evaluated our **Shuffle-Then-Assemble** method by performing visual relationship prediction on two benchmark datasets. We conducted experiments under extensive settings such as supervised, weakly-supervised, and zero-shot learning, each of which has various ablative baselines and state-of-the-art methods. We also visualized qualitative object-agnostic features maps compared against others.

3.3.1 Datasets and Metrics

We used two publicly available datasets: VRD (Visual Relationships Dataset[80]) and VG (Visual Genome V1.2 dataset [56]).

VRD dataset. It contains 5,000 images with 100 object categories and 70 relationships. In total, VRD contains 37,993 relationship triplet annotations with 6,672 unique triplets and 24.25 relationships per object category. We followed the same training/test split as in [80], *i.e.*, 4,000 training images and 1,000 test images, where 1,877 triplets are only in the test set for zero-shot evaluations.

VG dataset. VG is annotated by crowd workers and thus the relations labeling are noisy, *e.g.*, free-language and typos. Therefore, we used the pruned version provided by Zhanget *al.* [57]. As a result, VG contains 99,658 images with 200 object categories and 100 predicates, 1,174,692 relation annotations with 19,237 unique relations and 57 predicates per object category. We followed the same 73,801/25,857 training/test split. And this dataset contains 2,098 relationships which never occur in the training set, which can be used for zero-shot evaluations. Though this dataset has more annotation than VRD, it is also heavily long-tailed and incomplete.

FIGURE 3.4: We evaluate relationship prediction task using four different experiment settings: supervised, supervised (Det), weakly-supervised and zero-shot. “?” denotes the relationship to be predicted.

Metrics. As conventions [57, 80], we used Recall@50 ($\mathbf{R@50}$) and Recall@100 ($\mathbf{R@100}$) as evaluation metrics. $\mathbf{R@K}$ computes the fraction of times a true relationship is predicted in the top K confident relation predictions in an image. Note that precision and average precision (AP) are not proper metrics as visual relations are labeled incompletely and they will penalize the detection if we do not have that particular ground truth.

3.3.2 Settings

In our experiments, we only focused on the relationship prediction task, *i.e.*, classifying any two object regions into relationship classes. The reasons are two-fold. First, relationship prediction plays the core role in relationship detection, a more comprehensive task that also needs to detect the two objects. Second, we can exclude the influence of object detection performance, as the improvement of object detection can greatly improve the relationship detection scores [57]. To offer a testbed for more practical application domains of relationship prediction, we designed the following 4 settings according to different pairwise modeling fashions:

Supervised. This setting is the standard supervised relationship prediction. As shown in FIGURE 3.4 (a), for training, all the objects are provided with ground truth boxes and the relationship between objects are given; for testing, a pair of objects with bounding boxes are given and their relationship is to be predicted.

FIGURE 3.5: Qualitative feature maps of two relationships on VRD. For each one, three kinds of feature maps are visualized by averaging over the 512 channels. We can see that by using the proposed **Shuffle-Then-Assemble** (STA), the RoI features are less likely biased to the objects and more focused on the regions of interaction of the two objects. Moreover, the observation is consistent with diverse relationship appearances.

Supervised (Det). The above setting assumes a perfect object bounding box detector at testing. However, as shown in FIGURE 3.4 (b), a more practical setting is to use detected object bounding boxes by off-the-shelf object detectors. We used Faster RCNN [6] to detect around 100 objects in an image. Note that when calculating Recall@50 and Recall@100, we only count the detected object pairs that both of the objects have.

Weakly-Supervised. Compared to Supervised setting, we discard the one-to-one

paired object annotation with respect to a relationship. As shown in FIGURE 3.4 (c), at training, given objects with bounding boxes, we do not know which object relates to which one. Therefore, we adopted an average-pooling image-level relationship loss:

$$\mathcal{L}_{weak} = - \sum_{i=1}^N \sum_{j=1}^N \sum_{r=1}^R [y_{ijr} \log S(i, j, r) + (1 - y_{ijr}) \log(1 - S(i, j, r))]; \quad (3.5)$$

where N is the number of objects, y_{ijr} is 1 if the object pair (i, j) has the r -th relationship, and $S(i, j, r)$ is the relationship score in Eq. (3.1). Note that the testing stage of this setting is the same as that of Supervised setting.

Zero-Shot. This setting is the same as Supervised setting except that at testing we want to predict object pairs whose triplet combination is unseen during training. As shown in FIGURE 3.4 (d), though object sheep, road, and relationship on are individually seen at training, but the composition “sheep on road” is novel to test.

3.3.2.1 Comparing Methods

We compared the proposed **Shuffle-Then-Assemble (STA)** pre-training strategy with the following ablative baselines:

Base. We directly used RoI features which were extracted from the base CNN for relationship prediction.

Base+OA. We did not pre-train OA conv-layers ϕ by **Shuffle-Then-Assemble** strategy and directly fine-tuned ϕ and $MLP(\cdot)$ by minimizing the cross-entropy loss of Eq. (3.1).

STA w/o FT. After pre-training ϕ by **Shuffle-Then-Assemble** strategy, the parameters of ϕ were fixed. When training the network by minimizing Eq. (3.1), only parameters of $MLP(\cdot)$ were updated.

STA w/o Res. The transformation network in FIGURE 3.3 was no longer a residual network that the shortcut removed. And the other settings were the same with STA.

We also compared with state-of-the-art visual relationship prediction methods such as **VTransE** [57], **Lu’s-V** [80], **Lu’s-VLK** [80], and **Peyre’s-A** [76]. Note that

FIGURE 3.6: Performances (R@100%) of relation classification of the four relation types using the different methods in the supervised setting.

TABLE 3.1: The performances (Recall@K%) of compared methods on two datasets under Supervised setting and Supervised (Det) setting.

Dataset	VRD		VG		VRD(Det)		VG(Det)	
Metric	R@50	R@100	R@50	R@100	R@50	R@100	R@50	R@100
Base	39.25	39.25	52.48	52.61	37.83	37.83	50.12	50.31
Base+OA	43.29	43.29	58.35	58.53	40.78	40.78	57.03	57.31
STA w/o FT	44.30	44.30	58.14	58.32	41.12	41.12	56.88	57.02
STA w/o Res	46.83	46.83	62.08	62.32	44.85	44.85	61.12	61.30
STA	48.03	48.03	62.71	62.94	45.65	45.65	61.27	61.51
Lu's-V [80]	7.11	7.11	—	—	—	—	—	—
Lu's-VLK [80]	47.87	47.87	—	—	—	—	—	—
VTransE [57]	44.76	44.76	62.63	62.87	—	—	—	—
Peyre's-A [76]	46.30	46.30	—	—	—	—	—	—

TABLE 3.2: The performances (Recall@K%) of compared methods on two datasets under Weakly Supervised setting (WS) and Zero-Shot setting(ZS).

Dataset	VRD(WS)		VG(WS)		VRD(ZS)		VG(ZS)	
Metric	R@50	R@100	R@50	R@100	R@50	R@100	R@50	R@100
Base	29.36	29.36	45.78	46.01	14.10	14.10	11.04	11.04
Base+OA	31.47	31.47	47.46	47.72	16.53	16.53	13.09	13.09
STA w/o FT	32.84	32.84	47.23	47.39	18.24	18.24	13.72	13.72
STA w/o Res	35.10	35.10	50.89	51.13	19.01	19.01	18.03	18.03
STA	35.89	35.89	51.73	51.92	20.57	20.57	18.90	18.90
Peyre's A [76]	34.03	34.03	—	—	16.10	16.10	—	—

except for Lu's-VLK which is a multimodal model, all the methods compared here are visual models.

3.3.3 Results and Analysis

TABLE 3.1 and 3.2 show the performances of compared methods on two datasets of different experiment settings. As we can see, the proposed STA has the best

TABLE 3.3: Computed ratios of two kinds of feature maps

Dataset	OA	Base CNN	Dataset	OA	Base CNN
VRD	0.5027	0.4245	VG	0.4850	0.4132

performances compared with the other baselines and state-of-the-art models on both datasets. For example, compared to the Base+OA, the proposed STA can boost the Recall@100 of supervised, weakly supervised, and zero-shot relationship prediction by absolute 4.75%, 4.42%, 4.04%, respectively on VRD, and 4.41%, 4.2%, 5.81%, respectively on VG.

Comparing the results of Base+OA with Base, we can see that by adding OA conv-layers, the performances are improved. This observation is basically as expected since the number of parameters have been increased and thus the representation ability of the whole network is improved. By comparing the performance of STA w/o FT with Base+OA, we can find that, even OA conv-layers are not fine-tuned, the features which are pre-trained by **Shuffle-Then-Assemble** still have comparable performance with the Base+OA. And when the pre-trained OA conv-layers are further fine-tuned (STA w/o Res, STA), the performances will have a considerable boost. Such observations show that the success of the proposed method is not only due to the added small network (OA conv-layers), but also thanks to the proposed **Shuffle-Then-Assemble** pre-training strategy.

FIGURE 3.6 shows the R@100 of relationship prediction of the four relation types which are comparative, preposition, verb and spatial. From this, we can see that the proposed STA has the best performance in each relationship type on both datasets.

Analysis of feature maps. FIGURE 3.5 shows six qualitative examples of feature maps generated by three different strategies. By comparing the STA’s feature maps with Base and Base+OA, we can find that STA’s feature maps focus more on the overlap regions between subjects and objects. For example, in the second row, STA’s feature maps put more attention on people’s feet, which would provide cues for predicting the right relationship “stand on”.

The ratio: $(\sum_{i \in R_{over}} f(i)) / (\sum_{i \in R_{joint}} f(i))$, in TABLE 3.3, is also used to quantitatively validate that whether our model can focus more on the overlapped region or not. In this formula, $f(\cdot)$ is the normalized joint feature map of subject and object

region (averaged over channels, minimum value and maximum value are set to 0 and 1 respectively), R_{over} and R_{joint} mean the overlapped region and the joint region of that feature map respectively. We compare the ratios computed by Object-Agnostic (OA) feature and Base CNN feature on both VRD and VG datasets. From the quantitative results we can see that the proposed **Shuffle-Then-Assemble** pre-training strategy can help the relationship model capture attention on the shared regions between subject and object, resulting in better generalizability.

Analysis of Zero-Shot Setting. Since the main purpose of this research is to learn object-agnostic feature maps for more generalizable relationship models, one practical task is zero-shot learning. From TABLE 3.2, we can see that the proposed STA has the best performance on both datasets compared with other baselines and one state-of-the-art. This result can further validate the effectiveness of the proposed **Shuffle-Then-Assemble** pre-training strategy. From the qualitative examples in FIGURE 3.7, we can demonstrate that the reason why STA achieves better performance is due to the learned OA feature maps.

Failure mode. Our model will fail when one relationship depends heavily on the combination of two very specific objects. For example, for relationship “read” in VG dataset, the accuracy is 0.0769 by using our method and 0.2115 by using baseline. For this relationship, the subjects and objects are almost “person” and “book”. Under this condition, the object-related features will be very useful for predicting the relationship and if such information is deleted, the model will not recognize the right relationship.

3.3.4 Discussion and Future Work

For image captioning, the features extracted from the proposed STA-based encoder will not produce as better results as some frequently used feature extractor [1]. The major reason is that the features of SAT neglect the attribute knowledge of the objects, while the objects’ attributes are quite important for generating descriptive captions. We will try to deploy the SAT pre-training strategy into [1] for testing the results in the future days.

FIGURE 3.7: Qualitative feature maps of two zero-shot relationships on VRD. For each one, two feature maps of Base+OA with wrong prediction and STA with correct prediction are visualized by averaging over the 512 channels. We can see that by using the proposed Shuffle-Then-Assemble (STA), the RoI features are less likely biased to the objects and more focused on the regions of interaction of the two objects.

Chapter 4

Auto-encoding and Distilling Scene Graphs for Image Captioning¹

4.1 Introduction

Modern image captioners usually deploy end-to-end encoder-decoder frameworks [1, 32, 38, 156, 157], *i.e.*, the encoder transforms an image into vector representations and then the decoder generates captions by these representations. Such frameworks have experienced several significant developments due to its invention is inspired from neural machine translation [37], *e.g.*, the reinforced mechanism for sequence decoding [28, 29] and the top-bottom [24] and the bottom-up [1] visual attentions for dynamic encoding. However, there is a crux issue that has never been substantially resolved: when an unseen image scene is fed into the framework, only a simple and trivial caption about the salient objects will be gotten, such as “a dog on the floor”, which is no better than just a list of detected objects [32]. It is particularly embarrassing since the booming “mid-level” vision techniques nowadays can already detect and segment almost everything in an image [7, 158, 159].

¹This chapter is published as Xu Yang, Hanwang Zhang, Jianfei Cai. “Auto-encoding and Distilling Scene Graphs for Image Captioning.”, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Accepted. Digital Object Identifier: 10.1109/TPAMI.2020.3042192.

FIGURE 4.1: The sketch of our SGAE, which incorporates the auto-encoding of scene graphs (blue arrows) into the conventional encoder-decoder framework for image captioning (red arrows). The trainable shared dictionary $\mathcal{D}$ encodes the language inductive bias. In $\mathcal{D} \rightarrow \mathcal{S}$, the dark dot line from the blue arrow to the red arrow indexes the Knowledge Distillation strategy, which teaches the encoder-decoder to reason as the auto-encoder. The colors of the words in $\mathcal{S}$ correspond to the nodes in image and sentence scene graphs.

Humans are good at composing sentences about a visual scene. Unsurprisingly, cognitive evidence [160] demonstrates that the visually grounded language generation is largely attributed to “high-level” symbolic reasoning instead of the end-to-end manner, *i.e.*, once humans abstract the symbols from a scene, the generation will be almost *disentangled* from the visual perception. Take FIGURE 4.1 as the example, given “helmet-on-human” and “dirt-road”, humans can say “a man with a helmet in countryside” by using commonsense knowledge like “a dirt road appears in countryside”. Actually, we can consider such collocations and contextual inference in human language as the *inductive bias* that is apprehended by humans in everyday life, which makes humans perform better than machines in high-level reasoning [87, 161]. However, it is ineffective to directly exploit the inductive bias, *e.g.*, early template/rule-based caption models [162, 163], is well-known to achieve low performances compared to the modern encoder-decoder ones since there is a large gap between visual perception and language composition.

To alleviate such problems, we incorporate language inductive bias as additional

knowledge into the encoder-decoder framework for image captioning, which benefits from the complementary strengths of both symbolic reasoning and end-to-end multi-modal training. Specifically, scene graphs [82, 89] are used to bridge the gap between the two worlds. In this chapter, we define a scene graph ($\mathcal{G}$) as a unified representation that connects 1) the objects (or entities), 2) their attributes, and 3) their relationships in an image ($\mathcal{I}$) or a sentence ($\mathcal{S}$) by directed edges. We apply the spatial Graph Convolutional Networks (GCNs) [88, 164, 165] to embed the graph structure into the vector representations for seamlessly integrating them into the encoder-decoder frameworks. We expect such graph representations to transfer the language inductive bias from the pure language domain to the vision-language domain, then more descriptive captions can be generated.

In this chapter, we propose the Scene Graph Auto-Encoder (SGAE) to encode the language inductive bias. As shown in FIGURE 4.1, blue arrows demonstrates such sentence self-reconstruction network, which is in the $\mathcal{S} \rightarrow \mathcal{G}_{\mathcal{S}} \rightarrow \mathcal{D} \rightarrow \mathcal{S}$ pipeline. In this pipeline, $\mathcal{D}$ is a trainable dictionary set for the re-encoding purpose of the node features, the $\mathcal{S} \rightarrow \mathcal{G}_{\mathcal{S}}$ module is a fixed off-the-shelf scene graph language parser [84], $\mathcal{G}_{\mathcal{S}} \rightarrow \mathcal{D}$ is an Attentional GCN (AGCN) [60] to selectively encode the scene graph into embeddings, and $\mathcal{D} \rightarrow \mathcal{S}$ is a trainable RNN-based language decoder [1].

Note that $\mathcal{D}$ consists of descriptive and imaginary collocations, which is the language inductive bias extracted from training SGAE. $\mathcal{D} \rightarrow \mathcal{S}$ learns to compose these collocations into human-like captions. By sharing such $\mathcal{D}$ containing ample language inductive bias in the encoder-decoder pipeline $\mathcal{I} \rightarrow \mathcal{G}_{\mathcal{I}} \rightarrow \mathcal{D} \rightarrow \mathcal{S}$ (red arrows in FIGURE 4.1) and teaching this encoder-decoder to reason as the auto-encoder by Knowledge Distillation (KD) (vertical dark dot line $\mathcal{D} \rightarrow \mathcal{S}$ in FIGURE 4.1, also see Section 4.4.2), the language inductive bias is transferred to guide the end-to-end image captioning. Specifically, we introduce a visual scene graph detector [58] for the $\mathcal{I} \rightarrow \mathcal{G}_{\mathcal{I}}$ module and the $\mathcal{G}_{\mathcal{I}} \rightarrow \mathcal{D}$ module in the captioning pipeline is a multi-modal AGCN (MAGCN) to complement necessary visual cues that are missing due to the imperfect visual detection. Interestingly, $\mathcal{D}$ can be thought as a working memory [98] that helps re-key the encoded nodes from $\mathcal{I}$ or $\mathcal{S}$ to more generic representations with smaller domain gaps (see Section 4.3.3).

The proposed SGAE-based captioning model is implemented by applying the recently released Faster RCNN based visual encoder [6] and LSTM based language

decoder [1] with an RL-based training strategy [29]. We carry extensive experiments on MS-COCO [151] to validate the superiority of using SGAE in image captioning. Specifically, we achieve an absolute 9-point CIDEr-D improvement over a strong baseline Up-Down [1]. Our *single-model* achieves 129.6 CIDEr-D on the Karpathy split, which reaches a new state-of-the-art score and achieves 126.6 CIDEr-D on the official test server, which is even comparable to many ensemble models. Moreover, we apply two more challenging settings to validate the transferability of our SGAE, which are transferring inductive bias from other language corpora and unpaired image captioning. In particular, when using another Web language corpus to train the auto-encoder, consistent improvements in CIDEr-D can be observed in CIDEr-D than the captioner without this corpus. Our SGAE also obtains a 10.6-point improvement in CIDEr-D than the baseline on the unpaired image captioning task.

To summarize up, we have the following technical contributions in this model:

- We propose a novel Scene Graph Auto-Encoder (SGAE) to transfer the language inductive bias, which is treated as additional knowledge.
- We design a multi-modal Attentional Graph Convolutional Network to incorporate scene graphs into visual representations.
- We develop an SGAE-based encoder-decoder image captioner, which contains a shared dictionary and a Knowledge Distillation objective, both helping the language decoder generate better captions.
- We carry various experiments on different captioning tasks and settings to confirm the effectiveness of our SGAE.

4.2 Encoder-Decoder Revisited

As illustrated in FIGURE 4.2, given an image $\mathcal{I}$, the target of image captioning is to generate a natural language sentence $\mathcal{S} = \{s_1, s_2, \dots, s_T\}$ describing the image.

FIGURE 4.2: Top: the conventional encoder-decoder. Bottom: our proposed encoder-decoder, where the novel SGAE embeds the language inductive bias in the dictionary set, which is shared to the image captioner, and in the decoder $\mathcal{S}$ -RNN of SGAE, whose knowledge is distilled into the decoder $\mathcal{I}$ -RNN of the image captioner.

We can formulate the encoder-decoder image captioner as:

$$\begin{aligned}
 \text{Encoder: } \mathcal{V} &\leftarrow \mathcal{I}, \\
 \text{Map: } \hat{\mathcal{V}} &\leftarrow \mathcal{V}, \\
 \text{Decoder: } \mathcal{S} &\leftarrow \hat{\mathcal{V}}.
 \end{aligned} \tag{4.1}$$

In this formulation, the encoder is usually set to a Convolutional Neural Network (CNN) [6, 36] for extracting the image feature $\mathcal{V}$; the map is the widely used attention mechanism [1, 24] that re-encodes the visual features into more informative $\hat{\mathcal{V}}$ which is dynamic to language generation; the decoder can be set to an RNN-based language decoder that generates the caption $\mathcal{S}$. Given a ground-truth caption $\mathcal{S}^*$

for $\mathcal{I}$, the model can be trained by minimizing cross-entropy loss:

$$L_{XE} = -\log P(\mathcal{S}^*), \quad (4.2)$$

or by maximizing a reinforcement learning (RL) based reward [29] as:

$$R_{RL} = \mathbb{E}_{\mathcal{S}^s \sim P(\mathcal{S})}(r(\mathcal{S}^s; \mathcal{S}^*)), \quad (4.3)$$

where r is a sentence-level metric for the sampled sentence $\mathcal{S}^s$ and the ground-truth $\mathcal{S}^*$, which is usually set to the CIDEr-D [166] metric.

This encoder-decoder framework is the core pillar underpinning almost all the modern neural network based image captioners since S&T [20]. However, it is widely shown less effective for generating descriptive captions [32]. In our model, we exploit the language inductive bias as additional knowledge and incorporate it into the encoder-decoder for more human-like image captioning, which is sketched in the bottom part of FIGURE 4.2. The proposed framework can be formulated as:

$$\begin{aligned} \textbf{Encoder: } & \mathcal{V} \leftarrow \mathcal{I}, \\ \textbf{Map: } & \hat{\mathcal{V}} \leftarrow R(\mathcal{V}, \mathcal{G}_{\mathcal{I}}; \mathcal{D}), \mathcal{G}_{\mathcal{I}} \leftarrow \mathcal{V}, \\ \textbf{Decoder: } & \mathcal{S} \leftarrow \hat{\mathcal{V}}. \end{aligned} \quad (4.4)$$

By comparing Eq. (4.1) and Eq. (4.4), it can be found that the Map module is modified that we introduce the scene graph $\mathcal{G}_{\mathcal{I}}$ into a re-encoder R , which is parameterized by a shared dictionary set $\mathcal{D}$. Specifically, a Scene Graph Auto-Encoder (SGAE) is proposed to learn the dictionary set $\mathcal{D}$ during the sentence self-reconstruction (see Section 4.3) with the help of scene graphs. In this process, the decoder $\mathcal{S}$ -RNN is used to reconstruct the sentences from their scene graphs. After that, an encoder-decoder is equipped with this proposed SGAE, which is our overall image captioner (see Section 4.4). In particular, a novel Multi-modal Attentional Graph Convolutional Network (MAGCN) is used to re-encode the image features (see Section 4.4.1), and the dictionary set $\mathcal{D}$ and a Knowledge Distillation strategy are exploited to train the encoder-decoder pipeline, which narrows the gap between vision and language (see Section 4.4.2).

4.3 Auto-Encoding Scene Graphs

In this section, we will discuss how to learn $\mathcal{D}$ through self-reconstructing a sentence $\mathcal{S}$ from its scene graph. As sketched in FIGURE 4.2, the pipeline of reconstructing $\mathcal{S}$ is also an encoder-decoder framework. Thus, we can formulate SGAE by slightly abusing the notations as:

$$\begin{aligned} \text{Encoder: } \mathcal{X} &\leftarrow \mathcal{G}_{\mathcal{S}} \leftarrow \mathcal{S}, \\ \text{Map: } \hat{\mathcal{X}} &\leftarrow R(\mathcal{X}; \mathcal{D}), \\ \text{Decoder: } \mathcal{S} &\leftarrow \hat{\mathcal{X}}. \end{aligned} \tag{4.5}$$

Next, we introduce each part in Eq. (4.5).

4.3.1 Scene Graphs

We discuss how to deploy the step $\mathcal{G}_{\mathcal{S}} \leftarrow \mathcal{S}$, *i.e.*, from sentence to scene graph. Formally, a scene graph can be represented by a tuple $\mathcal{G}_{\mathcal{S}} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ and $\mathcal{E}$ are the node and edge sets, respectively. There are three kinds of nodes in $\mathcal{N}$: object node o , attribute node a , and relationship node r . In this chapter, o_i , r_{ij} , and $a_{i,l}$ denote the i -th object, the relationship between o_i and o_j , the l -th attribute of o_i , respectively. Each node in $\mathcal{N}$ is represented by a d -dimensional learnable embedding vector of the corresponding semantic label, *i.e.*, $\mathbf{e}_o$, $\mathbf{e}_a$, and $\mathbf{e}_r$ for object, attribute, and relationship label embedding, respectively. In the experiments, d is set to 1,000. The edges in $\mathcal{E}$ are formulated as follows:

- if an object o_i has an attribute $a_{i,l}$, a directed edge is connected from $a_{i,l}$ to o_i ;
- if there is one relationship triplet $\langle o_i - r_{ij} - o_j \rangle$ in the graph, two directed edges are connected from o_i to r_{ij} and from r_{ij} to o_j , respectively.

One example of $\mathcal{G}_{\mathcal{S}}$ is sketched in FIGURE 4.3, which contains 7 nodes in $\mathcal{N}$ and 6 directed edges in $\mathcal{E}$.

We exploit Spice [84] to parse sentence scene graphs $\mathcal{G}_{\mathcal{S}}$ from captions. In this parser, a syntactic dependency tree is first built and then a rule-based method [83] is used for changing the tree to a scene graph.

FIGURE 4.3: The sketch of the proposed Attentional Graph Convolutional Network. In particular, this is a spatial graph convolutional network, where the colored neighborhood is “convolved” into the resultant embedding. The dashed lines in (b) and (c) denote that these knowledge flows are softly filtered by the calculated attention weights.

4.3.2 Attentional Graph Convolutional Network

Here we discuss the deployment about the process $\mathcal{X} \leftarrow \mathcal{G}_S$ in Eq. (4.5), *i.e.*, how to build a new set of context-aware embeddings $\mathcal{X}$ from the original node embeddings e_o , e_a , and e_r . Specifically, there are three types of d -dimensional embeddings in $\mathcal{X}$: relationship embedding $\mathbf{x}_{r_{ij}}$ for r_{ij} , object embedding $\mathbf{x}_{o_i}$ for o_i , and attribute embedding $\mathbf{x}_{a_i}$ for o_i . In the experiments, d is set to 1,000. There are four *spatial graph convolutions*: g_r , g_a , g_s , and g_o are used to build the context-aware embedding set $\mathcal{X}$. In the deployment, all these four functions share the same structure while their parameters are independent: a vector concatenation input to two fully-connected layers with a ReLU in between (FC-ReLU-FC). We apply two-layer perceptrons instead of one-layer perceptron for getting stronger representation power [88].

Relationship Embedding $\mathbf{x}_{r_{ij}}$: Given a relationship triplet $\langle o_i - r_{ij} - o_j \rangle$ in $\mathcal{G}_S$, we incorporate this context into the relationship embedding $\mathbf{x}_{r_{ij}}$ as:

$$\mathbf{x}_{r_{ij}} = g_r([e_{o_i}, e_{r_{ij}}, e_{o_j}]), \quad (4.6)$$

where $[\cdot, \cdot]$ denotes the concatenation operation. One example about this relationship embedding is sketched in FIGURE 4.3 (a).

Attribute Embedding $\mathbf{x}_{a_i}$: Given one object node o_i with all its attributes $a_{i,1:N_{a_i}}$ in $\mathcal{G}_S$, where N_{a_i} is the number of attributes this object o_i has, an Attentional GCN (AGCN) [60] is used to incorporate the context of this object with its

attributes into the attribute embedding $\mathbf{x}_{a_i}$:

$$\mathbf{x}_{a_i} = \sum_{l=1}^{Na_i} p_l^a g_a([\mathbf{e}_{o_i}, \mathbf{e}_{a_{i,l}}]), \quad (4.7)$$

where p_l^a is the adaptive soft attention weight for the l -th attribute:

$$\begin{aligned} q_l^a &= \mathbf{w}_a^T \tanh(\mathbf{W}_a[\mathbf{e}_{o_i}, \mathbf{e}_{a_{i,l}}]), \\ \mathbf{p}^a &= \{p_1^a, p_2^a, \dots, p_{Na_i}^a\} = \text{Softmax}(\{q_1^a, q_2^a, \dots, q_{Na_i}^a\}), \end{aligned} \quad (4.8)$$

where $\mathbf{w}_a$ is a trainable vector and $\mathbf{W}_a$ is a trainable matrix.

We apply this AGCN because not all the attributes are equally important to the final caption and hence we use the context of o_i with its attribute $a_{i,l}$ to decide whether this attribute is important or not. By this AGCN, the knowledge flow in the graph can be softly filtered for generating more accurate captions. One example of this attribute embedding is sketched in FIGURE 4.3 (b).

Object Embedding $\mathbf{x}_{o_i}$: In $\mathcal{G}_S$, an object o_i can act as “a subject” or “an object” in relationships, which means o_i will play distinguishable roles denoted by different edge directions. Then, different functions should be used to incorporate such knowledge, *i.e.*, the functions g_s and g_o . To avoid the ambiguous meaning of the same “predicate” in a different context, knowledge of the whole relationship triplets where o_i appears should be incorporated into $\mathbf{x}_{o_i}$. Here is one simple example for ambiguity, in $\langle \text{hand-with-cup} \rangle$, the predicate “with” may mean “hold”, while in $\langle \text{head-with-hat} \rangle$, “with” may mean “wear”. Moreover, similar to the attribute embedding, AGCN is also applied here to modulate the knowledge flows. Therefore, $\mathbf{x}_{o_i}$ is got as:

$$\begin{aligned} \mathbf{x}_{o_i} &= \sum_{o_j \in \text{obj}(o_i)} p_j^o g_o([\mathbf{e}_{o_i}, \mathbf{e}_{o_j}, \mathbf{e}_{r_{ij}}]) \\ &+ \sum_{o_k \in \text{sbj}(o_i)} p_k^s g_s([\mathbf{e}_{o_k}, \mathbf{e}_{o_i}, \mathbf{e}_{r_{ki}}]), \end{aligned} \quad (4.9)$$

where $o_j \in \text{obj}(o_i)$ and $o_k \in \text{sbj}(o_i)$ indicate (o_i, o_j) and (o_k, o_i) form subject-object pairs, respectively, *e.g.*, $\text{obj}(o_1) = \{o_2\}$ in FIGURE 4.3 (c); and p_j^o, p_k^s are soft attention weights computed in similar ways with the same trainable vector $\mathbf{w}_o$ and

FIGURE 4.4: The sketch of the re-encoder function R . The black dashed block demonstrates this function. The top part shows how “imagination” is achieved by re-encoding: the green line draws the generated phrase by re-encoding, while the red line draws the one without re-encoding. For convenience, we only sketch the attribute dictionary $\mathcal{D}_A$ here.

matrix $\mathbf{W}_o$, *e.g.*, p_j^o is:

$$\begin{aligned} q_j^o &= \mathbf{w}_o^T \tanh(\mathbf{W}_o[\mathbf{e}_{o_i}, \mathbf{e}_{o_j}, \mathbf{e}_{r_{ij}}]), \\ \mathbf{p}^o &= \{p_1^o, p_2^o, \dots, p_{N_{obj_i}}^o\} = \text{softmax}(\{q_1^o, q_2^o, \dots, q_{N_{obj_i}}^o\}), \end{aligned} \quad (4.10)$$

where N_{obj_i} is the size of the set $obj(o_i)$. FIGURE 4.3 (c) shows this object embedding.

4.3.3 Dictionary

Here we detail how to learn the dictionary set $\mathcal{D}$ and how to use it to re-encode $\hat{\mathcal{X}} \leftarrow R(\mathcal{X}; \mathcal{D})$ in Eq. (4.5). The crux idea is motivated from using the working memory, which preserves a global knowledge base for latter inference. Such an idea is widely used in various learning tasks like textual QA [100], VQA [99], and one-shot classification [98]. In this chapter, our $\mathcal{D}$ targets to incorporate language inductive bias in language composition, which is achieved by the dictionary learning during the sentence self-reconstruction. Here we use three different dictionaries $\mathcal{D} = \{\mathcal{D}_R, \mathcal{D}_A, \mathcal{D}_O\}$ to respectively preserve relationship, attribute, and object embeddings. We remove the subscript of $\mathcal{D}$ for conveniently introducing how it works in our pipeline. In particular, a dictionary is denoted as a $d \times K$ matrix

$\mathcal{D} = \{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_K\}$, where K is the number of the elements. In the experiment, we set K to 5,000 for each dictionary. Given an embedding vector $\mathbf{x} \in \mathcal{X}$, the re-encoder function R is formulated as:

$$\hat{\mathbf{x}} = R(\mathbf{x}; \mathcal{D}) = \mathcal{D}\boldsymbol{\alpha} = \sum_{k=1}^K \alpha_k \mathbf{d}_k, \quad (4.11)$$

where $\boldsymbol{\alpha} = \text{softmax}(\mathcal{D}^T \mathbf{x})$ is the differentiable attention operation in memory network [100, 102] and since the computations are all differentiable, $\mathcal{D}$ can be learned by end-to-end training. FIGURE 4.4 sketches such a operation.

After learning, intuitively, the dictionary set $\mathcal{D}$ preserves the “additional knowledge” which humans generally apply to generate descriptive captions. After inserting $\mathcal{D}$ into the image captioner pipeline, even the provided image scene graphs are noisy and simple, the captioner will apply the preserved “additional knowledge” in $\mathcal{D}$ to transform them into embeddings for more descriptive captions. FIGURE 4.6 illustrates a few examples about this process, such “additional knowledge” can summarize “a street with many cars” as “busy” in (a); add one more descriptive phrase “lush green” to decorate the trees in (b); achieve certain commonsense reasoning that infers “a ripe banana” from “a yellow and dotted banana” in (c); and use one more suitable action “perch” for the bird to replace the trivial relation “on” in (d).

