VISUALLY GROUNDED LANGUGAGE UNDERSTANDING AND GENERATION

A Dissertation
Presented to
The Academic Faculty

By

Jiasen Lu

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Interactive Computing

Georgia Institute of Technology

May 2020

VISUALLY GROUNDED LANGUAGE UNDERSTANDING AND GENERATION

Approved by:

Dr. Devi Parikh
College of Computing
Georgia Institute of Technology

Dr. Dhruv Batra
College of Computing
Georgia Institute of Technology

Dr. Mark Riedl
College of Computing
Georgia Institute of Technology

Dr. Judy Hoffman
College of Computing
Georgia Institute of Technology

Dr. Jason J. Corso
Department of Electrical Engineering and Computer Science
University of Michigan

Date Approved: Jan 6, 2020

To my parents, Zhenshui Lu and Zengguo Li

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor, Devi Parikh, for guiding and supporting me through my research. The works in this dissertation would not have been possible without her vision, immense knowledge, openness and determination. Thanks for believing in and providing me an opportunity to start the research journey five years ago. I still remember the excitement I felt the first time I visited Virginia Tech. Thanks for teaching me first hand the values of doing creative and original research, and for giving me the freedom and flexibility to find out my own research interests. I could not have imagined having a better advisor and mentor for my Ph.D. study.

I would also like to thank Dhruv Batra, for all the advice and support through the years. I always felt lucky that I have had not only one, but two advisors. Since my first project, Dhruv has provided a tremendous amount of invaluable advice. Thanks for teaching amazing and fun classes on deep learning at Virginia Tech. Thanks for scheduling the lab retreat, where we can think about our goal of life besides the research.

I would like to thank my committee, Jason Corso, Mark Riedl, and Judy Hoffman. Their insightful comments and suggestions leveraging knowledge from multiple areas pushed me to think more broadly and helped me polish the ideas and the dissertation.

I would also like to thank Caiming Xiong for being a wonderful mentor and internship advisor, shaping so many of my tastes and outlook around the research. Thanks for sharing your vision across my Ph.D. study. I would like to thank Stefan Lee for all the advice and supports over the last few years. Thanks for hosting the weekly board game session and I really enjoyed all the games we played together!

I would like to thank Jianwei Yang for being a great collaborator and roommate, and for being a constant presence through good and bad times. I would like to thank

Ying Ding, Sheng Li, and Chao Lan. Without them, I could not study computer vision and machine learning during my undergraduate studies.

I would like to thank all my collaborators: Michael Cogswell, Xiao Lin, Stanislaw Antol, Aishwarya Agarwal, Vedanuj Goswami – as well as everyone in the Computer Vision and Machine Learning and Perception (CVMLP) Labs. I would like to thank Qing Sun, Peng Zhang, Zhile Re, Ramprasaath Selvaraju, Ramakrishna Vedantam, Yash Goyal, Akrit Mohapatra, Harsh Agarwal, Abhishek Das, Arjun Chandrasekharan, Ashwin Kalyan, Prithvijit Chattopadhyay, Samyak Datta, Nirbhay Modhe, Peter Anderson for all the brainstorming sessions over the years and for being a part of an amazing community in the CVMLP labs.

TABLE OF CONTENTS

Acknowledgments v

List of Tables xiii

List of Figures xvii

Chapter 1: Introduction 1

1.1 Visual Question Answering 3

1.1.1 Dataset and Baselines for VQA 4

1.1.2 Hierarchical Co-Attention for Image Question Grounding . . . 5

1.2 Neural Image captioning 5

1.2.1 Knowing When to Look for Image Caption Generation 6

1.2.2 Neural Baby Talk: Explicitly Grounding on Object Detection 7

1.3 Visual Dialog 9

1.3.1 Discriminant Perceptual Loss for Visual Dialog 9

1.3.2 Learning Image-Discriminative Dialog Policies from VQA . . . 11

1.4 Vision and Language Pretraining 12

1.4.1 ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations 13

1.4.2 12-in-1: Multi-Task Vision and Language Representation Learning 14

Chapter 2: Background 16

2.1 Background: Visual Question Answering 16

2.2 Background: Neural Image Captioning 17

2.3 Background: Visual Dialog 19

2.4 Background: Bidirectional Encoder Representations from Transformers 20

Chapter 3: Situating the Work 23

3.1 Visual Question Answering 23

3.1.1 VQA datasets 23

3.1.2 Image Attention 24

3.1.3 Language Attention 25

3.2 Neural Image Captioning 26

3.2.1 Template-based approaches 26

3.2.2 Neural-based approaches 26

3.3 Visual Dialog 27

3.3.1 GANs for sequence generation 27

3.3.2 Modeling image and text attention in visual dialog 28

3.3.3 Visual Question Generation 28

3.3.4 Latent Action Spaces for dialog generation 29

3.3.5 Reference Games. 29

3.4 Vision and Language Pretraining 30

3.4.1 Self-Supervised Learning 30

3.4.2 Recent Works on Vision-And-Language Pretraining 30

3.4.3 Multi-Task Learning. 31

3.4.4 Multi-Task V&L Learning. 31

Chapter 4: Visual Question Answering 33

4.1 VQA Baselines and Methods 33

4.1.1 Baselines 33

4.1.2 Methods 34

4.1.3 Results 37

4.2 Hierarchical Co-Attention for Visual Question Answering 42

4.2.1 Approach 43

4.2.2 Results 48

4.2.3 Discussion 54

Chapter 5: Neural Image Captioning 55

5.1 Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning 55

5.1.1 Approach 57

5.1.2 Implementation Details 61

5.1.3 Results 62

5.1.4 Discussion 70

5.2 Neural Baby Talk 70

5.2.1 Approach 72

5.2.2 Implementation Details 77

5.2.3 Experimental Results 79

5.2.4 Discussion . . . 86

Chapter 6: Visual Dialog . . . 87

6.1 Best of Both Worlds: Transferring Knowledge from Discriminative Learning to a Generative Visual Dialog Model . . . 88

6.1.1 Approach: Backprop Through Discriminative Losses for Generative Training . . . 90

6.1.2 Experiments . . . 95

6.1.3 Discussion . . . 101

6.2 Dialog without Dialog: Learning Image-Discriminative Dialog Policies from Single-Shot Question Answering Data . . . 102

6.2.1 Dialog-based Image Guessing Game . . . 104

6.2.2 Dialog without Dialog . . . 108

6.2.3 Experiments . . . 111

6.2.4 Discussion . . . 121

Chapter 7: Vision and Language Pretraining . . . 122

7.1 ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks . . . 123

7.1.1 ViLBERT . . . 124

7.1.2 Experimental Settings . . . 127

7.1.3 Results and Analysis . . . 131

7.1.4 Discussion . . . 135

7.2 12-in-1: Multi-Task Vision and Language Representation Learning . . . 135

7.2.1 Vision-and-Language Tasks . . . 137

7.2.2 Approach . . . 139

7.2.3 Experiments and Results . . . 144

7.2.4 Analysis and Ablation Study . . . 149

7.2.5 Discussion . . . 152

Chapter 8: Conclusion . . . 154

Appendix A: Appendix for Knowing When to Look . . . 157

A.1 COCO Categories Mapping List for Weakly-Supervised Localization . 157

A.2 Adaptive attention across different datasets . . . 157

A.3 More Visualization of Attention . . . 158

A.3.1 Visualization of Weakly Supervised Localization . . . 158

Appendix B: Appendix for Neural Baby Talk . . . 161

Appendix C: Appendix for Dialog without Dialog . . . 165

C.1 Additional Results . . . 165

C.2 Mechanical Turk Studies . . . 165

Appendix D: Appendix for Multi-Task Vision and Language Representation Learning . . . 169

D.1 Datasets . . . 169

D.2 Improvements over ViLBERT Pretraining . . . 169

D.3 Model Architecture . . . 171

D.4 Implementation Details . . . 171

D.5 Multi-Task Training . . . 172

D.6 Comparison with other SOTA . . . 172

D.7 Full Breakdown of Ablation Study 173

D.8 Multi-task visual grounding consistency 174

D.9 Qualitative Results 176

D.10 Attention Visualizations 176

References 198

List of Publications 199

LIST OF TABLES

4.1 Accuracy of our methods for the open-ended and multiple-choice tasks on the VQA test-dev for real images. Q = Question, I = Image, C = Caption. (Caption and BoW Q + C results are on val). See text for details. 37

4.2 Open-ended test-dev results for different question types on real images (Q+C is reported on val). Machine performance is reported using the bag-of-words representation for questions. Questions types are determined by the one or two words that start the question. The percentage of questions for each type is shown in parentheses. Last and second last columns respectively show the average human age and average degree of commonsense required to answer the questions (as reported by AMT workers), respectively. See text for details. 39

4.3 Accuracy of ablated versions of our best model (deeper LSTM Q + norm I) for the open-ended and multiple-choice tasks on the VQA test-dev for real images. Q = Question, I = Image. See text for details. 42

4.4 Results on the VQA dataset. "-" indicates the results is not available. 50

4.5 Results on the COCO-QA dataset. "-" indicates the results is not available. 51

4.6 Ablation study on the VQA dataset using Ours^a+VGG. 52

5.1 Performance on Flickr30k test splits. † indicates ensemble models. **B-n** is BLEU score that uses up to n-grams. Higher is better in all columns. For future comparisons, our ROUGE-L/SPICE Flickr30k scores are 0.467/0.145 63

5.2 Performance on COCO test splits. † indicates ensemble models. **B-n** is BLEU score that uses up to n-grams. Higher is better in all columns. For future comparisons, our ROUGE-L/SPICE COCO scores are 0.549/0.194 63

5.3 Performance on the test portion of Karpathy *et al.* [19]'s splits on Flickr30k Entities dataset. 81

5.4 Performance on the test portion of Karpathy *et al.* [19]'s splits on COCO dataset. $*$ directly optimizes the CIDEr Metric, † uses better image features, and are thus not directly comparable. 81

5.5 Performance on the test portion of the robust image captioning split on COCO dataset. 83

5.6 Evaluation of captions generated using the proposed method. G means greedy decoding, and T1−2 means using constrained beam search [168] with 1−2 top detected concepts. $*$ is the result using VGG-16 [172] and † is the result using ResNet-101. 85

6.1 Results (generative) on VisDial dataset. 99

6.2 Results (discriminative) on VisDial dataset. 99

6.3 Adversarial training results on VisDial dataset. 101

6.4 Performance of our models and baselines in different experimental settings. From setting `A` to setting `F`, agents are tasked with generalizing further from the source data. Our method strikes a balance between guessing game performance and interpretability. 113

6.5 Human evaluation of language quality – question fluency (top) and relevance (bottom). Each row compares a pair of agent-generated questions, asking users which (or possibly neither) is more fluent/relevant. The values report the percentage of times the option represented by that column was chosen. 115

6.6 Various ablations of our training curriculum. 119

7.1 Transfer task results for our ViLBERT model compared with existing state-of-the-art and sensible architectural ablations. † indicates models without pretraining on Conceptual Captions. For VCR and VQA which have private test sets, we report test results (in parentheses) only for our full model. Our full ViLBERT model outperforms task-specific state-of-the-art models across all tasks. 133

7.2 Ablation study of the depth of our model with respect to the number of Co-TRM→TRM blocks (shown in a dashed box in Fig. 7.1). We find that different tasks perform better at different network depths – implying they may need more or less context aggregation. 134

7.3 Transfer task results for ViLBERT as a function of the percentage of the Conceptual Captions dataset used during pre-training. We see monotonic gains as the pretraining dataset size grows. 134

7.4 Percentage of row-task test images that are present in column-tasks train/val images. 138

7.5 Comparison of our multi-task models to single-task performance. We find multi-task training (rows 3-5) provides significant gains over single-task training (rows 1-2) while reducing the parameter count from over 3 billion to 270 million. Further, following multi-task training by task-specific fine-tuning (rows 6-9) further gains can be made at the cost of increased parameters. 143

7.6 Pair-wise (left) and triple-wise (right) inter-group representative task analysis. Each entry is the relative performance change from single-task training for the row-task when jointly trained with the column-task(s). 144

7.7 Comparison to recent SOTA. For image retrieval (IR) COCO and Flickr we report R1 scores on the 1K test set. 147

7.8 Comparison with other multi-task models. VQA score is on test-dev and the retrieval tasks on their respective 1K test split. For Flickr Grounding (FG) we report R1 on Flickr30K test. 148

7.9 Ablations on our design choices and comparison to curriculum and anti-curriculum learning multi-task approaches. 150

A.1 Visual grounding probabilities of the same word on COCO and Flickr30K datasets. 158

B.1 COCO category mapping list for visual words. 162

B.2 COCO category mapping list for visual words (continued). 163

B.3 COCO category mapping list for visual words (continued). 163

D.1 Number of images in the train+val and test sets before and after cleaning. We use the training part of the cleaned dataset in the multi-task experiments. Note that this is not the number of training samples but the number of images in the dataset. 170

D.2 Training details including sample sizes, testing metric and hyperparameters for single task and multi-task training. 173

D.3 Comparison of Ours$_{\texttt{ST}}$ (Table. 7.5 `Row 1`) and Ours$_{\texttt{AT->ST}}$ (Table. 7.5 `Row 8`) models on full dataset with other SOTA methods. Results for RefCOCO and RefCOCO+ are reported on the full test split (testA + testB). Refer to Sec D.6 for more details. 175

D.4 Full per task accuracy for the different ablation studies. RC is Retrieval COCO and RF is Retrieval Flickr30k. Mean of G2 is taken over the Recall@1 scores. We can see that with task token per dataset and `DSG` achieve the best performance. 175

D.5 Full per task accuracy for Fig. 7.6 showing different Dynamic Stop-and-Go Iteration Gaps (Δ). Mean of G2 is taken over the Recall@1 scores. 176

LIST OF FIGURES

1.1 Examples of free-form, open-ended questions collected for images via Amazon Mechanical Turk. Note that commonsense knowledge is needed along with a visual understanding of the scene to answer many questions. 4

1.2 Examples of image captions. Note that there are visual words (in blue) and non-visual words (in black) based on whether the words have an explicit visual corresponding in the image. We only consider Noun and Adjective words in the examples. 6

1.3 From left to right is the generated caption using the same captioning model but with different detectors: 1) No detector; 2) A weak detector that only detects "person" and "sandwich"; 3) A detector trained on COCO [5] categories (including "teddy bear"). 4) A detector that can detect novel concepts (e.g. "Mr. Ted" and "pie" that never occurred in the captioning training data). Different colors show a correspondence between the visual word and grounding regions. 8

1.4 (Top - 2 pools) We train our questioner to ask questions that can discriminate between pairs of images by mimicing questions from the VQAv2 dataset. (Bottom - 1 pool) Our proposed model generalizes to new settings in a way that humans can understand without additional language supervision (*i.e.*, without dialog). 12

2.1 Standard encoder transformer block. 21

4.1 Our best performing model (deeper LSTM Q + norm I). This model uses a two layer LSTM to encode the questions and the last hidden layer of VGGNet [139] to encode the images. The image features are then ℓ_2 normalized. Both the question and image features are transformed to a common space and fused via element-wise multiplication, which is then passed through a fully connected layer followed by a softmax layer to obtain a distribution over answers. 35

4.2 Pr (system is correct | answer) for 50 most frequent ground truth answers on the VQA validation set (plot is sorted by accuracy, not frequency). System refers to our best model (deeper LSTM Q + norm I). 40

4.3 Pr (answer | system is correct) for 50 most frequently predicted answers on the VQA validation set (plot is sorted by prediction frequency, not accuracy). System refers to our best model (deeper LSTM Q + norm I). 40

4.4 Flowchart of our proposed hierarchical co-attention model. Given a question, we extract its word level, phrase level and question level embeddings. At each level, we apply co-attention on both the image and question. The final answer prediction is based on all the co-attended image and question features. 43

4.5 (a) Parallel co-attention mechanism; (b) Alternating co-attention mechanism. 45

4.6 (a) Hierarchical question encoding (Sec. 4.2.1); (b) Encoding for predicting answers (Sec. 4.2.1). 47

4.7 Visualization of image and question co-attention maps on the COCO-QA dataset. From left to right: original image and question pairs, word level co-attention maps, phrase level co-attention maps and question level co-attention maps. For visualization, both image and question attentions are scaled (from red:high to blue:low). Best viewed in color. 53

5.1 Our model learns an adaptive attention model that automatically determines when to look (**sentinel gate**) and where to look (**spatial attention**) for word generation. 56

5.2 A illustration of soft attention model from [18] (a) and our proposed spatial attention model (b). 58

5.3 An illustration of the proposed model generating the t-th target word y_t given the image. 60

5.4 Visualization of generated captions and image attention maps on the COCO dataset. Different colors show a correspondence between attended regions and underlined words. First 2 rows are success cases, last row are failure examples. Best viewed in color. 65

5.5 Visualization of generated captions, visual grounding probabilities of each generated word, and corresponding spatial attention maps produced by our model. 66

5.6 Rank-probability plots on COCO (left) and Flickr30k (right) indicating how likely a word is to be visually grounded when it is generated in a caption. 67

5.7 Localization accuracy over generated captions for top 45 most frequent COCO object categories. “Spatial Attention” and “Adaptive Attention” are our proposed spatial attention model and adaptive attention model, respectively. The COCO categories are ranked based on the align results of our adaptive attention, which cover 93.8% and 94.0% of total matched regions for spatial attention and adaptive attention, respectively. 68

5.8 From left to right is the generated caption using the same captioning model but with different detectors: 1) No detector; 2) A weak detector that only detects “person” and “sandwich”; 3) A detector trained on COCO [5] categories (including “teddy bear”). 4) A detector that can detect novel concepts (e.g. “Mr. Ted” and “pie” that never occurred in the captioning training data). Different colors show a correspondence between the visual word and grounding regions. 71

5.9 One block of the proposed approach. Given an image, proposals from any object detector and current word “A”, the figure shows the process to predict the next visual word “cat”. 74

5.10 Language model used in our approach. 79

5.11 Generated captions and corresponding visual grounding regions on the standard image captioning task (Top: COCO, Bottom: Flickr30k). Different colors show a correspondence between the visual words and grounding regions. Grey regions are the proposals not selected in the caption. First 3 columns show success and last column shows failure cases (words are grounded in the wrong region). 80

5.12 Generated captions and corresponding visual grounding regions for the robust image captioning task. “cat-remote”, “man-bird”, “dog-skateboard” and “orange-bird” are co-occurring categories excluded in the training split. First 3 columns show success and last column shows failure case (orange was not mentioned). 83

5.13 Generated captions and corresponding visual grounding regions for the novel object captioning task. “zebra”, “tennis racket”, “bus” and “pizza” are categories excluded in the training split. First 3 columns show success and last column shows a failure case. 84

6.1 (a): Model architecture (b): Given the image, history, and question, D's score for different candidate answers. Note that the multiple plausible responses all score high. The candidate in the blue box is the true response and in green is the response generated by G. 89

6.2 Structure of the proposed encoder. 92

6.3 Qualitative comparison. “Ours” are samples from G-DIS model with different gumbel noise z. Images from the COCO dataset 101

6.4 A single round of our q-bot which decomposes into the modules described in Section 6.2.1. This factorization allows us to fine-tune just the intention of the model for task performance, limiting language drift. 106

6.5 Qualitative comparison of dialogs generated by our model with those generated by Non-Var Cont and Stage 1 baselines. Top / middle /bottom rows are image pool from COCO / AWA / CUB images respectively. Our model pretrained on VQA (COCO image) generates more interpretable questions for the DwD task which is semantic meaning and generalize well to out-of-domain images. 117

7.1 Our ViLBERT model consists of two parallel streams for visual (green) and linguistic (purple) processing that interact through novel co-attentional transformer layers. This structure allows for variable depths for each modality and enables sparse interaction through co-attention. Dashed boxes with multiplier subscripts denote repeated blocks of layers. . . 124

7.2 Our co-attention transformer layer 125

7.3 We train ViLBERT on the Conceptual Captions [38] dataset under two training tasks to learn visual grounding. In masked multi-modal learning, the model must reconstruct image region categories or words for masked inputs given the observed inputs. In multi-modal alignment prediction, the model must predict whether or not the caption describes the image content. 126

7.4 Examples for each vision-and-language task we transfer ViLBERT to in our experiments. 129

7.5 We introduce an approach for effective multi-task learning, training a single model on 12 popular vision-and-language datasets. This single model performs at par or even better than independent task-specific state-of-the-art approaches for many tasks. 136

7.6 Left: Visualization of Dynamic stop-and-go during multi-task training. Solid line indicates in the `go` mode while thin line indicates `stop` mode. Right: Mean accuracy (normalized group-wise for easier comparison) for each group with different iter-gap Δ for Dynamic stop-and-go . . 150

7.7 Multi-Task training acts as a regularizer. 153

7.8 Our single model (Our$_{\texttt{AT}}$) can perform a multitude of V&L tasks: caption and image retrieval, question answering, grounding phrases, guessing image regions based on a dialog, verifying facts about a pair of images, natural language inferences from an image, etc. Here we show outputs of our model for a variety of inputs (that mimic tasks from the 12 datasets it has been trained on). 153

A.1 Image attention visualization of word "of" on several images. For each image pair, left: output of spatial attention model (no visual sentinel), right: output of our adaptive attention model (with visual sentinel). . 157

A.2 Visualization of generated captions and image attention maps on the COCO dataset. Different colors show a correspondence between attended regions and underlined words. 159

A.3 Example of generated caption, spatial attention and visual grounding probability. 159

A.4 Visualization of generated captions and weakly supervised localization result. Red bounding box is the ground truth annotation, blue bounding box is the predicted location using spatial attention map. 160

C.1 Task performance (guessing game accuracy) over rounds of dialog. Performance increases over rounds for all models except the Stage 1 models.166

C.2 167

C.3 168

C.4 168

D.1 Architecture of the our model for V&L multi-task learning. We augment the input query with a task token to learn the task-aware feature embedding. 172

D.2 Training curves on *train* set for $\text{Ours}_{\texttt{ST}}$ (Table 7.5 `Row 2`) vs $\text{Ours}_{\texttt{AT}}$ (Table 7.5 `Row 4`) models for all the 12 tasks in our experiments. Green lines show single-task training($\text{Ours}_{\texttt{ST}}$) and blue lines show multi-task training($\text{Ours}_{\texttt{AT}}$). Note that all these training are with the *Clean V&L* setup. We can observe that for some of the tasks the training for $\text{Ours}_{\texttt{ST}}$ are shorter as they have fewer number of iterations when trained alone. Please refer to Sec. D.5 for more details. 174

D.3 **Our single multi-task model can solve multiple task consistently and correctly.** Additional qualitative examples of our single model $\text{Our}_{\texttt{AT}}$ on multitude of V&L tasks: caption and image retrieval, question answering, grounding phrases, guessing image regions based on a dialog, verifying facts about a pair of images, natural language inferences from an image, etc. Here we show outputs of our model for a variety of inputs (that mimic tasks from the 12 datasets it has been trained on). 177

D.4 Failure cases of our single `AT` model on multitude of V&L tasks. Failure cases mostly occur when the model encounters `counting` questions or difficult referring expressions and phrases for fine grained recognition. 178

D.5 Visualizations of image to sentence attention for the pretrained model on conceptual caption dataset. Given the image to sentence co-attention, we use the most attended word to represent its semantic meaning, and show the patches corresponding to the visual words ('boy', 'mom', 'pet', 'white', 'sheep'). Different colors show a correspondence between attended regions and underlined words. We can see that the model learns meaningful concept through pretraining. 179

D.6 Visualizations of the attentions of the pretrained model on conceptual caption dataset using BertVis toolbox. From left to right: Image and associate caption, sentence to sentence self-attention for all layers and all heads, sentence to sentence self-attention for Layer 1 Head 1 and Layer 2 Head 0. Our model learns the previous words attention pattern, bag of words attention pattern and next words attention pattern. . . 179

D.7 Visualizations of the attentions of Our$_{\texttt{AT}}$ model using BertVis toolbox on each tasks. From left to right are image and associate sentence, sentence to sentence self-attention, sentence to image co-attention image to sentence co-attention image to image self-attention. Dashed orange bounding boxes in the image are the referring expression outputs regardless of tasks. The model learns to use significant different sentence to sentence self-attention pattern for different tasks. 180

D.8 Visualizations of the attentions of Our$_{\texttt{AT}}$ model using BertVis toolbox on each tasks. From left to right are image and associate sentence, sentence to sentence self-attention, sentence to image co-attention image to sentence co-attention image to image self-attention. Dashed orange bounding boxes in the image are the referring expression outputs regardless of tasks. The model learns to use significant different sentence to sentence self-attention pattern for different tasks. 181

Thesis Statement

Inducing appropriate grounding in models improves multi-modal AI capabilities. We can induce appropriate grounding in models via novel attention mechanisms, using outputs of other vision tasks, novel training paradigms and learning from external large-scale data.

SUMMARY

The goal of this thesis is to study how inducing appropriate *grounding* improves multi-modal AI capabilities in the context of 'vision-and-language'. In pursuit of this overarching goal, I will look at these four tasks: visual question answering, neural image captioning, visual dialog and vision and language pretraining.

In visual question answering, we collected a large scale visual question answering dataset and I will study various baselines to benchmark these tasks. To jointly reason about image and question, I then propose a novel co-attention mechanism that can learn fine-grained grounding to answer the question.

In the second part, I will address the model designs for grounded caption generation given an image. A key focus will be to augment the model with the ability to know when to look at the image when generating each word. For the words which have explicit visual correspondence (*e.g.*, 'puppy' and 'tie'), we further proposed a novel approach that reconciles classical slot filling approaches with modern neural captioning approaches. As a result, our model can produce natural language explicitly grounded in entities that object detectors find in the image.

In the third part, I will explore the training paradigms to learn better visual grounding for visual dialog. I will study both sides of the visual dialog agents – questioner and answerer. For modeling answerer which answers visual questions in dialog, we will introduce a novel discriminant perceptual loss that transfers knowledge from a discriminative model a generative model. For modeling questioner, we will consider an image guessing game as a test-bed for balancing task performance and language drift. We propose a Dialog without Dialog task, which requires agents to generalize from single round visual question generation with full supervision to a multi-round dialog-based image guessing game without direct language supervision. We will study a new training paradigm that first learns "how to speak" and then

learns ”what to speak”. Our visually-grounded dialog models that can adapt to new tasks while exhibiting less linguistic drift.

Finally, we will study more general multi-modal AI models that can learn visual groundings from massive meta-data on the internet. Our work represents a shift away from learning groundings between vision and language only as part of task training and towards treating visual grounding as a pretrainable and transferable capability. We will also explore the multi-task vision and language representation learning. Our results not only show that a single model can perform all 12 vision and language tasks, but also that joint training can lead to improvements in task metric compared to single-task training with the same architecture.

CHAPTER 1
INTRODUCTION

"... spend the summer linking a camera to a computer and getting the computer to describe what it saw."

The goal of a 1966 first-year undergraduate summer research project for Gerald Sussman [1]

The world around us involves multiple modalities – we see objects, feel texture, hear sounds, smell odors and so on. In order for Artificial Intelligence (AI) to make progress in understanding the world around us, it needs to be able to interpret and reason about multiple modalities. In 1966, Minsky at MIT asked his undergraduate student to let the computer describe what is saw. Since this now famously ambitious summer project, steady progress has been made towards systems that can demonstrate their visual understanding by generating or responding to natural language in the context of images, videos, or even full 3D environments [2, 3, 4, 5, 6, 7]. These approaches and corresponding tasks have come to be referred to under the common banner of 'vision-and-language'.

In recent years, the advent of deep learning techniques has resulted in exciting progress on individual aspects of this problem. One the vision side alone, driven by the advance in training deep Convolutional Neural Networks (CNN) [8], machines can now reliably recognize whether an image or video contains one of over a thousand object categories [9]. An attractive byproduct of this progress has been in the realization that the visual features learnt from such large-scale dataset [10] have strong representational power and are useful as generic image features for various of visual understanding tasks.

In parallel, machines have learned to translate from French to English and identify the sentiment of a sentence by using recurrent neural networks (RNN) [11]. More recently, language model pre-training on large-scale text corpus (*e.g.* Wikipedia [12]) such as ELMO [13], GPT [14] and BERT [15] has been shown to be effective for improving many natural language processing tasks. For example, BERT based system [16] beats humans on the general language understanding evaluation benchmark (GLUE) [17].

In both domains, the learnt visual and linguistic representations can provide useful information for target tasks, like dog breed sensitive image features or a well-calibrated semantic distance between words. While visual and linguistic understanding is of course essential to vision-and-language tasks, equally important is how they related to one another – *i.e.* how to induce appropriate *grounding* given the heterogeneity of the data. For example, a perfect visual representation of dog breeds is of little use if a downstream vision-and-language model fails to associate it with appropriate phrases like "beagle" or "shepherd".

In this thesis we study how inducing *appropriate grounding* improves multi-modal AI capabilities in the context of 'vision-and-language'. We first walk through different approaches by different vision and language tasks – starting from the task of answering visual question about an image. Next we address image captioning, where the goal is to generate image description. Third we address visual dialog, which requires an AI agent to hold a meaningful dialog with humans in natural, conversational language about visual content. At last, we study how to learn task-agnostic visiolinguistic representations for different vision and language tasks.

Definition of *'Appropriate Grounding'*: 'grounding' refers to the connection between symbols and any intended referents (meaning) [12]. In this thesis, 'appropriate grounding' refers to slightly different abilities for different tasks. We define those below.

1. For visual question answering, *'appropriate grounding'* refers to jointly modeling visual attention and question attention to answer the questions.

2. For image captioning, *‘appropriate grounding’* refers to utilizing the outputs of other visual models (e.g., object detectors) to generate descriptions of an image.

3. For the answering agent in visual dialog, *‘appropriate grounding’* refers to the ability of a generative visual dialog model to be consistent with the discriminate visual dialog model.

4. For the question agent in visual dialog, *‘appropriate grounding’* refers to the human interpretability of the generated questions.

5. For vision and language pretraining, *‘appropriate grounding’* refers to refers to learn the connection between words and visual patches from large-scale external data that is not specific to a task at hand.

1.1 Visual Question Answering

Given an image and a natural language question about the image, the task of visual question answering (VQA) requires the model to provide an accurate natural language answer. Mirroring real-world scenarios, such as helping the visually impaired, both the questions and answers are open-ended. Visual questions selectively target different areas of an image, including background details and underlying context. As a result, a system that succeeds at VQA typically needs a more detailed understanding of the image and more complex reasoning than a system producing generic image captions. We provide a dataset containing $\sim$0.25M images, $\sim$0.76M questions, and $\sim$10M answers. To explore the difficulty of the dataset, we first implement various baselines and a novel neural approach which uses a hadamard product to fuse visual and linguistic representations. We further propose a novel co-attention mechanism that jointly reasons about image attention and question attention to correctly answer the question. Below I discuss my work on tackling each of these problems in visual question answering.