4.3.4 Unpaired Image Captioning

In this section, we discuss how to use our SGAE to solve a more challenge task: unpaired image captioning [51], by which we can confirm the transferability of SGAE between the pure language domain and the vision-language domain. In unpaired image captioning, only an image set $\mathcal{I}$ and a language corpus $\mathcal{S}$ are available, while we do not have any paired image-caption samples for training the image captioner. To solve this task, an auto-encoder is first trained during reconstructing sentences $\mathcal{S}$ from the scene graphs $\mathcal{G}_{\mathcal{S}}$ parsed from them. Significantly, we only train SGAE in this unpaired setting and at the inference stage, after extracting image scene graphs $\mathcal{G}_{\mathcal{I}}$ from the images, they will be directly fed into the trained auto-encoder

for getting the captions. This process can be formulated as:

$$\begin{aligned} \text{Training: } \mathcal{S} &\leftarrow \hat{\mathcal{X}} \leftarrow \mathcal{D} \leftarrow \mathcal{X} \leftarrow \mathcal{G}_S \leftarrow \mathcal{S}, \\ \text{Testing: } \mathcal{S} &\leftarrow \hat{\mathcal{X}} \leftarrow \mathcal{D} \leftarrow \mathcal{X} \leftarrow \mathcal{G}_I \leftarrow \mathcal{I}. \end{aligned} \quad (4.12)$$

The experiment results in Section 4.5.4.2 confirm the effectiveness of our SGAE in unpaired image captioning compared with some state of the art models.

4.3.5 Training of SGAE

We deployed the top-down attention structure [1] as the $\mathcal{S}$ -RNN for reconstructing $\mathcal{S}$, which contains two LSTM layers and one attention module in between. This structure has proven to be a powerful one and most state-of-the-art captioning models use it as the backbone, *e.g.*, GCN-LSTM [2], RFNet [167], and LBPf [39].

Here we first show the prototypical architecture of our $\mathcal{S}$ -RNN and then specify the values of the variables in this network. Specifically, at time step t , given the last word s_{t-1} and the embedding set $\mathcal{X}$, $\mathcal{S}$ -RNN calculates the word probability vector $\mathbf{z}_t$ as follows:

$$\begin{aligned} \text{Input: } \mathbf{i}_t &= [\text{Embed}(s_{t-1}), \mathbf{h}_2^{t-1}] \\ \text{LSTM1: } \mathbf{h}_1^t &= \text{LSTM}_1(\mathbf{i}_t; \mathbf{h}_1^{t-1}), \\ \text{Attention: } \hat{\mathbf{x}}^t &= \text{ATT}(\mathcal{X}, \mathbf{h}_1^t), \\ \text{LSTM2: } \mathbf{h}_2^t &= \text{LSTM}_2([\mathbf{h}_1^t, \hat{\mathbf{x}}^t]; \mathbf{h}_2^{t-1}), \\ \text{Output: } \mathbf{z}_t &= \text{Softmax}(\text{FC}(\mathbf{h}_2^t)), \end{aligned} \quad (4.13)$$

where $[\cdot]$ means the concatenation operation; $\text{Embed}(\cdot)$ is a learnable word embedding layer; $\text{LSTM1}(\cdot)$ and $\text{LSTM2}(\cdot)$ are two different LSTM layers whose hidden unit sizes are both set to 1,000; and the attention sub-network ATT is:

$$\begin{aligned} \text{Input: } \mathcal{X} &= \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}, \mathbf{h}_1^t, \\ \text{Attention Weights: } b_n^t &= \boldsymbol{\omega}_b^T \tanh(\mathbf{W}_v \mathbf{x}_n + \mathbf{W}_h \mathbf{h}_1^t), \\ \boldsymbol{\beta}^t &= \{\beta_1^t, \beta_2^t, \dots, \beta_N^t\} = \text{softmax}(\mathbf{b}^t), \\ \text{Output: } \hat{\mathbf{x}}^t &= \sum_{n=1}^N \beta_n^t \mathbf{x}_n, \end{aligned} \quad (4.14)$$

where $\mathbf{W}_v$, $\mathbf{W}_h$ are trainable matrices and $\boldsymbol{\omega}_b$ is a trainable vector.

Given a sentence $\mathcal{S}$, we first extracted its scene graph and then used AGCN operations (Eq. (4.6), (4.7), (4.9)) to compute relationship, attribute, and object embeddings and then grouped them as three embedding sets $\mathcal{X}_R, \mathcal{X}_A, \mathcal{X}_O$. For example, $\mathcal{X}_R$ contains all the relationship embeddings $\mathbf{x}_{r_{ij}}$ computed by Eq. (4.6). These embedding sets were input into three dictionaries $\mathcal{D}_R, \mathcal{D}_A, \mathcal{D}_O$ to get the re-encoded embedding sets $\hat{\mathcal{X}}_R, \hat{\mathcal{X}}_A, \hat{\mathcal{X}}_O$ by Eq. (4.11) (e.g., $\hat{\mathcal{X}}_A = R(\mathcal{X}_A; \mathcal{D}_A)$), respectively. Then we grouped $\hat{\mathcal{X}}_R, \hat{\mathcal{X}}_A, \hat{\mathcal{X}}_O$ together as the embedding set $\mathcal{X}$ in Eq. (4.14) to calculate the attended vector $\hat{\mathbf{x}}^t$. Then we used Eq. (4.13) to calculate the word probability vector $\mathbf{z}_t$ and trained the network by cross-entropy loss (Eq. (4.2)) where the ground-truth probability vector $\mathbf{z}_t^*$ is an one-hot vector corresponding to the ground-truth word s_t^* .

The whole SGAE (AGCN, $\mathcal{S}$ -RNN, and the dictionary set) were all randomly initialized. When we trained this SGAE, in the first 10 epochs, we removed the dictionary set in the pipeline to train a rudiment AGCN and $\mathcal{S}$ -RNN. In the next 30 epochs, we inserted the dictionary set into the pipeline to jointly train the dictionary, AGCN, and $\mathcal{S}$ -RNN. Removing the dictionary set at the beginning was to avoid learning trivial values since the graph embeddings ($\mathcal{X}_R, \mathcal{X}_A, \mathcal{X}_O$) had not learnt meaningful inductive bias at that moment.

The learning rate was initialized to $5e^{-4}$ for all parameters and decayed by 0.8 for every 5 epochs. The batch size was set to 100 and Adam optimizer [168] was used. Note that the training of SGAE is unsupervised, that is, SGAE offers the potential never-ending learning from large-scale unsupervised inductive bias for $\mathcal{D}$. Some preliminary studies are reported in Section 4.5.2.2. Note that these training strategies were used in both paired and unpaired image captioning.

4.4 SGAE-based Encoder-Decoder

In this section, we introduce the overall model: SGAE-based Encoder-Decoder image captioner as in FIGURE 4.2 and Eq. (4.4).

4.4.1 Multi-modal AGCN

The original image features extracted by CNN are not ready for the dictionary re-encoding (Eq. (4.11)) due to the large gap between vision and language. To this end, we propose a Multi-modal Attentional Graph Convolution Network (MAGCN) to map the visual features $\mathcal{V}$ into a set of scene graph-modulated features $\mathcal{V}'$.

Here, the scene graph $\mathcal{G}_{\mathcal{I}}$ is extracted by an image scene graph parser that contains an object proposal detector, an attribute classifier, and a relationship classifier. In our implementation, we use Faster-RCNN as the object detector [6], MOTIFS relationship detector [58] as the relationship classifier, and we use our attribute classifier: an FC-ReLU-FC-Sigmoid network head for multi-label classification. The image-parsed $\mathcal{G}_{\mathcal{I}}$ is different from the sentence-parsed $\mathcal{G}_{\mathcal{S}}$ in the sense that the object node o_i here contains knowledge from two domains: the semantic domain (the predicted category of this object) and the visual domain (the RoI feature extracted from the pre-trained Faster RCNN). Thus we have two different representations: the label embedding $\mathbf{e}_{o_i}$ (*e.g.*, $\mathbf{e}_{o_i} = \text{Embed}(\text{"dog"})$ for category "dog") and the RoI feature $\mathbf{v}_{o_i}$ to represent o_i . Since the modality differences are important in a vision-language task like image captioning, we hope to get a fused feature which contains the knowledge of such modality differences. Thus, following [169], we fuse $\mathbf{e}_{o_i}$ with $\mathbf{v}_{o_i}$ into a new node feature $\mathbf{u}_{o_i}$:

$$\mathbf{u}_{o_i} = \text{ReLU}(\mathbf{W}_1 \mathbf{e}_{o_i} + \mathbf{W}_2 \mathbf{v}_{o_i}) - (\mathbf{W}_1 \mathbf{e}_{o_i} - \mathbf{W}_2 \mathbf{v}_{o_i})^2, \quad (4.15)$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are the trainable matrices, here the $(\cdot)^2$ is the elementwise square operation. In this formula, the first term is a general operation to fuse the features of two different modalities, while the second part calculates their differences. The rest attribute and relationship node representations $\mathbf{u}_a$ and $\mathbf{u}_r$ are the learnable label embeddings of the corresponding attribute and relationship labels which are predicted by our attribute and relationship classifiers, respectively. The differences between the two scene graphs generated from $\mathcal{I}$ and $\mathcal{S}$ are visualized in FIGURE 4.1, where $\mathcal{G}_{\mathcal{I}}$ is usually much simpler and noisier than $\mathcal{G}_{\mathcal{S}}$, *e.g.*, there are so many relationships "on" in $\mathcal{G}_{\mathcal{I}}$.

Similar to AGCN used in Section 4.3.2, MAGCN also has an ensemble of four functions f_r , f_a , f_s and f_o , each of which has the same structure: FC-ReLU-FC with independent parameters. The computations of the relationship, attribute and

FIGURE 4.5: The sketch of Knowledge Distillation, where **Con** denotes the concatenation operation.

object embeddings are similar to Eq. (4.6), Eq. (4.7), and Eq. (4.9) with the inputs $\{\mathbf{u}_o, \mathbf{u}_a, \mathbf{u}_r\}$. By replacing the functions $g_{\#}$ by $f_{\#}$ and the inputs $\mathbf{e}_{\#}$ by $\mathbf{u}_{\#}$, we can calculate the multi-modal graph embeddings of $\mathcal{G}_{\mathcal{I}}$. Similarly, as in Section 4.3.5, the embeddings are grouped into three sets: $\mathcal{V}'_R, \mathcal{V}'_A, \mathcal{V}'_O$ and these three sets are further grouped into an embedding set $\mathcal{V}'$. Then we adopt Eq. (4.11) to re-encode each representation $\mathbf{v}'$ in $\mathcal{V}'$ with respect to the corresponding dictionary (*e.g.*, using $\mathcal{D}_R$ to re-encode the embeddings in $\mathcal{V}'_R$) to get $\hat{\mathbf{v}}$ and group these re-encoded representations as $\hat{\mathcal{V}}$:

$$\hat{\mathcal{V}} = R(\mathcal{V}'; \mathcal{D}). \quad (4.16)$$

4.4.2 Knowledge Distillation

We deploy another top-down attention structure [1] $\mathcal{I}$ -RNN for generating captions and the network structure is the same as Eq. (4.13) and (4.14) but with different inputs and parameters. Specifically, in $\mathcal{I}$ -RNN, the input feature set $\mathcal{X}$ in Eq. (4.14) is set to $\tilde{\mathcal{V}}$, which is the concatenation of $\mathcal{V}'$ and $\hat{\mathcal{V}}$:

$$\tilde{\mathcal{V}} = \{\tilde{\mathbf{v}}_1, \dots, \tilde{\mathbf{v}}_N\} = \{[\mathbf{v}'_1, \hat{\mathbf{v}}_1], \dots, [\mathbf{v}'_N, \hat{\mathbf{v}}_N]\}, \quad (4.17)$$

where $[\cdot, \cdot]$ means the concatenation operation and N is the number of the representations in $\mathcal{V}'$ (or $\hat{\mathcal{V}}$). In this way, $\mathcal{V}'$ and $\hat{\mathcal{V}}$ respectively provide visual clues from the image and high-level semantic contexts from language inductive bias to the decoder for correct and human-like captions.

Compared with $\mathcal{S}$ -RNN (Section 4.3.5) where the input embedding $\mathbf{x}$ is a d -dimensional representation, the input embedding of $\mathcal{I}$ -RNN $\tilde{\mathbf{v}}$ is a $2d$ -dimensional representation. Therefore, the sizes of the LSTM layers and the attention modules in $\mathcal{I}$ -RNN are different from $\mathcal{S}$ -RNN. As a result, we can not directly use the parameters of $\mathcal{S}$ -RNN to initialize $\mathcal{I}$ -RNN, which weakens the transferability

of the inductive bias. Fortunately, Knowledge Distillation [105] provides another perspective about knowledge which is the mapping from input vectors to output vectors of a trained system. Inspired by this, we can treat the inductive bias as the mapping from input to output of the trained $\mathcal{S}$ -RNN and distill such inductive bias to $\mathcal{I}$ -RNN for more descriptive captions.

Specifically, when we apply Knowledge Distillation to train $\mathcal{I}$ -RNN, as shown in FIGURE 4.5, we first input the re-encoded representation set $\hat{\mathcal{V}}$ of an image into the trained $\mathcal{S}$ -RNN, *i.e.*, setting $\mathcal{X}$ to $\hat{\mathcal{V}}$ in Eq. (4.14) and then calculating Eq. (4.13) to get a series of word probabilities $\hat{\mathcal{Z}} = \{\hat{z}_1, \dots, \hat{z}_T\}$ with the temperature set to 5 in the softmax layer, where $\hat{z}_t$ is the output of the softmax layer at time step t . Meanwhile, we input the concatenation feature set $\tilde{\mathcal{V}}$ of the same image into $\mathcal{I}$ -RNN, *i.e.*, setting $\mathcal{X}$ to $\tilde{\mathcal{V}}$ in Eq. (4.14) to get a series of word probabilities $\tilde{\mathcal{Z}} = \{\tilde{z}_1, \dots, \tilde{z}_T\}$ with the same temperature. Then, we calculate the KL divergence between $\hat{\mathcal{Z}}$ and $\tilde{\mathcal{Z}}$ as the Knowledge Distillation objective:

$$L_{KD} = \text{KL}(\hat{\mathcal{Z}}||\tilde{\mathcal{Z}}) = \sum_{t=1}^T \text{KL}(\hat{z}_t||\tilde{z}_t), \quad (4.18)$$

which teaches $\mathcal{I}$ -RNN to reason as $\mathcal{S}$ -RNN.

4.4.3 Training and Inference

Following the common practice in deep-learning feature transfer [170, 171], we used the SGAE pre-trained $\mathcal{D}$ in auto-encoder as the initialization for the $\mathcal{D}$ in our overall encoder-decoder for image captioning (Eq.(4.4)). In particular, we intentionally used a very small learning rate to fine-tune $\mathcal{D}$ to impose the sharing purpose. The overall training loss is hybrid: we first used $L_{XE} + 25L_{KD}$ for 20 epochs and then used $-R_{RL} + 5L_{KD}$ for another 80 epochs; where L_{XE} and R_{RL} are cross-entropy loss (Eq. (4.2)) and RL-based reward (Eq. (4.3)), respectively. We set the importance weight of L_{KD} to 25 in the first 20 epochs since the magnitudes of the gradients of L_{KD} scale as $1/5^2$ (temperature is set to 5) [105] and then empirically set this importance weight to 5 in the other 80 epochs. The learning rate was initialized to $5e^{-5}$ for $\mathcal{D}$ and $5e^{-4}$ for the other parameters. All these learning rates were decayed by 0.8 for every 5 epochs. The batch size was set to

100 and Adam optimizer [168] was used. For inference in language generation, we adopted the beam search strategy [29] with a beam size of 5.

4.5 Experiments

4.5.1 Datasets and Metrics

MS-COCO [151]. There are two splits of MS-COCO: the official online test split and the 3rd-party Karpathy split [172]. The first split has 82,783/40,504/40,775 train/val/test images and the second split has 113,287/5,000/5,000 train/val/test images. Each of image has about 5 human labelled captions for training. For captions, we used the following steps to pre-process them: we first tokenized the texts on white space; then we change all the words to lowercase; we also deleted the words which appear less than 5 times; at last, we trimmed each caption to a maximum of 16 words. This results in a vocabulary of 10,369 words.

Visual Genome [56] (VG). This dataset has abundant scene graph annotations, *e.g.*, object categories, object attributes, and pairwise relationships, which can be exploited to respectively train the object proposal detector, the attribute classifier, and the relationship classifier, as our image scene graph parser (see Section 4.4.1). Since the object, attribute, and relationship annotations are very noisy in VG dataset, we filtered them by keeping the objects, attributes, and relationships which appear more than 2,000 times in the training set. After filtering, the remained 305 objects, 103 attributes, and 64 relationships were used to train our object detector, attribute classifier and relationship classifier, respectively.

SentiCap [173]. Compared with MS-COCO and VG which only contain factual descriptions (*e.g.*, “a girl” or “a street”) of the images, this dataset has a quite different text distribution: it provides 4,892 positive (*e.g.*, “a beautiful girl”) and 3,977 negative (*e.g.*, “an ugly street”) sentiment captions.

We used a similar way as in MS-COCO to pre-process the captions in VG and SentiCap. It is noteworthy that we only exploited these additional text descriptions in the ablation studies in Section 4.5.2.2, which were not used for the training in the other experiments.

Metrics. We employed five standard automatic evaluation metrics to measure the similarities between the generated captions and the ground-truth captions: CIDEr-D [166], BLEU [174], METEOR [175], ROUGE [176] and SPICE [84].

4.5.2 Ablation Studies

We conducted extensive ablation studies for architecture (Section 4.5.2.1), language corpus (Section 4.5.2.2), and sentence reconstruction quality (Section 4.5.2.3). For simplicity, we use **SGAE-KD** to denote our SGAE-based encoder-decoder captioning model trained by the Knowledge Distillation objective L_{KD} (Eq. (4.18)) to distinguish from **SGAE**, where L_{KD} was not used.

4.5.2.1 Architecture

Comparing Methods. To quantify the importance of the proposed AGCN, MAGCN, the dictionary set $\mathcal{D}$, and the Knowledge Distillation objective, we gradually incorporated these techniques into a strong baseline and compared their performances:

Base: We followed the pipeline given in Eq. (4.1) and the network in Eq. (4.13) without using AGCN, MAGCN, $\mathcal{D}$, and L_{KD} . This baseline is the benchmark for other ablation baselines. Importantly, this is also the reimplemented version of Up-Down model [1].

MAGCN: We added MAGCN to compute the multi-modal embedding set $\mathcal{V}'$ for validating the importance of MAGCN. We also compared MAGCN with the one used in our preliminary work [3]: **MGCN**, where attentional gates were not incorporated into the GCN. Noteworthy, this is one kind of the reimplemented version of GCN-LSTM model [2] with a much smaller batch size (100) compared with the original one (1024).

$\mathcal{D}$ w/o AGCN: We learned $\mathcal{D}$ by using Eq. (4.5), while AGCN was not used and only the word embeddings of $\mathcal{S}$ were input to the decoder. Also, MAGCN was not used in Eq. (4.4). This baseline is designed for validating the importance of AGCN and MAGCN.

TABLE 4.1: The performances of various methods on MS-COCO Karpathy split. The metrics: B@N, M, R, C and S denote BLEU@N, METEOR, ROUGE-L, CIDEr-D and SPICE. The best results are marked in boldface.

Models	B@1	B@4	M	R	C	S
SCST [29]	—	34.2	26.7	55.7	114.0	—
LSTM-A [40]	78.6	35.5	27.3	56.8	118.3	20.8
StackCap [31]	78.6	36.1	27.4	—	120.4	—
Up-Down [1]	79.8	36.3	27.7	56.9	120.1	21.4
CAVP [38]	—	38.6	28.3	58.5	126.3	21.6
RFNet [167]	80.4	37.9	28.3	58.3	125.7	21.7
LBPF [39]	80.5	38.3	28.5	58.4	127.6	22.0
CNM [157]	80.8	38.7	28.4	58.7	127.4	21.8
GCN-LSTM [2]	80.5	38.2	28.5	58.3	127.6	22.0
Base	79.9	36.8	27.7	57.0	120.6	20.9
MGCN	80.2	37.2	27.9	57.5	123.4	21.2
MAGCN	80.3	37.4	28.0	57.6	123.9	21.3
$\mathcal{D}$ w/o AGCN	80.2	37.3	27.8	58.0	124.2	21.4
$\mathcal{D}\#1$ +AGCN	80.5	37.5	28.0	58.2	125.9	21.5
$\mathcal{D}\#3$ +GCN	80.5	37.5	28.1	58.3	126.0	21.4
$\mathcal{D}\#3$ +AGCN	80.6	37.8	28.1	58.5	126.4	21.7
SGAE-pre [3]	80.8	38.4	28.4	58.6	127.8	22.1
SGAE	80.9	38.5	28.5	58.8	128.4	22.2
SGAE-KD	81.0	38.8	28.8	58.8	129.6	22.4

$\mathcal{D}\#3$ +AGCN: Compared to Base, $\mathcal{D} = \{\mathcal{D}_R, \mathcal{D}_A, \mathcal{D}_O\}$ was learned by using AGCN in Eq. (4.5). And MAGCN was not used in Eq. (4.4). This baseline is designed for validating the importance of the shared $\mathcal{D}$. We also compared it with another two baselines: $\mathcal{D}\#3$ +GCN where attentional gates were not used and $\mathcal{D}\#1$ +AGCN where only one dictionary $\mathcal{D}$ was used.

SGAE: When we used all the AGCN, MAGCN, and the dictionary set $\mathcal{D}$, the baseline is called SGAE. We also compare it with the one in our preliminary version **SGAE-pre** [3], which used GCN, MGCN, and only one dictionary.

SGAE-KD: When we applied L_{KD} (Eq. (4.18)) to train $\mathcal{I}$ -RNN based on the baseline SGAE, we got our integral model.

Results. The bottom of TABLE 4.1 shows the performances of the ablation baselines on MS-COCO Karpathy split. Compared with Base and our preliminary version SGAE-pre, our SGAE-KD can boost the CIDEr-D by absolute 9 and 1.8, respectively. By comparing MAGCN, $\mathcal{D}$ w/o AGCN, and $\mathcal{D}\#1$ +AGCN with Base,

TABLE 4.2: The performances of using different language corpora.

Models	B@1	B@4	M	R	C	S
Base	79.9	36.8	27.7	57.0	120.6	20.9
VG	80.4	38.1	28.3	58.4	125.1	21.7
COCO	81.0	38.8	28.8	58.8	129.6	22.4
Senti#Pos	80.1	37.0	27.9	57.2	122.8	21.2
Senti#Neg	80.2	37.2	28.0	57.4	123.1	21.2
VG+COCO	81.3	38.9	29.1	58.9	130.3	22.5

we can find that all the performances are improved, which demonstrate that the proposed MAGCN, AGCN, and $\mathcal{D}$ are all indispensable for improving the performances. In addition, by refining the graph convolution operations (MAGCN vs. MGCN and AGCN vs. GCN) and partitioning the dictionary to more fine-grained ones ($\mathcal{D}\#3+\text{AGCN}$ vs. $\mathcal{D}\#1+\text{AGCN}$), we also achieve certain improvements. These observations validate the importance of the refinements of the components in SGAE. Furthermore, by applying the Knowledge Distillation strategy to teach $\mathcal{I}$ -RNN to reason as $\mathcal{S}$ -RNN, the performance of the whole system is also improved, which suggests that the transferability of the language inductive bias is further enhanced.

Qualitative Examples. FIGURE 4.6 shows 6 examples of the generated captions using different models. We can see that compared with captions generated by Up-Down (Base) [1], the descriptions of GCN-LSTM (MAGCN) [2] usually contain more about object attributes and pairwise relationships. For captions generated by SGAE [3], they are more complex and descriptive. For example, in FIGURE 4.6 (a), the word “busy” is used to describe the heavy traffic; in (b) the scene “forest” can be deduced from “trees”; and in (c), the attribute “ripe” is inferred from the color of the banana. It can also be observed that our SGAE-KD generates more descriptive captions which contain more interesting details than SGAE, *e.g.*, the action “perch” which is more exclusive for the animal bird is generated in (d); the word “pile” more accurately describes the state of the rocks in (e); and the word “spray” instead of “stand” is used to describe the action between the hydrant and the water in (f).

FIGURE 4.6: Qualitative examples of different methods. The comparing methods are Base, the reimplemented version of Up-Down [1], MAGCN, the reimplemented version of GCN-LSTM [2], SGAE, the upgraded version of our preliminary work [3], and our SGAE-KD. 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.

FIGURE 4.7: Captions generated by using different language corpora. The id refers to the image id in MS-COCO.

4.5.2.2 Language Corpus

Comparing Methods. To test the performances of transferring the inductive bias from other language corpora, we also learned $\mathcal{D}$ from the text corpora of VG [56] and SentiCap [173]. Among them, VG has a similar language corpus as COCO while SentiCap’s corpus is very dissimilar. By learning $\mathcal{D}$ from them, we can get a more comprehensive understanding of the effects brought by the pre-trained $\mathcal{D}$. After learning $\mathcal{D}$, we inserted it in Eq. (4.16) to get $\hat{\mathcal{V}}$ for captioning. The results are demonstrated in TABLE 4.2, where **VG/COCO/VG+COCO** learn $\mathcal{D}$ by

texts of VG/COCO/VG+COCO and **Senti#Pos/Senti#Neg** use positive and negative captions from SentiCap, respectively.

Quantitative Results. By comparing Base with the models using additional text corpora in TABLE 4.2, we find that using additional texts can boost the performances even when a dissimilar corpus like SentiCap is used (Senti#Pos vs. Base or Senti#Neg vs. Base). We can also see that the more similar the language corpus is, the higher the scores can be achieved. For example, COCO is obviously better than the other single language corpus model (COCO vs. VG) and the similar corpus VG outperforms the dissimilar ones (VG vs. Senti#Pos or Senti#Neg). Such observations suggest that $\mathcal{D}$ can memorize common inductive bias from the additional unmatched language corpus or specific inductive bias from a matched language corpus for generating better captions. Also, when we compose two similar corpora VG and COCO to learn $\mathcal{D}$, a consistent improvement can be achieved compared with the model with only one corpus (COCO+VG vs COCO).

Human Evaluation. The purpose for introducing human evaluation is to avoid the performance bias towards COCO since we measure the quantitative performance based on ground-truth captions of COCO, while our dictionary $\mathcal{D}$ could come from a different corpus with a very different style from COCO. In particular, we conducted human evaluation with 50 annotators who are fluent in English and unaware of our work. We divided 50 annotators into 5 groups and asked each group to respectively compare the following pairwise methods: Base vs. COCO, Base vs. VG, Base vs. Senti, COCO vs. VG, COCO vs. Senti. We randomly sampled 100 images from the COCO test set and assigned these same images to each group with the captions generated by the corresponding two comparing methods. For Senti captions, one half of them were generated by Senti#Pos and another half were generated by Senti#Neg. For each group with 10 annotators, it was further divided into two sub-groups that every 5 annotators were given the same 50 images for comparing.

During the comparisons, each annotator was asked that which one of the two captions is more descriptive, *e.g.*, containing or inferring more interesting details (“a lush forest” vs. “a forest”), using more exclusive words (“a seagull” vs. “a bird”), using more summary phrases (“busy street” vs. “many people in a street”). We set three options for the annotators to vote: “the first caption is better”, “the second caption is better”, and “two captions are similar”. In this process, the

TABLE 4.3: Results of human evaluation, where each row compares two methods. In each column, “1st”/“2nd” lists the percentage of votes which prefer the fist/second method in that row and “Similar” denotes the votes for “two captions are similar”.

Models	1st	2nd	Similar
COCO vs. Base	69%	12%	19%
VG vs. Base	65%	14%	21%
Senti vs. Base	73%	11%	16%
COCO vs. VG	36%	29%	35%
COCO vs. Senti	34%	38%	28%

annotators did not know where these captions are from. After collecting the votes, we calculated the ratio and reported them in TABLE 4.3.

Analyses of Human Evaluation. 1) For the comparisons of COCO/VG/Senti vs. Base, we can see that the models using $\mathcal{D}$ generate more descriptive captions, getting 69%/65%/73% votes, respectively. 2) The comparison of COCO vs. VG suggests using COCO is better, although the gap is not as large as that in the case of COCO vs. Base. 3) More interestingly, the comparison of COCO vs. Senti suggests that the generated captions by using Senti are a little more descriptive than using COCO. This might be because the language inductive bias from the sentiment captions evokes the annotators’ emotions to vote more for Senti, despite the sentiment words do not return high quantitative scores in TABLE 4.2.

Qualitative Examples. FIGURE 4.7 shows 4 examples in human evaluation, where the captions are generated by four methods. Generally, compared with captions generated by Base, the captions of VG, COCO, and Senti are more descriptive. Specifically, the captions generated by different language corpora reflect the inductive bias of such corpora: COCO and VG use some factual language to describe the image while Senti prefers to convey different sentiments. For example, in FIGURE 4.7 (a), COCO and VG describe there is “a fire hydrant” and its colour, while Senti tags this “fire hydrant” with the adjective “ugly”, which reflects the negative sentiment.

TABLE 4.4: The performances of the sentence reconstruction using different scene graphs.

Models	B@1	B@4	M	R	C	S
$\hat{\mathcal{X}}$	90.3	53.8	34.3	66.5	153.2	30.6
$\mathcal{X}$	93.9	65.2	38.5	71.8	177.0	34.3
SGAE-KD	81.0	38.8	28.8	58.8	129.6	22.4

TABLE 4.5: The performances of various methods on MS-COCO Karpathy split trained by cross-entropy loss only.

Models	B@1	B@4	M	R	C	S
SCST [29]	—	30.0	25.9	53.4	99.4	—
LSTM-A [40]	73.4	32.6	25.4	54.0	100.2	18.6
StackCap [31]	76.2	35.2	26.5	—	109.1	—
Up-Down [1]	77.2	36.2	27.0	56.4	113.5	20.3
RFNet [167]	77.4	37.0	27.9	57.3	116.3	20.8
LBPF [39]	77.8	37.4	28.1	57.5	116.4	21.2
GCN-LSTM [2]	77.3	36.8	27.9	57.0	116.3	20.9
CNM [157]	77.6	37.1	27.9	57.3	116.6	20.8
Base	76.8	36.1	27.1	56.3	113.1	20.3
MAGCN	77.3	36.4	27.3	56.7	114.4	20.6
SGAE-pre [3]	77.6	36.9	27.7	57.2	116.7	20.9
SGAE	77.8	37.0	27.9	57.3	116.9	21.0
SGAE-KD	78.2	37.3	28.1	57.4	117.1	21.3

FIGURE 4.8: Two examples of unpaired image captioning. The id refers to the image id in MS-COCO.

4.5.2.3 Sentence Reconstruction

Comparing Methods. We investigated how well the sentences are reconstructed in training SGAE (Eq. (4.5)) with or without the re-encoder $R(\mathcal{X}; \mathcal{D})$, that is, we denote $\hat{\mathcal{X}}$ as the pipeline using $\mathcal{D}$ and $\mathcal{X}$ as the pipeline directly reconstructing sentences from their scene graph embeddings. Such results are given in TABLE 4.4.

TABLE 4.6: The performances of various methods on the online MS-COCO test server.

Model	B@3		B@4		M		R-L		C-D	
Metric	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
SCST [29]	47.0	75.9	35.2	64.5	27.0	35.5	56.3	70.7	114.7	116.0
LSTM-A [40]	47.6	76.5	35.6	65.2	27.0	35.4	56.4	70.5	116.0	118.0
StackCap [31]	46.8	76.0	34.9	64.6	27.0	35.6	56.2	70.6	114.8	118.3
Up-Down [1]	49.1	79.4	36.9	68.5	27.6	36.7	57.1	72.4	117.9	120.5
RFNet [167]	50.1	80.4	38.0	69.2	28.2	37.2	58.2	73.1	122.9	125.1
CAVP [177]	50.0	79.7	37.9	69.0	28.1	37.0	58.2	73.1	121.6	123.8
SGAE-pre [3]	50.1	79.6	37.8	68.7	28.1	37.0	58.2	73.1	122.7	125.5
CNM [157]	—	—	37.9	68.4	28.1	36.9	58.3	72.9	123.0	125.3
SGAE-KD	50.1	79.9	38.2	69.3	28.7	37.9	58.4	73.5	124.5	126.6

Analysis. As we can see, the performances of using direct scene graph features $\mathcal{X}$ are much better than those ($\hat{\mathcal{X}}$) imposed with $\mathcal{D}$ for re-encoding. This is reasonable since $\mathcal{D}$ regularizes the reconstruction and thus creates knowledge loss in this pure language domain. Interestingly, the gap between $\hat{\mathcal{X}}$ and SGAE-KD suggests that we should develop a more powerful image scene graph parser for improving the quality of $\mathcal{G}_{\mathcal{I}}$ in Eq. (4.4) and design a stronger re-encoder for extracting more preserved inductive bias when only low-quality visual scene graphs are available.

4.5.3 Comparisons with State-of-The-Arts

Comparing Methods. Though there are various captioning models developed in recent years, we only compare SGAE-KD with some encoder-decoder methods trained by cross-entropy loss (Eq. (4.2)) and RL-based reward (Eq. (4.3)) due to their superior performances. Specifically, we compare our methods with **SCST** [29], **StackCap** [31], **Up-Down** [1], **LSTM-A** [40], **GCN-LSTM** [2], **CAVP** [38], **RFNet** [167], **LBPF** [39], and **CNM** [157]. Among these methods, SCST and Up-Down are two strong baselines since almost all the following captioning models use the advanced self-critic reward provided by SCST and the visual features provided by Up-Down. Compared with SCST, StackCap proposes a more complex RL-based reward for learning captions with more details. LSTM-A, GCN-LSTM, CAVP, and CNM try to exploit the knowledge of visual scene graphs, where LSTM-A and GCN-LSTM respectively exploit attribute and relationship information, CAVP tries to recognize the pairwise relationships in the decoder, and CNM

learns to collocate various types of modules to generate object, attribute, and relation words for composing the caption. The results of these methods are reported in TABLE 4.1, where the models are trained by cross-entropy loss (Eq. (4.2)) first and then by RL-based reward (Eq. (4.3)) and TABLE 4.5, where the models are only trained by cross-entropy loss (Eq. (4.2)).

Analysis. From TABLE 4.1 and 4.5, we can see that our single model outperforms the other image captioners. In TABLE 4.1, when both cross-entropy loss and RL-based reward are used, our single model achieves a new state-of-the-art score among all the compared methods in terms of CIDEr-D, reaching 129.6. By transferring the inductive bias from the pure language domain to the vision-language domain, our method still has better performances even our decoder is not as sophisticated as CVAP or LBPF, even we do not use more advanced RL-based reward as StackCap, and even we do not mix up different visual features as the input as RFNet or CNM. Moreover, compared with GCN-LSTM which sets the batch size to 1,024, our smaller batch size (100) still leads to higher performances. TABLE 4.6 reports the performances of different methods test on the official server. Compared with the published captioning methods, our single model achieves the highest CIDEr-D c40 score, 126.6.

4.5.4 Unpaired Image Captioning

4.5.4.1 Comparing Methods

We designed the following ablation methods to validate the contributions of the scene graphs and the dictionary set $\mathcal{D}$ in unpaired image captioning. To distinguish the unpaired image captioners with our paired image captioners in Section 4.5.2.1, we add the capital letter “U” in the names of the following models.

UBase: When training, we only input the node embeddings of the sentence scene graph $\mathcal{G}_S$ into the auto-encoder for reconstructing, where these embeddings were the word embeddings of the one-hot node labels. When testing, we only input the node embeddings of the image scene graph $\mathcal{G}_I$ into the auto-encoder for captioning. Also, we did not use the dictionary set $\mathcal{D}$ in this baseline.

TABLE 4.7: The performances of unpaired image captioning.

Models	B@1	B@4	M	R	C	S
Pivoting [51]	46.2	5.4	13.2	—	17.7	—
UIC [52]	53.8	15.6	16.6	39.9	46.7	9.6
SME [54]	59.7	16.6	18.3	43.8	53.8	10.8
UBase	47.8	7.2	14.1	32.2	23.4	7.3
UMAGCN [53]	53.9	12.9	16.8	40.0	43.9	12.4
USGAE	60.8	17.1	19.1	43.8	55.1	12.8
SGAE-KD	81.0	38.8	28.8	58.8	129.6	22.4

UMAGCN: Compared with UBase, we exploited the MAGCN for transforming both $\mathcal{G}_S$ and $\mathcal{G}_T$ to embeddings, which were input into the auto-encoder for captioning. Also, we did not use $\mathcal{D}$ in this baseline. This is an upgraded version of [53] without adversarial learning objective.

USGAE: We followed Eq. (4.12) to generate the captions. Compared with UMAGCN, the graph embeddings were re-encoded by R and then the re-mapped embeddings were used for captioning.

We also compare our USGAE with the following state-of-the-arts: **Pivoting** [51], **UIC** [52], and **SME** [54]. For fair comparisons, we only compare our USGAE with the models which are not trained by adversarial learning objectives since different methods have different adversarial learning strategies and it is beyond the scope of our research. Besides these models, we also list the results of SGAE-KD (Section 4.5.2.1) to show the gap between the unpaired case and the paired case.

4.5.4.2 Results and Analysis

TABLE 4.7 reports the performances of various methods of unpaired image captioning. We can observe that our USGAE achieves new state-of-the-art scores among all the unpaired image captioning methods. More importantly, our USGAE can complement other models by using our auto-encoder as initialization followed by fine-tuning with specific training strategies. By comparing UMAGCN with UBase and USGAE with UMAGCN, we can conclude that both the scene graph representations and the dictionary set play indispensable roles in transferring inductive bias from the pure language domain to the vision-language domain. Specifically, scene graph representations contain more comprehensive knowledge

than single-node embeddings, which build a stronger bridge between two domains. The dictionary set $\mathcal{D}$ filters the noisy $\mathcal{G}_{\mathcal{I}}$ by the re-encoder, thus narrowing the gap between two domains. Furthermore, compared with the paired image captioning SGAE-KD, the upside potential for unpaired image captioning is huge.

FIGURE 4.8 shows two examples about unpaired image captioning. We can observe that UBase only lists a few detected objects as the caption which does not contain any interesting details. In contrast, our USGAE can generate more descriptive captions, even comparable to SGAE-KD which is trained by paired supervisions.

Chapter 5

Hierarchical Scene Graph Encoder-Decoder for Image Paragraph Captioning¹

5.1 Introduction

The traditional image captioning task depicts an image by a single sentence, which is however, too short to detail the rich visual contents [20, 24, 32, 178]. So, a paragraph with more descriptive capacity is a better way to detail the distinctiveness of an image [43]. As shown in FIGURE 5.1 (a), the paragraph details not only more objects (“Sand”, “Tree”, and “Rock”) but also more attributes of these objects (“Golden Gog” and “Brown Rocks”) than a single sentence.

A paragraph is beyond a bag of sentences; they should be *coherent*, *i.e.*, neighboring sentences share some concepts, and *distinctive*, *i.e.*, they detail different aspects of the image, *e.g.*, FIGURE 5.1 (b) shows the coherent and distinctive topics of the paragraph in (a). Therefore, these new requirements of the paragraph rather than sentence invalidate the flat RNN based captioning models, which lack explicit topic guidance. To illustrate, if we treat the paragraph as a long sentence and directly apply the flat RNN [43, 48], the generated paragraph is usually full

¹This chapter is published as Xu Yang, Chongyang Gao, Hanwang Zhang, Jianfei Cai. “Hierarchical Scene Graph Encoder-Decoder for Image Paragraph Captioning.” Association for Computing Machinery (ACM)’s annual conference on multimedia (**ACMMM, Poster**). Seattle, United States, 2020.

FIGURE 5.1: Illustrations of our motivation. (a) Comparisons between single sentence captioning and the informative paragraph. (b) Topics and script of the paragraph in (a). (c) A paragraph generated by flat RNN. (d) The paragraph generation process of HRNN without hierarchical constraint where the top and bottom parts denote the higher and lower RNNs, respectively. (e) The paragraph generation process of our HSGED, where the grey color means this part will not be attended.

of redundant sentences as in FIGURE 5.1 (c), where all the sentences repeat the same topic “Human-Near-Dog”. To achieve the topic guidance, researchers propose Hierarchical RNN (HRNN) [179–181] that contains multiple RNNs, where the higher-level RNNs abstract the topic as the guidance for the lower-level RNNs to generate the corresponding sentence. In this way, a more informative paragraph can be generated, as in FIGURE 5.1 (d), the generated sentences cover more topics than those generated by the flat RNN in FIGURE 5.1 (c).

However, the multi-level RNNs in HRNN are built without any hierarchical constraints [43–46, 49], which results in two problems. 1) The topics are not coherent and distinctive since they are formed by randomly sampling some object sub-sets without any global constraint. 2) The generated sentences are not corresponding to the given topics since the input of the lower-level RNNs is the whole image instead of the sub-regions constrained by the topic. So, such HRNN tends to degenerate to flat RNN and generates redundant summary sentences, *e.g.*, in FIGURE 5.1 (d), both the first and third sentences describe “Man” and “Dog”, even though the third topic is about “Dog” and “Sand”.

The crux of these problems is to find a “script” as in FIGURE 5.1 (b), which connects the topics to provide hierarchical constraints for the HRNN. Recent studies show that Scene Graphs [61], which connects local objects as a global graph in terms of the object relations, can serve as the script to provide hierarchical knowledge for solving complex tasks, *e.g.*, image retrieval [55], image generation [89], image captioning [3], and visual reasoning [90]. Since the neighboring sub-graphs of a scene graph share some concepts and each sub-graph is distinctive, if we form the topic flows from a scene graph and align each topic to a sub-graph, the generated paragraph will naturally be coherent and distinctive.

Motivated by the above conjecture, we propose Hierarchical Scene Graph Encoder-Decoder (HSGED) to exploit the scene graph as the topic script and transfer its hierarchical topological knowledge into the text domain for better paragraphs. Specifically, the high-level RNNs can follow scene graphs to generate the topics, each of which is represented by a local compact sub-graph. Compared with the HRNNs which generate sub-set topics without any global constraints, the topic flows in our HSGED are naturally coherent and distinctive, *e.g.*, the topic flows in Fig 5.1 (e) are “Human-With-Dog”, “Dog-On-Sand”, and “Hill-Near-Sand”, showing that the neighborhood topics are not only closely related but also distinctive. The sub-graph topic retains a sentence about a corresponding region, *e.g.*, the second topic in FIGURE 5.1 (e) is about “Human”, “Dog”, and “Sand”, which is more likely to be treated as a compact part by humans instead of “Human” and “Tree” as in (d).

Since the paragraph dataset is limited in size and for fair comparisons, we follow the previous studies [43–45] to construct the decoder of our HSGED by two RNNs which are sentence scene graph RNN (SSG-RNN) and word scene graph RNN (WSG-RNN). Specifically, the scene graph is transformed into a set of sub-graph level embeddings by a graph neural network [88]. When generating a new sentence, SSG-RNN adaptively attends to a few sub-graph embeddings based on the context knowledge to form the new topic (see Section 5.2.2.2). Also, we design an irredundant attention strategy to encourage the new topics to be formed from the undescribed sub-graphs (see Section 5.2.2.3), *e.g.*, as shown in Fig 5.1 (e) where the new topic comes from the colorful undescribed regions. Given the generated sub-graph topic, WSG-RNN composes a new sentence by focusing on the region constrained by the selected compact sub-graphs, which is achieved by a

FIGURE 5.2: The illustrations of three sub-graphs around different objects. The dash lines connect the shared nodes which facilitate the system to generate coherent sub-graph topics.

novel inheriting attention strategy (see Section 5.2.2.5). Furthermore, we design an efficient sentence-level loss to encourage the topics further to follow the scene graph for a human-like order. Extensive experiments on Stanford image paragraph dataset [43] show that our HSGED generates more coherent and distinctive paragraphs. In particular, we achieve a new state-of-the-art 36.02 CIDEr-D score [166] and have absolute 5.24 points boost than a strong baseline (see Section 5.3).

5.2 Hierarchical Scene Graph Encoder-Decoder

Our Hierarchical Scene Graph Encoder-Decoder (HSGED) belongs to the encoder-decoder framework [20, 24, 43]. The hierarchical encoder transforms the scene graph into node-level and sub-graph level embeddings (see Section 5.2.1.1), which are respectively input into Sentence Scene Graph RNN (SSG-RNN) to abstract the sub-graph topic and Word Scene Graph RNN (WSG-RNN) to generate the corresponding sentence (see Section 5.2.2). Specifically, we design two attention strategies that can remove redundancies for attention in SSG-RNN for more distinctive sub-graph topics (see Section 5.2.2.3) and enhancing attention in WSG-RNN for more grounded sentences based on the sub-graph topics (see Section 5.2.2.5).

5.2.1 Hierarchical Scene Graph Encoder

5.2.1.1 Scene Graphs

The scene graph is constructed by using directed edges to connect three different nodes: object node o_i , denoting the i -th object; attribute node a_{il} , denoting the l -th attribute of o_i ; and relation node r_{ij} , denoting the pairwise relation between o_i and o_j . We assign directed edges from object o_i to all of its attributes a_{il} and from object o_i to relation r_{ij} and from relation r_{ij} to object o_j to form the scene graph. FIGURE 5.2 demonstrates one scene graph, which contains four object nodes, three relation nodes, and two attribute nodes. In this way, the scene graph contains rich semantic knowledge brought from the semantic labels, *e.g.*, “Brown” and “Near”, and topological knowledge brought from the connectivity of the graph, *e.g.*, “Dog $\rightarrow$ Near $\rightarrow$ Bike” and “Dog $\rightarrow$ On $\rightarrow$ Road”.

Node Level Embeddings: $\mathbf{X}$ For three types of nodes, we use different representations as their node-level embeddings: the linear transformation of visual feature $\mathbf{x}_o$ (RoI feature from Faster RCNN through an FC layer) for object node and learnable label embeddings of attribute and relation labels $\mathbf{x}_a$, $\mathbf{x}_r$ for attribute and relation nodes, respectively. We use visual features as $\mathbf{x}_o$ since they contain more visual clues and they empirically achieve better performance than label embeddings. All these representations are grouped into one **node-level embedding set $\mathbf{X}$** :

$$\mathbf{X} = \{\mathbf{x}_a, \mathbf{x}_r, \mathbf{x}_o\}, \quad (5.1)$$

where each embedding corresponds to one node, *e.g.*, the scene graph in FIGURE 5.2 contains 9 embeddings (4, 2, 3 for object, attribute and relation nodes). This node-level embedding set will be input into WSG-RNN for generating sentences (see Section 5.2.2).

Sub-graph Level Embeddings: $\mathbf{U}$ Since the sentences of a high-quality paragraph should describe distinctive aspects of an object, *e.g.*, the relationships with other objects, we adopt the sub-graph level embedding which could facilitate our system to achieve such goal. Specifically, we define the sub-graph around the object o_i as a graph connecting the following nodes: o_i , which is this object itself; a_{il} , which are this object’s attributes; r_{ij} and r_{ki} , which are the potential relationships between this object and the other objects; and o_j and o_k , which are the objects

FIGURE 5.3: The sketches of three sub-graph level embeddings.

that have potential relationships with o_i . For example, a sub-graph around the object “Dog” in FIGURE 5.2 connects the following nodes: “Dog”, “Brown”, “On”, “Road”, “Near”, and “Bike”. We incorporate such sub-graphs into embeddings by an efficient Graph Neural Network (GNN) [87, 88] and input them into SSG-RNN for abstracting sub-graph topics (see Section 5.2.2.2).

We denote the i -th object’s sub-graph embedding as $\mathbf{u}_i$ and group all the objects’ sub-graph embeddings to the **sub-graph level embedding set** $\mathbf{U}$, *e.g.*, for the scene graph in FIGURE 5.2, $\mathbf{U}$ contains 4 sub-graph embeddings for 4 objects. Each $\mathbf{u}_i$ concatenates three distinctive embeddings: the object, attribute and relation embeddings $\mathbf{u}_{o_i}$, $\mathbf{u}_{a_i}$ and $\mathbf{u}_{r_i}$, which are respectively computed as:

$$\mathbf{u}_{o_i} = \mathbf{x}_{o_i}, \quad (5.2)$$

$$\mathbf{u}_{a_i} = \sum_{a_{il} \in \text{Attr}(o_i)} f_A(\mathbf{x}_{o_i}, \mathbf{x}_{a_{il}}), \quad (5.3)$$

$$\mathbf{u}_{r_i} = \sum_{r_{ij} \in \text{Sbj}(o_i)} f_S(\mathbf{x}_{o_i}, \mathbf{x}_{r_{ij}}, \mathbf{x}_{o_j}) + \sum_{r_{ki} \in \text{Obj}(o_k)} f_O(\mathbf{x}_{o_k}, \mathbf{x}_{r_{ki}}, \mathbf{x}_{o_i}), \quad (5.4)$$

$$\mathbf{u}_i = \text{Concat}(\mathbf{u}_{o_i}, \mathbf{u}_{a_i}, \mathbf{u}_{r_i}), \quad (5.5)$$

where $\mathbf{x}_*$ are the node-level embeddings in Eq. (5.1); $\text{Sbj}(o_i)$ (or $\text{Obj}(o_i)$) in Eq. (5.4) means the set of relations where o_i acts as the subject (or object) node,

e.g., $sbj(o_{Dog}) = \{r_{Near}, r_{On}\}$; Concat in Eq. (5.5) means the concatenation operation, and f_A , f_S and f_O are the independent sub-networks with the same structure: FC-ReLU-FC-ReLU. We use such two-layer perceptrons followed by a summation pooling in GNN to achieve stronger representation power [88]. FIGURE 5.3 sketches the three operations for the sub-graph level embedding, *e.g.*, the relation embedding incorporates the knowledge of “Dog”, “Road”, “On”, “Bike”, and “Near”.

Importantly, sub-graph level embeddings have two significant advantages. 1) It contains rich semantic knowledge brought from the attribute and relation labels. 2) It preserves useful topological knowledge to determine whether two objects are closely related. For example, as in FIGURE 5.2, the sub-graphs around “Dog” and “Bike” share the “Dog”, “Bike”, and “Near”, and then the sub-graph embedding $\mathbf{u}_{Dog}$ is more closely related to $\mathbf{u}_{Bike}$ instead of related to $\mathbf{u}_{Wall}$ due to the relation embedding (Eq. (5.4)). Such relatedness encourages SSG-RNN to generate *coherent* sub-graph topics: after describing “Dog”, “Bike” is more likely to be described.

5.2.2 Hierarchical Decoders

5.2.2.1 Top-Down Attention Network

We use the Top-Down attention network [1] as the prototype to design our SSG-RNN and WSG-RNN. Here we first briefly revisit this architecture and then detail how to revise it to get our SSG-RNN (see Section 5.2.2.2) and WSG-RNN (see Section 5.2.2.4).

Basically, a top-down attention network contains two LSTM layers and one attention sub-network, as shown in FIGURE 5.4 (a). Given the input vector $\mathbf{z}_t$ at time step t , it can be formalized as:

$$\begin{aligned}
 \text{Context vector: } \mathbf{h}_1^t &= \text{LSTM}_1(\mathbf{z}_t; \mathbf{h}_1^{t-1}), \\
 \text{Attention: } \hat{\mathbf{v}}^t &= \text{ATT}(\mathbf{V}, \mathbf{h}_1^t), \\
 \text{Output: } \mathbf{h}_2^t &= \text{LSTM}_2(\text{Concat}(\mathbf{h}_1^t, \hat{\mathbf{v}}^t); \mathbf{h}_2^{t-1}),
 \end{aligned} \tag{5.6}$$

where the attention sub-network ATT is:

$$\begin{aligned}
&\textbf{Input: } \mathbf{V}, \mathbf{h}_1^t, \\
&\textbf{Attention weight: } \begin{aligned} a_i^t &= \omega_a^T \tanh(\mathbf{W}_v \mathbf{v}_i + \mathbf{W}_h \mathbf{h}_1^t), \\ \boldsymbol{\alpha}^t &= \text{softmax}(\mathbf{a}^t), \end{aligned} \\
&\textbf{Output: } \hat{\mathbf{v}}^t = \mathbf{V} \boldsymbol{\alpha}^t,
\end{aligned} \tag{5.7}$$

where $\mathbf{W}_v$, $\mathbf{W}_h$ are trainable matrices and ω_a is a trainable vector. In the following sections about SSG-RNN and WSG-RNN, we specify different values for $\mathbf{z}_t$ and $\mathbf{V}$. Then, we revise the attention sub-network with different strategies to detail how SSG-RNN abstracts sub-graph topics and how WSG-RNN generates the corresponding sentences.

5.2.2.2 SSG-RNN

To avoid confusion, we use n as the sentence index and t as the word index. SSG-RNN is used to generate a topic vector $\mathbf{c}_n$ which covers the content of the n -th sentence. The topic vector has two utilities in the whole system. 1) It passes interdependent knowledge between neighbor sentences for more coherent paragraphs. 2) It constrains the content of the sentence to be generated. To achieve the first goal, we input the last sentence topic vector $\mathbf{c}_{n-1}$ into SSG-RNN to pass interdependent knowledge and to achieve the second goal, we input the generated new topic vector $\mathbf{c}_n$ into WSG-RNN.

Specifically, in SSG-RNN, the input vector $\mathbf{z}_t$ in Eq. (5.6) is:

$$\mathbf{z}_t = \text{Concat}(\mathbf{c}_{n-1}, \mathbf{W}_\Sigma w_{t-1}, \bar{\mathbf{u}}, \mathbf{h}_2^{t-1}), \tag{5.8}$$

where $\mathbf{W}_\Sigma$ is a trainable word embedding matrix; w_{t-1} is the $t - 1$ -th word; $\bar{\mathbf{u}}$ is the mean pooling of sub-graph level embedding set $\mathbf{U}$ (Eq. (5.5)); and $\mathbf{h}_2^{t-1}$ is the output of the second LSTM layer in SSG-RNN at time step $t - 1$.

In SSG-RNN, we replace the input feature set $\mathbf{V}$ of the attention sub-network (Eq. (5.7)) by the sub-graph level embedding set $\mathbf{U}$, where each embedding covers the sub-graph of an object. For attention computation, SSG-RNN adaptively focuses on a few sub-graphs and then produces a sub-graph level attended embedding $\hat{\mathbf{u}}^t$, which will be used to compute our sub-graph topic vector by the second LSTM

FIGURE 5.4: In HSGED, SSG-RNN and WSG-RNN are both built based upon the top-down attention network with different attention mechanisms and inputs: sub-graph level and node level embedding sets $\mathbf{U}$, $\mathbf{X}$, respectively.

in Eq. (5.6). Once the input word w_t is the full stop symbol “.”, we compute $\mathbf{c}_n$ as:

$$\mathbf{c}_n = \text{ReLU}(\text{FC}(\mathbf{h}_2^t)). \quad (5.9)$$

For adaptive attending on the sub-graphs, the generated topic vector captures distinctive semantic knowledge of the attended sub-graphs and also preserves topological knowledge for facilitating coherence between neighboring sentences. The coherence is achieved due to the relation embedding (Eq. (5.4)), which incorporates an object with its related objects into one embedding, thus facilitating the system to choose the neighboring sub-graph according to the last attended sub-graphs. For example, in FIGURE 5.1(e), the second sub-graph is about “Human”, “Dog” and “Sand”. SSG-RNN exploits the node “Sand” as the clue to generate the next sub-graph topic about “Sand”, “Tree”, and “Hill”.

5.2.2.3 Irredundant Attention

To further encourage each topic vector to focus on rarely described sub-graphs, we substitute the attention sub-network in Eq. (5.7) with the following irredundant attention strategy [182]. When SSG-RNN generates the n -th topic vector $\mathbf{c}_n$, it does not directly input the computed attention weights $\mathbf{a}$ into the softmax layer (Eq. (5.7)) for computing attended vectors. Instead, it distracts the current attention from the previous frequently attended sub-graph level embeddings to get the irredundant attention weights β :

$$b_i^n = \begin{cases} \exp(a_i^{\tau_{n-1}}) & \text{if } n = 1 \\ \frac{\exp(a_i^{\tau_{n-1}})}{\sum_{m=1}^{n-1} \exp(a_i^{\tau_m})} & \text{otherwise} \end{cases} \quad (5.10)$$

$$\beta^n = \text{softmax}(\mathbf{b}^n), \quad (5.11)$$

where τ_m is the time step of the full stop symbol of sentence s_m . Then we compute the irredundant attention vector $\hat{\mathbf{u}}^n$:

$$\hat{\mathbf{u}}^n = \mathbf{U}\beta^n, \quad (5.12)$$

which will be used to generate the distinctive topic vector $\mathbf{c}_n$ as in Section 5.2.2.2.

If the i -th sub-graph level embedding $\mathbf{u}_i$ is frequently attended during generating previous $n - 1$ topic vectors, then $\sum_{m=1}^{n-1} \exp(a_i^{\tau_{m-1}})$ will be large and b_i^n will be small. So, the current topic vector will be less likely to focus on the i -th sub-graph, and thus the generated sentences will be less repetitive.

It is noteworthy that this irredundant attention will not limit our model for attending the salient objects for multiple times as needed, *e.g.*, as in FIGURE 5.1 (a), the concept ‘‘Human’’ is described twice in the first two sentences of the ground-truth paragraph. Our model can achieve multi-visit of the same salient object o_i since its knowledge will not only be incorporated into $\mathbf{u}_{r_i}$, but also into $\mathbf{u}_{r_k}$ if o_i and o_k own potential relations (Eq. (5.4)). As a result, the salient object o_i will also be mentioned when the sub-graph of o_k is chosen as the topic, *e.g.*, as in FIGURE 5.1 (e), the concept ‘‘Human’’ is described in the first two sentences of our generated paragraph.

5.2.2.4 WSG-RNN

Given a topic vector $\mathbf{c}_n$, WSG-RNN works to complete the corresponding sentence $\mathbf{s}_n$. Here the input vector $\mathbf{z}_t$ in Eq. (5.6) is:

$$\mathbf{z}_t = \text{Concat}(\mathbf{c}_n, \mathbf{W}_\Sigma w_{t-1}, \bar{\mathbf{x}}, \mathbf{h}_2^{t-1}), \quad (5.13)$$

where $\mathbf{c}_n$ is the topic vector of the n -th sentence to regularize the word generation; $\bar{\mathbf{x}}$ is the mean of node-level embeddings (Eq. (5.1)); and $\mathbf{h}_2^{t-1}$ is the hidden state of the second LSTM layer in WSG-RNN.

In WSG-RNN, the feature set $\mathbf{V}$ in Eq. (5.7) is set to the node-level embedding set $\mathbf{X}$. In this way, WSG-RNN puts attention to the nodes and generates the words based on them. After computing the attended node-level embedding $\hat{\mathbf{x}}$ by the inheriting attention network (Eq. (5.16)), the second LSTM layer outputs its hidden state $\mathbf{h}_2^t$ in Eq. (5.6) to predict the word distribution at time step t :

$$P(w_t | w_{1:t-1}) = \text{softmax}(\text{FC}(\mathbf{h}_2^t)). \quad (5.14)$$

5.2.2.5 Inheriting Attention

To generate the sentence more closely corresponding to the selected sub-graphs, inheriting attention is applied in WSG-RNN to constrain the attention of WSG-RNN more on the nodes in the selected sub-graphs. To achieve this, the node level attention weights γ are first inherited from the sub-graph level attention weights β (Eq. (5.12)) as follows:

$$\begin{aligned} \text{Attribute Node: } \gamma_{a_{il}} &= \beta_i, \\ \text{Object Node: } \gamma_{o_i} &= \text{Mean}_{j \in \text{Cover}(o_i)}(\beta_j), \\ \text{Rela Node: } \gamma_{r_{ij}} &= \text{Mean}_{j \in \text{Cover}(r_{ij})}(\beta_j), \end{aligned} \quad (5.15)$$

where the attribute node attention $\gamma_{a_{il}}$ is directly inherited from the attention weight β_i of the sub-graph around o_i . Since the object and relation nodes are covered by many sub-graphs, we average the attention weights of these sub-graphs as the inherited attention weights to both object and relation nodes, where $j \in \text{Cover}(o_i)$ in Eq. (5.15) means that the object o_i is covered by the sub-graph which

is around the object o_j . For example, as in FIGURE 5.2, “Bike” is covered by the sub-graphs of “Dog”, “Bike”, and “Wall”, and then we average the attention weights of these three sub-graphs as the inherited attention weight of the node o_{Bike} .

After inheriting γ from β , we can compute the attended node level embedding $\hat{x}$ as:

$$\hat{x} = X \text{Softmax}(\gamma .* \alpha), \quad (5.16)$$

where $.*$ denotes element-wise production and α is computed from the WSG-RNN attention network in Eq. (5.7).

If a sub-graph is not selected by SSG-RNN for generating the current topic vector, the attention weight of the corresponding sub-graph level embedding is small. By inheriting attention, the attention weights of the nodes in this sub-graph will also be small, so these nodes will be less likely to be selected to complete the current sentence. In this way, compared with HRNN methods without any hierarchical constrains, our WSG-RNN generates more grounded sentences.

5.2.3 Training Objectives

Given a ground-truth paragraph $\mathcal{P}^* = \{w_{1:T}^*\}$, we can end-to-end train our HSGED by maximizing the likelihood of the ground-truth paragraph $\mathcal{P}^*$ as in Regions-Hierarchical [43]. For convenience, we denote the predicted word distribution as $P(w_t)$ (Eq. (5.14)). Maximizing the likelihood equals to minimizing the cross-entropy loss:

$$L_{word} = - \sum_{t=1}^T \log P(w_t^*). \quad (5.17)$$

This is a **word-level loss** since it directly encourages the generated words to be the same as the ground-truth words.

Though this word-level loss is simple and efficient, it causes the exposure bias problem [183] which damages the performance due to the mismatch between training and test. To alleviate this mismatch problem, a reinforcement learning (RL) based reward [29, 48] can be used to train the paragraph generator:

$$L_{para} = -\mathbb{E}_{w_t^s \sim P(w)}[r(\mathcal{P}^s; \mathcal{P}^*)], \quad (5.18)$$

where r is CIDEr-D [166] metric between the paragraph $\mathcal{P}^s = \{w_{1:T}^s\}$ sampled from Eq. (5.14) and the ground-truth paragraph $\mathcal{P}^*$. This is a **paragraph-level loss** since it encourages the whole generated paragraph to be similar to the ground-truth paragraph.

Though this paragraph-level loss improves the quality of the generated paragraphs, there are still two shortcomings. First, this paragraph-level loss neglects the sequence of the sentences, while a coherent paragraph requires the generated sentences to be listed in proper sequences. Second, the paragraph-level loss does not make sufficient use of the training paragraphs since it only computes a reward for the whole paragraph.

Therefore, we propose a **sentence-level loss**, which is the sum of CIDEr-D scores between each sampled sentence $\mathbf{s}_n^s$ and the ground-truth sentence $\mathbf{s}_n^*$:

$$L_{sen} = -\mathbb{E}_{w_t^s \sim P(w)} \left[\sum_{n=1}^N r(\mathbf{s}_n^s; \mathbf{s}_n^*) \right]. \quad (5.19)$$

By enforcing each sampled $\mathbf{s}_n^s$ to be similar to the ground-truth $\mathbf{s}_n^*$, the narrative logic is encouraged to be the same as the human-like sequence, which is also a kind of hierarchical constraint. Also, the training paragraphs are more sufficiently utilized because each sentence provides a direct supervision.

In the experiment, to utilize the advantages of the word-level, paragraph-level and sentence-level losses, we combine them as:

$$L_{comb} = w_{word}L_{word} + w_{sen}L_{sen} + w_{para}L_{para}. \quad (5.20)$$

where w_{word} , w_{sen} , w_{para} are the weights of L_{word} , L_{sen} and L_{para} , respectively.

5.3 Experiments

5.3.1 Datasets, Settings, and Metrics

5.3.1.1 Datasets

Stanford Image Paragraph Dataset [43] is a mainstream large-scale dataset for image paragraph generation. This dataset contains 19,551 images and each image is paired with one paragraph. The whole dataset contains 14,575/2,487/2,489 image-paragraph pairs for training/validation/test, respectively. On average, each paragraph contains 5.7 sentences and 11.9 words.

Visual Genome [56] (VG). We use the annotations of this dataset to train our scene graph generator, which includes objects’ categories, attributes, and pairwise relations. Specifically, by removing the objects, attributes, and relations that appear less than 2,000 times in the training set, we use the remaining 305 objects, 103 attributes, and 64 relations to train our object detector, attribute classifier, and relation classifier. It is noteworthy that for fair testing, we filter out the images of VG which also exist in the test set of Stanford image paragraph dataset.

5.3.1.2 Settings

The dimensions of $\mathbf{x}_o/\mathbf{x}_a/\mathbf{x}_r/\mathbf{u}_o/\mathbf{u}_a/\mathbf{u}_r$ in Section 5.2.1.1 and 5.2.1.1 are all set to 1,000. The dimensions of the hidden states ($\mathbf{h}_*$) of all the LSTM layers are set to 1,000. The sizes of $\mathbf{W}_v/\mathbf{W}_h$ (Eq. (5.7)) are set to $3,000 \times 512/1,000 \times 512$ in SSG-RNN and $1,000 \times 512/1,000 \times 512$ in WSG-RNN. The size of $\mathbf{W}_\Sigma$ in Eq. (5.8) is $1,000 \times 4,962$. To parse the scene graph, we use Faster-RCNN as the object detector [6] and MOTIFS as the relation classifier [58]. Our attribute classifier is an FC-ReLU-FC-Sigmoid network aiming to multi-classify the attributes.

When training the whole pipeline, we use the word-level loss (Eq. (5.17)) in the first 30 epochs and then use the combination loss (Eq. (5.20)) where w_{sen} is 0.1 and w_{para} is 1 in the next 70 epochs. When the word-level/combination losses are used, the learning rates are initialized as $3e^{-4}/2e^{-5}$ and decayed by 0.85 for every 5 epochs. We set the batch size to 100 and use Adam optimizer [155]. At the

FIGURE 5.5: Two qualitative examples. The bottom left and bottom right parts show the sub-graph topics and the generated paragraph, respectively. The colors highlight the alignments between the sentences and the sub-graph topics.

inference stage, we use both beam search and trigram repetition penalty [48] to sample paragraphs.

5.3.1.3 Metrics

Following the previous methods [43, 44, 48], we use three standard metrics: CIDEr-D [166], BLEU [174], and METEOR[175], to measure the similarities between the generated paragraphs and the ground-truth paragraphs. Because the ground-truth paragraphs are created by humans which are naturally coherent, the generated paragraphs are likely to be more coherent if the similarity scores are higher. Besides, we measure the distinctiveness from two aspects: diversity and fine-grain degree. Diversity is measured by a metric [184] which is derived from the kernelized latent semantic analysis of CIDEr-D. Fine-grain degree is measured by the statistics of the part-of-speech. Moreover, we conduct the human evaluation on coherence, diversity and fine-grain degree of the paragraphs generated by different methods.

5.3.2 Ablative Studies

We carry extensive ablative studies by gradually constructing our model from the original Up-Down model [1, 48] so as to evaluate the effectiveness of each individual component. In particular, we construct the following baselines.

TABLE 5.1: The performances of various baselines on Stanford image paragraph dataset. The metrics: B@N, M, and C, denote BLEU@N, METEOR, and CIDEr-D, respectively.