1.1.1 Dataset and Baselines for VQA

We first introduce the task of *free-form* and *open-ended* Visual Question Answering (VQA). A VQA system takes as input an image and a free-form, open-ended, natural language question about the image and produces a natural language answer as the output. As shown in Fig. 1.1, open-ended questions require a potentially vast set of AI capabilities – fine-grained recognition (*e.g.*, “What color are her eyes?”), object detection (*e.g.*, ”What is the mustache made of?”) activity recognition (*e.g.*, “Is this man crying?”), knowledge based reasoning (*e.g.*, “Is this a vegetarian pizza?”), and commonsense reasoning (*e.g.*, “Does this person have 20/20 vision?”). We present a large dataset containing ∼0.25M images, ∼0.76M questions, and ∼10M answers. As part of the VQA initiative, we offer several approaches that use a combination of both text and state-of-the-art visual features. Thus, in Section 4.1 I first implement various baselines to explore the difficulty of the VQA dataset.

Figure 1.1: Examples of free-form, open-ended questions collected for images via Amazon Mechanical Turk. Note that commonsense knowledge is needed along with a visual understanding of the scene to answer many questions.

1.1.2 Hierarchical Co-Attention for Image Question Grounding

While it is desirable to model "where to look," using visual attention to answer visual questions, we argue that the problem of identifying "which words to listen to," or *question attention*, is equally important. For instance, given the questions "how many horses are in this image?" and "how many horses can you see in this image?". They have the same meaning, essentially captured by the first three words. A machine that attends to the first three words would arguably be more robust to irrelevant linguistic variations. Motivated by this observation, in Section 4.2 I study different mechanisms that jointly reason about visual attention and question attention, which refers to *co-attention*. The core contribution of this work is in developing a novel co-attention mechanism to learn grounded features for visual question answering. Our results suggest the approach offers consistent improvements over baselines that only perform image attention.

1.2 Neural Image captioning

Image captioning is also a challenging problem that lies at the intersection of computer vision and natural language processing. It involves generating a natural language sentence that accurately summarizes the contents of an image. A word in an image caption can be either visual or non-visual based on whether it has visual meaning or not. For example, as shown in Fig. 1.2, given an image and the corresponding caption "A puppy with a tie is sitting at table with a cake," the non-visual words "a" and "with" do not have corresponding visual signals. On the other hand, the visual words "puppy" and "tie" do have an explicit visual correspondence in the image (*i.e.*, to know which words have visual meaning.) An important prerequisite to appropriate grounding is the ability to knowing when to look at the image. Of course, when looking at the image, the model also needs to decide which image region it should attend to. Thus, I describe below my line of work on appropriately grounding (or not) the visual words and non-visual words. For the visual words, instead of relying on the language model, I further propose a novel approach that grounds on concepts exist in object detections.

Figure 1.2: Examples of image captions. Note that there are visual words (in blue) and non-visual words (in black) based on whether the words have an explicit visual corresponding in the image. We only consider Noun and Adjective words in the examples.

1.2.1 Knowing When to Look for Image Caption Generation

Visual attention-based neural encoder-decoder models [18, 19] learns to "attend" to selective regions while generating a description. Similar to human vision, which fixates when you perceive the visual world, the attention mechanism typically produces a spatial map highlighting image regions relevant to each generated word. Most attention models for image captioning and visual question answering attend to the image at every time step, irrespective of which word is going to be emitted next [18, 19]. However, not all words in the caption have corresponding visual signals. Consider the example (right image) in Fig. 1.2 that shows an image and its generated caption "A white bird perched on top of a red stop sign". The words "a" and "of" do not have corresponding canonical visual signals. Moreover, language correlations make the visual signal unnecessary when generating words like "on" and "top" following "perched", and "sign" following "a red stop". Motivated by this observation, I next focus on the problem of selectively grounding visual signals for caption generation. We introduce a new Long Short-Term Memory (LSTM) [20] extension, which produces an additional "visual sentinel" vector instead of a single hidden state. The "visual sentinel", an additional latent representation of the decoder's memory, provides a fallback option to the decoder. We further design a new sentinel gate, which decides how much new information the decoder wants to get from the image as opposed to relying on the visual sentinel when generating the next word.

1.2.2 Neural Baby Talk: Explicitly Grounding on Object Detection

While there are many recent extensions of this basic idea to include attention [18, 21, 22], it is well-understood that models still lack visual grounding (*i.e.*, do not associate named concepts to pixels in the image). They often tend to 'look' at different regions than humans would and tend to copy captions from training data [23]. For instance, in Fig. 1.3 a neural image captioning approach [24] describes the image as "A dog is sitting on a couch with a toy." This is not quite accurate. But if one were to *really* squint at the image, it (arguably) does perhaps look like a scene where a dog *could* be sitting on a couch with a toy. It certainly is common to find dogs sitting on couches with toys. A-priori, the description is reasonable. Existing neural captioning models tend to produce generic *plausible* captions based on the language model[1] that match a first-glance gist of the scene.

If we take a step back – do we really need the language model to do the heavy lifting in image captioning? Given the unprecedented progress we are seeing in object recognition[2] (e.g., object detection, semantic segmentation, instance segmentation, pose estimation), it seems like the vision pipeline can certainly do better than relying on just a first-glance gist of the scene. In fact, today's state-of-the-art object detectors

[1]frequently, directly reproduced from a caption in the training data.

[2]e.g., 11% absolute increase in average precision in object detection in the COCO challenge in 2017.

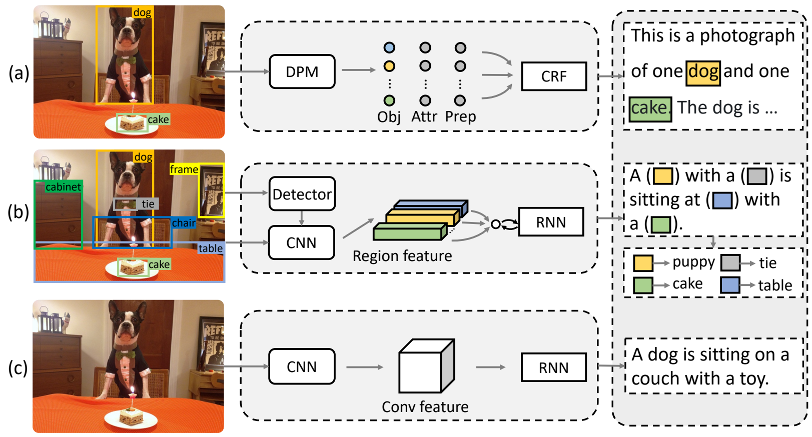


Figure 1.3: From left to right is the generated caption using the same captioning model but with different detectors: 1) No detector; 2) A weak detector that only detects "person" and "sandwich"; 3) A detector trained on COCO [5] categories (including "teddy bear"). 4) A detector that can detect novel concepts (e.g. "Mr. Ted" and "pie" that never occurred in the captioning training data). Different colors show a correspondence between the visual word and grounding regions.

can successfully detect the table and cake in the image in Fig. 1.3(c)! The caption ought to be able to talk about the table and cake *actually detected* as opposed to letting the language model hallucinate a couch and a toy simply because that sounds plausible. Interestingly, some of the first attempts at image captioning [25, 24] – before the deep learning "revolution" – relied heavily on outputs of object detectors and attribute classifiers to describe images. Inspired by this observation, I introduce a novel framework that reconciles these methodologies. It produces natural language *explicitly* grounded in entities found by object detectors. It is a neural approach that generates a sentence "template" with slot locations explicitly tied to image regions. These slots are then filled by object recognizers with concepts found in the regions. The entire approach is trained end-to-end. This results in natural sounding and grounded captions.

1.3 Visual Dialog

Despite rapid progress at the intersection of vision and language – in particular, in image captioning and visual question answering – it is clear that we are far from this grand goal of an AI agent that can 'see' and 'communicate' [4]. In captioning, the machine simply talks at the human with no dialog or input from the human, while VQA still represents only a single round of dialog. As a step towards conversational visual AI, the task of visual dialog requires computers to communicate naturally with human in grounded language to achieve a collaborative objective. For visual dialog, it usually contains two sides – questioner and answerer. The canonical visual dialog task [4] lies in modeling answer side, where an agent answers a sequence of questions grounded in an image, and need to reason about both visual content and dialog history. Modeling the questioner side is actually more challenging – the agent needs to learn how to ask meaningful and visually grounded questions to achieve a goal [26, 27]. A popular approach to these tasks has been to observe humans engaging in dialogs like the ones we would like to automate and then train agents to mimic these human dialogs by minimizing the cross-entropy of the human questions or responses. However, a recurring problem with maximum likelihood estimation (MLE) trained generative neural dialog models is that they tend to produce 'safe' and generic responses (*'I don't know'*, *'I can't tell'*) and these models are typically fragile and generalize poorly to new tasks. Thus, I describe below my line of work on introducing novel training paradigms for generating grounded questions and responses in the context of visual dialog.

1.3.1 Discriminant Perceptual Loss for Visual Dialog

The standard training paradigm for neural dialog models is maximum likelihood estimation (MLE) or equivalently, minimizing the cross-entropy (under the model) of a 'ground-truth' human response. Across a variety of domains, a recurring problem with MLE trained neural dialog models is that they tend to produce 'safe', generic responses, such as *'Not sure'* or *'I don't know'* in text-only dialog [28], and *'I can't see'* or *'I can't tell'* in visual dialog [4, 26]. One reason for this emergent behavior is that the space of possible next utterances in a dialog is *highly* multi-modal (there are many possible paths a dialog may take in the future). In the face of such highly multi-modal output distributions, models 'game' MLE by latching on to the head of the distribution or mimicing most frequent responses, which by nature tend to be generic and widely applicable.

One promising alternative to MLE training proposed by recent work [29, 30] is *sequence-level training* of neural sequence models. Specifically, using reinforcement learning to optimize task-specific sequence metrics such as BLEU [31], ROUGE [32], CIDEr [33]. Unfortunately, in the case of dialog, *all existing* automatic metrics correlate poorly with human judgment [34], which renders this alternative infeasible for dialog models. In Section 6.1, Inspired by the success of adversarial training [35], we propose to train a *generative* visual dialog model (G) to produce sequences that score highly under a *discriminative* visual dialog model (D). The discriminative dialog model receives as input a candidate list of possible responses and learns to sort this list from the training dataset. The generative dialog model (G) aims to produce a sequence that D will rank the highest in the list.

Note that while our proposed approach is inspired by adversarial training, there are a number of subtle but crucial differences over generative adversarial networks (GANs). Unlike traditional GANs, one novelty in our setup is that our discriminator has *access to more information* than G – specifically, D receives a list of candidate responses and explicitly learns to reason about similarities and differences across candidates.

1.3.2 Learning Image-Discriminative Dialog Policies from VQA

While training dialog agents with a discriminant perceptual loss indeed increases task performance, language quality suffers even for similar tasks. It tends to drifts from human language, becoming ungrammatical and loosing human interpretable semantics – sometimes even turning into unintelligible code. Though bots might understand it, humans cannot, so humans will not be able to use it either. Both effects have been observed in earlier work [26, 36].

In Section 6.2, we consider an image guessing game as a test-bed for balancing task performance and language drift. Our Dialog without Dialog (DwD) task requires agents to generalize from single round visual question generation with full supervision to a multi-round dialog based image guessing game without direct language supervision. Specifically, as illustrated in Fig. 1.4 (top), agents are trained to mimic human-generated, visually-grounded questions that when answered can discern which of two images is secretly indicated to the answerer. We then develop techniques to transfer these agents to a multi-round, QA-based image guessing game over pools of various sizes, difficulties, and even image domains.

To solve this task we propose a an architecture for the questioner agent, Q-bot, that decomposes generating question intent from the words used to express that intent. It does this by introducing a discrete latent representation that is the only input to the language decoder. We pair this with an incremental learning curriculum that adapts the single round Q-bot to dialog in stages – first learning simply to follow the dialog and then to influence question intention. We show that our model can be fine-tuned to increase task performance while maintaining human interpretable language. To measure interpretability we take a two pronged approach, getting humans to evaluate our questions on one hand, and using automatic metrics on the other. Humans evaluate question fluency and relevance while our automatic metrics evaluate fluency, relevance, and diversity to help scale our analysis.

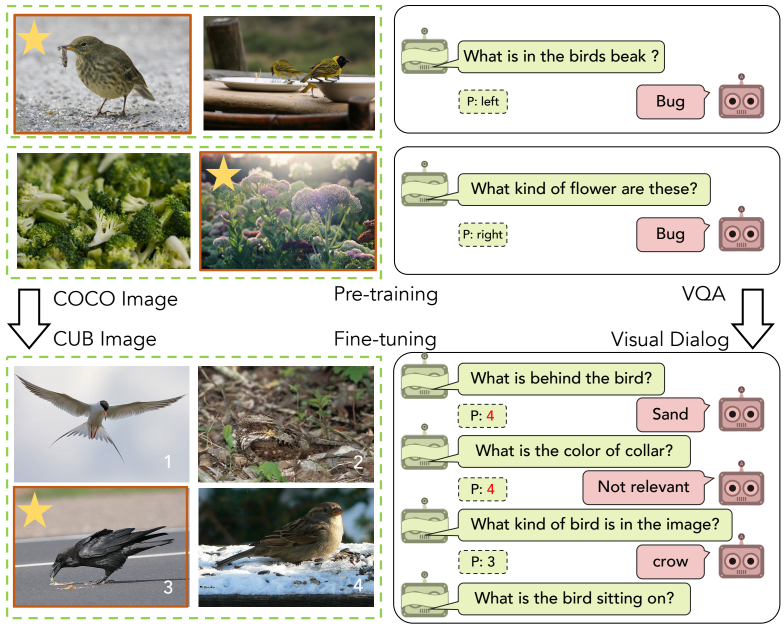


Figure 1.4: (Top - 2 pools) We train our questioner to ask questions that can discriminate between pairs of images by mimicing questions from the VQAv2 dataset. (Bottom - 1 pool) Our proposed model generalizes to new settings in a way that humans can understand without additional language supervision (*i.e.*, without dialog).

1.4 Vision and Language Pretraining

A compelling reason to study language and vision jointly is the promise of language as a universal and natural interface for visual reasoning problems – useful both in specifying a wide range of problems and in communicating AI responses. However, the current research landscape for visually-grounded language understanding is a patchwork of many specialized tasks like question answering or caption generation, each supported by a handful of datasets. As such, progress in this field has been measured by the independent improvement of bespoke models designed and trained for each of these specific tasks and datasets.

A general multi-modal AI model cannot emerge within a paradigm that focuses on the particularities of a single dataset, metric, and tasks. In the vision and language pretraining chapter, I aim to build a general multi-modal AI model and training paradigm that has following properties: 1: it can utilize large webly supervised dataset to effectively learn the visiolinguistic representations; 2: it has a unified structure and interface which can be shared across different vision and language tasks; 3: it is trained with an effective multi-task training paradigm which can handle datasets that vary greatly in size and difficulty. Thus, I describe below my line of work on introducing pretrain-then-transfer learning approaches to learn vision and language representations. I also introduce a novel multi-task training paradigm I use to jointly train 12 tasks simultaneously and achieve state of the art performance on 7 out of 12 vision and language tasks.