Models	B@1	B@2	B@3	B@4	M	C
Base	43.42	27.56	17.41	10.39	17.22	30.78
Base(SLL)	43.63	27.74	17.58	10.44	17.42	31.90
GNN	43.77	27.83	17.66	10.65	17.51	32.48
GNN(SLL)	44.02	27.98	17.87	10.90	17.78	33.57
HRNN	43.54	27.65	17.53	10.42	17.44	31.36
HSGED-IRA-INA	44.17	28.36	17.85	11.04	17.89	33.67
HSGED-INA	44.21	28.42	18.04	11.09	18.06	34.42
HSGED	44.33	28.47	18.09	11.10	18.11	35.13
HSGED(SP, SLL)	44.20	28.39	18.02	11.07	18.02	34.15
HSGED(SLL)	44.51	28.69	18.28	11.26	18.33	36.02

5.3.2.1 Comparing Methods

Base: We treat each paragraph as one long sentence and directly use the Up-Down model (Eq. (5.6)) as the Flat RNN to generate them. This baseline is the benchmark for the other ablative baselines.

GNN: We use Graph Neural Network (GNN) to compute the sub-graph level embedding set $\mathbf{U}$ (Eq. (5.5)) and input it into the decoder of a Flat RNN as Base for generating paragraphs. **HRNN:** We use the hierarchical RNNs to generate paragraphs, while we only input node level embedding set $\mathbf{X}$ (Eq. (5.1)) into the two RNNs without irredundant and inheriting attentions. **HSGED-IRA-INA:** We input the node/sub-graph level embedding sets $\mathbf{X}/\mathbf{U}$ into SSG/WSG-RNNs, respectively, while we do not deploy irredundant and inheriting attentions. **HSGED-INA:** Compared with HSGED-IRA-INA, we apply irredundant attention and still do not use inheriting attention. **HSGED:** We use the integral architecture sketched in FIGURE 5.4 (b). **SP:** We replace the original semantic scene graph with the spatial scene graph, where each box is connected by its nearest 5 boxes with their relative spatial positions as labels. **SLL:** We use the sentence-level loss defined in Eq. (5.19) to train some of the baselines.

5.3.2.2 Results on Similarity

TABLE 5.1 shows the performances of the different ablative baselines. Here we use trigram repetition penalty [48] to sample paragraphs. Compared with Base,

TABLE 5.2: The diversity scores (the larger, the more diverse) of various baselines. GT denotes ground-truth.

	Base	HRNN	HSGED-IRA-INA	HSGED-INA	HSGED	HSGED(SLL)	GT
Diversity	0.786	0.791	0.810	0.823	0.836	0.840	0.847

our integral model, HSGED(SLL), boosts the CIDEr-D by 5.24. By comparing Base(SLL) and GNN with Base, we observe that the performances are boosted, which confirms the utility of sentence-level loss and sub-graph level embedding. More importantly, by alternately comparing HRNN, HSGED-IRA-INA, HSGED-INA, and HSGED, we also observe the uninterrupted improvements, which substantially validate the superiorities of our proposed SSG-RNN and WSG-RNN, especially the irredundant and inheriting attentions. Another interesting observation is that the improvement of HRNN compared with Base is marginal (0.58 CIDEr-D), while the improvement of our HSGED-IRA-INA is more obvious (2.89 CIDEr-D), which proves that only applying a naive HRNN structure without using any hierarchical constraint is not enough for capturing the hierarchical knowledge in both the image and the paragraph and makes HRNN degrade to the flat RNN. FIGURE 5.5 shows two examples of the paragraphs generated by HSGED(SLL) and HRNN, where HSGED’s paragraphs are clearly more coherent and distinctive.

5.3.2.3 Results on Diversity

TABLE 5.2 shows the diversity scores [184] of the paragraphs generated by different models. We also compute the diversity score of the ground truth paragraphs of the test set, which is denoted as GT. Comparing the results of HSGED-IRA-INA, HSGED-INA, HSGED, HSGED(SLL) in TABLE 5.2, we can see that the diversity scores are improved, which demonstrates the effectiveness of the irredundant attention, inheriting attention, and sentence-level loss in improving the paragraph diversity.

5.3.2.4 Results on Fine-grain Degree

To test the fine-grain degree, we calculate the ratios of the non-repetitive noun, verb, preposition and adjective among paragraphs generated by Base, HRNN, HSGED-IRA-INA and HSGED(SLL), and draw two corresponding radar charts

TABLE 5.3: The object and relation SPICE score of various baselines on Stanford image paragraph dataset.

	Base	HRNN	HSGED-IRA-INA	HSGED
Object	0.32	0.34	0.38	0.41
Relation	0.15	0.15	0.21	0.23

FIGURE 5.6: The radar charts illustrate the ratios of four part of speech (PoS) in paragraphs generated by four methods.

in FIGURE 5.6, where (a) and (b) use the length of the paragraph and the length of the paragraph without repetition as the denominator, respectively. Furthermore, we calculate the Object and Relation SPICE [84] scores presented in TABLE 5.3. Obviously, paragraphs generated by HSGED(SLL) have the greatest ratios of noun, verb, preposition and adjective in both radar charts, and HSGED achieves the highest Object and Relation SPICE score: 0.41 and 0.23. From these results, we find that HSGED(SLL) generates the most abundant semantic information and HSGED generates richer objects and relations than the baseline HRNN.

5.3.2.5 Human Evaluation.

To further demonstrate that our model could generate more coherent, fine-grained, and diverse paragraphs, we conduct a human evaluation with 20 workers. We sample 50 images from the test set and assign the paragraphs generated by HRNN, HSGED-IRA-INA and HSGED(SLL) to workers. Then the workers are asked to choose the most coherent, fine-grained, and diverse paragraphs. The results are shown in FIGURE 5.7. Compared with HRNN and HSGED-IRA-INA, the paragraphs generated by HSGED(SLL) are considered as the most coherent, fine-grained and diverse captions, obtaining respectively 41.6%, 45.1% and 55% votes.

FIGURE 5.7: The pie charts show the results of the human evaluation on the quality of the generated paragraphs.

5.3.3 Comparisons with State-of-The-Arts

5.3.3.1 Comparing Methods.

We compare our HSGED(SLL) with several state-of-the-art models: **Regions-Hierarchical** [43], **RTT-GAN** [45], **DCPG** [44], **HCAVP** [46], **DHPV** [49], **CAE-LSTM** [47], **TDPG** [48] and **CRL** [50]. Among these methods, RTT-GAN, DCPG, Regions-Hierarchical, DHPV, Hierarchical CAVP and CAE-LSTM use HRNNs with different technique details. RTT-GAN and DCPG-VAE use additional generative models, *e.g.*, GAN and VAE, for improving the coherence and diversity of the paragraphs. For RTT-GAN (plus), they use additional image captioning data from MS-COCO dataset [151] to train their model. Compared with them, TDPG uses a Flat RNN to generate paragraphs while it proposes the trigram repetition penalty in sampling process for reducing repetitions. When comparing with TDPG, we report the results of using the trigram repetition penalty sample strategy. And when comparing with the other methods, we report the results of using beam search with a beam size of 5. CRL uses a different reinforcement learning method.

TABLE 5.4: The performances of various state-of-the-art methods. The top and middle sections show the results sampled by beam search and trigram repetition penalty, respectively. The model in the bottom section uses a different RL reward.

Models	B@1	B@2	B@3	B@4	M	C
Regions-Hierarchical [43]	41.90	24.11	14.23	8.69	15.95	13.52
RTT-GAN [45]	41.99	24.86	14.89	9.03	17.12	16.87
RTT-GAN (Plus) [45]	42.06	25.35	14.92	9.21	18.39	20.36
DCPG [44]	42.12	25.18	14.74	9.05	17.81	19.95
DCPG-VAE [44]	42.38	25.52	15.15	9.43	18.62	20.93
HCAVP [46]	41.38	25.40	14.93	9.00	16.79	20.94
DHPV [49]	43.35	26.73	16.92	10.99	17.02	22.47
HSGED-IRA-INA	43.76	26.44	16.86	9.79	17.14	23.85
HSGED(SLL)	44.22	26.93	17.31	10.35	17.45	25.52
CAE-LSTM [47]	—	—	—	9.67	18.82	25.15
TDPG [48]	43.54	27.44	17.33	10.58	17.86	30.63
HSGED(SLL)	44.51	28.69	18.28	11.26	18.33	36.02
CRL [50]	43.12	27.03	16.72	9.95	17.42	31.47

5.3.3.2 Result Analysis.

TABLE 5.4 reports the performances of our HSGED(SLL) and the other comparing methods. From the table, we observe that our HSGED(SLL) achieves two new state-of-the-art CIDEr-D scores, 25.52 and 36.02, for the cases of using the beam search and the trigram repetition penalty strategy, respectively. In particular, compared with the methods which use HRNN without applying hierarchical constraints, our method outperforms them in almost all metrics, despite that we do not use additional generative models like in RTT-GAN and DCPG-VAE, or train our model with additional language data like in RTT-GAN(Plus), or use more complex LSTM like in HCAVP. The comparison between our HSGED(SLL) and DHPV, which exploits a more complex sentence-level loss, suggests that when providing suitable hierarchical constraints, even with a simple sentence-level loss, our method can still generate better paragraphs. Although CRL uses different reinforcement learning methods, HSGED(SLL) still performs better.

5.3.4 Discussion.

The proposed HSGED exploits the scene graphs as the input, then when the scene graph parser is not perfect that the quality of the generated scene graph is poor, then the errors will be passed into the paragraph generator and the generated

paragraphs will be affected. Thus, we need to carefully parse the scene graphs from an image for our HSGED to generate high quality paragraphs.

Chapter 6

Learning to Collocate Visual-Linguistic Modules for Image Captioning¹

6.1 Introduction

Let's describe three images in FIGURE 6.1a. Most of people will compose sentences varying vastly from image to image. In fact, the ability of using diverse language to describe the colorful visual world is a gift to humans, but a formidable challenge to machines. Although recent advances in visual representation learning [6, 36] and language modeling [22, 116] demonstrate the impressive power of modeling the diversity in their respective modalities, it is still far from being resolved to establish a robust cross-modal connection between them. Indeed, image captioning is not the only visual reasoning task that is solved by easily exploiting the dataset bias even without looking at the image [185] and almost all existing models for visual reasoning tasks such as visual Q&A [123, 186, 187] have been spotted mode collapse to certain dataset idiosyncrasies and failed to reproduce the diversity of our world — the more complex the task is, the more severe the collapse will be, such as image paragraph generation [188], scene graph generation [63], and visual dialog [134, 189]. For example, in MS-COCO [151] captioning training set, as the

¹This chapter is published as Xu Yang, Hanwang Zhang, Jianfei Cai. “Learning to Collocate Neural Modules for Image Captioning.” Proceedings of the IEEE Conference on International Conference on Computer Vision (ICCV, **Poster**). Seoul, Korea, 2019.

FIGURE 6.1: The motivation of our learning to Collocate Visual-Linguistic Neural Modules (CVLNM) for image captioning: the principle of the modular design, which regularizes the diverse training effectively.

co-occurrence chance of “man” and “standing” is 11% (FIGURE 6.2c), a state-of-the-art captioner [1] is very likely to generate “man standing”, regardless of their actual relationship, “milking”, which is a rare one of only 0.023%.

Alas, unlike a visual concept in ImageNet which has 650 training images on average [142] for image classification, a specific sentence in MS-COCO is only corresponding to *one single* image [151], which is extremely scarce in the conventional view of supervised training. However, it is more than enough for us humans — anyone with normal vision (analogous to pre-trained CNN encoder) and language skills

FIGURE 6.2: By comparing our CVLNM with the baseline Module/O, which only uses OBJECT module in the encoder (an upgraded version of Up-Down [1]), we have three interesting findings in tackling the dataset bias: (a) more accurate grammar, where % in the bottom box for each image denotes the frequency of a certain pattern in MS-COCO, (b) more descriptive attributes, and (c) more accurate object interactions. The ratio $./.$ denotes the percentage of co-occurrence, *e.g.*, “sheep+field” / “sheep” = 28% means that “sheep” and “field” contributes the 28% occurrences of “sheep”. We can see that CVLNM outperforms Module/O even with highly biased training samples.

(analogous to pre-trained language decoder) that does NOT need many training samples to perform captioning. We humans can achieve this since we can decompose a sentence into different parts and fill suitable contents into each part. To see this, given a sentence pattern including four parts in Figure 6.1b, even a baby can fill suitable contents to complete three captions for three diverse images. Therefore, even though substantial progress has been made in the past few years since Show&Tell [20], there is still a crucial step missing between vision and language for modern image captioners [1, 27, 32].

Such missing crux is the *principle of modular design* which is strongly suggested by the psychophysics evidence that our brain follows [112]. Such evidence shows that our brain can split up a large computation as a collection of parts that are nearly independent of one another. Inspired by this, researchers have begun to exploit the modular principle to design modern vision systems where the training of them can be further regularized that each module only needs to learn its related supervision instead of being entangled by unrelated knowledge, which makes the system less likely to collapse into the dataset idiosyncrasies [112, 113]. Apart from solving dataset bias, modular design also enriches the flexibility and extensibility of a learning system that researchers can insert suitable modules for achieving specific goals [119, 121].

Motivated by these advantages, we follow the modular design principle [112, 113] and propose learning to Collocate Visual-Linguistic Neural Modules (CVLNM) to decompose captioning into a series of sub-tasks that each one is solved by a

specific module. To cover the frequently appeared part-of-speech of a sentence, we design an empirically complete set of modules in the encoder, including OBJECT module for nouns, ATTRIBUTE module for adjectives, RELATION module for verbs, prepositions, and quantifiers, and FUNCTION module for other function words (see Section 6.2.1). Among them, the former three are visual modules and the last one is a linguistic module. We apply different inductive biases to design these modules and in this way, they can learn distinguishable knowledge [114] to solve specific sub-tasks, *i.e.*, generate the module-related words. Moreover, our CVLNM is flexible in that we can insert other modules for specific goals. In this research, we plug a linguistic REASON module into the decoder to approximate human-like commonsense reasoning for more descriptive captions (see Section 6.2.4).

Though the idea of neural module networks has appeared in other vision-language tasks such as VQA [119], where the question is parsed into a module structure like COLOR(FIND(‘chair’)) from the question “What color is the chair?”, for image captioning, the case is more challenging as only partially observed sentences are available during captioning and we cannot parse a perfect module structure from such sentence. To address this challenge, we design a self-attention based module controller to dynamically collocate modules during captioning conditioned on the visual and linguistic context knowledge (see Section 6.2.2). By this controller, our CVLNM can discover the frequently appeared syntax patterns from the training dataset and use them to regularize the caption training and generation. During the caption generation, at each time step, we first choose a few modules and then use their knowledge to generate the words. FIGURE 6.1c sketches such captioning process where we choose the most responsible module as the example.

Since the module layout is dynamically parsed in image captioning, it is far from perfect compared with the layout in the other visual reasoning tasks where the auxiliary sentences are available [121, 123, 129]. To enhance the effectiveness and robustness of our captioner, the following techniques are also applied to the controller. 1) The module controller is designed to softly fuse four distinguishable modules in the encoder to comprehensively exploit their knowledge (see Section 6.2.2.2). 2) A part-of-speech based syntax loss is imposed on the module controller to make the generated module layout more faithful to human-like syntax pattern, *e.g.*, an adjective is usually used before a noun and in such a case the syntax loss will encourage ATTRIBUTE module to be selected before OBJECT module when captioning

(see Section 6.2.3). And the meantime, this loss encourages each module to learn its module-specific knowledge for further decomposing the image captioning task.

Extensive discussions and human evaluations are offered in Section 6.3.2 to validate the effectiveness and robustness of CVLNM on the challenging MS-COCO image captioning benchmark. Overall, we achieve a new state-of-the-art 129.5 CIDEr-D score on Karpathy split and 127.8 c40 on the official server. More importantly, by following the principle of modular design, we find that our CVLNM is less likely to collapse into dataset bias. For example, compared to a strong non-modular baseline Up-Down [1], our CVLNM generates 1) more accurate grammar (FIGURE 6.2a) thanks to the joint reasoning of FUNCTION and OBJECT module, 2) more descriptive attributes (FIGURE 6.2b) due to ATTRIBUTE module, and 3) more accurate interactions (FIGURE 6.2c) due to RELATION module. Moreover, we find that when only one training sentence of each image is provided, our CVLNM will suffer less performance deterioration compared with the strong non-module baseline.

Our contributions can be summarized as follows:

- Our CVLNM is the first module network for image captioning, which is a generic framework with principled module and controller designs. This enriches the spectrum of using neural modules for vision-language tasks.
- We develop an advanced module controller and a syntax based loss for robustifying the dynamic module collocations and meanwhile encouraging each distinguishable module to learn their module-specific knowledge.
- To show the extensibility of our framework, we plug a REASON module in the decoder to approximate commonsense reasoning for more descriptive captions.
- We conduct extensive experiments with various metrics. Experiment results show the effectiveness and robustness of our CVLNM.

6.2 Collocation of Visual-Linguistic Neural Modules

FIGURE 6.3 shows the encoder-decoder structure of our learning to Collocates Visual-Linguistic Neural Modules (CVLNM) model. The encoder contains a CNN

FIGURE 6.3: The encoder-decoder pipeline of our image captioner that Collocates Visual-Linguistic Neural Modules (CVLNM). The dash lines from RNN to FUNCTION module and the module controller mean that both of these sub-networks require the contextual knowledge of the partially observed sentences, which is accumulated by the RNN.

and four distinguishable neural modules to extract different features for language decoding (see Section 6.2.1). Our decoder has a module controller (see Section 6.2.2) that softly fuses these features into a single feature for further commonsense reasoning and language decoding by the followed linguistic REASON module (see Section 6.2.4) and RNN, respectively. We also impose a part-of-speech based syntax loss on the controller to make it more faithful to human-like module collocations and meantime to regularize the distinguishable training of each module (see Section 6.2.3). Besides composing captions, the RNN also accumulates the context of the partially observed sentence as the input to FUNCTION module and the controller for providing linguistic information (See Section 6.2.5).

6.2.1 Distinguishable Feature Extraction Modules

Following the principle of modular design, as in the encoder of FIGURE 6.3, we design three visual and one linguistic neural modules to cover the frequently appeared part-of-speech of the captioning dataset, *i.e.*, OBJECT module for nouns; ATTRIBUTE module for adjectives; RELATION module for verbs, prepositions, and quantifiers; and FUNCTION module for other function words. In this way, the complex captioning task is decomposed to easier sub-tasks addressed by specific modules. As a result, the visual reasoning can be more robusified because different words are predicted from distinguishable modules that the whole system is less likely to collapse to the word co-occurrence as in Fig 6.2.

6.2.1.1 Object Module

It is a visual module designed to transform the CNN features to a feature set $\mathcal{V}_O$ containing the knowledge about object categories, *i.e.*, $\mathcal{V}_O$ facilitates the nouns predictions like “person” or “dog”. The input of this module is the feature set $\mathcal{R}_O \in \mathbb{R}^{N \times d_r}$, which is extracted by a ResNet-101 Faster R-CNN [6] pre-trained on the object detection task using the object annotations of VG dataset [56], where N is the number of detected objects and d_r is the dimension of an ROI feature. Formally, this module can be written as:

$$\begin{aligned} \textbf{Input: } & \mathcal{R}_O, \\ \textbf{Output: } & \mathcal{V}_O = \text{LeakyReLU}(\text{FC}(\mathcal{R}_O)), \end{aligned} \tag{6.1}$$

where $\mathcal{V}_O \in \mathbb{R}^{N \times d_v}$ is the output feature set.

6.2.1.2 Attribute Module

It is a visual module designed to transform the CNN features to a feature set $\mathcal{V}_A$ containing attribute knowledge for generating adjectives like “black” or “small”. The input of this module is $\mathcal{R}_A \in \mathbb{R}^{N \times d_r}$, which is also extracted by a ResNet-101 Faster R-CNN, while this network is pre-trained on the attribute classification task using the attribute annotations of VG dataset. Formally, this module can be written as:

$$\begin{aligned} \textbf{Input: } & \mathcal{R}_A, \\ \textbf{Output: } & \mathcal{V}_A = \text{LeakyReLU}(\text{FC}(\mathcal{R}_A)), \end{aligned} \tag{6.2}$$

where $\mathcal{V}_A \in \mathbb{R}^{N \times d_v}$ is the output of this module.

6.2.1.3 Relation Module

It is a visual module designed to transform CNN features to a feature set $\mathcal{V}_R$ containing knowledge about potential relations between objects, which would help generate verbs like “ride”, prepositions like “on”, or quantifiers like “three”. This module is built upon the multi-head self-attention mechanism [116], which automatically seeks the relations among the objects during the attention computations between the corresponding features. Here, we use $\mathcal{R}_O$ in Eq. (6.1) as the input because these kinds of features are widely applied as the input for successful relation detection [57, 58]. This module is formulated as:

$$\begin{aligned} \textbf{Input: } & \mathcal{R}_O, \\ \textbf{MH-SA: } & \mathcal{M} = \text{MH-SA}(\mathcal{R}_O), \\ \textbf{Output: } & \mathcal{V}_R = \text{LeakyReLU}(\text{MLP}(\mathcal{M})), \end{aligned} \tag{6.3}$$

where $\text{MH-SA}(\cdot)$ is the multi-head self-attention mechanism, $\text{MLP}(\cdot)$ is a feed-forward network containing two fully connected layers with a ReLU activation layer in between [116]: FC-ReLU-FC, and $\mathcal{V}_R \in \mathbb{R}^{N \times d_v}$ is the output of this module. Specifically, the following steps are deployed to compute the multi-head self-attention. We first use the scaled dot-product to compute k head matrices $\text{head}_i \in \mathbb{R}^{N \times d_k}$:

$$\text{head}_i = \text{Softmax}\left(\frac{\mathcal{R}_O \mathbf{W}_i^1 (\mathcal{R}_O \mathbf{W}_i^2)^T}{\sqrt{d_k}}\right) \mathcal{R}_O \mathbf{W}_i^3, \tag{6.4}$$

where $\mathbf{W}_i^1, \mathbf{W}_i^2, \mathbf{W}_i^3 \in \mathbb{R}^{d_r \times d_k}$ are all trainable matrices, $d_k = d_r/k$ and k is the number of head matrices. Then these k head matrices are concatenated and linearly projected to the feature set $\mathcal{M} \in \mathbb{R}^{N \times d_r}$:

$$\mathcal{M} = [\text{head}_1, \dots, \text{head}_k] \mathbf{W}_C, \tag{6.5}$$

where $[\cdot]$ means concatenation operation and $\mathbf{W}_C \in \mathbb{R}^{d_r \times d_r}$ is a trainable matrix.

FIGURE 6.4 demonstrates how this self-attention based RELATION module generates relation specific words. For example, in the middle figure, at the third time step, RELATION module focuses more on the “paw” part (red box) of the bird, and meantime the clues about “bird” (yellow box) and “tree” (blue box) are incorporated to the feature of the “paw” part by multi-head self-attention. By

FIGURE 6.4: Three visualizations show how RELATION module generates relation specific words. The red box in each image is the attended image region when RELATION module generates a relation specific word. The thickness of lines connecting different boxes is determined by the soft attention weights computed by the self-attention technique in Eq. (6.4), *i.e.*, the larger the attention weight between two boxes, the thicker the line between these boxes.

exhaustively considering these visual clues produced by RELATION module, a more accurate action “perch” is generated by our CVLNM to describe the action of the bird.

6.2.1.4 Function Module

It is a linguistic module designed to produce a single feature $\hat{\mathbf{v}}_F$ for generating function words like “a” or “and”. The input of this module is the linguistic context vector $\mathbf{c} \in \mathbb{R}^{d_c}$ accumulated in the RNN (which is $\mathbf{h}_t^2$ in Eq. (6.16)), as the dashed line drawn in FIGURE 6.3. We use $\mathbf{c}$ as the input because it contains rich linguistic context knowledge of the partially generated captions and such knowledge instead of visual knowledge is more beneficial for generating function words. This module is formulated as:

$$\begin{aligned} \text{Input: } & \mathbf{c}, \\ \text{Output: } & \hat{\mathbf{v}}_F = \text{LeakyReLU}(\text{FC}(\mathbf{c})), \end{aligned} \tag{6.6}$$

where $\hat{\mathbf{v}}_F \in \mathbb{R}^{d_v}$ is the output feature.

FIGURE 6.5: The detailed structure of our module controller, which consist of three attention networks and a self-attention (SA) network to generate four soft weights to fuse the attended features of four modules (i.e. $\hat{v}_O$, $\hat{v}_A$, $\hat{v}_R$, $\hat{v}_F$) into a single feature $\hat{v}$.

6.2.2 Module Controller

Yet, it is still an open question on how to define a perfect complete set of neural modules for visual reasoning [119, 127], we believe that a variety of complex tasks can be approximately accomplished by adaptively selecting modules from an empirically complete module set conditioned on the visual and linguistic context [122]. Motivated by this, we design a module controller for dynamically collocating the modules under the condition that only partially generated caption is available. FIGURE 6.5 shows the detail of the module controller, which contains three attention networks and one self-attention based soft-weight generator. The fused feature vector $\hat{v}$ is the output of this module controller, which will be input into the followed REASON module and RNN for the next step reasoning (FIGURE 6.3). Next, we describe how each component of our module controller works.

6.2.2.1 Module Attention

Before the soft fusion, three attention networks are used to respectively transform three visual modules' output into three more informative features:

$$\begin{aligned}
\textbf{Object Attention: } \hat{\mathbf{v}}_O &= \text{Att}_{Obj}(\mathcal{V}_O, \mathbf{h}), \\
\textbf{Attribute Attention: } \hat{\mathbf{v}}_A &= \text{Att}_{Attr}(\mathcal{V}_A, \mathbf{h}), \\
\textbf{Relation Attention: } \hat{\mathbf{v}}_R &= \text{Att}_{Rela}(\mathcal{V}_R, \mathbf{h}),
\end{aligned} \tag{6.7}$$

where $\hat{\mathbf{v}}_O, \hat{\mathbf{v}}_A, \hat{\mathbf{v}}_R \in \mathbb{R}^{d_v}$ are the transformed features of $\mathcal{V}_O, \mathcal{V}_A$, and $\mathcal{V}_R$ produced by OBJECT, ATTRIBUTE, and RELATION modules (Section 6.2.1), respectively; $\mathbf{h} \in \mathbb{R}^{d_c}$ is the query vector produced by the RNN (which is $\mathbf{h}_t^1$ in Eq. (6.16)); and three attention networks own the same structure while the parameters are not shared:

$$\begin{aligned}
\textbf{Input: } & \mathcal{V}, \mathbf{h}, \\
\textbf{Attention weight: } & a_i = \boldsymbol{\omega}_a^T \tanh(\mathbf{W}_v \mathbf{v}_i + \mathbf{W}_h \mathbf{h}), \\
& \boldsymbol{\alpha} = \text{softmax}(\mathbf{a}), \\
\textbf{Output: } & \hat{\mathbf{v}} = \mathcal{V} \boldsymbol{\alpha},
\end{aligned} \tag{6.8}$$

where $\mathbf{W}_h \in \mathbb{R}^{d_a \times d_c}$, $\mathbf{W}_v \in \mathbb{R}^{d_a \times d_v}$ and $\boldsymbol{\omega}_a \in \mathbb{R}^{d_a}$ are all trainable parameters.

6.2.2.2 Soft Fusion

Since we only have partially observed caption as the source for parsing the module layout, the parsed layout is far from perfect compared with some other vision-language tasks like VQA or visual grounding. To enhance the robustness of our CVLNM, we softly fuse these module outputs to provide more comprehensive vision and language knowledge for better captions. After computing the features from each module: $\hat{\mathbf{v}}_O, \hat{\mathbf{v}}_A, \hat{\mathbf{v}}_R$ (Eq. (6.7)), and $\hat{\mathbf{v}}_F$ (Eq. (6.6)), the controller generates four soft weights to fuse them.

Since the module collocation at each time step is highly related to the previous module collocations, we build this generator upon the multi-head self-attention network with the previous module collocations $\mathcal{Z}$ as input. We apply self-attention instead of LSTM (which is used in our preliminary work [178]) since the current module collocation is possibly related to a more remote module collocation and self-attention is good at dealing with such long-range dependencies [116, 190].

At time step t , $\mathcal{Z} = \{z_{0:t-1}\} \in \mathbb{R}^{t \times d_z}$ where z_0 is the embedding of the start token and z_τ is the embedding of the chosen modules at time step τ (Eq. (6.12)). We first deploy multi-head self-attention upon $\mathcal{Z}$ to build the connections among collocations in different steps:

$$\hat{\mathcal{Z}} = \text{MH-SA}(\mathcal{Z}), \quad (6.9)$$

where $\text{MH-SA}(\cdot)$ has the similar formula as Eq. (6.3) with different parameters. Then the multi-modal context vector $\mathbf{x} \in \mathbb{R}^{d_z}$ is used as the query vector to get the current soft collocation weights and $\mathbf{x}$ is:

$$\mathbf{x} = \text{ReLU}(\text{FC}([\hat{\mathbf{v}}_O, \hat{\mathbf{v}}_A, \hat{\mathbf{v}}_R, \mathbf{c}])), \quad (6.10)$$

where $\mathbf{c}$ is the context vector accumulated in the RNN (which is $\mathbf{h}_t^2$ in Eq. (6.16) in our experiments). We use $\mathbf{x}$ as the query vector because both the visual clues $(\hat{\mathbf{v}}_O, \hat{\mathbf{v}}_A, \hat{\mathbf{v}}_R)$ and the linguistic context $(\mathbf{c})$ of the partially generated captions are indispensable for successfully collocating modules at a new time step. The process of soft fusion is formulated as:

$$\begin{aligned} \text{Input: } & \hat{\mathcal{Z}}, \mathbf{x}, \\ \text{MH: } & \text{head}_i = \text{Softmax}\left(\frac{\mathbf{x} \mathbf{W}_l^1 (\hat{\mathcal{Z}} \mathbf{W}_l^2)^T}{\sqrt{d_z}}\right) \hat{\mathcal{Z}} \mathbf{W}_l^3, \\ & \hat{\mathbf{x}} = [\text{head}_1, \dots, \text{head}_j] \mathbf{W}_Z, \\ \text{Soft Vector: } & \mathbf{w} = \{w_O, w_A, w_R, w_F\} = \text{Softmax}(\text{FC}(\hat{\mathbf{x}})), \\ \text{Output: } & \hat{\mathbf{v}} = [w_O \hat{\mathbf{v}}_O, w_A \hat{\mathbf{v}}_A, w_R \hat{\mathbf{v}}_R, w_F \hat{\mathbf{v}}_F], \end{aligned} \quad (6.11)$$

where $\mathbf{W}_l^1, \mathbf{W}_l^2, \mathbf{W}_l^3 \in \mathbb{R}^{d_z \times d_j}$, $\mathbf{W}_Z \in \mathbb{R}^{d_z \times d_z}$ are all trainable matrices, $d_j = d_z/j$ is the dimension of each head vector, and j is the number of the head vectors; $\mathbf{w} \in \mathbb{R}^4$ is the soft fusion vector; and the output fusion vector is $\hat{\mathbf{v}} \in \mathbb{R}^{4d_v}$.

After generating the fusion weights $\mathbf{w}$, we compute the t -th module embedding $\mathbf{z}_t$ as:

$$\mathbf{z}_t = w_O \mathbf{e}_O + w_A \mathbf{e}_A + w_R \mathbf{e}_R + w_F \mathbf{e}_F, \quad (6.12)$$

where $\mathbf{e}_O, \mathbf{e}_A, \mathbf{e}_R, \mathbf{e}_F \in \mathbb{R}^{d_z}$ are four learnable label embeddings² corresponding to OBJECT, ATTRIBUTE, RELATION, and FUNCTION modules, respectively. The positional encoding [116] is also used to inject the position knowledge of the module

²Learnable label embeddings mean that they are 4-dimensional one hot vectors multiplied with a learnable $4 \times d_z$ matrix.

layout, whose size is set to d_z . This position encoding and $\mathbf{z}_t$ are summed together to get the module embedding, which is used in Eq. (6.9), while for convenience, we still use $\mathbf{z}_t$ to denote it.

6.2.3 Syntax Loss

To further encourage each module to learn distinguishable knowledge, *e.g.*, OBJECT module focuses more on object categories instead of visual attributes, and make the module layout faithful to human-like collocations, *e.g.*, adjectives are usually be said before nouns, we design a part-of-speech based syntax loss and impose it on the module controller.

We build this loss by extracting the words’ part-of-speech (*e.g.*, adjectives, nouns, or verbs) from the ground-truth captions by a part-of-speech tagger tool [191]. According to the extracted part-of-speech, we assign each word an one hot vector $\mathbf{w}^* \in \mathbb{R}^4$, indicating which module should be chosen for generating this word. In particular, we assign OBJECT module to nouns (NN like “bus”); ATTRIBUTE module to adjectives (ADJ like “green”); RELATION module to verbs (VB like “drive”), prepositions (PREP like “on”), and quantifiers (CD like “three”); and FUNCTION module to the other words (CC like “and”).

Given these expert-guided module layout $\mathbf{w}^*$, our syntax loss L_s is defined as the cross-entropy value between $\mathbf{w}^*$ and soft fusion weights $\mathbf{w}$ which are calculated from the module controller (Eq. (6.11)):

$$L_s = - \sum_{i=1}^4 w_i^* \log w_i. \quad (6.13)$$

Importantly, when this loss is imposed on the module controller, the soft fusion strategy in Section 6.2.2.2 approximates to the hard selection that only one feature extraction module will be responsible for the generated word. For example, when the noun “dog” is the supervision, only $w_O^* = 1$ and w_O in Eq. (6.11) is encouraged to be 1 due to this syntax loss. Then the $\hat{\mathbf{v}}_O$ part in the fusion representation $\hat{\mathbf{v}}$ will get more back-propagated gradients from the noun “dog”, and thus OBJECT module is more encouraged to learn from this noun. In this way, this syntax loss regularizes the training of the feature extraction modules to make them more distinguishable. Such argument is validated in Section 6.3.2 where we find that this

loss indeed increases the recalls of each module-specific words and the accuracy of the module collocations.