1.4.1 ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations

To learn these joint visual-linguistic representations, we look to recent successes in self-supervised learning which have captured rich semantic and structural information from large, unlabelled data sources by training models to perform so-called 'proxy' tasks. These proxy tasks leverage structure within the data to generate supervised tasks automatically (*e.g.* colorizing images [37] or reconstructing masked words in text [15]). In Section 7.1, we present a joint model for learning task-agnostic visual grounding from paired visiolinguistic data which we call Vision & Language BERT (ViLBERT for short). Our approach extends the recently developed BERT [15] language model to jointly reason about text and images. Our key technical innovation is introducing separate streams for vision and language processing that communicate through co-attentional transformer layers. This structure can accommodate the differing processing needs of each modality and provides interaction between modalities at varying representation depths. We demonstrate that this structure outperforms a single-stream unified model in our experiments.

In analogy to the training tasks in [15], we train our model using the Conceptual Captions dataset [38] and two proxy tasks: predicting the semantics of masked words and image regions given the unmasked inputs, and predicting whether an image and text segment correspond. We apply our pretrained model as a base for four established vision-and-language tasks – visual question answering [39], visual commonsense reasoning [40], referring expressions [2], and caption-based image retrieval [41] – setting state-of-the-art on all four tasks.

1.4.2 12-in-1: Multi-Task Vision and Language Representation Learning

In the previous section I introduced a general architectures for vision-and-language which reduces architectural differences across tasks [42, 43, 44, 45, 46, 47, 48]. The model pretrains common architectures on self-supervised tasks to learn general visio-linguistic representations then finetunes for specific datasets; however, the result is still a menagerie of independent task-specific models rather than a single unified model. This is dissatisfying in practice – the model that understands questions cannot ground noun phrases, the grounding model cannot retrieve images based on a description, and so forth. Further, this approach does not scale well as each new task requires storing a new model.

Beyond being intellectually dissatisfying, this task-based fracturing leaves quite a lot on the table. While individual tasks present different challenges and diverse interfaces, the underlying associations between language and visual concepts are often common across tasks. For example, learning to ground the referring expression "small red vase" requires understanding the same concepts as answering the question "What color is the small vase?". Training multiple tasks jointly can potentially pool these different sources of grounding supervision. Further, developing models that can perform well on a wide range of tasks simultaneously can help guard against the research community overfitting to specific datasets and metrics.

In section Section 7.2, I introduce a multi-task model for discriminative vision-and-language tasks based on the recently proposed ViLBERT[42] model. We consider four categories of tasks – training jointly on a total of 12 different datasets. Our results not only show that a single model can perform all these tasks, but also that joint training can lead to improvements on task metrics compared to single-task training with the same architecture. Our model attains improvements of 0.25 to 4.19 absolute points from multi-task training – improving over corresponding single-task models for 11 out of 12 tasks.

CHAPTER 2
BACKGROUND

We will first introduce some necessary background material which will be useful to understand the proposed models. Specifically, I will first introduce tasks and baseline models for visual question answering [3], image captioning [49] and visual dialog [4]. Then, I will brief describe other related vision and language tasks explored in the thesis. Finally, I will talk about a self supervised learning models called bidirectional encoder representations from transformers (BERT). You can skip this chapter if you are familiar with these topics.

2.1 Background: Visual Question Answering

We collected one of the most widely used dataset for visual question answering, commonly referred to simply as VQA. It comprises two parts, one using natural images named VQA-real, and a second one with cartoon images named VQA-abstract. The real part comprises 123,287 training and 81,434 test images, respectively, sourced from COCO [5]. We tested and evaluated a number of user interfaces for collecting such "interesting" questions. To bias against generic image-independent questions, subjects were instructed to ask questions that require the image to answer. Overall, it contains 614,163 questions, each having 10 answers from 10 different annotators.

For testing, we offer two modalities for answering the questions: (i) **open-ended** and (ii) **multiple-choice**. For the open-ended task, the generated answers are evaluated using the following accuracy metric:

$$\text{accuracy} = \min\left(\frac{\#\text{ humans that provided that answer}}{3}, 1\right)$$

i.e., an answer is deemed 100% accurate if at least 3 workers provided that exact answer.[1] Before comparison, all responses are made lowercase, numbers converted to digits, and punctuation & articles removed. We avoid using soft metrics such as Word2Vec [50], since they often group together words that we wish to distinguish, such as "left" and "right". For multiple-choice task, 18 candidate answers are created for each question. As with the open-ended task, the accuracy of a chosen option is computed based on the number of human subjects who provided that answer (divided by 3 and clipped at 1).

2.2 Background: Neural Image Captioning

For image captioning, we start by briefly describing the encoder-decoder image captioning framework [51, 18]. Given an image and the corresponding caption, the encoder-decoder model directly maximizes the following objective:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \sum_{(\boldsymbol{I},\boldsymbol{y})} \log p(\boldsymbol{y}|\boldsymbol{I};\boldsymbol{\theta}) \tag{2.1}$$

where $\boldsymbol{\theta}$ are the parameters of the model, $\boldsymbol{I}$ is the image, and $\boldsymbol{y} = \{y_1, \ldots, y_t\}$ is the corresponding caption. Using the chain rule, the log likelihood of the joint probability distribution can be decomposed into ordered conditionals:

$$\log p(\boldsymbol{y}) = \sum_{t=1}^{T} \log p(y_t|y_1, \ldots, y_{t-1}, \boldsymbol{I}) \tag{2.2}$$

where we drop the dependency on model parameters for convenience.

In the encoder-decoder framework, with recurrent neural network (RNN), each conditional probability is modeled as:

$$\log p(y_t|y_1, \ldots, y_{t-1}, \boldsymbol{I}) = f(\boldsymbol{h}_t, \boldsymbol{c}_t) \tag{2.3}$$

[1] In order to be consistent with 'human accuracies', machine accuracies are averaged over all $\frac{9}{10}$ sets of human annotators

where f is a nonlinear function that outputs the probability of y_t. $\boldsymbol{c}_t$ is the visual context vector at time t extracted from image $\boldsymbol{I}$. $\boldsymbol{h}_t$ is the hidden state of the RNN at time t. In this paper, we adopt Long-Short Term Memory (LSTM) instead of a vanilla RNN. The former have demonstrated state-of-the-art performance on a variety of sequence modeling tasks. $\boldsymbol{h}_t$ is modeled as:

$$\boldsymbol{h}_t = \mathrm{LSTM}(\boldsymbol{x}_t, \boldsymbol{h}_{t-1}, \boldsymbol{m}_{t-1}) \tag{2.4}$$

where $\boldsymbol{x}_t$ is the input vector. $\boldsymbol{m}_{t-1}$ is the memory cell vector at time $t-1$. we can describe the LSTM with the following equations:

$$\begin{aligned}
\boldsymbol{i}_t &= \sigma(W_i \cdot [\boldsymbol{x}_{t-1}, \boldsymbol{h}_{t-1}]) \\
\boldsymbol{f}_t &= \sigma(W_f \cdot [\boldsymbol{x}_{t-1}, \boldsymbol{h}_{t-1}]) \\
\boldsymbol{o}_t &= \sigma(W_o \cdot [\boldsymbol{x}_{t-1}, \boldsymbol{h}_{t-1}]) \\
\boldsymbol{m}_t &= \boldsymbol{f}_t \odot \boldsymbol{m}_{t-1} + \boldsymbol{i}_t \odot tanh(W_c \cdot [\boldsymbol{x}_{t-1}, \boldsymbol{h}_{t-1}]) \\
\boldsymbol{h}_t &= \boldsymbol{o}_t \cdot \boldsymbol{m}_t
\end{aligned} \tag{2.5}$$

Commonly, context vector, $\boldsymbol{c}_t$ is an important factor in the neural encoder-decoder framework, which provides visual evidence for caption generation [52, 51, 18]. These different ways of modeling the context vector fall into two categories: vanilla encoder-decoder and attention-based encoder-decoder frameworks:

- First, in the vanilla framework, $\boldsymbol{c}_t$ is only dependent on the encoder, a Convolutional Neural Network (CNN). The input image $\boldsymbol{I}$ is fed into the CNN, which extracts the last fully connected layer as a global image feature [52, 51]. Across generated words, the context vector $\boldsymbol{c}_t$ keeps constant, and does not depend on the hidden state of the decoder.

- Second, in the attention-based framework, $\boldsymbol{c}_t$ is dependent on both encoder and decoder. At time t, based on the hidden state, the decoder would attend to the specific regions of the image and compute $\boldsymbol{c}_t$ using the spatial image features from a convolution layer of a CNN. In [18, 21], they show that attention models can significantly improve the performance of image captioning.

2.3 Background: Visual Dialog

Visual Dialog: A visual dialog model is given as input an image $\boldsymbol{I}$, caption $\boldsymbol{c}$ describing the image, a dialog history till round $t-1$, $\boldsymbol{H} = (\underbrace{\boldsymbol{c}}_{H_0}, \underbrace{(\boldsymbol{q}_1, \boldsymbol{a}_1)}_{H_1}, \ldots, \underbrace{(\boldsymbol{q}_{t-1}, \boldsymbol{a}_{t-1})}_{H_{t-1}})$, and the followup question $\boldsymbol{q}_t$ at round t. The visual dialog agent needs to return a valid response to the question.

Given the problem setup, there are two broad classes of methods – generative and discriminative models. Generative models for visual dialog are trained by maximizing the log-likelihood of the ground truth answer sequence $\boldsymbol{a}_t^{gt} \in \mathcal{A}_t$ given the encoded representation of the input $(\boldsymbol{I}, \boldsymbol{H}, \boldsymbol{q}_t)$. On the other hand, discriminative models receive both an encoding of the input $(\boldsymbol{I}, \boldsymbol{H}, \boldsymbol{q}_t)$ *and* as additional input a list of 100 candidate answers $\mathcal{A}_t = \{\boldsymbol{a}_t^{(1)}, \ldots, \boldsymbol{a}_t^{(100)}\}$. These models effectively learn to sort the list. Thus, by design, they cannot be used at test time without a list of candidates available.

Image Guessing Game: Visual conversational agents are AI agents trained to understand and communicate about the contents of a scene via a natural language dialog. Chattopadhyay et.al. [53] propose to evaluate visual conversational agents by a human-AI game called GuessWhich. At the start of the game, the visual conversational agent (Alice) is provided an image which is unknown to a human. The human then identifies the secret image from a pool of images by asking Alice a sequence of questions that Alice answers. Machine-machine versions of this game have also been studied [26] where both the questioning and answering agents are bots. The idea was that via self-play, these bots could become better conversational agents.

Concretely, we formulate a game between a questioner bot (Q-BOT) and an answerer bot (A-BOT). The A-BOT is assigned a secret image from a pool of images taken from the COCO dataset [5] unknown to the Q-BOT. The Q-BOT is provided a caption of a target image and is allowed to communicate in natural language with the A-BOT. The objective of this cooperative game is for Q-BOT to ask an intelligent question to guess the secret image. Our setting is very similar to [53], which evaluates conversational agents. The difference is instead of recruiting human players, we develop Q-BOT to mimic the human behavior.

2.4 Background: Bidirectional Encoder Representations from Transformers

The BERT model introduced by [15] is an attention-based bidirectional language model. When pretrained on a large language corpus, BERT has proven to be very effective for transfer learning to multiple natural language processing tasks.

The BERT model operates on sequences of word tokens $w_0, \ldots, w_T$. These tokens are mapped to learned encodings and passed through L "encoder-style" transformer blocks [54] to produce final representations $h_0, \ldots, h_T$. Let $H^{(l)}$ be a matrix with rows $h_0^{(l)}, \ldots, h_T^{(l)}$ corresponding to the intermediate representations after the l-th layer. Abstracting some internal details found in [54], we depict the computation of a single encoder-style transformer block in Fig. 2.1 consisting of a multi-headed attention block followed by a small fully-connected network, both wrapped in residual adds. Note that the intermediate representation $H^{(l)}$ is used to compute three matrices – Q, K, and V – corresponding to queries, keys, and values that drive the multi-headed attention block. Specifically, the dot-product similarity between queries and keys determines attentional distributions over value vectors. The resulting weight-averaged value vector forms the output of the attention block.

Text Representation. BERT operates over sequences of discrete tokens comprised

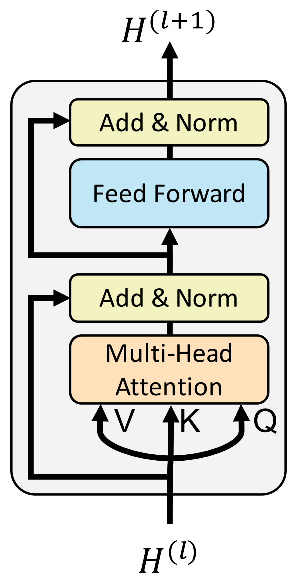


Figure 2.1: Standard encoder transformer block.

of vocabulary words and a small set of special tokens: `SEP`, `CLS`, and `MASK`. For a given token, the input representation is a sum of a token-specific learned embedding [55] and encodings for position (*i.e.* token's index in the sequence) and segment (*i.e.* index of the token's sentence if multiple exist).

Training Tasks and Objectives. The BERT model is trained end-to-end on a large language-corpus under two tasks: *masked language modelling* and *next sentence prediction*.

The masked language modelling task randomly divides input tokens into disjoint sets corresponding to masked X_M and observed X_O tokens (approximately 15% of tokens being masked). Masked tokens are replaced with a special `MASK` token 80% of the time, a random word 10%, and unaltered 10%. The BERT model is then trained to reconstruct these masked tokens given the observed set. Specifically, a linear layer is learned to map the final representations at each index (*e.g.* h_i) to a distribution over the vocabulary and the model is trained under a cross-entropy loss.

In next sentence prediction, the BERT model is passed two text segments A and B following the format $\{\texttt{CLS}, w_{A1}, \ldots, w_{AT}, \texttt{SEP}, w_{B1}, \ldots, w_{B\mathcal{T}}, \texttt{SEP}\}$ and is trained to predict whether or not B follows A in the source text. Specifically, a linear layer operating on the final representation for the `CLS` token (*i.e.* $h_{\texttt{CLS}}$) is trained to minimize a binary cross-entropy loss on this label.

CHAPTER 3
SITUATING THE WORK

I first discuss related work on visual question answering, then image captioning followed by visual dialog. Finally, I will discuss related work on self supervised learning and multi-task learning for vision and language pretraining. In VQA, I will cover prior dataset on VQA, and image attention and language attention models. In image captioning, I will cover related work using template-based approaches, and neural-based approaches. For visual dialog tasks, I will cover prior work on using generative adversarial networks for sequence generation and related work on attention models. I will also cover prior works related to visual question generation, dialog generation using latent action space and reference game (image guessing game). In vision and language pretraining, I will discuss related work on self supervised learning and multi-task learning.

3.1 Visual Question Answering

3.1.1 VQA datasets

Several recent papers have begun to study visual question answering [56, 57, 58, 59]. However, unlike our work, these are fairly restricted (sometimes synthetic) settings with small datasets. For instance, [57] only considers questions whose answers come from a predefined closed world of 16 basic colors or 894 object categories. [56] also considers questions generated from templates from a fixed vocabulary of objects, attributes, relationships between objects, *etc.* In contrast, our proposed task involves *open-ended*, *free-form* questions and answers provided by humans. Our goal is to increase the diversity of knowledge and kinds of reasoning needed to provide correct answers. Critical to achieving success on this more difficult and unconstrained task, our VQA dataset is *two orders of magnitude* larger than [56, 57] (>250,000 *vs.* 2,591 and 1,449 images respectively). The proposed VQA task has connections to other related work: [58] has studied joint parsing of videos and corresponding text to answer queries on two datasets containing 15 video clips each. [59] uses crowdsourced workers to answer questions about visual content asked by visually-impaired users. In concurrent work, [60] proposed combining an LSTM for the question with a CNN for the image to generate an answer. In their model, the LSTM question representation is conditioned on the CNN image features at each time step, and the final LSTM hidden state is used to sequentially decode the answer phrase. In contrast, the model developed in this paper explores "late fusion" – *i.e.*, the LSTM question representation and the CNN image features are computed independently, *fused* via an element-wise multiplication, and then passed through fully-connected layers to generate a softmax distribution over output answer classes. [61] generates abstract scenes to capture visual common sense relevant to answering (purely textual) fill-in-the-blank and visual paraphrasing questions. [62] and [63] use visual information to assess the plausibility of common sense assertions. [64] introduced a dataset of 10k images and prompted captions that describe specific aspects of a scene (*e.g.*, individual objects, what will happen next). Concurrent with our work, [65] collected questions & answers in Chinese (later translated to English by humans) for COCO images. [66] automatically generated four types of questions (object, count, color, location) using COCO captions.

3.1.2 Image Attention

Instead of directly using the holistic entire-image embedding from the fully connected layer of a deep CNN (as in [39, 67, 60, 68]), a number of recent works have explored image attention models for VQA. Zhu *et al.* [69] add spatial attention to the standard

LSTM model for pointing and grounded QA. Andreas *et al.* [70] propose a compositional scheme that consists of a language parser and a number of neural modules networks. The language parser predicts which neural module network should be instantiated to answer the question. Some other works perform image attention multiple times in a stacked manner. In [71], the authors propose a stacked attention network, which runs multiple hops to infer the answer progressively. To capture fine-grained information from the question, Xu *et al.* [72] propose a multi-hop image attention scheme. It aligns words to image patches in the first hop, and then refers to the entire question for obtaining image attention maps in the second hop. In [73], the authors generate image regions with object proposals and then select the regions relevant to the question and answer choice. Xiong *et al.* [74] augments dynamic memory network with a new input fusion module and retrieves an answer from an attention based GRU. In concurrent work, [23] collected 'human attention maps' that are used to evaluate the attention maps generated by attention models for VQA. Note that all of these approaches model visual attention alone, and do not model question attention. Moreover, [72, 71] model attention sequentially, i.e., later attention is based on earlier attention, which is prone to error propagation. In contrast, we conduct co-attention at three levels independently.

3.1.3 Language Attention

Though no prior work has explored question attention in VQA, there are some related works in natural language processing (NLP) in general that have modeled language attention. In order to overcome difficulty in translation of long sentences, Bahdanau *et al.* [75] propose RNNSearch to learn an alignment over the input sentences. In [76], the authors propose an attention model to circumvent the bottleneck caused by fixed width hidden vector in text reading and comprehension. A more fine-grained attention mechanism is proposed in [77]. The authors employ a word-by-word neural attention mechanism to reason about the entailment in two sentences. Also focused on modeling sentence pairs, the authors in [78] propose an attention-based bigram CNN for jointly performing attention between two CNN hierarchies. In their work, three attention schemes are proposed and evaluated. In [79], the authors propose a two-way attention mechanism to project the paired inputs into a common representation space.

3.2 Neural Image Captioning

Image captioning has many important applications ranging from helping visually impaired users to human-robot interaction. As a result, many different models have been developed for image captioning. In general, those methods can be divided into two categories: template-based [25, 24, 80] and neural-based [81, 52, 82, 51, 19, 83, 22, 21].

3.2.1 Template-based approaches

Template-based approaches generate caption templates whose slots are filled in based on outputs of object detection, attribute classification, and scene recognition. Farhadi *et al.* [25] infer a triplet of scene elements which is converted to text using templates. Kulkarni *et al.* [24] adopt a Conditional Random Field (CRF) to jointly reason across objects, attributes, and prepositions before filling the slots. [80] uses more powerful language templates such as a syntactically well-formed tree, and add descriptive information from the output of attribute detection.

3.2.2 Neural-based approaches

Neural-based approaches are inspired by the success of sequence-to-sequence encoder-decoder frameworks in machine translation [84, 85, 75] with the view that image captioning is analogous to translating images to text. Kiros *et al.* [81] proposed a feed forward neural network with a multimodal log-bilinear model to predict the next word given the image and previous word. Other methods then replaced the feed forward neural network with a recurrent neural network [52, 86]. Vinyals *et al.* [51] use an LSTM instead of a vanilla RNN as the decoder. However, all these approaches represent the image with the last fully connected layer of a CNN. Karpathy *et al.* [87] adopt the result of object detection from R-CNN and output of a bidirectional RNN to learn a joint embedding space for caption ranking and generation.

Recently, attention mechanisms have been introduced to encoder-decoder neural frameworks in image captioning. Xu *et al.* [18] incorporate an attention mechanism to learn a latent alignment from scratch when generating corresponding words. [88, 21] utilize high-level concepts or attributes and inject them into a neural-based approach as semantic attention to enhance image captioning. Yang *et al.* [89] extend current attention encoder-decoder frameworks using a review network, which captures the global properties in a compact vector representation and are usable by the attention mechanism in the decoder.

3.3 Visual Dialog

3.3.1 GANs for sequence generation

Generative Adversarial Networks (GANs) [35] have shown to be effective models for a wide range of applications involving continuous variables (*e.g.* images) *c.f* [90, 91, 92, 93]. More recently, they have also been used for discrete output spaces such as language generation – *e.g.* image captioning [94, 95], dialog generation [28], or text generation [96] – by either viewing the generative model as a stochastic parametrized policy that is updated using REINFORCE with the discriminator providing the reward [96, 94, 95, 28], or (closer to our approach) through continuous relaxation of discrete variables through Gumbel-Softmax to enable backpropagating the response from the discriminator [97, 95].

There are a few subtle but significant differences w.r.t. to our application, motivation, and approach. In these prior works, both the discriminator and the generator are trained in tandem, and from scratch. The goal of the discriminator in those settings has primarily been to discriminate 'fake' samples (*i.e.* generator's outputs) from 'real' samples (*i.e.* from training data). In contrast, we would like to transfer knowledge from the discriminator to the generator. We start with pre-trained D and G models suited for the task, and then transfer knowledge from D to G to further improve G, while keeping D fixed. As we show in our experiments, this procedure results in G producing diverse samples that are close in the embedding space to the ground truth, due to perceptual similarity learned in D. One can also draw connections between our work and Energy Based GAN (EBGAN) [98] – without the adversarial training aspect. The "energy" in our case is a deep metric-learning based scoring mechanism, instantiated in the visual dialog application.

3.3.2 Modeling image and text attention in visual dialog

In the context of visual dialog, [4] uses attention to identify utterances in the dialog history that may be useful for answering the current question. However, when modeling the image, the entire image embedding is used to obtain the answer. In contrast, our proposed encoder HCIAE (Section 6.1.1) localizes the region in the image that can help reliably answer the question. In particular, in addition to the history and the question guiding the image attention, our visual dialog encoder also reasons about the history when identifying relevant regions of the image. This allows the model to implicitly resolve co-references in the text and ground them back in the image.

3.3.3 Visual Question Generation

Other approaches like [99] and [100] also aim to ask questions with limited question supervision. They give Q-bot access to an oracle to which it can ask any question and get a good answer back. This feedback allows these models to ask questions that are more useful for teaching A-bot [99] or generating scene graphs [100], but they require a domain specific oracle and do not take any measures to encourage interpretability. We are also interested in generalizing with limited supervision, using a standard VQAv2 [101] trained A-bot as a flawed oracle, but we focus on maintaining interpretability of generated questions and not just their usefulness.

3.3.4 Latent Action Spaces for dialog generation

Of particular interest to us a line of work that uses represents dialogs using latent action spaces [102, 103, 27, 104, 105, 27, 106]. Recent work use these representations have been used to discover interpretable language [102] and to perform zero-shot dialog generation [103], though neither works consider visually grounded language as in our approach. Most relevant is [107], which focuses on the difference between word level feedback and latent action level feedback. Like us, they use a variationally constrained latent action space (like our z) to generate dialogs and find that by providing feedback to the latent actions instead of the generated words (as opposed to the approaches in [26] and [36]) they achieve better dialog performance. Our variational prior is similar to the Full ELBO considered there In contrast to [107], we consider generalization from non-dialog data and generalization to new modalities.

3.3.5 Reference Games.

The task we use to study question generation follows a body of work that uses reference games to study language and its interaction with other modalities [108]. Our particular task is most similar to those in [109] and [110]. In particular, [109] collects a dataset for goal oriented visual dialog using a similar image reference game and [110] uses a similar guessing game we use to evaluate how well humans can interact with A-bot.

3.4 Vision and Language Pretraining

3.4.1 Self-Supervised Learning

There has been substantial recent interest in both vision [111, 112, 113, 114, 115, 116] and language around self-supervised representation learning. In this paradigm, deep models are trained for tasks where regularities in existing data can be turned into supervision automatically. While there has been progress on the vision side, self-supervised image representations still lag behind those from models trained under image classification tasks. Self-supervised language models on the other hand have resulted in significant improvements over prior work [15, 14, 117, 118]. In this work, we develop a model and proxy tasks for learning joint visual-linguistic representations – extending the popular BERT [15] model.

Most related to our approach is concurrent work on learning joint representations between video and language [119]. In this work, self-supervised tasks paralleling our own are derived from cooking videos paired with text-to-speech transcribed audio. They present a unified BERT architecture for both the visual and linguistic inputs similar to the Single-Stream baseline we consider here. They apply the learned model to two tasks on cooking videos: zero-shot activity recognition and blank-filling on audio transcripts. In contrast, we learn representations of images and descriptive text on a wide range of images from the web and focus extensively on transfer learning from this model for well-established vision-and-language tasks.

3.4.2 Recent Works on Vision-And-Language Pretraining

Since our paper released on arXiv, a few other useful preprints have recently been released on similar vision-and-language cross-modality pre-training directions. LXMERT [43] uses a more specific design for the cross-modality model. Instead of using webly supervised Conceptual Caption [38] dataset, LXMERT uses in-domain datasets (*i.e.*

COCO [49] and Visual Genome [120]) for pre-training. VisualBERT [44] directly extend BERT [15] for vision and language domain. VisualBERT uses both out-of-domain and in-domain dataset for pre-training and applies MLM object only on the language side. Unicoder [46] focuses exclusively on image caption retrieval tasks with online hardest negative mining. More recent preprints including VLBERT [47], Unified VLP [48] and UNITER [48] also show promising improvements in this research direction of joint visio-linguistic pretraining.

3.4.3 Multi-Task Learning.

There has been substantial interest in multi-task learning [121, 122], *i.e.* training a single model for multiple tasks at once. Advances in multi-task learning have been developed in the context of vision [123, 124, 125, 126], language [127, 128, 129, 16, 130], and robotics [131, 132, 133]. Among them, Standley *et al.* [134] studies how different vision tasks are related to each other. McCann *et al.* [129] pose ten natural language processing (NLP) tasks as question answering tasks. MT-DNN [16] combines multi-task learning with pretraining [15] to improve the learning of text representations. Despite this progress, it is still challenging to train a single model on many tasks that can outperform or even match their single-task counterparts. To enhance the training scheme, BAM [135] applies knowledge distillation where single-task models teach the multi-task model. Raffel *et al.* [130] explore different sampling strategies for NLP tasks. We focus on multi-task learning for V&L tasks.

3.4.4 Multi-Task V&L Learning.

Recent work [136, 137, 138] also explores multi-task learning in V&L. HDC [137] trains a multi-task network on multiple datasets and uses a hyper-parameter search method to determine which layer output should be taken for each task. Our method does not need any hyperparameter search to choose outputs for different tasks and outperforms both [136] and [137]. [138] is a concurrent work that does multi-task training on 12 dialogue datasets (only two with images). Our work differs in that we focus on a variety of vision and language tasks.

CHAPTER 4

VISUAL QUESTION ANSWERING

In this chapter, we will discuss my line of work in visual question answering. Since we first introduced the VQA dataset, to explore the difficulty of the dataset, we first implement various baselines and a novel neural approach that uses Hadamard product to fuse the visual and linguistic representations. Our baselines consist of random, prior, per Q-type prior, and nearest neighbor. We further develop a 2-channel vision + language model that culminates with a softmax over K possible outputs.

Next, we will motivate the co-attention framework for VQA that jointly reasons for image and question attention. In addition, our model reasons about the question (and consequently the image via the co-attention mechanism) in a hierarchical fashion via a novel 1-dimensional convolution neural networks (CNN). Our model improves the state-of-the-art on the VQA dataset from 60.3% to 60.5%, and from 61.6% to 63.3% on the COCO-QA dataset. By using ResNet, the performance is further improved to 62.1% for VQA and 65.4% for COCO-QA.

4.1 VQA Baselines and Methods

In this section, we explore the difficulty of the VQA dataset for the MS COCO images using several baselines and novel methods. We train on VQA train+val. Unless stated otherwise, all human accuracies are on test-standard, machine accuracies are on test-dev, and results involving human captions (in gray font) are trained on train and tested on val (because captions are not available for test).

4.1.1 Baselines

We implemented the following baselines:

1. **random:** We randomly choose an answer from the top 1K answers of the VQA train/val dataset.

2. **prior ("yes"):** We always select the most popular answer ("yes") for both the open-ended and multiple-choice tasks. Note that "yes" is always one of the choices for the multiple-choice questions.

3. **per Q-type prior:** For the open-ended task, we pick the most popular answer per question type (see the appendix for details). For the multiple-choice task, we pick the answer (from the provided choices) that is most similar to the picked answer for the open-ended task using cosine similarity in Word2Vec[50] feature space.

4. **nearest neighbor:** Given a test image, question pair, we first find the K nearest neighbor questions and associated images from the training set. See appendix for details on how neighbors are found. Next, for the open-ended task, we pick the most frequent ground truth answer from this set of nearest neighbor question, image pairs. Similar to the "per Q-type prior" baseline, for the multiple-choice task, we pick the answer (from the provided choices) that is most similar to the picked answer for the open-ended task using cosine similarity in Word2Vec [50] feature space.

4.1.2 Methods

For our methods, we develop a 2-channel vision (image) + language (question) model that culminates with a softmax over K possible outputs. We choose the top $K = 1000$ most frequent answers as possible outputs. This set of answers covers 82.67% of the train+val answers. We describe the different components of our model below:

Image Channel: This channel provides an embedding for the image. We experiment with two embeddings –

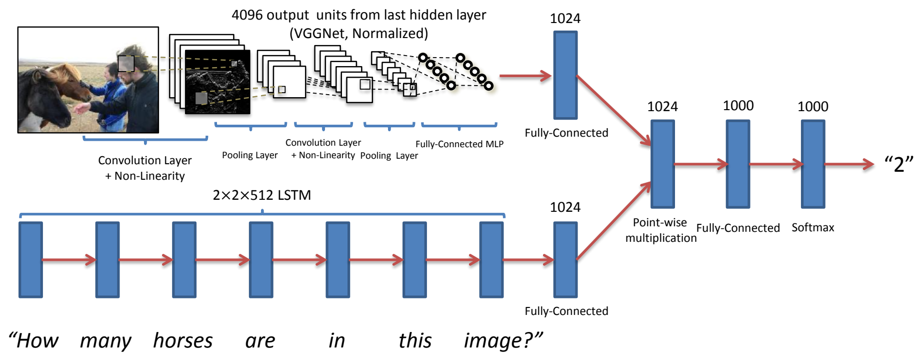


Figure 4.1: Our best performing model (deeper LSTM Q + norm I). This model uses a two layer LSTM to encode the questions and the last hidden layer of VGGNet [139] to encode the images. The image features are then ℓ_2 normalized. Both the question and image features are transformed to a common space and fused via element-wise multiplication, which is then passed through a fully connected layer followed by a softmax layer to obtain a distribution over answers.