6.2.4 Reason Module

In addition to distinguishing different aspects of an image during captioning, we humans can also achieve commonsense reasoning to summarize these aspects or even infer something that is not evidently detected from the image. For example, when we observe some persons holding umbrellas in a wet road, we can infer that it is raining even the raindrops are not obvious (which means OBJECT module can hardly extract evident knowledge about the object “raindrop”). To equip our image captioner with such commonsense reasoning, we plug a linguistic REASON module into the decoder as sketched in FIGURE 6.3. This module is inserted between the controller and the RNN to imitate that we humans usually achieve commonsense reasoning after grasping the comprehensive knowledge of a visual scene, where such knowledge is preserved in the fused representation $\hat{\mathbf{v}}$ (Eq. (6.11)) of the four different modules.

We exploit ConceptNet [192] dataset, which is originally built for helping computers to achieve commonsense reasoning, to build our REASON module. We collect the relation triplets from this dataset and embed them into a key-value memory network [100, 102]: $\mathcal{M} = \{\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_K\} \in \mathbb{R}^{d_m \times K}$, where each element $\mathbf{m}_k$ is the embedding of a relation triplet. The relation triplet is represented as “subject-predicate-object”, *e.g.*, “umbrella-related to-rain” or “eagle-type of-bird”, and we design an efficient relation embedding operation to transform it into the continuous embedding $\mathbf{m}$:

$$\mathbf{m} = \text{ReLU}(\text{FC}([\mathbf{e}_s, \mathbf{e}_p, \mathbf{e}_o])), \quad (6.14)$$

where $\mathbf{e}_s, \mathbf{e}_p, \mathbf{e}_o \in \mathbb{R}^{d_e}$ denote trainable word embeddings of the subject, predicate, and object, respectively.

Given this memory network, we apply attention computation to imitate commonsense reasoning: the fused representation $\hat{\mathbf{v}}$ acts as the query and the embedding

$\mathbf{m}_k$ of $\mathcal{M}$ acts as both the key and the value:

$$\begin{aligned}
&\textbf{Input: } \hat{\mathbf{v}}, \mathcal{M} = \{\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_K\} \\
&\textbf{Query-Key Attention: } \beta = \text{softmax}(\mathcal{M}^T \mathbf{W}_v \hat{\mathbf{v}}), \\
&\textbf{Output: } \mathbf{v}' = \mathcal{M} \beta = \sum_{k=1}^K \beta_k \mathbf{m}_k,
\end{aligned} \tag{6.15}$$

where $\mathbf{W}_v \in \mathbb{R}^{d_m \times 4d_v}$ is a trainable matrix and $\mathbf{v}'$ is the output of REASON module, which provides commonsense knowledge for the RNN to captioning.

6.2.5 Training and Inference of CVLNM

By using ResNet-101 [36] based Faster-RCNN as CNN, the five neural modules, the module controller, and Top-Down architecture [1] as RNN (see FIGURE 6.3), our CVLNM can predict the probability of the next word s_t given the image and the last word s_{t-1} . More specifically, Top-Down RNN contains two LSTM layers and it generates the word distribution $P(s_t)$ as follows:

$$\begin{aligned}
&\textbf{Input: } s_{t-1}, \hat{\mathbf{v}}, \mathbf{v}' \\
&\textbf{LSTM1: } \mathbf{h}_t^1 = \text{LSTM1}([\text{Embed}(s_{t-1}), \mathbf{h}_{t-1}^2]) \\
&\textbf{LSTM2: } \mathbf{h}_t^2 = \text{LSTM2}([\hat{\mathbf{v}}, \mathbf{v}', \mathbf{h}_t^1]) \\
&\textbf{Output: } P(s_t) = \text{Softmax}(\text{FC}(\mathbf{h}_t^2)),
\end{aligned} \tag{6.16}$$

where $\hat{\mathbf{v}}, \mathbf{v}'$ are the output of the module controller (Eq. (6.11)) and REASON module (Eq. (6.15)); $\text{Embed}(\cdot)$ is the learnable word embedding layer; $\text{LSTM1}(\cdot)$ and $\text{LSTM2}(\cdot)$ are two different LSTM layers. In Eq. (6.7), the query vector $\mathbf{h} = \mathbf{h}_t^1$ and in Eq. (6.6) and Eq. (6.10), the context vector $\mathbf{c} = \mathbf{h}_t^2$.

Given the ground-truth caption $\mathcal{S}^* = \{s_{1:T}^*\}$ with its extracted part-of-speech tags $\mathcal{W}^* = \{\mathbf{w}_{1:T}^*\}$, we can end-to-end train our CVLNM by minimizing the syntax loss L_s defined in Eq. (6.13) and the language loss L_l between the generated captions and the ground-truth captions. Given the predicted word distribution $P(s)$, we can define the language loss L_l as the cross-entropy loss:

$$L_l = L_{XE} = - \sum_{t=1}^T \log P(s_t^*), \tag{6.17}$$

or the negative reinforcement learning (RL) based reward [29]:

$$L_l = L_{RL} = -\mathbb{E}_{s_t^s \sim P(s)}[r(s_{1:T}^s; s_{1:T}^*)], \tag{6.18}$$

where r is a sentence-level metric, *e.g.*, CIDEr-D [166], for the sampled sentence $\mathcal{S}^s = \{s_{1:T}^s\}$ and the ground-truth $\mathcal{S}^* = \{s_{1:T}^*\}$. Then the total loss becomes

$$L = L_l + \lambda L_s, \quad (6.19)$$

where λ is the trade-off weight. During inference, we adopt the beam search strategy [29] with a beam size of 5 to sample the caption from the predicted distribution $P(s)$.

6.3 Experiments

6.3.1 Datasets and Settings

MS-COCO [151]. This dataset provides one official split for researchers to test the effectiveness of their models: 82,783, 40,504 and 40,775 images for training, validation and test, respectively. The 3rd-party Karpathy split [172] is used for the off-line test, which has 113,287, 5,000, 5,000 images for training, validation and test, respectively. We deployed our CVLNM and its comparison methods on both splits to validate the effectiveness. The captions of COCO were pre-processed by the following steps: the texts were first tokenized on white spaces; all the letters were changed to lowercase; the words were removed if they appear less than 5 times and then we had a vocabulary with 10,369 words; each caption was trimmed to a maximum of 16 words.

Visual Genome [56] (**VG**). We first exploited the object annotations of VG to pre-train a ResNet-101 Faster RCNN to extract $\mathcal{V}_O$, which is input into OBJECT module (Section 6.2.1.1). Then we continued to use the attribute annotations to fine-tune this Faster RCNN and extracted $\mathcal{V}_A$ for ATTRIBUTE module (Section 6.2.1.2). When training this Faster-RCNN, we did not use all the object and attribute annotations because the dataset is noisy. We filtered the noisy annotations by keeping the labels which appear more than 2,000 times in the training set and then 305 object and 103 attribute labels remained. Importantly, since some images co-exist in both VG and COCO, we filtered out these images and their annotations of VG, which also appear in COCO test set.

TABLE 6.1: The sizes of different trainable parameters.

symbol	equation	number	symbol	equation	number
d_r	Eq. (6.1)	2048	d_v	Eq. (6.1)	1000
k	Eq. (6.4)	8	d_k	Eq. (6.4)	256
d_c	Eq. (6.6)	1000	d_a	Eq. (6.8)	512
d_z	Eq. (6.11)	1000	j	Eq. (6.11)	8
d_j	Eq. (6.11)	125	d_e	Eq. (6.14)	1000
d_m	Eq. (6.15)	1000	K	Eq. (6.15)	10,000

ConceptNet [192]. ConceptNet is a freely-available semantic dataset, which contains abundant commonsense triplets for facilitating researchers to develop human-level AI agents. Each triplet is formed as “subject-predicate-object” and is also assigned with a weight to measure its importance. We exploited these weighted triplets to build our REASON module (Section 6.2.4) for approximating commonsense reasoning. We searched the related triplets from ConceptNet by using the words in caption vocabulary as the keys and preserved the top 10,000 searched commonsense triplets into the memory network $\mathcal{M}$ according to the importance weight of each triplet.

Settings. TABLE 6.1 summarizes the sizes of various training parameters used in our experiments. We used Adam optimizer [155] to train the whole model. The learning rate was initialized to $5e^{-4}$ and was decayed by 0.8 for every 5 epochs. The cross-entropy loss Eq. (6.17) and the RL-based loss Eq. (6.18) were in turn used as the language loss to train our CVLNM for 35 epochs and 65 epochs, respectively. In the experiments, we found that the performance is non-sensitive to the trade-off weight λ in Eq. (6.19). By default, we empirically set the trade-off weight $\lambda = 1$ and $\lambda = 0.5$ when the cross-entropy loss and the RL-based loss were used as the language loss, respectively. The batch size was set to 100.

6.3.2 Ablation Studies

We conducted extensive ablations for confirming the effectiveness of each component in this research by gradually incorporating them into the pipeline, including four distinguishable modules in the encoder (see Section 6.2.1), the module controller (see Section 6.2.2), the part-of-speech based syntax loss (see Section 6.2.3), and REASON module (see Section 6.2.4). The full model is called **CVLNM**, which

will be used to compare with state-of-the-art image captioners in Section 6.3.3. We arrange the ablation studies by proposing research questions (**Q**) in Section 6.3.2.1, specifying miscellaneous metrics in Section 6.3.2.2, and providing the corresponding empirical answers (**A**) in Section 6.3.2.3.

6.3.2.1 Research Questions and the Corresponding Baselines

Q1: Will each visual module introduced in Section 6.2.1, *i.e.* OBJECT, ATTRIBUTE, and RELATION modules, learn distinguishable knowledge and then generate more accurate module-specific words, *e.g.*, will OBJECT module generate more accurate nouns?

To answer it, the following baselines were designed: we only used a single visual module in the encoder to extract features and deployed the top-down attention LSTM [1] as the decoder for captioning. When OBJECT, ATTRIBUTE, and RELATION modules were used, the baselines are denoted as **Module/O**, **Module/A**, and **Module/R**, respectively. Note that the baseline Module/O is an upgraded version of Up-Down model [1].

Q2: Will the qualities of the generated captions be improved when the modules in the encoder are fused? How to fuse them to achieve a better performance?

To answer them, the following three strategies were designed, where each one used a specific type of fusion weights to fuse the modules. Specifically, when we set all the fusion weights as 1, the baseline is called **Col/1**, which equals to the model without the module controller; when learnable soft fusion weight vector $\mathbf{w}$ (Eq. (6.11)) was used, the baseline is called **Col/S**; and when Gumbel-Softmax layer [193] was used for transferring soft weight vector $\mathbf{w}$ into one hot vector to achieve hard selection, the baseline is called **Col/H**. In both Col/S and Col/H, we input the previous module layout $\mathcal{Z}$ into the self-attention based module controller to generate the fusion weights $\mathbf{w}$.

Q3: Will inputting previous module layout $\mathcal{Z}$ (Eq. (6.11)) into the module controller generate better soft fusion weights? And will self-attention preserve more long-range dependencies between remote module collocations than LSTM (which is deployed in our preliminary work CNM [178]) for better module collocations?

Then, more importantly, will better soft fusion weights encourage each visual feature extraction module to learn its own distinguishable knowledge?

To answer them, we compared the following baselines. When we built the controller upon the self-attention mechanism with/without $\mathcal{Z}$ as input, the baselines are named as **SA+ $\mathcal{Z}$ /SA**. In the same vein, when the module controller is built upon LSTM, the baselines are **LSTM+ $\mathcal{Z}$ /LSTM** based on whether we used $\mathcal{Z}$ as the input or not. In these baselines, the encoder contains all the four modules, the soft fusion strategy was applied, and we did not use REASON module and the syntax loss. Note that SA+ $\mathcal{Z}$ equals to the baseline Col/S in Q2.

Q4: Will the expert knowledge of part-of-speech collocations provided by the syntax loss L_s (Eq. (6.13)) benefit the model? Also, will such loss encourage each feature extraction module to learn more distinguishable knowledge?

To answer them, we compared the models by training them with and without L_s . When we added the syntax loss to the baseline SA+ $\mathcal{Z}$, we have the baseline **SA+ $\mathcal{Z}$ + L_s** .

Q5: Will REASON module (see Section 6.2.4) in the decoder generate more human-like captions?

To answer it, we inserted REASON module into SA+ $\mathcal{Z}$ to get **SA+ $\mathcal{Z}$ +Reason** and compared these two baselines.

Q6: Will the integrated model achieve the best performances among all the baselines?

We compared our full model **CVLNM** with the other baselines. Note that when incorporating REASON module into the baseline SA+ $\mathcal{Z}$ + L_s or applying the syntax loss in the baseline SA+ $\mathcal{Z}$ +Reason, we get CVLNM.

6.3.2.2 Evaluation Metrics for Answering Questions

To quantitatively answer the above research questions, we applied the following miscellaneous metrics to comprehensively test the effectiveness and robustness of our CVLNM.

1) We measured the similarities between the generated captions and the ground-truth captions by five standard metrics, which are CIDEr-D [166], BLEU [174], METEOR [175], ROUGE [176], and SPICE [84]. CIDEr-D is the most robustness one among them with the strongest discriminative power. The higher these similarity metrics are, the more human-like the generated captions are.

2) We exploited CHAIRs and CHAIRi [185] to quantitatively measure the bias degree of the generated captions to some objects. The lower CHAIRs and CHAIRi are, the less biased the captions are. FIGURE 6.2 gives some examples of the biased problem. TABLE 6.2 reports the results on the five similarity metrics and the two bias degree metrics, where the models are trained following the settings in Section 6.3.1.

3) To confirm whether each feature extraction module learns its module-specific knowledge, we calculated two metrics which are the recall of each module-specific word and the accuracy of the predicted module layout. To calculate the recalls of the module-specific words, *e.g.*, the nouns for the OBJECT module, we first counted all the irredundant nouns from all the ground-truth captions of the i -th image and denoted this number as N_{gt}^i , and counted how many of these nouns irredundantly appeared in the predicted caption as N_{pre}^i . Then we calculated $\sum_i N_{pre}^i / \sum_i N_{gt}^i$ as the recall of nouns over all the images in the test set. Specifically, we calculated the recalls of five different types of words: nouns for OBJECT module; adjectives for ATTRIBUTE module; and verbs, prepositions, and quantifiers for RELATION module, and reported them in TABLE 6.3. To measure the accuracy of the predicted module layout, we inspected whether a word is generated by exploiting more knowledge from the corresponding module, *e.g.*, whether a noun is generated from OBJECT module. Mathematically, we treated the module with the maximum soft fusion weight w (Eq. (6.11)) as the predicted module and inspected whether this module matches with the ground-truth module w^* (see Section 6.2.3). Such accuracy is reported in TABLE 6.4.

4) Apart from them, we also carried human evaluation to test the qualities of the generated captions. Specifically, we invited 20 workers for human evaluation. We exhibited 50 images sampled from the test set for each worker and asked them to pairwise compare the captions generated from three models: Module/O, SA+ $\mathcal{Z}$ + L_s , and CVLNM. The captions were compared from two aspects: the language itself that whether the generated captions are fluent and descriptive (the top

TABLE 6.2: The performances of various baselines on Karpathy split. The metrics: B@4, M, R, C, S, CHs, and CHi denote BLEU@4, METEOR, ROUGE-L, CIDEr-D, SPICE, CHAIRs, and CHAIRi, respectively. The symbols $\uparrow$ and $\downarrow$ mean the higher the better and the lower the better, respectively.

Models	B@4 $\uparrow$	M $\uparrow$	R $\uparrow$	C $\uparrow$	S $\uparrow$	CHs $\downarrow$	CHi $\downarrow$
Module/O	37.5	27.7	57.5	123.1	21.0	13.8	9.4
Module/A	37.3	27.4	57.1	121.9	20.9	14.3	9.8
Module/R	37.9	27.8	57.8	123.8	21.2	13.6	9.3
Col/1	38.2	27.9	58.1	125.6	21.2	13.5	9.1
Col/H	38.4	28.2	58.3	126.1	21.3	11.7	8.2
Col/S (SA+ $\mathcal{Z}$)	38.5	28.3	58.4	127.3	21.6	11.3	7.8
LSTM	38.2	28.0	58.2	125.7	21.2	12.4	8.5
LSTM+ $\mathcal{Z}$	38.4	28.1	58.3	126.3	21.4	12.3	8.4
SA	38.3	28.0	58.4	126.7	21.2	11.9	8.2
SA+ $\mathcal{Z}$ +Reason	38.7	28.3	58.6	128.6	21.7	11.0	7.6
SA+ $\mathcal{Z}$ + L_s	38.9	28.4	58.5	128.4	21.9	10.8	7.4
CVLNM	39.4	28.7	59.1	129.5	22.2	10.5	7.0

TABLE 6.3: The recalls (%) of five part-of-speech words, where nouns correspond to OBJECT module; adjectives correspond to ATTRIBUTE module; and verbs, prepositions, and quantifiers correspond to RELATION module.

Models	nouns	adjectives	verbs	prepositions	quantifiers
Module/A	42.4	12.4	20.2	41.7	14.3
Module/O	44.5	11.5	21.8	42.6	17.1
Module/R	44.3	11.3	22.8	43.5	22.3
LSTM	45.2	13.1	23.1	43.6	23.4
SA	45.5	13.5	23.4	43.9	24.2
SA+ $\mathcal{Z}$	46.3	14.8	23.8	44.2	26.8
SA+ $\mathcal{Z}$ + L_s	47.6	16.5	24.4	45.1	29.6

TABLE 6.4: The accuracy (%) of the predicted module layout by the module controller. The column “Average” means the average accuracy of all four modules.

Models	OBJECT	ATTRIBUTE	RELATION	FUNCTION	Average
LSTM	56.3	39.7	75.3	27.7	59.4
SA	62.2	42.3	78.2	29.1	64.2
SA+ $\mathcal{Z}$	66.2	46.8	83.8	32.5	68.6
SA+ $\mathcal{Z}$ + L_s	74.9	55.7	93.0	39.0	77.6

three pie charts in FIGURE 6.6); and the relevance that whether the generated captions match with the images (the bottom three pie charts in FIGURE 6.6).

FIGURE 6.6: Each pie chart shows the comparison of two methods in human evaluation.

6.3.2.3 Empirical Answers

A1: From TABLE 6.3, we find that each distinguishable module prefers to generate more corresponding module-specific words, *e.g.*, among three single-module based models, Module/O has the highest recall of nouns. Moreover, in TABLE 6.4, we find that even when a simple module controller (the baseline LSTM) is applied to predict the module layout, the accuracy is still much better than randomly choosing (which gives rise to 25% average accuracy)³. Both observations validate that *by using diverse inductive biases to design four modules, they can indeed learn module-specific knowledge for generating more corresponding words and for the controller to predict better module layout.*

A2: As shown in TABLE 6.2, when we fuse the modules by three different strategies, the performances are all improved compared with the single-module baselines,

³Intuitively, by randomly choosing, at each time, the probability of choosing the right one from 4 modules is $1/4 = 25\%$.

FIGURE 6.7: Visualizations of the caption generation processes of three models: Module/O, SA+Z+L_s, and CVLNM. Note that CVLNM is equal to incorporating REASON module into the baseline SA+Z+L_s. Different colours refer to different modules, *i.e.*, red for OBJECT module, blue for ATTRIBUTE module, purple for RELATION module, and black for FUNCTION module. Since Module/O is the single-module based model, we directly use black boxes to show the attended regions. For simplicity, we only show the module with the largest soft module collocation weight (Eq. (6.11)) and the corresponding region with the largest attention weight (Eq. (6.8)).

e.g., Col/1 is better than Module/R, which answers the first part of Q2 that *fusing modules will generate better captions*.

To answer the second part of Q2, we first observe that in TABLE 6.2, both hard selection (Col/H) and soft fusion (Col/S) strategies achieve better performances than equally exploiting the modules (Col/1), which suggests that *the modules should be discriminatively fused*. Furthermore, we find that Col/S outperforms Col/H. One possible reason is that the module layout is parsed from the partially generated

caption, which is not perfect enough for high quality captions and the complement of the other modules is necessary.

A3: In TABLE 6.2, we can find that by separately inputting previous module layout $\mathcal{Z}$ into the controller (SA+ $\mathcal{Z}$ vs. SA or LSTM+ $\mathcal{Z}$ vs. LSTM) and building such controller based on self-attention network (SA vs. LSTM or SA+ $\mathcal{Z}$ vs. LSTM+ $\mathcal{Z}$), the performances are respectively improved. Such comparisons prove that both of two refinements improve the quality of the generated captions. More importantly, from TABLE 6.3 and 6.4, we find that the recall of each specific word and the module layout accuracy are both improved when self-attention is applied (SA vs LSTM) and when $\mathcal{Z}$ is used (SA+ $\mathcal{Z}$ vs SA). All of these observations confirm that compared with the module controller in our preliminary work [178] which corresponds to the baseline LSTM, *applying self-attention and inputting $\mathcal{Z}$ will enhance the effectiveness of the module controller, which then gives rise to more distinguishable modules and better captions.*

A4: By comparing SA+ $\mathcal{Z}$ + L_s with SA+ $\mathcal{Z}$ in Tables 6.2, 6.3, and 6.4, we find that SA+ $\mathcal{Z}$ + L_s achieves better results in terms of the similarity metrics, the bias degree, the recall of each type of words, and the module layout accuracy. Among these observations, the improvements of the word recalls and the module accuracy validate that *the syntax loss indeed encourages each module to learn more distinguishable knowledge*, and higher CIDEr scores and lower bias metrics suggest that *more distinguishable modules lead to better and less biased captions.*

A5: In TABLE 6.2 and Fig 6.6, we respectively observe that when REASON module is applied, the similarity scores are improved (SA+ $\mathcal{Z}$ +REASON vs. SA+ $\mathcal{Z}$) and the captions are considered better by humans (CVLNM vs. SA+ $\mathcal{Z}$ + L_s), especially in terms of language descriptiveness, which suggest that REASON module improves the quality of the generated captions by approximating human-like commonsense reasoning. Also, as shown in Figure 6.7 (a), we observe that in CVLNM, the word “rain” is generated by the region covering “umbrella” and “wet floor”, while when REASON module is not used, the word “rain” is not generated in SA+ $\mathcal{Z}$ + L_s .

A6: In Tables 6.2, 6.3, 6.4 and FIGURE 6.6, we find that by sequentially incorporating the module controller, the syntax loss, and REASON module into the pipeline, more distinguishable modules can be learned, lower bias degree can be achieved, and better captions are generated. FIGURE 6.7 gives two examples to

TABLE 6.5: The performances of various methods on MS-COCO Karpathy split trained by cross-entropy loss only (left part) and the mixture of cross-entropy loss and self-critical reward (right part).

Loss	Cross-Entropy					Cross-Entropy & Self-Critical				
Models	B@4	M	R	C	S	B@4	M	R	C	S
SCST [29]	30.0	25.9	53.4	99.4	—	34.2	26.7	55.7	114.0	—
StackCap [31]	35.2	26.5	—	109.1	—	36.1	27.4	—	120.4	—
NBT [32]	34.7	27.1	—	108.9	20.1	—	—	—	—	—
Up-Down [1]	36.2	27.0	56.4	113.5	20.3	36.3	27.7	56.9	120.1	21.4
RFNet [167]	37.0	27.9	57.3	116.3	20.8	37.9	28.3	58.3	125.7	21.7
GCN-LSTM [2]	36.8	27.9	57.0	116.3	20.9	38.2	28.5	58.3	127.6	22.0
CAVP [46]	—	—	—	—	—	38.6	28.3	58.5	126.3	21.6
LBPF [39]	37.4	28.1	57.5	116.4	21.2	38.3	28.5	58.4	127.6	22.0
SGAE [3]	36.9	27.7	57.2	116.7	20.9	38.4	28.4	58.6	127.8	22.1
CNM [178]	37.1	27.9	57.3	116.6	20.8	38.9	28.4	58.8	127.9	22.0
CVLNM	37.3	28.2	57.6	117.1	21.2	39.4	28.7	59.1	129.5	22.2

TABLE 6.6: The performances of various compared image captioners on the online MS-COCO test server.

Model	B@3		B@4		M		R-L		C-D	
Metric	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
SCST [29]	47.0	75.9	35.2	64.5	27.0	35.5	56.3	70.7	114.7	116.0
LSTM-A [40]	47.6	76.5	35.6	65.2	27.0	35.4	56.4	70.5	116.0	118.0
StackCap [31]	46.8	76.0	34.9	64.6	27.0	35.6	56.2	70.6	114.8	118.3
Up-Down [1]	49.1	79.4	36.9	68.5	27.6	36.7	57.1	72.4	117.9	120.5
RFNet [167]	50.1	80.4	38.0	69.2	28.2	37.2	58.2	73.1	122.9	125.1
CAVP [177]	50.0	79.7	37.9	69.0	28.1	37.0	58.2	73.1	121.6	123.8
SGAE [3]	50.1	79.6	37.8	68.7	28.1	37.0	58.2	73.1	122.7	125.5
CNM [178]	50.2	79.8	38.4	69.3	28.2	37.2	58.4	73.4	123.8	126.0
CVLNM	50.5	80.4	38.5	70.0	28.7	38.0	58.5	73.7	125.2	127.8

show how CVLNM collocates the modules to generate the captions, which also demonstrates that our CVLNM has better interpretability than the single module based captioner.

6.3.3 Comparisons with the State-of-The-Art Models

Comparing Methods. We compared our CVLNM with the following state-of-the-art image captioners: **SCST** [29], **Up-Down** [1], **NBT** [32], **CAVP** [46], **RFNet** [167], **LBPF** [39], **GCN-LSTM** [2], and **SGAE** [3]. Among these methods, NBT, CAVP, GCN-LSTM, and Up-Down can be considered as simple versions of our CVLNM where NBT only uses an OBJECT module, CAVP and GCN-LSTM apply different RELATION modules, and Up-Down integrates OBJECT and

TABLE 6.7: The CIDEr-D deterioration of using fewer training sentences. The values in the bracket are the real CIDEr-D scores.

X	5	4	3	2	1
Module-O&X	0(123.1)	0.9(122.2)	2.3(120.8)	4.1(119.0)	6.8(116.3)
SA+Z&X	0(127.1)	0.5(126.6)	1.4(125.7)	3.0(124.1)	4.5(122.6)
SA+Z+L _s &X	0(128.4)	0.3 (128.1)	1.0 (127.4)	2.1 (126.3)	3.6 (124.8)

ATTRIBUTE modules. RFNet uses various CNN architectures to extract visual features, which owns a wider encoder compared with some other state-of-the-art methods, and its comparison with our CVLNM is to prove that only a wider encoder is not enough for better captions. SGAE not only exploits the additional annotations, *e.g.*, object, attribute, and relation labels to form visual scene graphs, but also uses Graph Neural Network (GNN) to embed such visual scene graphs to transfer linguistic inductive bias from pure language domain to vision-language domain for more descriptive captions. In CVLNM, REASON module has the similar function which transfers linguistic inductive bias. Thus, the comparison with SGAE is to show that by REASON module, our CVLNM can still outperform SGAE even though we did not use visual scene graphs and GNN. Moreover, we compared our CVLNM with our preliminary version CNM [178] to show that by a more advanced module controller and REASON module, better captions can be generated.

Results and Analysis. The left and right parts of TABLE 6.5 report the performances of various methods trained by cross-entropy loss and the mixture of cross-entropy loss and self-critical reward, respectively. From both parts, we can see that CVLNM outperforms the other state-of-the-art models and it achieves a new state-of-the-art 129.5 CIDEr-D score when the mixture training loss is applied. Specifically, by integrating four distinguishable feature extraction modules, an advanced module controller, the syntax loss, and REASON module, our CVLNM significantly outperforms the single-module models, *e.g.*, NBT, CAVP, and GCN-LSTM; the model with wider encoder, *e.g.*, RFNet; and GNN and scene graph based model, *e.g.*, SGAE. CVLNM outperforms CNM is due to the more advanced module controller and REASON module. Moreover, we submitted our CVLNM to the online server and the scores are reported in TABLE 6.6, which shows that our CVLNM has competitive performances and can achieve the highest CIDEr-D c40 score of 127.8, compared with the other models.

6.3.4 Few-Shot Image Captioning

Comparing Methods. We designed a few-shot image captioning to further validate the robustness of our CVLNM which follows the principle of modular design. In this setting, we randomly sampled X captions from all the annotated captions of one image to train our models. To avoid confusion with the baselines in Section 6.3.2, we add $\&X$ to each baseline to denote that only X captions of an image were provided for training, *e.g.*, **Module-O $\&2$** means 2 captions of each image were sampled to train Module-O. We compared the following ablated baselines to see how each component affects the robustness: 1) **Module-O $\&X$** to test the robustness of only one module, 2) **SA+ $\mathcal{Z}$ $\&X$** to test the soft fusion, and 3) **SA+ $\mathcal{Z}$ + L_s $\&X$** to test the syntax loss. Noteworthy, for fair comparisons with the baseline Module-O $\&X$, we did not pre-train ATTRIBUTE module with attribute labels and did not use REASON module in SA+ $\mathcal{Z}$ $\&X$ and SA+ $\mathcal{Z}$ + L_s $\&X$.

The results are reported in TABLE 6.7, where the values in the table are the deterioration of CIDEr-D scores compared with the model trained by all sentences (5 sentences), and the values in the bracket are the real CIDEr-D scores of the models.

Results and Analysis. From TABLE 6.7, we can find that all the models achieve lower performances when fewer training sentences are provided. Interestingly, we observe that our SA+ $\mathcal{Z}$ + L_s $\&X$ can almost halve the CIDEr-D deterioration compared to Module/O $\&X$. For example, when only 1 caption is available, the deterioration of SA+ $\mathcal{Z}$ + L_s is 3.6, while Module-O is 6.8. Also, it can be found that when we apply the soft fusion strategy and impose the syntax loss to further regularize the learning, the deterioration caused by fewer training samples becomes less severe. Such observations suggest that *by decomposing the captioning into a series of sub-tasks solved by distinguishable modules and by applying suitable fusion and learning strategies, the whole system becomes more robust.*

6.4 Conclusions

We have proposed to follow the crux of our human brain — the principle of modular design — for captioning. In particular, we presented a novel modular network method: learning to Collocate Visual-Linguistic Neural Modules (CVLNM),

which can generate captions by filling the contents into the collocated patterns. In CVLNM, we designed four modules with diverse inductive biases in encoder to extract features and one REASON module in decoder to approximate commonsense reasoning. A self-attention based module controller was designed to dynamically collocate the feature extraction modules for caption generation since only partially observed caption is available. A syntax based loss was imposed on the module controller to further guarantee each feature extraction module to learn module-specific knowledge as well as encouraging the controller to learn human-like sentence patterns. We validated the effectiveness and robustness of our CVLNM by extensive ablations and comparisons with the state-of-the-art models on MS-COCO. The experiment results substantially demonstrated that our CVLNM not only generates more human-like captions, but also suffers less deterioration when fewer captions are provided for training.

Chapter 7

Deconfounded Image Captioning: A Causal Retrospect ¹

7.1 Introduction

Recently, our vision-language community has paid more and more attention to dataset bias. On one hand, it shows that we have already achieved impressive or even super-human performances on many benchmark leader-boards. On the other hand, it also shows that we do not actually believe in these “super-human” systems due to the obvious dataset bias, *e.g.*, image captioners are likely to imply gender discrimination [34] (*e.g.*, generating “a man” instead of “a woman” even through the image actually shows “a woman is snowboarding”) and hallucinate what they do not see [185] (*e.g.*, generating “sitting on the bench” when only seeing people talking on the phone); VQA agents usually answer “Yes” when being asked “Are the apples red?” even without a look at the image [194].

The ideal solution to eliminate dataset bias is to collect a perfect and totally balanced dataset. However, since humans are living in a world filled with various biases, *e.g.*, long-tailed concept distributions [195, 196], reporting bias [197, 198], and language bias [199, 200], we are easily trapped in the loop: “making a dataset” – “it’s biased” – “making a new one” ². Promisingly, by observing that our biological

¹This chapter is submitted to TPAMI as **Xu Yang**, Hanwang Zhang, Jianfei Cai. “Deconfounded Image Captioning: A Causal Retrospect.”

²by Alexei Efros@CVPR2019 Computer Vision After 5 Years workshop (<https://futurecv.github.io/schedule.html>)

FIGURE 7.1: The toy example about how the confounder disturbs the learning of the true causal effect, where $\mathcal{X}$, $\mathcal{L}$, and $\mathcal{P}$ denote image features, label, and position, respectively. (a) A training dataset constructed with a bias that “0” is mostly on the left and “1” is mostly on the right. (b) If we train a digit classifier by the training set in (a), it may not capture the true causal effect $\mathcal{X} \rightarrow \mathcal{L}$, while exploiting the “position knowledge” to make predictions (*e.g.*, when recognizing pixels on the left, predicting the label as “0”) through the backdoor path $\mathcal{X} \leftarrow \mathcal{P} \rightarrow \mathcal{L}$ induced by the confounder $\mathcal{P}$. (c) As a result, if we feed this classifier with the images where “1” appears on the left and “0” on the right, the incorrect predictions will be made.

vision-language system works well even though the around real word is biased, we are prompted to evolve from diagnosis to treatment, by changing from collecting new datasets [13, 186, 194, 201] to designing unbiased models [34, 202] for confronting the bias. However, recent studies still lack a principled analysis about why a vision-language learning system easily collapses into dataset bias. In this chapter, we exploit the concepts of **confounder** to analysis this problem, and propose a principled solution to alleviate the negative effects brought by dataset bias based on the two principles of causal inference: **the backdoor and front-door adjustments** [35, 203, 204]. In particular, we use Image Captioning (IC) as the case study because it has the longest history (since Show&Tell [20] in early deep learning era) and the simplest cross-modal objective among all the vision-language tasks [11, 12, 20, 134, 205].