1. **I:** The activations from the last hidden layer of VGGNet [139] are used as 4096-dim image embedding.
2. **norm I:** These are ℓ_2 normalized activations from the last hidden layer of VGGNet [139].

Question Channel: This channel provides an embedding for the question. We experiment with three embeddings –

1. **Bag-of-Words Question (BoW Q)**: The top 1,000 words in the questions are used to create a bag-of-words representation. Since there is a strong correlation between the words that start a question and the answer, we find the top 10 first, second, and third words of the questions and create a 30 dimensional bag-of-words representation. These features are concatenated to get a 1,030-dim embedding for the question.
2. **LSTM Q:** An LSTM with one hidden layer is used to obtain 1024-dim embedding for the question. The embedding obtained from the LSTM is a concatenation of last cell state and last hidden state representations (each being 512-dim) from the hidden layer of the LSTM. Each question word is encoded

with 300-dim embedding by a fully-connected layer + tanh non-linearity which is then fed to the LSTM. The input vocabulary to the embedding layer consists of all the question words seen in the training dataset.

3. **deeper LSTM Q:** An LSTM with two hidden layers is used to obtain 2048-dim embedding for the question. The embedding obtained from the LSTM is a concatenation of last cell state and last hidden state representations (each being 512-dim) from each of the two hidden layers of the LSTM. Hence 2 (hidden layers) x 2 (cell state and hidden state) x 512 (dimensionality of each of the cell states, as well as hidden states) in Fig. 4.1. This is followed by a fully-connected layer + tanh non-linearity to transform 2048-dim embedding to 1024-dim. The question words are encoded in the same way as in LSTM Q.

Multi-Layer Perceptron (MLP): The image and question embeddings are combined to obtain a single embedding.

1. For **BoW Q + I** method, we simply concatenate the BoW Q and I embeddings.
2. For **LSTM Q + I**, and **deeper LSTM Q + norm I** (Fig. 4.1) methods, the image embedding is first transformed to 1024-dim by a fully-connected layer + tanh non-linearity to match the LSTM embedding of the question. The transformed image and LSTM embeddings (being in a common space) are then fused via element-wise multiplication.

This combined image + question embedding is then passed to an MLP – a fully connected neural network classifier with 2 hidden layers and 1000 hidden units (dropout 0.5) in each layer with tanh non-linearity, followed by a softmax layer to obtain a distribution over K answers. The entire model is learned end-to-end with a cross-entropy loss. VGGNet parameters are frozen to those learned for ImageNet classification and not fine-tuned in the image channel.

We also experimented with providing captions as input to our model. We assume that a human-generated caption is given as input. We use a bag-of-words repre-

Table 4.1: Accuracy of our methods for the open-ended and multiple-choice tasks on the VQA test-dev for real images. Q = Question, I = Image, C = Caption. (Caption and BoW Q + C results are on val). See text for details.

	Open-Ended				Multiple-Choice			
	All	Yes/No	Number	Other	All	Yes/No	Number	Other
prior ("yes")	29.66	70.81	00.39	01.15	29.66	70.81	00.39	01.15
per Q-type prior	37.54	71.03	35.77	09.38	39.45	71.02	35.86	13.34
nearest neighbor	42.70	71.89	24.36	21.94	48.49	71.94	26.00	33.56
BoW Q	48.09	75.66	36.70	27.14	53.68	75.71	37.05	38.64
I	28.13	64.01	00.42	03.77	30.53	69.87	00.45	03.76
BoW Q + I	52.64	75.55	33.67	37.37	58.97	75.59	34.35	50.33
LSTM Q	48.76	78.20	35.68	26.59	54.75	78.22	36.82	38.78
LSTM Q + I	53.74	78.94	35.24	36.42	57.17	78.95	35.80	43.41
deeper LSTM Q	50.39	78.41	34.68	30.03	55.88	78.45	35.91	41.13
deeper LSTM Q + norm I	**57.75**	**80.50**	**36.77**	**43.08**	**62.70**	**80.52**	**38.22**	**53.01**
Caption	26.70	65.50	02.03	03.86	28.29	69.79	02.06	03.82
BoW Q + C	54.70	75.82	40.12	42.56	59.85	75.89	41.16	52.53

sentation containing the 1,000 most popular words in the captions as the caption embedding (**Caption**). For **BoW Question + Caption (BoW Q + C)** method, we simply concatenate the BoW Q and C embeddings.

For testing, we report the result on two different tasks: open-ended selects the answer with highest activation from all possible K answers and multiple-choice picks the answer that has the highest activation from the potential answers.

4.1.3 Results

Table. 4.1 shows the accuracy of our baselines and methods for both the open-ended and multiple-choice tasks on the VQA test-dev for real images. As expected, the vision-alone model (I) that completely ignores the question performs rather poorly (open-ended: 28.13% / multiple-choice: 30.53%). In fact, on open-ended task, the vision-alone model (I) performs worse than the prior ("yes") baseline, which ignores both the image *and* question (responding to every question with a "yes").

Interestingly, the language-alone methods (per Q-type prior, BoW Q, LSTM Q) that ignore the image perform surprisingly well, with BoW Q achieving 48.09% on open-ended (53.68% on multiple-choice) and LSTM Q achieving 48.76% on open-ended (54.75% on multiple-choice); both outperforming the nearest neighbor baseline (open-ended: 42.70%, multiple-choice: 48.49%). Our quantitative results and analyses suggest that this might be due to the language-model exploiting subtle statistical priors about the question types (e.g. "What color is the banana?" can be answered with "yellow" without looking at the image). For a detailed discussion of the subtle biases in the questions, please see [140].

The accuracy of our **best model** (deeper LSTM Q + norm I (Fig. 4.1), selected using VQA test-dev accuracies) on VQA test-standard is 58.16% (open-ended) / 63.09% (multiple-choice). We can see that our model is able to significantly outperform both the vision-alone and language-alone baselines. As a general trend, results on multiple-choice are better than open-ended. All methods are significantly worse than human performance.

Our VQA demo is available on CloudCV [141] – `http://cloudcv.org/vqa`. This will be updated with newer models as we develop them.

To gain further insights into these results, we computed accuracies by question type in Table. 4.2. Interestingly, for question types that require more reasoning, such as "Is the" or "How many", the scene-level image features do not provide any additional information. However, for questions that can be answered using scene-level information, such as "What sport," we do see an improvement. Similarly, for questions whose answer may be contained in a generic caption we see improvement, such as "What animal". For all question types, the results are worse than human accuracies.

We also analyzed the accuracies of our best model (deeper LSTM Q + norm I) on a subset of questions with certain specific (ground truth) answers. In Fig. 4.2, we show the average accuracy of the model on questions with 50 most frequent ground truth answers on the VQA validation set (plot is sorted by accuracy, not frequency). We can see that the model performs well for answers that are common visual objects such as “wii”, “tennis”, “bathroom” while the performance is somewhat underwhelming for counts (*e.g.*, “2”, “1”, “3”), and particularly poor for higher counts (*e.g.*, “5”, “6”, “10”, “8”, “7”).

In Fig. 4.3, we show the distribution of 50 most frequently predicted answers when the system is correct on the VQA validation set (plot is sorted by prediction frequency, not accuracy). In this analysis, “system is correct” implies that it has VQA accuracy 1.0. We can see that the frequent ground truth answers (*e.g.*, “yes”, “no”, “2”, “white”, “red”, “blue”, “1”, “green”) are more frequently predicted than others when the model is correct.

Table 4.2: Open-ended test-dev results for different question types on real images (Q+C is reported on val). Machine performance is reported using the bag-of-words representation for questions. Questions types are determined by the one or two words that start the question. The percentage of questions for each type is shown in parentheses. Last and second last columns respectively show the average human age and average degree of commonsense required to answer the questions (as reported by AMT workers), respectively. See text for details.

Question	Open-Ended					Human Age	Commonsense
	K = 1000			Human		To Be Able	To Be Able
Type	Q	Q + I	Q + C	Q	Q + I	To Answer	To Answer (%)
what is (13.84)	23.57	34.28	43.88	16.86	73.68	09.07	27.52
what color (08.98)	33.37	43.53	48.61	28.71	86.06	06.60	13.22
what kind (02.49)	27.78	42.72	43.88	19.10	70.11	10.55	40.34
what are (02.32)	25.47	39.10	47.27	17.72	69.49	09.03	28.72
what type (01.78)	27.68	42.62	44.32	19.53	70.65	11.04	38.92
is the (10.16)	70.76	69.87	70.50	65.24	95.67	08.51	30.30
is this (08.26)	70.34	70.79	71.54	63.35	95.43	10.13	45.32
how many (10.28)	43.78	40.33	47.52	30.45	86.32	07.67	15.93
are (07.57)	73.96	73.58	72.43	67.10	95.24	08.65	30.63
does (02.75)	76.81	75.81	75.88	69.96	95.70	09.29	38.97
where (02.90)	16.21	23.49	29.47	11.09	43.56	09.54	36.51
is there (03.60)	86.50	86.37	85.88	72.48	96.43	08.25	19.88
why (01.20)	16.24	13.94	14.54	11.80	21.50	11.18	73.56
which (01.21)	29.50	34.83	40.84	25.64	67.44	09.27	30.00
do (01.15)	77.73	79.31	74.63	71.33	95.44	09.23	37.68
what does (01.12)	19.58	20.00	23.19	11.12	75.88	10.02	33.27
what time (00.67)	8.35	14.00	18.28	07.64	58.98	09.81	31.83
who (00.77)	19.75	20.43	27.28	14.69	56.93	09.49	43.82
what sport (00.81)	37.96	81.12	93.87	17.86	95.59	08.07	31.87
what animal (00.53)	23.12	59.70	71.02	17.67	92.51	06.75	18.04
what brand (00.36)	40.13	36.84	32.19	25.34	80.95	12.50	41.33

Table. 4.3 shows the accuracy of different ablated versions of our best model (deeper LSTM Q + norm I) for both the open-ended and multiple-choice tasks on

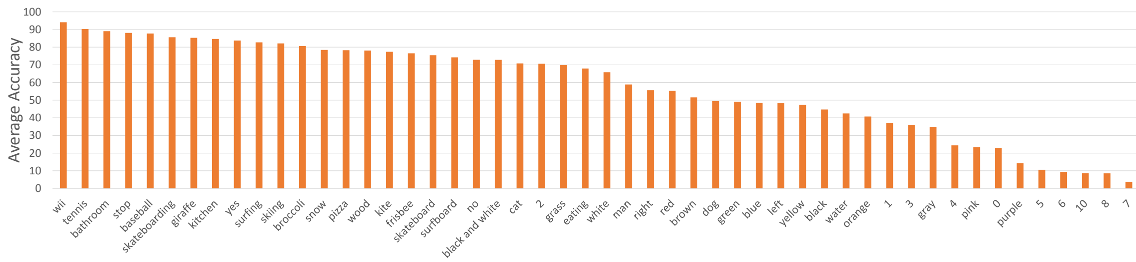


Figure 4.2: Pr (system is correct | answer) for 50 most frequent ground truth answers on the VQA validation set (plot is sorted by accuracy, not frequency). System refers to our best model (deeper LSTM Q + norm I).

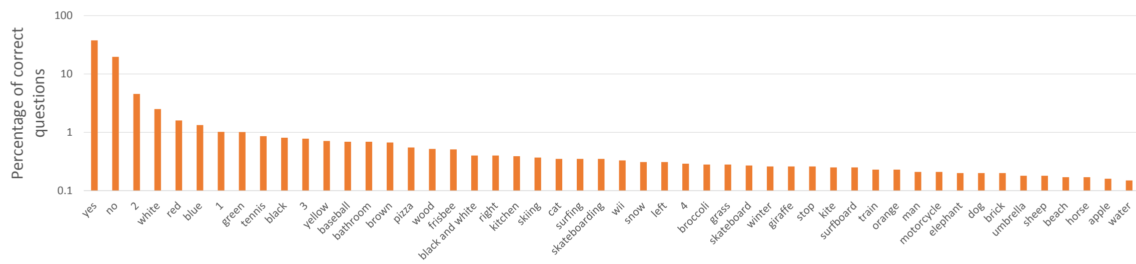


Figure 4.3: Pr (answer | system is correct) for 50 most frequently predicted answers on the VQA validation set (plot is sorted by prediction frequency, not accuracy). System refers to our best model (deeper LSTM Q + norm I).

the VQA test-dev for real images. The different ablated versions are as follows –

1. **Without I Norm:** In this model, the activations from the last hidden layer of VGGNet [139] are not ℓ_2-normalized. Comparing the accuracies in Table. 4.3 and Table. 4.1, we can see that ℓ_2-normalization of image features boosts the performance by 0.16% for open-ended task and by 0.24% for multiple-choice task.

2. **Concatenation:** In this model, the transformed image and LSTM embeddings are concatenated (instead of element-wise multiplied), resulting in doubling the number of parameters in the following fully-connected layer. Comparing the accuracies in Table. 4.3 and Table. 4.1, we can see that element-wise fusion performs better by 0.95% for open-ended task and by 1.24% for multiple-choice task.

3. **K = 500:** In this model, we use K = 500 most frequent answers as possible

outputs. Comparing the accuracies in Table. 4.3 and Table. 4.1, we can see that K = 1000 performs better than K = 500 by 0.82% for open-ended task and by 1.92% for multiple-choice task.

4. **K = 2000:** In this model, we use K = 2000 most frequent answers as possible outputs. Comparing the accuracies in Table. 4.3 and Table. 4.1, we can see that K = 2000 performs better then K = 1000 by 0.40% for open-ended task and by 1.16% for multiple-choice task.

5. **Truncated Q Vocab @ 5:** In this model, the input vocabulary to the embedding layer (which encodes the question words) consists of only those question words which occur atleast 5 times in the training dataset, thus reducing the vocabulary size from 14770 (when all question words are used) to 5134 (65.24% reduction). Remaining question words are replaced with UNK (unknown) tokens. Comparing the accuracies in Table. 4.3 and Table. 4.1, we can see that truncating the question vocabulary @ 5 performs better than using all questions words by 0.24% for open-ended task and by 0.17% for multiple-choice task.

6. **Truncated Q Vocab @ 11:** In this model, the input vocabulary to the embedding layer (which encodes the question words) consists of only those question words which occur atleast 11 times in the training dataset, thus reducing the vocabulary size from 14770 (when all question words are used) to 3561 (75.89% reduction). Remaining question words are replaced with UNK (unknown) tokens. Comparing the accuracies in Table. 4.3 and Table. 4.1, we can see that truncating the question vocabulary @ 11 performs better than using all questions words by 0.06% for open-ended task and by 0.02% for multiple-choice task.

7. **Filtered Dataset:** We created a filtered version of the VQA train + val dataset in which we only keep the answers with subject confidence "yes". Also, we keep only those questions for which at least 50% (5 out of 10) answers are annotated

Table 4.3: Accuracy of ablated versions of our best model (deeper LSTM Q + norm I) for the open-ended and multiple-choice tasks on the VQA test-dev for real images. Q = Question, I = Image. See text for details.

	Open-Ended				Multiple-Choice			
	All	Yes/No	Number	Other	All	Yes/No	Number	Other
Without I Norm	57.59	80.41	36.63	42.84	62.46	80.43	38.10	52.62
Concatenation	56.80	78.49	35.08	43.19	61.46	78.52	36.43	52.54
K = 500	56.93	80.61	36.24	41.39	60.78	80.64	37.44	49.10
K = 2000	58.15	80.56	37.04	43.79	63.86	80.59	38.97	55.20
Truncated Q Vocab @ 5	57.99	80.67	36.99	43.38	62.87	80.71	38.22	53.20
Truncated Q Vocab @ 11	57.81	80.42	36.97	43.22	62.72	80.45	38.30	53.09
Filtered Dataset	56.62	80.19	37.48	40.95	60.82	80.19	37.48	49.57

with subject confidence “yes”. The resulting filtered dataset consists of 344600 questions, compared to 369861 questions in the original dataset, thus leading to only 6.83% reduction in the size of the dataset. The filtered dataset has 8.77 answers per question on average. We did not filter the test set so that accuracies of the model trained on the filtered dataset can be compared with that of the model trained on the original dataset. The row “Filtered Dataset” in Table. 4.3 shows the performance of the deeper LSTM Q + norm I model when trained on the filtered dataset. Comparing these accuracies with the corresponding accuracies in Table. 4.1, we can see that the model trained on filtered version performs worse by 1.13% for open-ended task and by 1.88% for multiple-choice task.

4.2 Hierarchical Co-Attention for Visual Question Answering

In this section, I will explore how to extend from single modality attention to co-attention to improve VQA performance. As shown in Fig. 4.4, we propose a novel mechanism that jointly reasons about visual attention and question attention, which we refer to as co-attention. Unlike previous works, which only focus on visual attention, our model has a natural symmetry between the image and question, in the sense that the image representation is used to guide the question attention and the

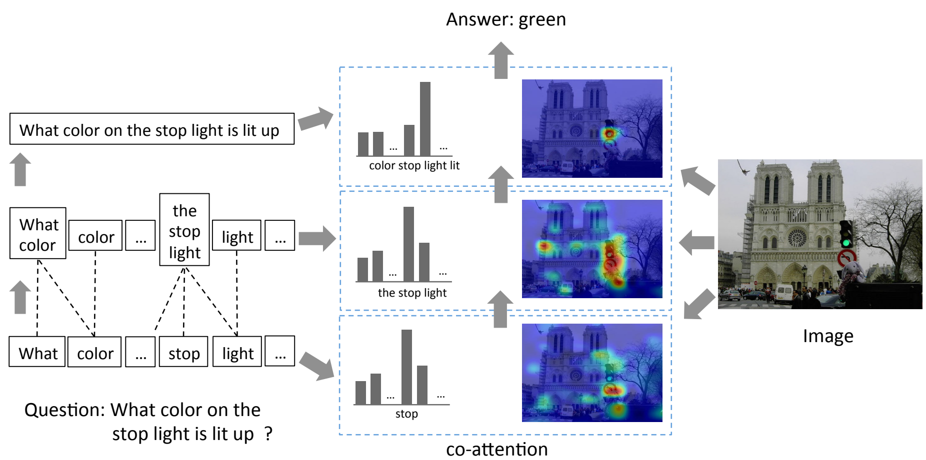


Figure 4.4: Flowchart of our proposed hierarchical co-attention model. Given a question, we extract its word level, phrase level and question level embeddings. At each level, we apply co-attention on both the image and question. The final answer prediction is based on all the co-attended image and question features.

question representation(s) are used to guide image attention.

We build a hierarchical architecture that co-attends to the image and question at three levels: (a) word level, (b) phrase level and (c) question level. At the word level, we embed the words to a vector space through an embedding matrix. At the phrase level, 1-dimensional convolution neural networks are used to capture the information contained in unigrams, bigrams and trigrams. Specifically, we convolve word representations with temporal filters of varying support, and then combine the various n-gram responses by pooling them into a single phrase level representation. At the question level, we use recurrent neural networks to encode the entire question. For each level of the question representation in this hierarchy, we construct joint question and image co-attention maps, which are then combined recursively to ultimately predict a distribution over the answers.

4.2.1 Approach

We begin by introducing the notation used in this chapter. To ease understanding, our full model is described in parts. First, our hierarchical question representation is described in Sec. 4.2.1 and the proposed co-attention mechanism is then described in Sec. 4.2.1. Finally, Sec. 4.2.1 shows how to recursively combine the attended question and image features to output answers.

Question Hierarchy

Given the 1-hot encoding of the question words $\boldsymbol{Q} = \{\boldsymbol{q}_1, \ldots, \boldsymbol{q}_T\}$, we first embed the words to a vector space (learnt end-to-end) to get $\boldsymbol{Q}^w = \{\boldsymbol{q}_1^w, \ldots, \boldsymbol{q}_T^w\}$. To compute the phrase features, we apply 1-D convolution on the word embedding vectors. Concretely, at each word location, we compute the inner product of the word vectors with filters of three window sizes: unigram, bigram and trigram. For the t-th word, the convolution output with window size s is given by

$$\hat{\boldsymbol{q}}_{s,t}^p = \tanh(\boldsymbol{W}_c^s \boldsymbol{q}_{t:t+s-1}^w), \quad s \in \{1, 2, 3\} \tag{4.1}$$

where $\boldsymbol{W}_c^s$ is the weight parameters. The word-level features $\boldsymbol{Q}^w$ are appropriately 0-padded before feeding into bigram and trigram convolutions to maintain the length of the sequence after convolution. Given the convolution result, we then apply max-pooling across different n-grams at each word location to obtain phrase-level features

$$\boldsymbol{q}_t^p = \max(\hat{\boldsymbol{q}}_{1,t}^p, \hat{\boldsymbol{q}}_{2,t}^p, \hat{\boldsymbol{q}}_{3,t}^p), \quad t \in \{1, 2, \ldots, T\} \tag{4.2}$$

Our pooling method differs from those used in previous works [142] in that it adaptively selects different gram features at each time step, while preserving the original sequence length and order. We use a LSTM to encode the sequence $\boldsymbol{q}_t^p$ after max-pooling. The corresponding question-level feature $\boldsymbol{q}_t^s$ is the LSTM hidden vector at time t.

Our hierarchical representation of the question is depicted in Fig. 4.6(a).

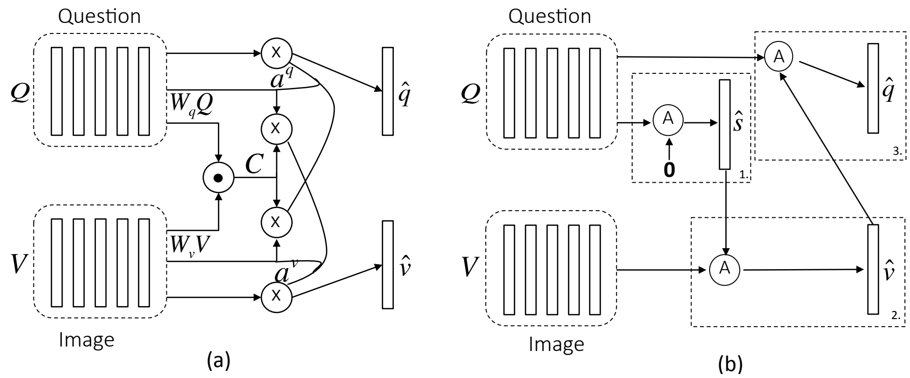


Figure 4.5: (a) Parallel co-attention mechanism; (b) Alternating co-attention mechanism.

Co-Attention

We propose two co-attention mechanisms that differ in the order in which image and question attention maps are generated. The first mechanism, which we call parallel co-attention, generates image and question attention simultaneously. The second mechanism, which we call alternating co-attention, sequentially alternates between generating image and question attentions. See Fig. 4.5. These co-attention mechanisms are executed at all three levels of the question hierarchy.

Parallel Co-Attention. Parallel co-attention attends to the image and question simultaneously. Similar to [72], we connect the image and question by calculating the similarity between image and question features at all pairs of image-locations and question-locations. Specifically, given an image feature map $\boldsymbol{V} \in \mathcal{R}^{d\times N}$, and the question representation $\boldsymbol{Q} \in \mathcal{R}^{d\times T}$, the affinity matrix $\boldsymbol{C} \in \mathcal{R}^{T\times N}$ is calculated by

$$\boldsymbol{C} = \tanh(\boldsymbol{Q}^T \boldsymbol{W}_b \boldsymbol{V}) \tag{4.3}$$

where $\boldsymbol{W}_b \in \mathcal{R}^{d\times d}$ contains the weights. After computing this affinity matrix, one possible way of computing the image (or question) attention is to simply maximize out the affinity over the locations of other modality, *i.e.* $\boldsymbol{a}^v[n] = \max_i(\boldsymbol{C}_{i,n})$ and $\boldsymbol{a}^q[t] = \max_j(\boldsymbol{C}_{t,j})$. Instead of choosing the max activation, we find that performance is improved if we consider this affinity matrix as a feature and learn to predict image and question attention maps via the following

$$\begin{aligned} \boldsymbol{H}^v &= \tanh(\boldsymbol{W}_v\boldsymbol{V} + (\boldsymbol{W}_q\boldsymbol{Q})\boldsymbol{C}), \quad \boldsymbol{H}^q = \tanh(\boldsymbol{W}_q\boldsymbol{Q} + (\boldsymbol{W}_v\boldsymbol{V})\boldsymbol{C}^T) \\ \boldsymbol{a}^v &= \mathrm{softmax}(\boldsymbol{w}_{hv}^T\boldsymbol{H}^v), \quad \boldsymbol{a}^q = \mathrm{softmax}(\boldsymbol{w}_{hq}^T\boldsymbol{H}^q) \end{aligned} \tag{4.4}$$

where $\boldsymbol{W}_v, \boldsymbol{W}_q \in \mathcal{R}^{k\times d}$, $\boldsymbol{w}_{hv}, \boldsymbol{w}_{hq} \in \mathcal{R}^k$ are the weight parameters. $\boldsymbol{a}^v \in \mathcal{R}^N$ and $\boldsymbol{a}^q \in \mathcal{R}^T$ are the attention probabilities of each image region $\boldsymbol{v}_n$ and word $\boldsymbol{q}_t$ respectively. The affinity matrix $\boldsymbol{C}$ transforms question attention space to image attention space (vice versa for $\boldsymbol{C}^T$). Based on the above attention weights, the image and question attention vectors are calculated as the weighted sum of the image features and question features, i.e.,

$$\hat{\boldsymbol{v}} = \sum_{n=1}^{N} a_n^v \boldsymbol{v}_n, \quad \hat{\boldsymbol{q}} = \sum_{t=1}^{T} a_t^q \boldsymbol{q}_t \tag{4.5}$$

The parallel co-attention is done at each level in the hierarchy, leading to $\hat{\boldsymbol{v}}^r$ and $\hat{\boldsymbol{q}}^r$ where $r \in \{w, p, s\}$.

Alternating Co-Attention. In this attention mechanism, we sequentially alternate between generating image and question attention. Briefly, this consists of three steps (marked in Fig. 4.5b): 1) summarize the question into a single vector $\boldsymbol{q}$; 2) attend to the image based on the question summary $\boldsymbol{q}$; 3) attend to the question based on the attended image feature.

Concretely, we define an attention operation $\hat{\boldsymbol{x}} = \mathcal{A}(\boldsymbol{X}; \boldsymbol{g})$, which takes the image (or question) features $\boldsymbol{X}$ and attention guidance $\boldsymbol{g}$ derived from question (or image) as inputs, and outputs the attended image (or question) vector. The operation can

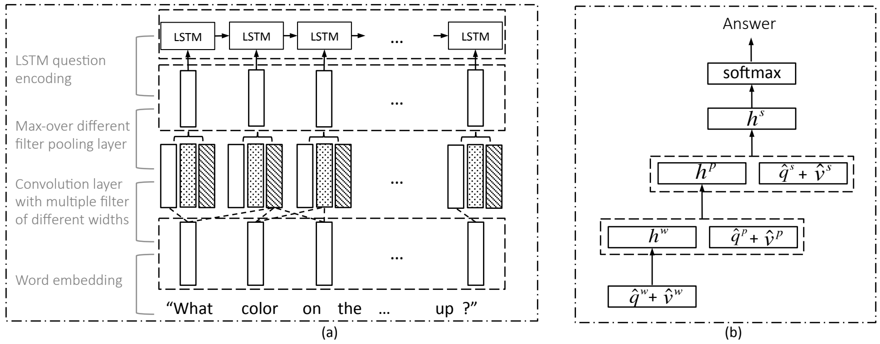


Figure 4.6: (a) Hierarchical question encoding (Sec. 4.2.1); (b) Encoding for predicting answers (Sec. 4.2.1).

be expressed in the following steps

$$
\begin{aligned}
\boldsymbol{H} &= \tanh(\boldsymbol{W}_x\boldsymbol{X} + (\boldsymbol{W}_g\boldsymbol{g})\mathbb{1}^T) \\
\boldsymbol{a}^x &= \text{softmax}(\boldsymbol{w}_{hx}^T\boldsymbol{H}) \\
\hat{\boldsymbol{x}} &= \sum a_i^x \boldsymbol{x}_i
\end{aligned}
\tag{4.6}
$$

where $\mathbb{1}$ is a vector with all elements to be 1. $\boldsymbol{W}_x, \boldsymbol{W}_g \in \mathcal{R}^{k\times d}$ and $\boldsymbol{w}_{hx} \in \mathcal{R}^k$ are parameters. $\boldsymbol{a}^x$ is the attention weight of feature $\boldsymbol{X}$.

The alternating co-attention process is illustrated in Fig. 4.5 (b). At the first step of alternating co-attention, $\boldsymbol{X} = \boldsymbol{Q}$, and $\boldsymbol{g}$ is $\boldsymbol{0}$; At the second step, $\boldsymbol{X} = \boldsymbol{V}$ where $\boldsymbol{V}$ is the image features, and the guidance $\boldsymbol{g}$ is intermediate attended question feature $\hat{\boldsymbol{s}}$ from the first step; Finally, we use the attended image feature $\hat{\boldsymbol{v}}$ as the guidance to attend the question again, i.e., $\boldsymbol{X} = \boldsymbol{Q}$ and $\boldsymbol{g} = \hat{\boldsymbol{v}}$. Similar to the parallel co-attention, the alternating co-attention is also done at each level of the hierarchy.

Encoding for Predicting Answers

Following [3], we treat VQA as a classification task. We predict the answer based on the co-attended image and question features from all three levels. We use a multi-layer perceptron (MLP) to recursively encode the attention features as shown in Fig. 4.6(b).

$$
\begin{aligned}
\boldsymbol{h}^w &= \tanh(\boldsymbol{W}_w(\hat{\boldsymbol{q}}^{\boldsymbol{w}} + \hat{\boldsymbol{v}}^{\boldsymbol{w}})) \\
\boldsymbol{h}^p &= \tanh(\boldsymbol{W}_p[(\hat{\boldsymbol{q}}^{\boldsymbol{p}} + \hat{\boldsymbol{v}}^{\boldsymbol{p}}), \boldsymbol{h}^w]) \\
\boldsymbol{h}^s &= \tanh(\boldsymbol{W}_s[(\hat{\boldsymbol{q}}^{\boldsymbol{s}} + \hat{\boldsymbol{v}}^{\boldsymbol{s}}), \boldsymbol{h}^p]) \\
\boldsymbol{p} &= \text{softmax}(\boldsymbol{W}_h \boldsymbol{h}^s)
\end{aligned}
\tag{4.7}
$$

where $\boldsymbol{W}_w, \boldsymbol{W}_p, \boldsymbol{W}_s$ and $\boldsymbol{W}_h$ are the weight parameters. $[\cdot]$ is the concatenation operation on two vectors. $\boldsymbol{p}$ is the probability of the final answer.