In the classic causal inference framework [35, 203, 204], if a variable is a common cause for the other two variables, such variable is called the confounder. One toy example about confounder in machine learning is given in Fig. 7.1, where the training dataset is constructed with a bias, *i.e.*, “0” mostly appears on the left and “1” mostly appears on the right. Such biased dataset entails two causal effects: the position $\mathcal{P}$ affects the image features $\mathcal{X}$ and $\mathcal{P}$ affects the digit label $\mathcal{L}$. For example, the left part of an image is likely to contain pixels configuring “0” and if the left part of an image contains black pixels, the label is likely to be “0”. In

FIGURE 7.2: Three examples show how the confounder $\mathcal{D}$ induces the spurious correlation. The probability denotes the percentage of co-occurrence of two words in the training set, *e.g.*, $P(\text{person}|ID_{Remote})$ means that “person” and “remote” contributes the 54% occurrences of “remote”. Here we use ID to signify the appearing visual concept. The top caption is generated by Up-Down [1] and the bottom one is generated by our UD-DICv1.0. Red and blue denote the wrong and right words, respectively, and green denotes the concept which may cause the wrong word.

the causal inference, we can draw two causal links to describe these causal effects: $\mathcal{P} \rightarrow \mathcal{X}$ and $\mathcal{P} \rightarrow \mathcal{L}$, where the tail is the cause of the head. If we want to build a digit classifier to learn the true causal effect $\mathcal{X} \rightarrow \mathcal{L}$ while employing the biased dataset to train this classifier (Fig. 7.1(a)), this classifier may simply recognize which part of the image contains more black pixels, *i.e.*, through $\mathcal{X} \leftarrow \mathcal{P}^3$, and then use this position knowledge to classify the label, *i.e.*, through $\mathcal{P} \rightarrow \mathcal{L}$. In this way, this classifier learns the **spurious correlation** between $\mathcal{X}$ and $\mathcal{L}$ through the **backdoor path** $\mathcal{X} \leftarrow \mathcal{P} \rightarrow \mathcal{L}$ induced by the confounder $\mathcal{P}$ as in Fig. 7.1(b). As a result, this classifier does not really exploit the digit shapes to predict the labels. During testing, if we provide some images where “1” appears on left or “0” appears on right as in Fig. 7.1(c), this classifier will make incorrect predictions. This is how the confounder disturbs the learning of the true causal effect.

Extending this toy example to the IC case, with a similar basic causal graph shown in Fig. 7.2, we denote $\mathcal{D}$ as a confounder due to dataset biases, $\mathcal{X}$ as image features to be learned and $\mathcal{L}$ as the resultant caption. In particular, we can treat the set of some frequently appearing concepts $\mathcal{D}$ in the training dataset as the confounder. On one hand, the more images of these concepts that a network is trained with, the more largely the learned representation $\mathcal{X}$ is affected. Thus, we have the causal effect $\mathcal{D} \rightarrow \mathcal{X}$. On the other hand, these concepts affect the frequency of some related words in the captions $\mathcal{L}$, *e.g.*, the most preferred concept ID_{Person} in MS COCO largely increases the frequency of the words “woman”, “man”, and some human related actions “sit”, “stand”. Thus, these concepts affect the captions

³It is noteworthy that the link direction only indexes the causal direction and the information can also pass from the head to the tail, *e.g.*, recognizing pixels from $\mathcal{X}$ to $\mathcal{P}$.

and we have $\mathcal{D} \rightarrow \mathcal{L}$. When we use the dataset to train a captioner, although we hope to learn the true causal effect $\mathcal{X} \rightarrow \mathcal{L}$, the captioner is likely to learn some spurious correlation between $\mathcal{X}$ and $\mathcal{L}$ due to the confounder $\mathcal{D}$, *i.e.*, over-exploiting the co-occurrence between the recognized concepts and some words to generate the biased captions. For example, as in Fig. 7.2 (c), after recognizing the concept ID_{Board} through $\mathcal{X} \leftarrow \mathcal{D}$, the captioner will exploit the higher co-occurrence between ID_{Board} with “man” through $\mathcal{D} \rightarrow \mathcal{L}$ to generate a wrong word.

So far, we have shown that $\mathcal{D}$ is a confounder and in Section 7.2.1 we will further show that if the **observational probability** $P(\mathcal{L}|\mathcal{X})$ is used as the training target, the captioner is likely to learn the spurious correlation. The rest is how to deconfound it — *Deconfounded Image Captioning* (**DIC**) — which is our goal in this chapter. A principled solution is to apply a new **interventional probability** as the training target: $P(\mathcal{L}|do(\mathcal{X}))$, which is fundamentally different from $P(\mathcal{L}|\mathcal{X})$ due to the **do-operator** $do(\cdot)$ [35, 203]. Intuitively, the do-operator cuts off the link $\mathcal{X} \leftarrow \mathcal{D}$ and then the backdoor path: $\mathcal{X} \leftarrow \mathcal{D} \rightarrow \mathcal{L}$ is blocked and the spurious correlation is eliminated. As a result, $\mathcal{X}$ and $\mathcal{L}$ are deconfounded and the captioner can learn the true causal effect $\mathcal{X} \rightarrow \mathcal{L}$. In Section 7.2.1 and 7.2.2, we will introduce two fundamental techniques to calculate $P(\mathcal{L}|do(\mathcal{X}))$, which are **the backdoor and front-door adjustments** [203, 204], respectively.

In addition to the contribution of introducing causal inference to analyze the dataset bias problem in IC, our technical contribution of this chapter is to propose a novel DIC framework called **DICv1.0** (see Section 7.4) based on both the backdoor and front-door adjustments. We apply it in two prevailing models: Up-Down [1] and AoANet [206], and help both of them boost the CIDEr-D scores from 126.4 to 129.5 and from 128.7 to 131.1, respectively, where the latter one is submitted to the MS COCO Caption test server and achieves a 128.4 CIDEr c40. Furthermore, another contribution is that we retrospect the major captioners in the 6-year-old IC community from the causal view (see Section 7.3). Interestingly, we will show how this causal retrospect enlightens us to develop our DICv1.0 as well as pointing us to a new future.

7.2 Preliminaries: Deconfounding

We hope to train an image captioner to learn the true causal effect $\mathcal{X} \rightarrow \mathcal{L}$: the captioner should reason the caption $\mathcal{L}$ from the image features $\mathcal{X}$ instead of exploiting the spurious correlations induced by the confounder $\mathcal{D}$. In this section, we review two main deconfounding techniques which help us achieve this goal: the backdoor and front-door adjustments [35, 203].

7.2.1 The Backdoor Adjustment

We first use the digit classifier example to illustrate that the observational probability $P(\mathcal{L}|\mathcal{X})$ often encourages the classifier to learn spurious correlation. By Bayes' theorem, we have

$$P(\mathcal{L}|\mathcal{X}) = \sum_p P(\mathcal{L}|\mathcal{X}, p)P(p|\mathcal{X}), \quad (7.1)$$

where p denotes the position of the black pixels, which is either “left” or “right”. Now assuming that $\mathcal{X}_0$ denotes the feature of an image containing “0” with label $\mathcal{L}_0$ and, by Eq. (7.1), we obtain:

$$P(\mathcal{L}_0|\mathcal{X}_0) = P(\mathcal{L}_0|\mathcal{X}_0, left)P(left|\mathcal{X}_0) + P(\mathcal{L}_0|\mathcal{X}_0, right)P(right|\mathcal{X}_0). \quad (7.2)$$

Due to the position bias in the dataset, we have $P(left|\mathcal{X}_0) \gg P(right|\mathcal{X}_0)$ and thus

$$P(\mathcal{L}_0|\mathcal{X}_0) \approx P(\mathcal{L}_0|\mathcal{X}_0, left)P(left|\mathcal{X}_0) \approx P(\mathcal{L}_0|left)P(left|\mathcal{X}_0). \quad (7.3)$$

We have the second approximation in Eq. (7.3) because $\mathcal{X}_0$ is learnable and the connection between “left” and $\mathcal{L}_0$ is strong. Maximizing $P(\mathcal{L}_0|\mathcal{X}_0)$ over the training set are likely to encourage $\mathcal{X}_0$ to converge to the position feature, since recognizing the position is easier than recognizing the digit shape, which is also referred as short-cut learning [207]. In this way, the classifier learns the spurious correlation induced by the backdoor path $\mathcal{X} \leftarrow \mathcal{P} \rightarrow \mathcal{L}$.

The backdoor adjustment calculates the interventional distribution $P(\mathcal{L}|do(\mathcal{X}))$ by revising the observational distribution in Eq. (7.1) as:

$$P(\mathcal{L}|do(\mathcal{X})) = \sum_p P(\mathcal{L}|\mathcal{X}, p)P(p). \quad (7.4)$$

By comparing Eq. (7.1) and Eq. (7.4), we can find that $P(p|\mathcal{X})$ becomes $P(p)$, which means that the position is not affected by $\mathcal{X}$ while being intervened to be “left” or “right” regardless of $\mathcal{X}$. Assuming $P(p = left) = P(p = right) = 0.5$, for the input $\mathcal{X}_0$, Eq. (7.4) becomes

$$P(\mathcal{L}_0|do(\mathcal{X}_0)) = \frac{1}{2} \sum_p P(\mathcal{L}_0|\mathcal{X}_0, p). \quad (7.5)$$

Although the probability of having $(\mathcal{X}_0, p = right)$ in the training set (say 10% or less) is much smaller than that of $(\mathcal{X}_0, p = left)$, it will not be ignored in Eq. (7.5) and maximizing Eq. (7.5) will encourage $\mathcal{X}_0$ to learn something that is different from the position feature and can lead to $\mathcal{L}_0$.

In the IC case, almost all the modern captioners are trained by the observational distribution $P(\mathcal{L}|\mathcal{X})$, where $\mathcal{X}$ are usually based the features extracted from visual backbones [6]. Various spurious correlations have been observed in the existing captioners as shown in Fig. 7.2. Treating the frequently appearing concepts as the confounder $\mathcal{D}$, we can decompose $P(\mathcal{L}|\mathcal{X})$ as:

$$P(\mathcal{L}|\mathcal{X}) = \sum_d P(\mathcal{L}|\mathcal{X}, d)P(d|\mathcal{X}), \quad (7.6)$$

where d is a concept ID in $\mathcal{D}$, *e.g.*, ID_{Apple} or ID_{Banana} . Since a captioning dataset usually contains more samples about certain concepts than the others, *e.g.*, MS COCO contains 80 frequently appearing objects [10, 151] and thus these objects are the more preferred concepts than the others, especially the attribute or relation concepts like ID_{Green} , ID_{Small} , ID_{Sit} , or ID_{Ride} .

Then during training, it is much easier for the visual encoders to recognize the objects and neglect the objects’ attributes or relations. After recognizing these objects, the captioner might only learn to predict the attributes or relations by their co-occurrences with those preferred concepts instead of really reasoning from the images. For example, if a dataset contains much more “red apple” than “green apple”, after recognizing ID_{Apple} , which means that $P(ID_{\text{Apple}}|\mathcal{X}) \approx 1$, Eq. (7.6) could degrade to $P(\mathcal{L}|ID_{\text{Apple}})$ to make the captioner build strong connection between the colour “red” with the concept ID_{Apple} without seeing the real colour. Fig. 7.2 shows three examples, where $\mathcal{X} \leftarrow \mathcal{D}$ in the causal graph denotes the recognizing of some preferred concepts and $\mathcal{D} \rightarrow \mathcal{L}$ denotes the learning or exploiting of the co-occurrence.

FIGURE 7.3: Compared with the backdoor graph, the front-door graph requires an additional mediator $\mathcal{Z}$ to transmit the knowledge between $\mathcal{X}$ and $\mathcal{L}$. To deconfound this front-door based captioner, we should chain together the two partial causal effects $P(\mathcal{Z} | \text{do}(\mathcal{X}))$ and $P(\mathcal{L} | \text{do}(\mathcal{Z}))$ by blocking two backdoor paths: $\mathcal{X} \leftarrow \mathcal{D} \rightarrow \mathcal{L} \leftarrow \mathcal{Z}$ and $\mathcal{Z} \leftarrow \mathcal{X} \leftarrow \mathcal{D} \rightarrow \mathcal{L}$, respectively. The first one is already blocked due to the collider at $\mathcal{L}$ and the second one can be blocked by cutting off the link $\mathcal{X} \rightarrow \mathcal{Z}$.

To deconfound the image captioning, we can also use the backdoor adjustment as Eq. (7.4) to split $\mathcal{D}$ into different concepts and calculate the average causal effects at these concepts:

$$P(\mathcal{L} | \text{do}(\mathcal{X})) = \sum_d P(\mathcal{L} | \mathcal{X}, d) P(d). \quad (7.7)$$

Then if we train our captioner by maximizing $P(\mathcal{L} | \text{do}(\mathcal{X}))$, the captioner will learn to maximize $P(\mathcal{L} | \mathcal{X}, d)$ at every concept instead of just one ID_{Apple} , *e.g.*, learning $\mathcal{L} = \text{green}$ from ID_{Banana} as well. In this way, the captioner is encouraged to build more connections between $\mathcal{X}$ and $\mathcal{L}$, *i.e.*, reasoning the caption from the image.

7.2.2 The Front-door Adjustment

In Eq. (7.7), we have seen that the backdoor adjustment requires us to split $\mathcal{D}$ into different levels. In other words, we need to know what the confounder is in advance in order to apply the backdoor adjustment. However, in real situations, dataset biases could be very complex and it is hard to know and disentangle different types of confounders. For example, in IC, it is an oversimplification to split $\mathcal{D}$ into individual concepts since dataset biases are not only brought by each single concept independently but more likely by the combinations of different concepts, *e.g.*, relation words are affected by both subject and object concepts, for which a sufficient split requires the combinational complexity. More seriously, since researchers

usually throw away the pre-training dataset and only use the pre-trained model, they have no idea which concepts appeared more frequently. Thus, it is almost impossible for us to get a reasonable split of $\mathcal{D}$ to apply the backdoor adjustment.

Fortunately, the front-door adjustment [203] can also be used to calculate $P(\mathcal{L}|do(\mathcal{X}))$ when we cannot get a reasonable split of $\mathcal{D}$. As shown in Fig. 7.3(a), to apply the front-door adjustment, an additional mediator $\mathcal{Z}$ should be inserted between $\mathcal{X}$ and $\mathcal{L}$ to construct a front-door path $\mathcal{X} \rightarrow \mathcal{Z} \rightarrow \mathcal{L}$ for helping transmit knowledge. Interestingly, it is not a weird idea to introduce the mediator in IC and we will see in Section 7.3 that many advanced image captioners have applied different kinds of mediators, *e.g.*, the prevailing visual attention mechanism [24] treats the position as the mediator.

Given such causal graph with a front-door path, we can use Bayes' theorem to get $P(\mathcal{L}|\mathcal{X})$:

$$P(\mathcal{L}|\mathcal{X}) = \sum_z P(\mathcal{Z} = z|\mathcal{X})P(\mathcal{L}|\mathcal{Z} = z). \quad (7.8)$$

However, this conditional probability is computed from the observational data that the spurious correlation between $\mathcal{X}$ and $\mathcal{L}$ still exists. To calculate the causal effect of $\mathcal{X}$ on $\mathcal{L}$, we can follow the front-door path to sum the causal effects of $\mathcal{X}$ on $\mathcal{Z}$ and $\mathcal{Z}$ on $\mathcal{L}$:

$$P(\mathcal{L}|do(\mathcal{X})) = \sum_z P(\mathcal{Z} = z|do(\mathcal{X}))P(\mathcal{L}|do(\mathcal{Z} = z)). \quad (7.9)$$

To calculate $P(\mathcal{Z}|do(\mathcal{X}))$, as in Fig. 7.3(b), we need to block the backdoor path between $\mathcal{X}$ and $\mathcal{Z}$: $\mathcal{X} \leftarrow \mathcal{D} \rightarrow \mathcal{L} \leftarrow \mathcal{Z}$. Fortunately, due to the collider at $\mathcal{L}$, this backdoor path is already blocked and thus we have:

$$P(\mathcal{Z} = z|do(\mathcal{X})) = P(\mathcal{Z} = z|\mathcal{X}). \quad (7.10)$$

The detail about why the collider at $\mathcal{L}$ blocks this backdoor path is given in Section A.2.3 of the supplementary material. Interestingly, we can see that due to this collider, we do not need to split $\mathcal{D}$ to block this backdoor path and thus we do not need to know the details of $\mathcal{D}$.

To calculate $P(\mathcal{L}|do(\mathcal{Z}))$, as in Fig. 7.3(b), we find that there is a backdoor path $\mathcal{Z} \leftarrow \mathcal{X} \leftarrow \mathcal{D} \rightarrow \mathcal{L}$. To block it, we can choose to cut off the link $\mathcal{Z} \leftarrow \mathcal{X}$ or $\mathcal{X} \leftarrow \mathcal{D}$. Since we do not know the details of $\mathcal{D}$, we have to cut off the link $\mathcal{Z} \leftarrow \mathcal{X}$

FIGURE 7.4: The causal retrospect of the major captioners. The past/present/future are colored by black/blue/red, respectively. We will look ahead the recursive future in Section 7.6. In these causal graphs, the shadowed variables signify that it is hard to split them for applying the backdoor adjustment, *e.g.*, D in (a) and S in (g).

by splitting $\mathcal{X}$ into different levels, *e.g.*, the features of all the images of the training dataset, and average the causal effects at these levels. Similar to Eq. (7.7), we have:

$$P(\mathcal{L}|\text{do}(\mathcal{Z} = z)) = \sum_{\mathbf{x}} P(\mathcal{L}|z, \mathbf{x})P(\mathbf{x}). \quad (7.11)$$

Overall, by applying Eq. (7.10) and (7.11) into Eq. (7.9), we have the front-door adjustment as:

$$P(\mathcal{L}|\text{do}(\mathcal{X})) = \sum_z P(z|\mathcal{X}) \sum_{\mathbf{x}} P(\mathcal{L}|z, \mathbf{x})p(\mathbf{x}). \quad (7.12)$$

To avoid too many variable symbols, we use $\mathcal{X}$ to specify the current image's representation and $\mathbf{x}$ to denote the potential visual feature in the representation space. When we use $P(\mathcal{L}|\text{do}(\mathcal{X}))$ (Eq. (7.12)) as the target to train a front-door based captioner, the induced spurious correlation is also eliminated.

7.3 Related Work: A Causal Retrospect

We follow Fig. 7.4 to retrospect the milestone image captioners proposed in recent six years from the causal view, which provides a novel perspective of understanding their contributions.

Show & Tell [20] (Fig. 7.4(a)). Compared with earlier template-based captioners like Baby Talk [162], Show & Tell is where we start to deploy deep neural network based encoder-decoder for captioning. Also, since Show & Tell, almost all the following captioners are built on the pre-trained model to narrow the gap between vision and language, while also intensifying the “preferred concepts” problem and encouraging the model to learn the spurious correlation.

Large-scale Training (Fig. 7.4(b)). A straightforward way to deconfound is to turn the confounder $\mathcal{D}$ into a non-confounder, *i.e.*, constructing an infinite and balanced dataset. In this way, the captioner can learn every possible visual-semantic pairs from the abundant training samples and as a result, the captioner will no longer collapse into biases. We can approximate this by scaling up the training data. Many studies fall into this large-scale training strategy, *e.g.*, Up-Down [1] exploited Visual Genome [56] with a larger label space, HIP [208] used more fine-grained segmentation annotations, and vision-language BERT frameworks [14, 209] trained their network by 3 million image-caption pairs from Conceptual Captions [13]. After training or pre-training with those large-scale datasets, these captioners’ performances are boosted. However, it is still almost impossible to collect an infinite and perfect dataset where no bias exists since even the real world is essentially unbalanced. In this way, large-scale training still do not deconfound the IC.

Attention Mechanisms (Fig. 7.4(c)). SAT [24] was the first work which applied the attention mechanism in IC, and then this mechanism was used and developed in every follow-up IC systems [1, 25, 27, 40, 46, 206, 210]. By this mechanism, captioners first scan over all the visual concepts of an image and then select suitable ones conditioned on the language context, *e.g.*, spatial attention [1, 24] and semantic attention [25] selected the most informative visual features and semantic labels, to generate the captions. The attention mechanism is an approximation of both the backdoor and front-door adjustments. Compared with the backdoor adjustment (Eq. (7.7)), where $P(\mathcal{L}|do(\mathcal{X}))$ averages over all the visual concepts of $\mathcal{D}$, while the attention mechanism only averages over the appearing visual concepts of the given image, which only construct a small subset of $\mathcal{D}$. Thus $\mathcal{D}$ is not totally deconfounded. From the perspective of the front-door adjustment, the attention mechanism treats the visual position as the mediator $\mathcal{Z}$ and the captioners generate the words from the selected positions at different time steps. However, it uses the observational conditional probability $P(\mathcal{L}|\mathcal{X})$ (Eq. (7.8)) instead of $P(\mathcal{L}|do(\mathcal{X}))$

(Eq. (7.12)) to compute the word distribution, which does not deconfound the confounder. Thus these captioners still easily collapse into dataset bias [185].

Sentence Patterns (Fig. 7.4(d)). Researchers also designed ICs which imitate humans to dynamically structure sentence patterns for captioning. In particular, they used diverse modules for different patterns, *e.g.*, NBT [32] designed two modules respectively for nouns and all the other words, and CNM [178] applied four fine-grained modules for nouns, adjectives, relation related words, and function words. During captioning, sentence patterns are dynamically composed for selecting suitable modules to generate the corresponding words. At first glance, this framework looks like a front-door IC where the mediator $\mathcal{Z}$ is sentence pattern. However, if we look closer, it can be discovered that since both the sentence patterns and the captions are learnt from the language resource of the dataset, both $\mathcal{Z}$ and $\mathcal{L}$ are affected by the concepts $\mathcal{D}$ in the language resource, *e.g.*, the more nouns appear, the more OBJECT module is trained. However, due to this causal link $\mathcal{D} \rightarrow \mathcal{Z}$, the front-door adjustment cannot be used and thus these captioners are still confounded.

Structured Attention (Fig. 7.4(e)). When humans describe an image, they can build a semantic structure (*e.g.*, scene graph) about this image and then turn this structure into the caption. Inspired by this, researchers [2, 3] proposed to learn a scene graph from the image and to apply Graph Neural Network [95–97] to transform this scene graph into a series of embeddings that each embedding corresponds to a sub-structure, *e.g.*, a triplet structure like “subject-predicate-object”. Then the captioners dynamically attend to these sub-structure embeddings to generate the words. These captioners also belong to the front-door causal graph where $\mathcal{Z}$ is the structure or the graph. However, such structures are usually affected by $\mathcal{D}$, *e.g.*, the learning and predicting of the relation is also biased by some preferred concepts [80]. Moreover, the applied scene graph parsers [58] are usually trained by VG [56], while VG is also used to pre-train these captioners. Thus we also have the causal link $\mathcal{D} \rightarrow \mathcal{Z}$ and the front-door adjustment cannot be used. Consequently, it is hard to deconfound such captioners and they are still encouraged to learn the spurious correlation.

FIGURE 7.5: (a) The causal graph of our DICv1.0 captioner, where $\mathcal{Z}$ and $\mathcal{S}$ denote the commonsense structure and the caption vocabulary, respectively. There are two confounders in this model, which are $\mathcal{D}$ and $\mathcal{S}$. (b) The sketch of our DICv1.0's decoder, where ATT/EXPT represent attention/expectation modules. (c) The sketch of EXPT($\mathcal{Z}$) (Eq. (7.20)), where $\otimes$ denotes the matrix multiplication $\bar{z} = Zp$.

7.4 Deconfounded Image Captioning

In this section, we discuss how to derive our DICv1.0 from the causal retrospect in Section 7.3 and how to implement our DICv1.0 into the prevailing encoder-decoder framework.

7.4.1 Choosing $\mathcal{Z}$

As discussed in Section 7.2.2, since $\mathcal{D}$ is very complex and almost unavailable, the backdoor adjustment cannot be deployed, we have to deconfound the IC by the front-door adjustment. To achieve this, the first challenge is the selection of the mediator $\mathcal{Z}$. Based on the retrospect in Section 7.3, we know there are three candidates of $\mathcal{Z}$: spatial position (in attention mechanism), sentence pattern, and some kind of the structure (in structured attention). For spatial position (Fig. 7.4(c)), it only provides the visual concepts of the given image, which is a small subset of $\mathcal{D}$, and thus it is not a good candidate. For sentence pattern (Fig. 7.4(d)), the calculation of its probability is almost impossible since it is decided by the whole caption, which is not available during the caption generation, so sentence pattern is also not a good candidate. For structured attention (Fig. 7.4(e)), it is designed to be decided by the image only, while researchers [2, 3] learn the structure generator from the same dataset as the captioning, which adds the link $\mathcal{D} \rightarrow \mathcal{Z}$ into the causal graph and as the result, this IC is confounded. However, if we only associate the structure with $\mathcal{X}$, we have a causal graph which cuts off the link $\mathcal{D} \rightarrow \mathcal{Z}$ as Fig. 7.4(f) and then we can apply the front-door adjustment.

To achieve this, we sample a commonsense structure set from ConceptNet [192] and treat it as the mediator $\mathcal{Z}$, *e.g.*, “car-at location-road” and “apple-is-red”. Given such a commonsense structure “subject-predicate-object”, we transform it into a semantic embedding $\mathbf{z}$:

$$\mathbf{z} = \text{ReLU}(\text{FC}([\mathbf{e}_s, \mathbf{e}_p, \mathbf{e}_o])), \quad (7.13)$$

where $\mathbf{e}_s, \mathbf{e}_p, \mathbf{e}_o$ denote the trainable word embeddings of the subject, predicate, and object, respectively, and FC denotes the fully connected layer. Compared with the original visual features, these semantic embeddings will naturally be more informative since they also belong to the semantic domain as the captions, which means that this commonsense structure set is a suitable candidate of the mediator. Intuitively, when a captioner uses this commonsense structure as the mediator, this captioner will first use the visual features to retrieve the related semantic embeddings ($\mathcal{X} \rightarrow \mathcal{Z}$) and then exploit the retrieved embeddings to generate the caption ($\mathcal{Z} \rightarrow \mathcal{L}$).

However, when we sample the related commonsense structures, we use the words from the caption vocabulary $\mathcal{S}$ as the keys (see Section 7.5.1), which makes $\mathcal{S}$ as an additional confounder between $\mathcal{Z}$ and $\mathcal{L}$ since the captions are also generated from this vocabulary. In this way, two additional causal links should be added: $\mathcal{S} \rightarrow \mathcal{Z}$ and $\mathcal{S} \rightarrow \mathcal{L}$. Now we have the complete causal graph in Fig. 7.5(a) where two confounders exist: $\mathcal{D}$ and $\mathcal{S}$. To calculate the causal effect $P(\mathcal{L}|do(\mathcal{X}))$, both two confounders should be deconfounded: $\mathcal{D}$ can be deconfounded by the front-door adjustment (see Section 7.2.2) and $\mathcal{S}$ can be deconfounded by the backdoor adjustment (Section 7.2.1) since $\mathcal{S}$ is an observable vocabulary.

This is our **DICv1.0** which deconfounds image captioning. Briefly, we provide the formula of $P(\mathcal{L}|do(\mathcal{X}))$ here and the detail derivations are given in Section B.1 of the supplementary material:

$$\begin{aligned} P(\mathcal{L}|do(\mathcal{X})) &= \sum_{\mathbf{s}} P(\mathbf{s}) \sum_{\mathbf{x}} P(\mathbf{x}) \sum_{\mathbf{z}} P(\mathbf{z}|\mathcal{X}) [P(\mathcal{L}|\mathbf{s}, \mathbf{x}, \mathbf{z})] \\ &= \mathbb{E}_{\mathbf{s}} \mathbb{E}_{\mathbf{x}} \mathbb{E}_{[\mathbf{z}|\mathcal{X}]} [P(\mathcal{L}|\mathbf{s}, \mathbf{x}, \mathbf{z})]. \end{aligned} \quad (7.14)$$

It can be found that this formula is the expected value of $P(\mathcal{L}|\mathbf{s}, \mathbf{x}, \mathbf{z})$ according to three variables $\mathbf{s}$, $\mathbf{x}$, and $\mathbf{z}$, which denote the word embeddings of the key words, the visual features, and the semantic embeddings, respectively. Notice that

$\mathcal{X}$ denotes the current image which we want to generate the caption from and $\mathbf{x}$ denotes the potential visual feature of the whole representation space.

To implement our DICv1.0 into the encoder-decoder framework, we parameterize $p(\mathcal{L}|\mathbf{s}, \mathbf{x}, \mathbf{z})$ as a network. The last layer of this network is a Softmax layer that implements $P(\mathcal{L}|\mathbf{s}, \mathbf{x}, \mathbf{z})$ as:

$$P(\mathcal{L}|\mathbf{s}, \mathbf{x}, \mathbf{z}) = \text{Softmax}[g(\mathbf{s}, \mathbf{x}, \mathbf{z})], \quad (7.15)$$

where $g(\cdot)$ is an embedding layer before the Softmax layer, *e.g.*, it can be an FC layer. However, this brings one challenge that in order to compute $P(\mathcal{L}|do(\mathcal{X}))$ in Eq. (7.14), we need a huge number of outputs sampled from this network to get the expectations of $P(\mathcal{L}|\mathbf{s}, \mathbf{x}, \mathbf{z})$. To solve this challenge, we propose a two-step approximation which allows us to forward the network only once to get an estimation of $P(\mathcal{L}|do(\mathcal{X}))$. The first step is called Normalized Weighted Geometric Mean (NWGM) approximation [24, 211, 212] which absorbs the expectations into the network (see Section 7.4.2). The second step is to sample finite $\mathbf{s}$, $\mathbf{x}$, and $\mathbf{z}$ from the respective representation spaces of $\mathcal{S}$, $\mathcal{X}$, and $\mathcal{Z}$ for estimating the expectations in Eq. (7.14) (see Section 7.4.3).

7.4.2 Normalized Weighted Geometric Mean

By NWGM approximation [24], the expectation of a Softmax unit is approximated as the Softmax of the expectation:

$$\begin{aligned} P(\mathcal{L}|do(\mathcal{X})) &= \mathbb{E}_{\mathbf{s}} \mathbb{E}_{\mathbf{x}} \mathbb{E}_{[\mathbf{z}|\mathcal{X}]} \{\text{Softmax}[g(\mathbf{s}, \mathbf{x}, \mathbf{z})]\} \\ &\approx \text{Softmax}\{\mathbb{E}_{\mathbf{s}} \mathbb{E}_{\mathbf{x}} \mathbb{E}_{[\mathbf{z}|\mathcal{X}]} [g(\mathbf{s}, \mathbf{x}, \mathbf{z})]\}. \end{aligned} \quad (7.16)$$

Furthermore, if $g(\cdot)$ is an FC layer, we have:

$$P(\mathcal{L}|do(\mathcal{X})) \approx \text{Softmax}\{g(\mathbb{E}_{\mathbf{s}}[\mathbf{s}], \mathbb{E}_{\mathbf{x}}[\mathbf{x}], \mathbb{E}_{[\mathbf{z}|\mathcal{X}]}[\mathbf{z}])\}. \quad (7.17)$$

Here we can put the expectations into the FC layer $g(\cdot)$ because the linear projection of the expectation of one variable equals to the expectation of the linear projection of that variable. More details about the derivations of Eq. (7.16) and (7.17) by NWGM approximation are given in Section B.2 of the supplementary material.

7.4.3 EXPT Modules for Expectations

Before this section, we formulate all the probability equations at the sentence level, *i.e.*, $\mathcal{L}$ denotes a whole caption. When we compute the probabilities at word level, for convenience, we still use $\mathcal{L}$ to denote the word and write the word distribution as $P(\mathcal{L}|do(\mathcal{X}), \mathbf{h})$, where $\mathbf{h}$ is a context vector which accumulates the knowledge of the previously generated words. By modifying Eq. (7.14) (see Section B.1 and B.2 of the supplementary material), the word distribution becomes:

$$\begin{aligned} & P(\mathcal{L}|do(\mathcal{X}), \mathbf{h}) \\ &= \sum_s P(\mathbf{s}|\mathbf{h}) \sum_x P(\mathbf{x}|\mathbf{h}) \sum_z P(\mathbf{z}|\mathcal{X}, \mathbf{h}) [P(\mathcal{L}|\mathbf{s}, \mathbf{x}, \mathbf{z}, \mathbf{h})] \\ &= \mathbb{E}_{[\mathbf{s}|\mathbf{h}]} \mathbb{E}_{[\mathbf{x}|\mathbf{h}]} \mathbb{E}_{[\mathbf{z}|\mathcal{X}, \mathbf{h}]} [P(\mathcal{L}|\mathbf{s}, \mathbf{x}, \mathbf{z}, \mathbf{h})]. \end{aligned} \quad (7.18)$$

Similar to Eq. (7.16) and (7.17), we can put the expectations into the Softmax by NWGM approximation:

$$P(\mathcal{L}|do(\mathcal{X}), \mathbf{h}) \approx \text{Softmax}\{g(\mathbb{E}_{[\mathbf{s}|\mathbf{h}]}[\mathbf{s}], \mathbb{E}_{[\mathbf{x}|\mathbf{h}]}[\mathbf{x}], \mathbb{E}_{[\mathbf{z}|\mathcal{X}, \mathbf{h}]}[\mathbf{z}])\}, \quad (7.19)$$

for which we should do next is to calculate: $\mathbb{E}_{[\mathbf{s}|\mathbf{h}]}[\mathbf{s}]$, $\mathbb{E}_{[\mathbf{x}|\mathbf{h}]}[\mathbf{x}]$, and $\mathbb{E}_{[\mathbf{z}|\mathcal{X}, \mathbf{h}]}[\mathbf{z}]$.

Here we define an **expt** module to estimate these expectations. For example, we use $\text{EXPT}(\mathbf{X}, \mathbf{h})$ to estimate $\mathbb{E}_{[\mathbf{x}|\mathbf{h}]}[\mathbf{x}]$:

$$\begin{aligned} \text{Input: } & \mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_K\}, \mathbf{h} \\ \text{Probability: } & q_k = \mathbf{x}_k^T \mathbf{h}, \\ & \mathbf{p} = \{p_1, \dots, p_K\} = \text{Softmax}(\{q_1, \dots, q_K\}) \\ \text{Output: } & \mathbb{E}_{[\mathbf{x}|\mathbf{h}]}[\mathbf{x}] \approx \sum_k p_k \mathbf{x}_k, \end{aligned} \quad (7.20)$$

where $\mathbf{h}$ is the context vector and $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_K\}$ is a dictionary with K elements. The reason we use such $\mathbf{X}$ is that it is costly and time-prohibitive to average over all the possible visual feature $\mathbf{x}$. In the experiments, we used a dictionary learning algorithm [213] to learn the initialization of this $\mathbf{X}$ from the visual features of the whole training dataset. Since all the operations of EXPT module are differentiable, $\mathbf{X}$ can also be updated during the end-to-end training. Similarly, as in Fig. 7.5(c), we use $\text{EXPT}(\mathbf{Z}, \text{Cat}(\hat{\mathbf{x}}, \mathbf{h}))$ to estimate $\mathbb{E}_{[\mathbf{z}|\mathcal{X}, \mathbf{h}]}[\mathbf{z}]$, where $\mathbf{Z}$ is the semantic embedding dictionary that each element $\mathbf{z}$ of it is computed by Eq. (7.13); $\text{Cat}(\cdot)$ denotes the concatenation operation; and $\hat{\mathbf{x}}$ is the attended feature of the image $\mathcal{X}$ at the current time step (see Section 7.4.4). When we

collect the commonsense structure set, we use the nouns, verbs, and adjectives of the caption vocabulary as the key words to search the related structures from ConceptNet. Meanwhile, these key words' trainable embeddings construct the dictionary $\mathbf{S}$, which is used in $\text{EXPT}(\mathbf{S}, \mathbf{h})$ to estimate $\mathbb{E}_{[\mathbf{s}|\mathbf{h}]}[\mathbf{s}]$. The numbers of elements of $\mathbf{X}$, $\mathbf{Z}$, and $\mathbf{S}$ are set to 10,000, 9,590, and 1,342.

7.4.4 Network Architectures

We incorporate our DICv1.0 into two models: Up-Down [1] and AoANet [206] and name them as **UD-DICv1.0** and **AoA-DICv1.0**, respectively. In both models, the visual encoder is a ResNet-101 Faster R-CNN [6] pre-trained on Visual Genome [56] as in Up-Down [1]. The decoders of two models have a similar architecture, which is sketched in Fig. 7.5(b), where the embedding layer represents the function $g(\cdot)$ in Eq. (7.15). The input of this decoder concatenates three terms: the mean pooling of the visual features of the image $\mathcal{X}$, the previous generated word's embedding, and the previous embedding layer's output. $\text{ATT}(\mathcal{X})$ represents an attention module which computes the attended vector $\hat{\mathbf{x}}$ from the visual feature $\mathcal{X}$. This $\hat{\mathbf{x}}$ is input into EXPT module ($\text{EXPT}(\mathbf{Z}, \text{Cat}(\hat{\mathbf{x}}, \mathbf{h}))$) to estimate $\mathbb{E}_{[\mathbf{z}|\mathcal{X}, \mathbf{h}]}[\mathbf{z}]$ and into the embedding layer since it is also a sample of the visual features. $\bar{\mathbf{z}}$, $\bar{\mathbf{s}}$, and $\bar{\mathbf{x}}$ denote the estimated vectors of $\mathbb{E}_{[\mathbf{z}|\mathcal{X}, \mathbf{h}]}[\mathbf{z}]$, $\mathbb{E}_{[\mathbf{s}|\mathbf{h}]}[\mathbf{s}]$, and $\mathbb{E}_{[\mathbf{x}|\mathbf{h}]}[\mathbf{x}]$ by the corresponding EXPT modules, respectively, and all of them are input into the embedding layer for computing the word distributions.

When UD-DICv1.0 or AoA-DICv1.0 is deployed, $\text{ATT}(\mathcal{X})$ is Top-Down Attention or Attention on Attention; the embedding layer is an LSTM or a GLU [214], respectively. Notice that although the embedding layer is not an FC layer, we still observe less bias and better performances compared with the original models (see Section 7.5.2.1). The details of the network architectures of UD-DICv1.0 and AoA-DICv1.0, *e.g.*, the architectures of ATT module (Top-Down Attention and Attention on Attention) and EXPT module and the size of each layer in these modules, are given in Section C of the supplementary material.

7.4.5 Training Objectives

When training both models, we minimize the cross-entropy loss by using $P(\mathcal{L}^*|do(\mathcal{X}))$ as the target in the first 35 epochs:

$$L_{XE} = -\log P(\mathcal{L}^*|do(\mathcal{X})), \quad (7.21)$$

where $\mathcal{L}^*$ denotes the ground-truth caption. After that, we minimize the RL-based loss [29] to train both models for another 65 epochs:

$$L_{RL} = -\mathbb{E}_{\mathcal{L}^s \sim P(\mathcal{L}|do(\mathcal{X}))} [r(\mathcal{L}^s; \mathcal{L}^*)], \quad (7.22)$$

where r is a sentence-level metric between the sampled sentence $\mathcal{L}^s$ and the ground-truth $\mathcal{L}^*$, *e.g.*, the CIDEr-D metric [166]. We use Adam optimizer [155] to train both models. The learning rate for both models is initialized to $5e^{-4}$ and is decayed by 0.8 for every 5 epochs. Importantly, the learning rate of all the dictionaries in EXPT modules, *e.g.*, $\mathbf{X}$, $\mathbf{Z}$, and $\mathbf{S}$, are set 10 times smaller than the other layers, *e.g.*, $5e^{-5}$. The batch sizes for UD-DICv1.0 and AoA-DICv1.0 are set to 100 and 10, respectively. When inference, we use beam search with a beam size of 5.

7.5 Experiments

7.5.1 Datasets

MS COCO [10]. We validated our models on MS COCO IC dataset. In particular, our models were tested on two different splits: Karpathy split [172] and the official online test split, which divide the whole dataset into 113,287/5,000/5,000 and 82,783/40,504/40,775 images for training/validation/test, respectively. We followed previous researches to pre-process our captions [1, 3]. At last we trimmed each caption to a maximum of 16 words and had a vocabulary of 10,369 words by removing the words which appear less than 5 times.

ConceptNet [192]. ConceptNet has abundant commonsense structures which have the form of “subject-predicate-object” and each of them is assigned with an importance weight. We used the nouns, verbs, and adjectives which appear more than 20 times in MS COCO captioning dataset as the key words to collect the related structures and all these 1,342 key words were used to construct the

dictionary $\mathcal{S}$. For each commonsense structure in ConceptNet, if its importance weight is larger than 2.5 and if both the subject and object of this structure are our key words, we collected it. Finally, we had a semantic structure set with 9,590 structures and they were used to build the dictionary $\mathcal{Z}$.

7.5.2 Ablation Studies

7.5.2.1 Deconfounding Techniques

We used Up-Down as the backbone, whose architecture is the same as Fig. 7.5 (b) but without the bottom three parallel LSTM layers and all the EXPT modules, and gradually incorporated new components into this backbone to design various ablation studies for validating the importance of different deconfounding techniques: the backdoor adjustment (Section 7.2.1), the front-door adjustment (Section 7.2.2), and our DICv1.0 (Section 7.4.4).

Comparing Methods.

UD: We re-implemented Up-Down [1] as our baseline, where only $\text{ATT}(\mathcal{X})$ exists in the decoder.

UD-BD: Compared with UD, we followed the backdoor adjustment (Eq. (7.7)) and estimated $\mathbb{E}_{[d|h]}[d]$ by $\text{EXPT}(\mathbf{D}, \mathbf{h})$ (Eq. (7.20)). We grouped the word embeddings of 80 visual concepts in MS COCO as the dictionary $\mathbf{D}$ since they are the most frequently appearing objects in the images and captions, which are likely to induce the spurious correlation. This baseline was designed to confirm the utility of the backdoor adjustment.

UD-FD/Cor: Compared with UD, we added $\text{EXPT}(\mathbf{Z}, \text{Cat}(\hat{\mathbf{x}}, \mathbf{h}))$ into the decoder. We can treat this as training a front-door IC by the observational correlation $P(\mathcal{L}|\mathcal{X})$ (Eq. (7.8)) and thus the spurious correlation still exists.

UD-FD: Compared with UD-FD/Cor, we added $\text{EXPT}(\mathbf{X}, \mathbf{h})$ into the decoder. This can be treated as training a front-door IC by the NWGM approximation of $P(\mathcal{L}|do(\mathcal{X}))$ (Eq. (7.12)) while neglecting the confounder $\mathcal{S}$. This baseline was used to confirm the utility of the front-door adjustment.

TABLE 7.1: The performances of various ablation studies on MS COCO Karpathy split. The metrics: B@4, M, R, C, S, CHs, CHi, A@Gen, A@Attr, and A@Act denote BLEU@4, METEOR, ROUGE-L, CIDEr-D, SPICE, CHAIRs, CHAIRi, the accuracy of gender, attribute, and action words. The symbols $\uparrow$ and $\downarrow$ mean the higher the better and the lower the better, respectively.

Models	B@4 $\uparrow$	M $\uparrow$	R $\uparrow$	C $\uparrow$	S $\uparrow$	CHs $\downarrow$	CHi $\downarrow$	A@Gen $\uparrow$	A@Attr $\uparrow$	A@Act $\uparrow$
UD	37.2	27.5	57.3	125.3	21.1	13.7	8.9	0.81	0.41	0.52
UD-BD	38.2	28.2	58.0	126.9	21.3	11.2	7.6	0.87	0.50	0.56
UD-FD/Cor	38.0	28.1	58.0	126.5	21.1	12.3	8.3	0.83	0.46	0.54
UD-FD	38.5	28.4	58.7	127.6	21.8	10.5	7.0	0.89	0.55	0.58
UD-DICv1.0	39.0	28.8	58.8	128.8	22.0	10.1	6.5	0.91	0.58	0.60

UD-DICv1.0: Compared with UD-FD, we added $\text{EXPT}(\mathcal{S}, \mathbf{h})$ into the decoder to get the integral framework of our DICv1.0 as in Fig. 7.5. This equals to train a front-door IC by using the NWGM approximation of Eq. (7.14), which deconfounds both two confounders: $\mathcal{D}$ and $\mathcal{S}$.

Analysis of the Similarities. We followed previous studies to use the following five metrics: CIDEr-D [166], BLEU [174], METEOR[175], ROUGE [176], and SPICE [84] to measure the similarities between the generated captions with the ground-truth captions. The similarity metrics of our UD-DICv1.0 and the baselines are reported in Table 7.1. Since the ground-truth captions are annotated by us humans and we are usually good at grasping the causal effect, similarity metrics are naturally suitable to evaluate whether a captioner grasps such causal effect.

Compared with the original UD, UD-DICv1.0 improves CIDEr-D from 125.3 to 128.8, which means that UD-DICv1.0 generates the most similar captions with the ground-truth. By comparing UD-BD with UD, we observe that UD-BD achieves a higher CIDEr-D score, which confirms the utility of the backdoor adjustment even in this baseline we only use 80 visual concepts as the split of $\mathcal{D}$, which is a simplification of the real situation. Also, due to this simplification, the effectiveness of the backdoor adjustment is limited and the front-door adjustment can achieve better performances, *i.e.*, UD-FD outperforms UD-BD, which suggests that the front-door adjustment is a better choice when the confounder $\mathcal{D}$ is very complex. By sequentially comparing UD-FD/Cor, UD-FD, and UD-DICv1.0, we can observe that the performances are also sequentially improved, which confirms the effectiveness of deconfounding $\mathcal{D}$ (UD-FD vs. UD-FD/Cor) and $\mathcal{S}$ (UD-DICv1.0 vs. UD-FD). Furthermore, such comparisons also suggest that incorporating a mediator like the commonsense knowledge and exploiting additional resource like ConceptNet

FIGURE 7.6: The accuracy of the gender, attribute, and action words when some specific visual concepts appear.

FIGURE 7.7: The pie charts each comparing two ICs.

are not enough for generating high-quality captions unless we discover the hidden confounders and use the backdoor and front-door adjustments to deconfound them.

Analysis of the Biases. To measure the bias degree of the generated captions, we not only applied CHs&CHi [185] to measure the object bias but also analyzed three more specific biases: gender bias, action bias, and attribute bias. We evaluated these biases by calculating the accuracy of these words when a specific visual concept appears in the ground-truth captions. Take the gender bias as an example, we calculated whether the gender word (*e.g.*, “she”, “he”, “girl”, or “boy”) is consistent between the ground-truth caption and the generated caption when a visual concept, *e.g.*, skateboard, appears. We averaged the accuracy of the gender word over 80 visual concepts and report the mean accuracy in Table 7.1. It is noteworthy that we considered the balance of the words that we separately calculated the accuracy of each word and averaged the results to obtain the mean accuracy.

FIGURE 7.8: Eight examples about gender, object, attribute, and action biases in captions generated by different models. “GT” denotes the ground truth caption. The visual concepts which may cause biases are colored by green. The red and blue words represent the inconsistent and consistent words, respectively. It can be found that our UD-DICv1.0 can generate the most consistent captions compared with UD and UD-BD.

From this table, we can observe that when better deconfounding techniques are used, CHs&CHi decreases and the accuracy of the gender, action, and attribute words increases, *e.g.*, UD-BD performs better than UD and UD-FD outperforms UD-FD/Cor. Also, it can be found that our UD-DICv1.0 achieves the lowest CHs&CHi and the highest accuracy, which means that UD-DICv1.0 generates the least biases induced by the biased training dataset.

The radar charts in Fig. 7.6 show the accuracy when some specific visual concepts appear. In the radar chart of gender words, we can find that the visual concept ID_{Bed} induces strong gender biases where the accuracy of UD is smaller than 0.6, while our UD-DICv1.0 can alleviate such bias by improving the accuracy to almost 0.9. Also, we can find that some biases are hardly alleviated, *e.g.*, the action

bias induced by ID_{Horse} where the accuracies of three models are lower than 0.4. There are at least two reasons for this. The first one is about language bias that many verbs have a similar meaning while we do not consider this situation when counting consistent words for the accuracy, *e.g.*, “horse-pull-carriage” and “horse-draw-carriage”. The second one is about the selection bias that humans may use different action words to describe an image, *e.g.*, when one human is near a horse, we may annotate it as “human-stand near-horse”, “human-walk with-horse”, or even “human-have-horse”. Such an example demonstrates the complexities of the bias constructions and the difficulties of the bias evaluations, which requires further investigations to disentangle different biases and develop more advanced metrics for evaluating them.

Human evaluation. Considering a better way of evaluating biases is human evaluation, we thus asked 20 humans to sort the captions generated by three models UD/UD-BD/UD-DICv1.0 according to the consistencies of these captions with the images. We chose 50 images to generate the captions and we specifically chose the images which are more likely to induce biases, *e.g.*, the images containing the visual concepts like $ID_{\text{Skateboard}}$ and ID_{Horse} . When these judges considered that two captions are similar, they sorted them with the same rank. After sorting, we pairwise compared the results to compute the pie charts shown in Fig. 7.7. We can see that the judges consider our UD-DICv1.0’s captions are more consistent than UD and UD-BD, which suggests that it generates more accurate and less biased captions. Fig. 7.8 visualizes some qualitative examples used in the human evaluation.

7.5.2.2 Network Size

Compared with the baseline UD, our UD-DICv1.0 has a larger network, which contains three additional parallel LSTM layers and three EXPT modules that introduce some additional dictionaries. To check the effects of these additional LSTM layers and dictionaries, we carried the following ablation studies. We set the number of LSTM layers to different values in UD and UD-DICv1.0 to measure their effects. Also, we changed the sizes of the dictionaries $\mathbf{X}$ and $\mathbf{Z}$ in UD-DICv1.0. The performances of these baselines are list in Table 7.2 where the column “LSTM” means the numbers of LSTM layers before the embedding layer, the columns “ $\mathbf{X}$ ” and “ $\mathbf{Z}$ ” denote the numbers of the elements in them. For convenience, we report the

TABLE 7.2: The performances of various ablation studies about the network size on MS COCO Karpathy split. For the convenience of the comparisons, we add an index before each model for better reference.

Models	LSTM	$\mathbf{X}$	$\mathbf{Z}$	C $\uparrow$	CHs $\downarrow$	CHi $\downarrow$
(1) UD	1	-	-	125.3	13.7	8.9
(2) UD	4	-	-	125.8	13.5	8.8
(3) UD-DICv1.0	1	10,000	9,590	127.9	10.6	6.8
(4) UD-DICv1.0	4	10,000	9,590	128.8	10.1	6.5
(5) UD-DICv1.0	4	5,000	9,590	128.4	10.3	6.7
(6) UD-DICv1.0	4	15,000	9,590	129.0	10.0	6.4
(7) UD-DICv1.0	4	10,000	5,000	128.3	10.4	6.8
(8) UD-DICv1.0	4	10,000	15,000	128.9	10.1	6.6

CIDEr-D, CHs, and CHi scores to measure the similarities and bias degrees of the compared models.

Results and Analysis. By comparing (1) with (2) in Table 7.2, we can find that although adding more LSTM layers in UD can improve the CIDEr-D score while the improvement is not as large as our UD-DICv1.0, whose CIDEr-D score is 128.8, and more importantly, simply adding LSTM layers will not alleviate the bias since CHs and CHi are not obviously decreased. Also, as shown in (3) and (4), if we only use one LSTM layer to generate the context vector, the performances will not be degraded significantly. These comparisons suggest that the improvements of our UD-DICv1.0, especially its ability to alleviate the biases, come more from the approximations of the deconfounding techniques than the additional LSTM layers. From (4) to (8) in Table 7.2, we change the size of $\mathbf{X}$ and $\mathbf{Z}$. By comparing their performances, we can find that the changes are marginal, which means that the size of the dictionaries does not largely affect the performances.

7.5.3 Comparisons with State of the Art

Comparing Methods. We start our comparisons from **Up-Down** [1], whose visual features are most frequently used by the subsequent captioners and so do we. We follow the causal retrospect in Fig. 7.4 to group the compared state of the art captioners into four groups: **large-scale pre-training: Up-Down** [1], **UD-HIP** [208], and **VLP** [209]; **attention mechanisms: CAVP** [177], **RFNet** [167],

TABLE 7.3: The performances of the state of the art ICs on Karpathy split. The top and bottom parts report the performances trained by CIDEr-D computed from 5 captions and the whole training set, respectively. “Group” specifies each IC’s category according to Fig. 7.4. The symbol [†] means the re-implemented model.

Models	Group	B@4	M	R	C	S
Up-Down [1]	b	36.3	27.7	56.9	120.1	21.4
Up-Down [†] [1]	b	37.2	27.5	57.3	125.3	21.1
RFNet [167]	c	37.9	28.3	58.3	125.7	21.7
CAVP [46]	c	38.6	28.3	58.5	126.3	21.6
LBPF [39]	c	38.3	28.5	58.4	127.6	22.0
CNM [178]	d	38.7	28.4	58.7	127.4	21.8
GCN-LSTM [2]	e	38.2	28.5	58.3	127.6	22.0
SGAE [3]	e	38.4	28.4	58.6	127.8	22.1
UD-DICv1.0	f	39.0	28.8	58.8	128.8	22.0
Up-Down [†] [1]	b	37.7	28.2	58.1	126.4	21.8
UD-HIP [208]	b	38.2	28.4	58.3	127.2	21.9
VLP [209]	b	39.5	—	—	129.3	23.2
AoANet [206]	c	38.9	29.2	58.8	129.8	22.4
AoANet [†] [206]	c	38.9	28.9	58.4	128.7	22.4
UD-DICv1.0	f	38.3	28.5	58.5	129.5	22.0
AoA-DICv1.0	f	39.5	29.6	59.0	131.1	22.6

LBPF [39], and **AoANet** [206]; **sentence patterns**: **CNM** [178]; and **structured attention**: **GCN-LSTM** [2] and **SGAE** [3]. It is noteworthy that two different CIDEr-D were used as the self-critical rewards in the previous captioners. The first one computes Inverse Document Frequency (IDF) from each image’s five ground-truth captions and the second one computes IDF from the captions of the whole training set. For fair comparisons, we also used two different CIDEr-D based self-critical rewards to train our DICs and compare them with the captioners trained by the same reward. The results are reported in Table 7.3, where the top and bottom parts list the results of the captioners trained by the first and second CIDEr-D based rewards, respectively.

Results and Analysis. From Table 7.3, we can find that our single-model UD-DICv1.0 and AoA-DICv1.0 achieve the best CIDEr-D scores: 128.8 and 131.1 when different training rewards are used. Compared with UD-HIP which pre-trained their captioner by object detection and segmentation, our UD-DICv1.0, which was only pre-trained by object detection, has better performances. Interestingly, compared with VLP which exploited 30 times more samples (3 million) than ours

TABLE 7.4: The performances of single methods on the online MS COCO test server. The top and bottom parts report the performances trained by CIDEr-D computed from 5 captions and the whole training set, respectively.

Model	B@4		M		R-L		C-D	
Metric	c5	c40	c5	c40	c5	c40	c5	c40
Up-Down [1]	36.9	68.5	27.6	36.7	57.1	72.4	117.9	120.5
CAVP [177]	37.9	69.0	28.1	37.0	58.2	73.1	121.6	123.8
RFNet [167]	38.0	69.2	28.2	37.2	58.2	73.1	122.9	125.1
SGAE [3]	37.8	68.7	28.1	37.0	58.2	73.1	122.7	125.5
CNM [178]	37.9	68.4	28.1	36.9	58.3	72.9	123.0	125.3
UD-DICv1.0	37.9	69.2	28.7	37.7	58.3	73.3	124.1	126.7
AoANet [206]	37.3	68.1	28.3	37.2	57.9	72.8	124.0	126.2
AoA-DICv1.0	38.8	70.5	28.8	38.2	58.6	73.9	126.2	128.4

(0.1 million) to pre-train their model, our UD-DICv1.0 and AoA-DICv1.0 still achieve competitive results. Both comparisons suggest that our DIC framework is more cost-effective and efficient than these large-scale pre-training captioners in boosting performances. Our DIC also outperforms the captioners with advanced attention mechanisms, *e.g.*, UD-DICv1.0 is better than RFNet, CAVP, and LBPF; and AoA-DICv1.0 is better than AoANet though we did not use more advanced training strategy as AoANet [206]. Finally, comparing our UD-DICv1.0 with the approximations of the front-door frameworks: sentence patterns and structured attention, we find that our UD-DICv1.0 is still the best, although we did not use multi-step reasoning as CNM and did not deploy complex graph convolution as GCN-LSTM and SGAE. We also compare our single-model UD-DICv1.0 and AoA-DICv1.0 with the other captioners on MS COCO online test set, where the results are reported in Table 7.4. It can be observed that our UD-DICv1.0 and AoA-DICv1.0 achieve the highest CIDEr-D c5 and c40 scores. All of these comparisons confirm the superiority of the proposed DIC framework over the captioners with weaker deconfounding approximations.

7.6 Epilogue

We used the causal perspective to offer an in-depth analysis about why modern captioners easily learn the spurious correlation. We discovered that some frequently appeared concepts act as a typical confounder that if we neglect such confounder

and train the captioners by the observational correlation $P(\mathcal{L}|\mathcal{X})$, the learned captioners collapse into the dataset bias. Then we introduced two techniques: the backdoor and front-door adjustments which can alleviate such problem by calculating the interventional probability $P(\mathcal{L}|do(\mathcal{X}))$ as the training target. We also retrospected the major progress in IC and found that many advanced captioners are the approximations of the backdoor and front-door adjustments. After that, we derived an effective method called DICv1.0 based on the causal retrospect and validated its effectiveness by incorporating it into two prevailing models: Up-Down and AoANet.

From the causal retrospect in Fig. 7.4, we see a promising future: *DICv1.0 is just a start!* For example, it is derived by assuming that $\mathcal{S}$ can be split into single key words for deconfounding, while this may also not be the real situation. As the red parts we draw in Fig. 7.4(g), if we cannot split $\mathcal{S}$, we can re-assign $\mathcal{S}$ to $\mathcal{D}$ and $\mathcal{Z}$ to $\mathcal{X}$, then we jump back into “where we start” and it is possible to recursively deploy all the previous techniques, including this chapter, which will be the “previous IC” in the recursive future.

Chapter 8

Conclusion and Future Works

8.1 Conclusion

In this thesis, we proposed three captioners which exploited three different kinds of knowledge for generating more descriptive and less biased captions or paragraphs. Before introducing how to incorporate additional knowledge into captioners, in Chapter 3, we introduced an object-agnostic relationship detector for generating less biased scene graph, which is one frequently used data structure of additional knowledge. Specifically, we designed a novel visual relationship feature learning strategy named as **Shuffle-Then-Assemble** for improving visual relationship models. The crux of this method is to discard the original one-to-one paired object alignments and then try to recover them in an unsupervised pair discovery fashion by using a cycle-consistent adversarial domain transfer method. As a result, the object category will less affect the relationship modelling and thus the resultant feature map is less likely biased to specific object compositions. We validated our **Shuffle-Then-Assemble** strategy on two visual relationship benchmarks, which both demonstrated that our strategy can consistently improve naive relationship prediction models. At that time, our work is the first attempt to tackle the small-data and incomplete-annotation challenges of visual relationship modelling without any external prior knowledge.

After introducing this scene graph generator for extracting scene graphs from images, in Chapter 4 and Chapter 5, we exploited these scene graphs as the bridges

to incorporate diverse additional knowledge into caption and paragraph generators, respectively. Specifically, in Chapter 4, we presented a novel unsupervised learning method: Scene Graph Auto-Encoder (SGAE) for embedding language inductive bias into a dictionary set, which can be shared as a re-encoder for language generation. Besides this, we also distilled the inductive bias into the language decoder of the image captioner to further improve the transferability. Thanks to these techniques, our image captioner generates more descriptive and human-like captions. We validated the SGAE-based framework by extensive ablations and comparisons with the state-of-the-art image captioners on MS-COCO captioning benchmark and the more challenging unpaired image captioning task. We believe that SGAE is a general solution for capturing the language inductive bias, which can also benefit other vision-language tasks.

In Chapter 5, we proposed Hierarchical Scene Graph Encoder-Decoder (HSGED) based encoder-decoder which follows the script constructed by a scene graph to generate the paragraph. In this way, the semantic and hierarchical knowledge of an image can be transferred into the language domain, so compared with traditional HRNNs without any hierarchical constraints, more coherent and distinctive paragraphs can be generated. Specifically, our HSGED contains two RNNs: SSG-RNN for generating sub-graph level topic vectors and WSG-RNN for completing the corresponding sentences. To further encourage distinctive sentences, irredundant attention and inheriting attention were respectively deployed into SSG-RNN and WSG-RNN as the additional hierarchical regularizations. We also designed a sentence-level loss to regularize the sequence of the generated sentences to be similar to the ground-truth paragraphs. Our extensive experiments demonstrated that the proposed model outperforms the state-of-the-art methods significantly and it can generate more coherent and distinctive paragraphs.

In Chapter 6, we exploited another kind of additional knowledge into the captioner, which is the sentence pattern. In this way, we imitated the humans inductive bias — sentences are composed by structuring patterns — for image captioning. In particular, we presented a novel modular network method: learning to Collocate Neural Modules (CNM), which can generate captions by filling the contents into collocated modules. As a result, the caption generation is expected to be disentangled from dataset bias. We validated our CNM by extensive ablations and comparisons with state-of-the-art models on MS-COCO.

In Chapter 7, we followed the causal view to retrospect the captioners proposed in recent years, which also includes our SGAE and CNM. Then we proposed a novel captioner: Deconfounded Image Captioning (DIC) to generate less biased captions. In DIC, we treated the commonsense knowledge as additional knowledge and built a front-door causal graph by using this knowledge as the mediator to transmit information from the images to the captions. With this graph, we exploited the backdoor and front-door adjustments for alleviating the negative effects brought by dataset bias and thus generated less biased captions. At last, we validated the effectiveness of our DICv1.0 by incorporating it into two prevailing models: Up-Down and AoANet.

From three major captioners, we can find that the better the additional knowledge, e.g., the higher the quality and the more similar between the external knowledge and the captioning domain, the better the captions, e.g., the more descriptive and less biased, can be generated. The strategies of using different additional knowledge are also important, when the more suitable ways are used, the better captions are generated. For example, the strategy used in DIC is better than the other two captioners which are SGAE and CNM since DIC applies the technique from causal inference while SGAE and CNM are both heuristic methods.

8.2 Future Works

In this thesis, we introduce different image captioners which exploit three diverse kinds of additional knowledge and by which more descriptive and less biased captions can be generated. However, some of the techniques proposed in this thesis can be further refined and some others can be extended to more general models for solving more vision-language tasks. We discuss two main directions for the future works.

8.2.1 Auto-Parsing Network (APN)

In Chapter 4 and Chapter 5, we introduce two captioners which exploit scene graphs to represent additional knowledge. However, such a strategy requires a large number of matched graph annotations [56, 215, 216] for training useful parsers [58, 217–219]; otherwise the domain shift will be induced, invalidating the parsed graphs. For example, a parser trained for extracting scene graphs from declarative sentence can not be used for extracting question sentences. In this way, the sentence scene graph parser trained for captioning can not be applied into the VQA systems.

To relieve the burden of exploiting hierarchical and sparse graph priors for incorporating additional knowledge, we need a network that does not require any additional graph annotation to learn how to parse the input sequences into trees during the end-to-end training, hence named as Auto-Parsing Network (APN). To design an APN, we can add sparsity assumptions into Transformer frameworks [116], which have been prevalently applied into vision-language tasks and impressive improvements have been observed in image captioning [210, 220–222], VQA [223], image grounding [224, 225], and visual reasoning [4, 226]. One major advantage of the Transformer is that it can incorporate the fully connected graph prior [87] via self-attention. Particularly, by the graph prior, even though vision and language data have quite different superficial forms, their structural commonalities can be automatically abstracted, embedded, and transferred to narrow the domain gap.

However, the underlying structures of vision and language data are usually hierarchical and sparse, which is different from fully connected graphs, *e.g.*, a sentence or an image can be parsed as constituent trees of words or objects, respectively [87, 227, 228]. Without sparse and hierarchical constraints, such a system may be overwhelmed by the trivial global dependencies and overlook the critical local context [229–231]. This is another reason that why we need an APN.

Here are some initial ideas about designing a Transformer based APN. We can regularize the self-attention operation by a linear chain Conditional Random Field (CRF) [232, 233], which is parameterized by differentiable attention operations. As shown in Figure 8.1, compared with the original self-attention, the CRF segments the input sequence into a tree by grouping the neighboring entities in clusters. After each iteration of segmenting, only the entities in the same cluster can attend to each other and thus the local context is embedded. Intuitively, each cluster can

FIGURE 8.1: The sub-graph (a) visualizes the segmentation of a fully connected graph containing five nodes: from s_1 to s_5 . The sub-graph (b) shows the process of constructing a hierarchical tree using APN.

be thought as the parent of the entities in it and these clusters together compose a new sequence. By stacking these regularized self-attention layers, the new sequence at a lower layer will be further segmented by the CRF at a higher layer. Via this iterative way, a tree can be formed and the local and global contexts are accordingly embedded at lower and higher layers, respectively. Once we build an encoder-decoder based on these regularized self-attention layers, the hierarchical structures of both source and target domains can be automatically parsed, embedded, and transferred. This APN can be used in various transformer based structures to solve many vision-language tasks like image captioning or visual question answering.

8.2.2 Causal Attention

In Chapter 7, we exploit the backdoor and front-door adjustments to design a deconfounded image captioner for generating less biased captions. In that chapter, we also retrospect different captioners from the causal view and discover that attention mechanism is an approximation of the front-door adjustment. Importantly, attention mechanism is widely used in different architectures for solving a large amount of machine learning and computer vision tasks [1, 24, 118, 234, 235]. Since we have discovered that attention mechanism is a weak approximation of the front-door adjustment, we can also develop an advanced attention mechanism based on the front-door adjustment and apply it into any architecture where attention mechanism is used. We can call the developed attention mechanism as causal attention since it is designed based on the causal framework.

FIGURE 8.2: Top: an example of image captioner with a self-attention and a top-down attention modules. Bottom: the corresponding causal graph. The reason why the prediction is “riding” but not “driving” is explained in Figure 8.4.

Although the formulation of attention mechanism varies from task to task, its nature can be summarized as the following common Q-K-V notation: given a *query* $\mathbf{q}$, the attention mechanism associates $\mathbf{q}$ to each feature *value* $\mathbf{v}_i$ by using the normalized attentive weight $\alpha_i \propto \mathbf{q}^T \mathbf{k}_i$, where $\mathbf{k}_i$ is the *key* function of the value; thus, the resultant selective feature value — attention — is $\sum_i \alpha_i \mathbf{v}_i$. In a modern view, the attention can be understood as a feature *transformer* that encodes input query $\mathbf{q}$ by using the given values $\mathbf{V} = \{\mathbf{v}_i\}$ [116].

Taking image captioning as an example in Figure 8.2, if $\mathbf{q}$ and $\mathbf{V}$ are both encoded from the input X , *e.g.*, the RoI features of an image, we call it *self-attention*; if $\mathbf{q}$ is changed to the sentence context, we call it *top-down attention*. Intuitively, self-attention is usually viewed as a non-local [236] (or graph [87]) convolution network that enriches each local value with global relationship features; top down-attention is used to enrich the context with the cross-domain relationship features [1]. Both of them can be combined and stacked into deep networks, serving as powerful multi-modal encoder-decoder transformer networks [14, 237].

As a bridge connecting the input feature X and the output label Y , the quality of attention — how reasonable the attentive weight α is — plays a crucial role for the overall performance. However, due to the fact that the attention weights are unsupervised, *e.g.*, there is no word-region grounding for the top-down attention

FIGURE 8.3: Before pre-training (*e.g.*, LXMERT [4]), attentions are correct (blue). After pre-training, attentions are wrong (red). This is because the co-occurrences of some concepts appear much more often than others, *e.g.*, “Sport+Man” appears 213 times more than “Sport+Screen” in the pre-training data.

or relationship dependency annotation for the self-attention, the weights will be inevitably misled by the dataset bias. For example, as shown in Figure 8.2, since there are many images captioned with “person riding horse” in the training data, self-attention learns to infer “riding” by building the dependency between “person” and “horse”. Then, given a test image with “person driving carriage”, this self-attention still tends to relate “person” with “horse” to infer “riding”, but ignoring the “carriage”. Unfortunately, such bias cannot be mitigated by simply enlarging the dataset scale, as most of the bias abides by the data nature — Zipf’s law [195] and social conventions [34] — there are indeed more “red apple” than “green apple” or “person standing” than “person dancing”. Therefore, as shown in Figure 8.3, large-scale pre-training may lead to even worse attentions.

The dataset bias is essentially caused by the confounder, a common cause that makes X and Y correlated even if X and Y have no direct causation. We illustrate this crucial idea in Figure 8.4. Suppose that the confounder C is the common sense¹ “person can ride horse”, $C \rightarrow X$ denotes that a visual scene is generated by such knowledge, *e.g.*, the dataset curator observes and captures the common sense; $X \rightarrow M$ denotes the fact that the objects $M = \{\text{person, horse}\}$ can be detected (*e.g.*, Faster

FIGURE 8.4: This expands the causal links of the confounding path $X \longleftrightarrow Y$ in Figure 8.2.

¹It is also well-known as the disentangled causal mechanism [238].

R-CNN [6]), whose object inventory is determined by $C \rightarrow M$; $M \rightarrow Y$ denotes the language generation for “person riding horse”. Note that besides the legitimate causal path from image X via object M to Y , the “backdoor” path $X \leftarrow C \rightarrow M \rightarrow Y$ also contributes an effect to Y . Therefore, if we only train the model based on the correlation $P(Y|X)$ without knowing the confounding effect, no matter how large the amount of training data is, the model can never identify the true causal effect from X to Y [35, 239]. For example, if the confounder distribution varies from training to testing, *e.g.*, the common sense “person can ride horse” is dominantly more often than the common sense “person can drive carriage” in training, but the latter is more often than the former in testing, then $P(Y|X)$ based on “person can ride horse” in training will be no longer applicable in testing [240].

To help the models identify the causal effect between X and Y for mitigating the bias caused by confounders, we can design a novel attention mechanism named as causal attention (CATT), which is based on the *front-door adjustment* principle that does not require the assumption of any observed confounder [241]. In this way, CATT can be applied in any domain where the attention resides.

List of the Publications

Journal Articles

- **Xu Yang**, Hanwang Zhang, Jianfei Cai. “Auto-encoding and Distilling Scene Graphs for Image Captioning.”, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Accepted.

Conference Proceedings

- **Xu Yang**, Chongyang Gao, Hanwang Zhang, Jianfei Cai. “Hierarchical Scene Graph Encoder-Decoder for Image Paragraph Captioning.” Association for Computing Machinery (ACM)’s annual conference on multimedia (**ACMMM, Poster**). Seattle, United States, 2020. Link: <https://dl.acm.org/doi/abs/10.1145/3394171.3413859>
- **Xu Yang**, Hanwang Zhang, Jianfei Cai. “Learning to Collocate Neural Modules for Image Captioning.” Proceedings of the IEEE Conference on International Conference on Computer Vision (**ICCV, Poster**). Seoul, Korea, 2019. Link: <https://arxiv.org/pdf/1904.08608.pdf>.
- **Xu Yang**, Kaihua Tang, Hanwang Zhang, Jianfei Cai. “Auto-Encoding Scene Graphs for Image Captioning.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (**CVPR, Oral**). Long Beach, USA, 2019. Link: <https://arxiv.org/pdf/1812.02378.pdf>.
- **Xu Yang**, Hanwang Zhang, Jianfei Cai. “Shuffle-Then-Assemble: Learning Object-Agnostic Visual Relationship Features.” Proceedings of the 15th European Conference on Computer Vision (**ECCV, Poster**). Munich, Germany. 2018. Link: <https://arxiv.org/pdf/1808.00171.pdf>.
- Xiangxi Shi, **Xu Yang**, Jiuxiang Gu, Shafiq Joty, and Jianfei Cai. “Finding It at Another Side: A Viewpoint-Adapted Matching Encoder for Change Captioning.” Proceedings of the 16th European Conference on Computer Vision (**ECCV, Poster**). Glasgow, UK, 2020. Link: <https://arxiv.org/pdf/2009.14352.pdf>

-
- Jiuxiang Gu, Shafiq Joty, Jianfei Cai, Handong Zhao, **Xu Yang**, Gang Wang. “Unpaired image captioning via scene graph alignments.” Proceedings of the IEEE Conference on International Conference on Computer Vision (**ICCV, Poster**). Seoul, Korea, 2019.

Bibliography

- [1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, volume 3, page 6, 2018. [xvi](#), [xvii](#), [xviii](#), [3](#), [10](#), [31](#), [33](#), [35](#), [36](#), [37](#), [44](#), [47](#), [50](#), [51](#), [52](#), [53](#), [56](#), [57](#), [67](#), [75](#), [83](#), [84](#), [86](#), [96](#), [99](#), [106](#), [112](#), [113](#), [119](#), [125](#), [126](#), [127](#), [132](#), [133](#), [134](#), [140](#), [141](#)
- [2] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Exploring visual relationship for image captioning. In *Computer Vision–ECCV 2018*, pages 711–727. Springer, 2018. [xvi](#), [10](#), [44](#), [50](#), [51](#), [52](#), [53](#), [56](#), [57](#), [106](#), [120](#), [121](#), [133](#)
- [3] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10685–10694, 2019. [xvi](#), [13](#), [50](#), [51](#), [52](#), [53](#), [56](#), [57](#), [63](#), [106](#), [120](#), [121](#), [126](#), [133](#), [134](#)
- [4] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019. [xix](#), [2](#), [139](#), [142](#)
- [5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. [1](#)
- [6] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. [1](#), [2](#), [9](#), [12](#), [20](#), [24](#), [27](#), [35](#), [37](#), [46](#), [74](#), [82](#), [88](#), [115](#), [125](#), [143](#)
- [7] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525. IEEE, 2017. [1](#), [33](#)
- [8] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. [1](#)
- [9] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. [1](#)

- [10] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. [1](#), [2](#), [115](#), [126](#)
- [11] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015. [1](#), [15](#), [16](#), [111](#)
- [12] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016. [1](#), [111](#)
- [13] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018. [2](#), [111](#), [119](#)
- [14] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265*, 2019. [2](#), [119](#), [141](#)
- [15] Polina Kuznetsova, Vicente Ordonez, Alexander C Berg, Tamara L Berg, and Yejin Choi. Collective generation of natural image descriptions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 359–368. Association for Computational Linguistics, 2012. [2](#), [9](#)
- [16] Margaret Mitchell, Xufeng Han, Jesse Dodge, Alyssa Mensch, Amit Goyal, Alex Berg, Kota Yamaguchi, Tamara Berg, Karl Stratos, and Hal Daumé III. Midge: Generating image descriptions from computer vision detections. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 747–756. Association for Computational Linguistics, 2012.
- [17] Siming Li, Girish Kulkarni, Tamara L Berg, Alexander C Berg, and Yejin Choi. Composing simple image descriptions using web-scale n-grams. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 220–228. Association for Computational Linguistics, 2011.
- [18] Girish Kulkarni, Visruth Premraj, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Baby talk: Understanding and generating image descriptions. In *Proceedings of the 24th CVPR*. Citeseer, 2011. [2](#), [9](#)

- [19] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014. 2, 10, 15
- [20] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015. 2, 10, 38, 61, 64, 84, 111, 119
- [21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2, 18, 21
- [22] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2, 82
- [23] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 2
- [24] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015. 3, 10, 15, 33, 37, 61, 64, 117, 119, 123, 140
- [25] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4659, 2016. 3, 10, 119
- [26] Zhe Gan, Chuang Gan, Xiaodong He, Yunchen Pu, Kenneth Tran, Jianfeng Gao, Lawrence Carin, and Li Deng. Semantic compositional networks for visual captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5630–5639, 2017. 3, 10
- [27] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 6, page 2, 2017. 3, 10, 84, 119
- [28] Marc’Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL <http://arxiv.org/abs/1511.06732>. 3, 11, 33
- [29] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *CVPR*, volume 1, page 3, 2017. 3, 11, 33, 36, 38, 49, 51, 56, 57, 72, 96, 97, 106, 126

- [30] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1171–1179, 2015. 3, 11
- [31] Jiuxiang Gu, Jianfei Cai, Gang Wang, and Tsuhan Chen. Stack-captioning: Coarse-to-fine learning for image captioning. *AAAI*, 2017. 3, 11, 51, 56, 57, 106
- [32] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Neural baby talk. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7219–7228, 2018. 4, 10, 33, 38, 61, 84, 106, 120
- [33] Xu Yang, Hanwang Zhang, and Jianfei Cai. Deconfounded image captioning: A causal retrospect. *arXiv preprint arXiv:2003.03923*, 2020. 4
- [34] Lisa Anne Hendricks, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach. Women also snowboard: Overcoming bias in captioning models. In *European Conference on Computer Vision*, pages 793–811. Springer, 2018. 4, 110, 111, 142
- [35] Judea Pearl. *Causality: models, reasoning and inference*, volume 29. Springer, 2000. 4, 111, 113, 114, 143
- [36] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 9, 15, 17, 18, 21, 37, 82, 96
- [37] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL <http://arxiv.org/abs/1409.0473>. 10, 33
- [38] Z. Zha, D. Liu, H. Zhang, Y. Zhang, and F. Wu. Context-aware visual policy network for fine-grained image captioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019. doi: 10.1109/TPAMI.2019.2909864. 10, 33, 51, 57
- [39] Yu Qin, Jiajun Du, Yonghua Zhang, and Hongtao Lu. Look back and predict forward in image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8367–8375, 2019. 10, 44, 51, 56, 57, 106, 133
- [40] Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, and Tao Mei. Boosting image captioning with attributes. In *IEEE International Conference on Computer Vision, ICCV*, pages 22–29, 2017. 10, 51, 56, 57, 106, 119
- [41] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4565–4574, 2016. 11

- [42] Guojun Yin, Lu Sheng, Bin Liu, Nenghai Yu, Xiaogang Wang, and Jing Shao. Context and attribute grounded dense captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6241–6250, 2019. [11](#)
- [43] Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei. A hierarchical approach for generating descriptive image paragraphs. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. [11](#), [61](#), [62](#), [63](#), [64](#), [72](#), [74](#), [75](#), [79](#), [80](#)
- [44] Moitreyia Chatterjee and Alexander G Schwing. Diverse and coherent paragraph generation from images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 729–744, 2018. [11](#), [75](#), [79](#), [80](#)
- [45] Xiaodan Liang, Zhiting Hu, Hao Zhang, Chuang Gan, and Eric P Xing. Recurrent topic-transition gan for visual paragraph generation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3362–3371, 2017. [11](#), [63](#), [79](#), [80](#)
- [46] Zheng-Jun Zha, Daqing Liu, Hanwang Zhang, Yongdong Zhang, and Feng Wu. Context-aware visual policy network for fine-grained image captioning. *IEEE transactions on pattern analysis and machine intelligence*, 2019. [62](#), [79](#), [80](#), [106](#), [119](#), [133](#)
- [47] Jing Wang, Yingwei Pan, Ting Yao, Jinhui Tang, and Tao Mei. Convolutional auto-encoding of sentence topics for image paragraph generation. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, 2019. [11](#), [79](#), [80](#)
- [48] Luke Melas-Kyriazi, Alexander Rush, and George Han. Training for diversity in image paragraph captioning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 757–761, 2018. [11](#), [61](#), [72](#), [75](#), [76](#), [79](#), [80](#)
- [49] Siying Wu, Zheng-Jun Zha, Zilei Wang, Houqiang Li, and Feng Wu. Densely supervised hierarchical policy-value network for image paragraph generation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 975–981. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/137. URL <https://doi.org/10.24963/ijcai.2019/137>. [11](#), [62](#), [79](#), [80](#)
- [50] Yadan Luo, Zi Huang, Zheng Zhang, Ziwei Wang, Jingjing Li, and Yang Yang. Curiosity-driven reinforcement learning for diverse visual paragraph generation. In *Proceedings of the 27th ACM International Conference on Multimedia*, page 2341–2350, 2019. [11](#), [79](#), [80](#)
- [51] Jiuxiang Gu, Shafiq Joty, Jianfei Cai, and Gang Wang. Unpaired image captioning by language pivoting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 503–519, 2018. [12](#), [43](#), [59](#)

- [52] Yang Feng, Lin Ma, Wei Liu, and Jiebo Luo. Unsupervised image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4125–4134, 2019. [12](#), [59](#)
- [53] Jiuxiang Gu, Shafiq Joty, Jianfei Cai, Handong Zhao, Xu Yang, and Gang Wang. Unpaired image captioning via scene graph alignments. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10323–10332, 2019. [12](#), [59](#)
- [54] Iro Laina, Christian Rupprecht, and Nassir Navab. Towards unsupervised image captioning with shared multimodal embeddings. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7414–7424, 2019. [12](#), [59](#)
- [55] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *CVPR*, 2015. [12](#), [13](#), [17](#), [63](#)
- [56] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2017. [12](#), [18](#), [19](#), [25](#), [49](#), [53](#), [74](#), [88](#), [97](#), [119](#), [120](#), [125](#), [139](#)
- [57] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. Visual translation embedding network for visual relation detection. In *CVPR*, 2017. [12](#), [17](#), [18](#), [20](#), [21](#), [25](#), [26](#), [28](#), [29](#), [89](#)
- [58] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin 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. [13](#), [35](#), [46](#), [74](#), [89](#), [120](#), [139](#)
- [59] Xu Yang, Hanwang Zhang, and Jianfei Cai. Shuffle-then-assemble: Learning object-agnostic visual relationship features. In *European Conference on Computer Vision*, pages 38–54. Springer, 2018.
- [60] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph rcnn for scene graph generation. In *European Conference on Computer Vision*, pages 690–706. Springer, 2018. [14](#), [35](#), [40](#)
- [61] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. 2017. [13](#), [17](#), [63](#)
- [62] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6619–6628, 2019.

- [63] Long Chen, Hanwang Zhang, Jun Xiao, Xiangnan He, Shiliang Pu, and Shih-Fu Chang. Counterfactual critic multi-agent training for scene graph generation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4613–4623, 2019. [12](#), [82](#)
- [64] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. [12](#)
- [65] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. [12](#)
- [66] Abhinav Gupta and Larry S Davis. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *ECCV*, 2008. [12](#)
- [67] Yu-Wei Chao, Zhan Wang, Yugeng He, Jiaxuan Wang, and Jia Deng. Hico: A benchmark for recognizing human-object interactions in images. In *ICCV*, 2015. [12](#)
- [68] Abhinav Gupta, Aniruddha Kembhavi, and Larry S Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *TPAMI*, 2009. [12](#)
- [69] Vignesh Ramanathan, Congcong Li, Jia Deng, Wei Han, Zhen Li, Kunlong Gu, Yang Song, Samy Bengio, Chuck Rosenberg, and Li Fei-Fei. Learning semantic relationships for better action retrieval in images. In *CVPR*, 2015.
- [70] Bangpeng Yao and Li Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *CVPR*, 2010. [12](#)
- [71] Mark Yatskar, Luke Zettlemoyer, and Ali Farhadi. Situation recognition: Visual semantic role labeling for image understanding. In *CVPR*, 2016. [12](#)
- [72] Yuval Atzmon, Jonathan Berant, Vahid Kezami, Amir Globerson, and Gal Chechik. Learning to generalize to new compositions in image understanding. In *EMNLP*, 2016.
- [73] Mohammad Amin Sadeghi and Ali Farhadi. Recognition using visual phrases. In *CVPR*, 2011. [20](#)
- [74] Chaitanya Desai and Deva Ramanan. Detecting actions, poses, and objects with relational phraselets. In *ECCV*, 2012. [12](#)
- [75] Hanwang Zhang, Zawlin Kyaw, Jinyang Yu, and Shih-Fu Chang. Ppr-fcn: Weakly supervised visual relation detection via parallel pairwise r-fcn. In *ICCV*, 2017. [13](#), [17](#), [18](#)
- [76] Julia Peyre, Ivan Laptev, Cordelia Schmid, and Josef Sivic. Weakly-supervised learning of visual relations. In *ICCV*, 2017. [13](#), [28](#), [29](#)

- [77] Yikang Li, Wanli Ouyang, Xiaogang Wang, et al. Vip-cnn: Visual phrase guided convolutional neural network. In *CVPR*, 2017. [13](#), [17](#), [21](#)
- [78] Ji Zhang, Mohamed Elhoseiny, Scott Cohen, Walter Chang, and Ahmed Elgammal. Relationship proposal networks. In *CVPR*, 2017.
- [79] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, 2018. [13](#)
- [80] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *ECCV*, 2016. [13](#), [15](#), [18](#), [19](#), [20](#), [25](#), [26](#), [28](#), [29](#), [120](#)
- [81] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. In *ICCV*, 2017. [13](#)
- [82] Yu-Siang Wang, Chenxi Liu, Xiaohui Zeng, and Alan Yuille. Scene graph parsing as dependency parsing. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 397–407. Association for Computational Linguistics, 2018. doi: 10.18653/v1/N18-1037. URL <http://aclweb.org/anthology/N18-1037>. [13](#), [35](#)
- [83] Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher D Manning. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the fourth workshop on vision and language*, pages 70–80, 2015. [13](#), [39](#)
- [84] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, pages 382–398. Springer, 2016. [13](#), [35](#), [39](#), [50](#), [78](#), [101](#), [128](#)
- [85] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015. [13](#)
- [86] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.
- [87] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018. [34](#), [66](#), [139](#), [141](#)
- [88] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=ryGs6iA5Km>. [13](#), [14](#), [35](#), [40](#), [63](#), [66](#), [67](#)

- [89] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1219–1228, 2018. [13](#), [35](#), [63](#)
- [90] Jiaxin Shi, Hanwang Zhang, and Juanzi Li. Explainable and explicit visual reasoning over scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8376–8384, 2019. [13](#), [63](#)
- [91] Wenbin Che, Xiaopeng Fan, Ruiqin Xiong, and Debin Zhao. Paragraph generation network with visual relationship detection. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1435–1443, 2018. [13](#)
- [92] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL <http://arxiv.org/abs/1312.6203>. [14](#)
- [93] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.
- [94] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015. [14](#)
- [95] Alessio Micheli. Neural network for graphs: A contextual constructive approach. *IEEE Transactions on Neural Networks*, 20(3):498–511, 2009. [14](#), [120](#)
- [96] James Atwood and Don Towsley. Diffusion-convolutional neural networks. In *Advances in neural information processing systems*, pages 1993–2001, 2016.
- [97] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pages 2014–2023, 2016. [14](#), [120](#)
- [98] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638, 2016. [14](#), [35](#), [42](#)
- [99] Caiming Xiong, Stephen Merity, and Richard Socher. Dynamic memory networks for visual and textual question answering. In *International conference on machine learning*, pages 2397–2406, 2016. [14](#), [42](#)
- [100] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448, 2015. [14](#), [42](#), [43](#), [95](#)

- [101] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. In *International conference on machine learning*, pages 1378–1387, 2016. [14](#)
- [102] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1409, 2016. [14](#), [43](#), [95](#)
- [103] Huijuan Xu and Kate Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *European Conference on Computer Vision*, pages 451–466. Springer, 2016. [14](#)
- [104] Drew A. Hudson and Christopher D. Manning. Compositional attention networks for machine reasoning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL <https://openreview.net/forum?id=S1Euwz-Rb>. [14](#)
- [105] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. [15](#), [48](#)
- [106] Dan Xu, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 675–684, 2018. [15](#)
- [107] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Sks9_ajex. [15](#)
- [108] Antonio Polino, Razvan Pascanu, and Dan Alistarh. Model compression via distillation and quantization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL <https://openreview.net/forum?id=S1XolQbRW>. [15](#)
- [109] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. [15](#)
- [110] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. [15](#)

- [111] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE, 2016. [15](#)
- [112] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Henry Holt and Co., Inc., New York, NY, USA, 1982. ISBN 0716715678. [15](#), [84](#)
- [113] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *TPAMI*, 2013. [15](#), [84](#)
- [114] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. *arXiv preprint arXiv:1811.12359*, 2018. [15](#), [85](#)
- [115] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. [15](#)
- [116] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. [15](#), [82](#), [89](#), [92](#), [93](#), [139](#), [141](#)
- [117] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018. [15](#)
- [118] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. [15](#), [140](#)
- [119] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 39–48, 2016. [16](#), [84](#), [85](#), [91](#)
- [120] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Learning to compose neural networks for question answering. In *Proceedings of NAACL-HLT*, pages 1545–1554, 2016.
- [121] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 804–813, 2017. [84](#), [85](#)
- [122] Ronghang Hu, Jacob Andreas, Trevor Darrell, and Kate Saenko. Explainable neural computation via stack neural module networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 53–69, 2018. [16](#), [91](#)

- [123] Jiaxin Shi, Hanwang Zhang, and Juanzi Li. Explainable and explicit visual reasoning over scene graphs. *arXiv preprint arXiv:1812.01855*, 2018. [16](#), [82](#), [85](#)
- [124] David Mascharka, Philip Tran, Ryan Soklaski, and Arjun Majumdar. Transparency by design: Closing the gap between performance and interpretability in visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4942–4950, 2018.
- [125] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. In *Advances in Neural Information Processing Systems*, pages 1031–1042, 2018. [16](#)
- [126] Daqing Liu, Hanwang Zhang, Zheng-Jun Zha, and Feng Wu. Explainability by parsing: Neural module tree networks for natural language visual grounding. *arXiv preprint arXiv:1812.03299*, 2018. [16](#)
- [127] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. Mattnet: Modular attention network for referring expression comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1307–1315, 2018. [91](#)
- [128] Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. Modeling relationships in referential expressions with compositional modular networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1115–1124, 2017.
- [129] Daqing Liu, Hanwang Zhang, Feng Wu, and Zheng-Jun Zha. Learning to assemble neural module tree networks for visual grounding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4673–4682, 2019. [16](#), [85](#)
- [130] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, 2016. [17](#), [20](#)
- [131] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *TPAMI*, 2017. [17](#)
- [132] Allan Jabri, Armand Joulin, and Laurens van der Maaten. Revisiting visual question answering baselines. In *ECCV*, 2016. [17](#)
- [133] Bolei Zhou, Yuandong Tian, Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Simple baseline for visual question answering. *arXiv preprint arXiv:1512.02167*, 2015. [17](#)
- [134] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. Visual dialog. In *CVPR*, 2017. [17](#), [82](#), [111](#)

- [135] Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. Modeling relationships in referential expressions with compositional modular networks. In *CVPR*, 2017. [17](#), [21](#)
- [136] Hanwang Zhang, Yulei Niu, and Shih-Fu Chang. Grounding referring expressions in images by variational context. In *CVPR*, 2018. [17](#)
- [137] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *CVPR*, 2017. [18](#)
- [138] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.
- [139] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, pages 1422–1430, 2015.
- [140] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. [22](#)
- [141] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016. [18](#)
- [142] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. [18](#), [21](#), [83](#)
- [143] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. [19](#), [22](#)
- [144] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jungkwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *ICML*, 2017.
- [145] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017.
- [146] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *ICCV*, 2017. [19](#), [22](#)
- [147] Bo Dai, Yuqi Zhang, and Dahua Lin. Detecting visual relationships with deep relational networks. In *CVPR*, 2017. [20](#)
- [148] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. [20](#)
- [149] Ross Girshick. Fast r-cnn. In *ICCV*, 2015. [20](#)

- [150] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017. [21](#)
- [151] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. [21](#), [36](#), [49](#), [79](#), [82](#), [83](#), [97](#), [115](#)
- [152] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schuster, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, 2018. [22](#)
- [153] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015. [24](#)
- [154] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017. [24](#)
- [155] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. [24](#), [74](#), [98](#), [126](#)
- [156] Ruotian Luo, Brian Price, Scott Cohen, and Gregory Shakhnarovich. Discriminability objective for training descriptive captions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6964–6974, 2018. [33](#)
- [157] Xu Yang, Hanwang Zhang, and Jianfei Cai. Learning to collocate neural modules for image captioning. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. [33](#), [51](#), [56](#), [57](#)
- [158] Ronghang Hu, Piotr Dollár, Kaiming He, Trevor Darrell, and Ross Girshick. Learning to segment every thing. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. [33](#)
- [159] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollar. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. [33](#)
- [160] David Marr. Vision: A computational investigation into the human representation and processing of visual information. mit press. *Cambridge, Massachusetts*, 1982. [34](#)
- [161] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, 2017. [34](#)
- [162] Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Babytalk: Understanding and generating simple image descriptions. In *CVPR*, 2011. [34](#), [119](#)

- [163] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. From captions to visual concepts and back. In *CVPR*, 2015. 34
- [164] Diego Marcheggiani and Ivan Titov. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, 2017. 35
- [165] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard S. Zemel. Gated graph sequence neural networks. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL <http://arxiv.org/abs/1511.05493>. 35
- [166] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015. 38, 50, 64, 73, 75, 97, 101, 126, 128
- [167] Wenhao Jiang, Lin Ma, Yu-Gang Jiang, Wei Liu, and Tong Zhang. Recurrent fusion network for image captioning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 499–515, 2018. 44, 51, 56, 57, 106, 132, 133, 134
- [168] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL <http://arxiv.org/abs/1412.6980>. 45, 49
- [169] Yan Zhang, Jonathon S. Hare, and Adam Prügel-Bennett. Learning to count objects in natural images for visual question answering. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=B12Js_yRb. 46
- [170] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019. 48
- [171] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *NIPS*, 2014. 48
- [172] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015. 49, 97, 126

- [173] Alexander Mathews, Lexing Xie, and Xuming He. Senticap: generating image descriptions with sentiments. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 3574–3580, 2016. [49](#), [53](#)
- [174] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002. [50](#), [75](#), [101](#), [128](#)
- [175] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005. [50](#), [75](#), [101](#), [128](#)
- [176] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004. [50](#), [101](#), [128](#)
- [177] Daqing Liu, Zheng-Jun Zha, Hanwang Zhang, Yongdong Zhang, and Feng Wu. Context-aware visual policy network for sequence-level image captioning. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 1416–1424. ACM, 2018. [57](#), [106](#), [132](#), [134](#)
- [178] Xu Yang, Hanwang Zhang, and Jianfei Cai. Learning to collocate neural modules for image captioning. *arXiv preprint arXiv:1904.08608*, 2019. [61](#), [92](#), [99](#), [105](#), [106](#), [107](#), [120](#), [133](#), [134](#)
- [179] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4584–4593, 2016. [62](#)
- [180] Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057*, 2015.
- [181] Rui Lin, Shujie Liu, Muyun Yang, Mu Li, Ming Zhou, and Sheng Li. Hierarchical recurrent neural network for document modeling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 899–907, 2015. [62](#)
- [182] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*, 2017. [70](#)
- [183] Marc’Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015. [72](#)

- [184] Qingzhong Wang and Antoni B Chan. Describing like humans: on diversity in image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4195–4203, 2019. [75](#), [77](#)
- [185] Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object hallucination in image captioning. *arXiv preprint arXiv:1809.02156*, 2018. [82](#), [101](#), [110](#), [120](#), [129](#)
- [186] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 1988–1997. IEEE, 2017. [82](#), [111](#)
- [187] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. *arXiv preprint arXiv:1812.01880*, 2018. [82](#)
- [188] Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei. A hierarchical approach for generating descriptive image paragraphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 317–325, 2017. [82](#)
- [189] Yulei Niu, Hanwang Zhang, Manli Zhang, Jianhong Zhang, Zhiwu Lu, and Ji-Rong Wen. Recursive visual attention in visual dialog. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. [82](#)
- [190] Nikita Kitaev and Dan Klein. Constituency parsing with a self-attentive encoder. *arXiv preprint arXiv:1805.01052*, 2018. [92](#)
- [191] Kristina Toutanova and Christopher D Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13*, pages 63–70. Association for Computational Linguistics, 2000. [94](#)
- [192] Hugo Liu and Push Singh. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226, 2004. [95](#), [98](#), [122](#), [126](#)
- [193] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016. [99](#)
- [194] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017. [110](#), [111](#)
- [195] William J Reed. The pareto, zipf and other power laws. *Economics letters*, 74(1):15–19, 2001. [110](#), [142](#)

- [196] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2537–2546, 2019. [110](#)
- [197] Jonathan Gordon and Benjamin Van Durme. Reporting bias and knowledge extraction. 2013. [110](#)
- [198] Ishan Misra, C Lawrence Zitnick, Margaret Mitchell, and Ross Girshick. Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2930–2939, 2016. [110](#)
- [199] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in neural information processing systems*, pages 4349–4357, 2016. [110](#)
- [200] Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356 (6334):183–186, 2017. [110](#)
- [201] Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don’t just assume; look and answer: Overcoming priors for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4971–4980, 2018. [111](#)
- [202] Remi Cadene, Corentin Dancette, Hedi Ben-younes, Matthieu Cord, and Devi Parikh. Rubi: Reducing unimodal biases in visual question answering. *arXiv preprint arXiv:1906.10169*, 2019. [111](#)
- [203] Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016. [111](#), [113](#), [114](#), [117](#)
- [204] Judea Pearl and Dana Mackenzie. *The Book of Why*. Basic Books, New York, 2018. ISBN 978-0-465-09760-9. [111](#), [113](#)
- [205] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6700–6709, 2019. [111](#)
- [206] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *International Conference on Computer Vision*, 2019. [113](#), [119](#), [125](#), [133](#), [134](#)
- [207] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *arXiv preprint arXiv:2004.07780*, 2020. [114](#)

- [208] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Hierarchy parsing for image captioning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2621–2629, 2019. [119](#), [132](#), [133](#)
- [209] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and vqa. *arXiv preprint arXiv:1909.11059*, 2019. [119](#), [132](#), [133](#)
- [210] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10578–10587, 2020. [119](#), [139](#)
- [211] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. [123](#)
- [212] Pierre Baldi and Peter Sadowski. The dropout learning algorithm. *Artificial intelligence*, 210:78–122, 2014. [123](#)
- [213] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*, pages 689–696. ACM, 2009. [124](#)
- [214] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 933–941. JMLR.org, 2017. [125](#)
- [215] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993. URL <https://www.aclweb.org/anthology/J93-2004>. [139](#)
- [216] Naiwen Xue, Fei Xia, Fu-Dong Chiou, and Marta Palmer. The penn chinese treebank: Phrase structure annotation of a large corpus. *Natural language engineering*, 11(2):207, 2005. [139](#)
- [217] Muhua Zhu, Yue Zhang, Wenliang Chen, Min Zhang, and Jingbo Zhu. Fast and accurate shift-reduce constituent parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 434–443, 2013. [139](#)
- [218] Danqi Chen and Christopher D Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 740–750, 2014.

- [219] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3716–3725, 2020. [139](#)
- [220] Yingwei Pan, Ting Yao, Yehao Li, and Tao Mei. X-linear attention networks for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10971–10980, 2020. [139](#)
- [221] Simao Herdade, Armin Kappeler, Kofi Boakye, and Joao Soares. Image captioning: Transforming objects into words. In *Advances in Neural Information Processing Systems*, pages 11137–11147, 2019.
- [222] Guang Li, Linchao Zhu, Ping Liu, and Yi Yang. Entangled transformer for image captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. [139](#)
- [223] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6281–6290, 2019. [139](#)
- [224] Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10502–10511, 2019. [139](#)
- [225] Xihui Liu, Zihao Wang, Jing Shao, Xiaogang Wang, and Hongsheng Li. Improving referring expression grounding with cross-modal attention-guided erasing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1950–1959, 2019. [139](#)
- [226] Chris Alberti, Jeffrey Ling, Michael Collins, and David Reitter. Fusion of detected objects in text for visual question answering. *arXiv preprint arXiv:1908.05054*, 2019. [139](#)
- [227] Noam Chomsky. *Aspects of the Theory of Syntax*, volume 11. MIT press, 2014. [139](#)
- [228] Zhuowen Tu, Xiangrong Chen, Alan L Yuille, and Song-Chun Zhu. Image parsing: Unifying segmentation, detection, and recognition. *International Journal of computer vision*, 63(2):113–140, 2005. [139](#)
- [229] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyu Zhou, Wenhui Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In *Advances in Neural Information Processing Systems*, pages 5243–5253, 2019. [139](#)

- [230] Mingzhou Xu, Derek F Wong, Baosong Yang, Yue Zhang, and Lidia S Chao. Leveraging local and global patterns for self-attention networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3069–3075, 2019.
- [231] Baosong Yang, Jian Li, Derek F Wong, Lidia S Chao, Xing Wang, and Zhaopeng Tu. Context-aware self-attention networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 387–394, 2019. [139](#)
- [232] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001. [139](#)
- [233] Christopher M Bishop. *Pattern recognition and machine learning*. Springer Science+ Business Media, 2006. [139](#)
- [234] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. [140](#)
- [235] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2020. [140](#)
- [236] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. [141](#)
- [237] Mark Chen, Alec Radford, Rewon Child, Jeff Wu, Heewoo Jun, Prafulla Dhariwal, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *Proceedings of the 37th International Conference on Machine Learning*, volume 1, 2020. [141](#)
- [238] Raphael Suter, Djordje Miladinovic, Bernhard Schölkopf, and Stefan Bauer. Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. In *International Conference on Machine Learning*, pages 6056–6065. PMLR, 2019. [142](#)
- [239] Donald B Rubin. Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association*, 100(469): 322–331, 2005. [143](#)
- [240] Judea Pearl and Elias Bareinboim. External validity: From do-calculus to transportability across populations. *Statistical Science*, pages 579–595, 2014. [143](#)
- [241] Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4): 669–688, 1995. [143](#)