Implementation Details

We use Torch [143] to develop our model. We use the Rmsprop optimizer with a base learning rate of 4e-4, momentum 0.99 and weight-decay 1e-8. We set batch size to be 300 and train for up to 256 epochs with early stopping if the validation accuracy has not improved in the last 5 epochs. For COCO-QA, the size of hidden layer $\boldsymbol{W_s}$ is set to 512 and 1024 for VQA since it is a much larger dataset. All the other word embedding and hidden layers were vectors of size 512. We apply dropout with probability 0.5 on each layer. Following [71], we rescale the image to 448×448, and then take the activation from the last pooling layer of VGGNet [139] or ResNet [144] as its feature.

4.2.2 Results

We evaluate the proposed model on two datasets, the VQA dataset [3] and the COCO-QA dataset [68].

VQA dataset [3] is the largest dataset for this problem, containing human annotated questions and answers on Microsoft COCO dataset [5]. The dataset contains 248,349 training questions, 121,512 validation questions, 244,302 testing questions, and a total of 6,141,630 question-answers pairs. There are three sub-categories according to answer-types including yes/no, number, and other. Each question has 10 free-response answers. We use the top 1000 most frequent answers as the possible outputs similar to [3]. This set of answers covers 86.54% of the train+val answers. For testing, we train our model on VQA train+val and report the test-dev and test-standard results from the VQA evaluation server. We use the evaluation protocol of [3] in the experiment.

COCO-QA dataset [68] is automatically generated from captions in the Microsoft COCO dataset [5]. There are 78,736 train questions and 38,948 test questions in the dataset. These questions are based on 8,000 and 4,000 images respectively. There are four types of questions including object, number, color, and location. Each type takes 70%, 7%, 17%, and 6% of the whole dataset, respectively. All answers in this data set are single word. As in [68], we report classification accuracy as well as Wu-Palmer similarity (WUPS) in Table 2.

Results

There are two test scenarios on VQA: open-ended and multiple-choice. The best performing method **deeper LSTM Q + norm I** from [3] is used as our baseline. For open-ended test scenario, we compare our method with the recent proposed **SMem** [72], **SAN** [71], **FDA** [145] and **DMN+** [74]. For multiple choice, we compare with **Region Sel.** [73] and **FDA** [145]. We compare with **2-VIS+BLSTM** [68], **IMG-CNN** [67] and **SAN** [71] on COCO-QA. We use Ours^p to refer to our parallel co-attention, Ours^a for alternating co-attention.

Table 4.4 shows results on the VQA test sets for both open-ended and multiple-choice settings. We can see that our approach improves the state of art from 60.4% (DMN+ [74]) to 62.1% (Ours^a+ResNet) on open-ended and from 64.2% (FDA [145]) to 66.1% (Ours^a+ResNet) on multiple-choice. Notably, for the question type *Other* and *Num*, we achieve 3.4% and 1.4% improvement on open-ended questions, and 4.0% and 1.1% on multiple-choice questions. As we can see, ResNet features outperform

Table 4.4: Results on the VQA dataset. "-" indicates the results is not available.

	Open-Ended					Multiple-Choice				
	test-dev				test-std	test-dev				test-std
Method	Y/N	Num	Other	All	All	Y/N	Num	Other	All	All
LSTM Q+I [3]	80.5	36.8	43.0	57.8	58.2	80.5	38.2	53.0	62.7	63.1
Region Sel. [73]	-	-	-	-	-	77.6	34.3	55.8	62.4	-
SMem [72]	80.9	37.3	43.1	58.0	58.2	-	-	-	-	-
SAN [71]	79.3	36.6	46.1	58.7	58.9	-	-	-	-	-
FDA [145]	**81.1**	36.2	45.8	59.2	59.5	**81.5**	39.0	54.7	64.0	64.2
DMN+ [74]	80.5	36.8	48.3	60.3	60.4	-	-	-	-	-
Ours^p+VGG	79.5	**38.7**	48.3	60.1	-	79.5	39.8	57.4	64.6	-
Ours^a+VGG	79.6	38.4	49.1	60.5	-	79.7	**40.1**	57.9	64.9	-
Ours^a+ResNet	79.7	**38.7**	**51.7**	**61.8**	**62.1**	79.7	40.0	**59.8**	**65.8**	**66.1**

or match VGG features in all cases. Our improvements are not solely due to the use of a better CNN. Specifically, FDA [145] also uses ResNet [144], but Ours^a+ResNet outperforms it by 1.8% on test-dev. SMem [72] uses GoogLeNet [146] and the rest all use VGGNet [139], and Ours+VGG outperforms them by 0.2% on test-dev (DMN+ [74]).

Table 4.5 shows results on the COCO-QA test set. Similar to the result on VQA, our model improves the state-of-the-art from 61.6% (SAN(2,CNN) [71]) to 65.4% (Ours^a+ResNet). We observe that parallel co-attention performs better than alternating co-attention in this setup. Both attention mechanisms have their advantages and disadvantages: parallel co-attention is harder to train because of the dot product between image and text which compresses two vectors into a single value. On the other hand, alternating co-attention may suffer from errors being accumulated at each round.

Ablation Study

In this section, we perform ablation studies to quantify the role of each component in our model. Specifically, we re-train our approach by ablating certain components:

- *Image Attention alone*, where in a manner similar to previous works [71], we do

Table 4.5: Results on the COCO-QA dataset. "-" indicates the results is not available.

Method	Object	Number	Color	Location	Accuracy	WUPS0.9	WUPS0.0
2-VIS+BLSTM [68]	58.2	44.8	49.5	47.3	55.1	65.3	88.6
IMG-CNN [67]	-	-	-	-	58.4	68.5	89.7
SAN(2, CNN) [71]	64.5	48.6	57.9	54.0	61.6	71.6	90.9
Ours^p+VGG	65.6	49.6	61.5	56.8	63.3	73.0	91.3
Ours^a+VGG	65.6	48.9	59.8	56.7	62.9	72.8	91.3
Ours^a+ResNet	**68.0**	**51.0**	**62.9**	**58.8**	**65.4**	**75.1**	**92.0**

not use any question attention. The goal of this comparison is to verify that our improvements are not the result of orthogonal contributions. (say better optimization or better CNN features).

- *Question Attention alone*, where no image attention is performed.
- *W/O Conv*, where no convolution and pooling is performed to represent phrases. Instead, we stack another word embedding layer on the top of word level outputs.
- *W/O W-Atten*, where no word level co-attention is performed. We replace the word level attention with a uniform distribution. Phrase and question level co-attentions are still modeled.
- *W/O P-Atten*, where no phrase level co-attention is performed, and the phrase level attention is set to be uniform. Word and question level co-attentions are still modeled.
- *W/O Q-Atten*, where no question level co-attention is performed. We replace the question level attention with a uniform distribution. Word and phrase level co-attentions are still modeled.

Table 4.6 shows the comparison of our full approach w.r.t these ablations on the VQA validation set (test sets are not recommended to be used for such experiments). The **deeper LSTM Q + norm I** baseline in [3] is also reported for comparison. We

Table 4.6: Ablation study on the VQA dataset using Ours^a+VGG.

	validation			
Method	Y/N	Num	Other	All
LSTM Q+I	**79.8**	32.9	40.7	54.3
Image Atten	79.8	33.9	43.6	55.9
Question Atten	79.4	33.3	41.7	54.8
W/O Q-Atten	79.6	32.1	42.9	55.3
W/O P-Atten	79.5	34.1	45.4	56.7
W/O W-Atten	79.6	34.4	45.6	56.8
Full Model	79.6	**35.0**	**45.7**	**57.0**

can see that image-attention-alone does improve performance over the holistic image feature (**deeper LSTM Q + norm I**), which is consistent with findings of previous attention models for VQA [74, 71].

Comparing the full model w.r.t. ablated versions without word, phrase, question level attentions reveals a clear interesting trend – the attention mechanisms closest to the 'top' of the hierarchy (*i.e.* question) matter most, with a drop of 1.7% in accuracy if not modeled; followed by the intermediate level (*i.e.* phrase), with a drop of 0.3%; finally followed by the 'bottom' of the hierarchy (*i.e.* word), with a drop of 0.2% in accuracy. We hypothesize that this is because the question level is the 'closest' to the answer prediction layers in our model. Note that *all* levels are important, and our final model significantly outperforms not using any linguistic attention (1.1% difference between Full Model and Image Atten). The question attention alone model is better than LSTM Q+I, with an improvement of 0.5% and worse than image attention alone, with a drop of 1.1%. Ours^a further improves if we performed alternating co-attention for one more round, with an improvement of 0.3%.

Qualitative Results

We now visualize some co-attention maps generated by our method in Fig. 4.7. At the word level, our model attends mostly to the object regions in an image, e.g., heads,

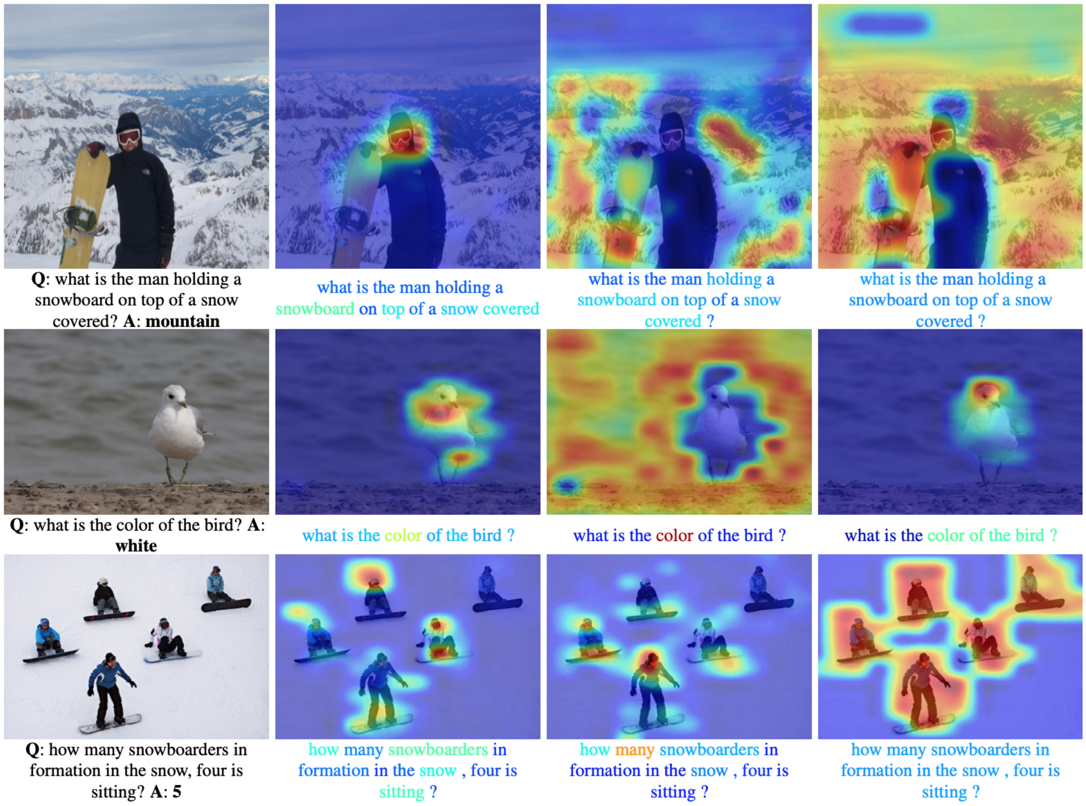


Figure 4.7: Visualization of image and question co-attention maps on the COCO-QA dataset. From left to right: original image and question pairs, word level co-attention maps, phrase level co-attention maps and question level co-attention maps. For visualization, both image and question attentions are scaled (from red:high to blue:low). Best viewed in color.

bird. At the phrase level, the image attention has different patterns across images. For the first two images, the attention transfers from objects to background regions. For the third image, the attention becomes more focused on the objects. We suspect that this is caused by the different question types. On the question side, our model is capable of localizing the key phrases in the question, thus essentially discovering the question types in the dataset. For example, our model pays attention to the phrases "what color" and "how many snowboarders". Our model successfully attends to the regions in images and phrases in the questions appropriate for answering the question, e.g., "color of the bird" and bird region. Because our model performs co-attention at three levels, it often captures complementary information from each level, and then combines them to predict the answer.

4.2.3 Discussion

In this chapter, we proposed a hierarchical co-attention model for visual question answering. Co-attention allows our model to attend to different regions of the image as well as different fragments of the question. We model the question hierarchically at three levels to capture information from different granularities. The ablation studies further demonstrate the roles of co-attention and question hierarchy in our final performance. Through visualizations, we can see that our model co-attends to interpretable regions of images and questions for predicting the answer. Though our model was evaluated on visual question answering, it can be potentially applied to other tasks involving vision and language.

CHAPTER 5

NEURAL IMAGE CAPTIONING

In this chapter, we will study neural image captioning and how inducing appropriate grounding improves the sequence generation model. The rationale for why grounding is useful for sequence generation is based on the idea that the model should know 'when' and 'where' to look at the image when generating descriptions. However, even augmenting with these skills, existing models still lack visual grounding (*i.e.* do not associate named concepts to pixels in the image). In the second work, we will build on top of the first work and propose a novel framework that can produce natural language explicitly grounded in entities that object detectors find in the image. Our approach reconciles classical slot filling approaches (that are generally better grounded in images) with modern neural captioning approaches (that are generally more natural sounding and accurate). Our approach first generates a sentence 'template' with slot locations explicitly tied to specific image regions. These slots are then filled in by visual concepts identified in the regions by object detectors.

5.1 Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning

Automatically generating captions for images has emerged as a prominent interdisciplinary research problem in both academia and industry. [19, 87, 51, 18]. It can aid visually impaired users, and make it easy for users to organize and navigate through large amounts of typically unstructured visual data. In order to generate high quality captions, the model needs to incorporate fine-grained visual clues from the image. Recently, visual attention-based neural encoder-decoder models [18, 19] have been explored, where the attention mechanism typically produces a spatial map highlighting

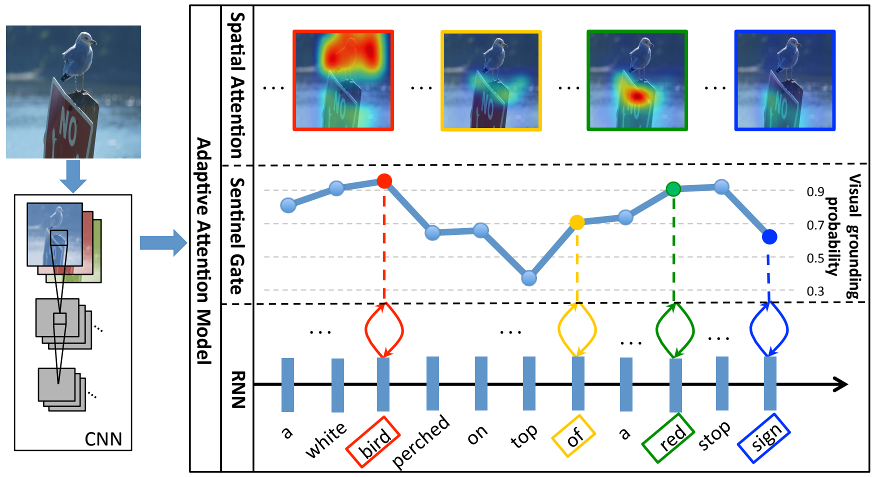


Figure 5.1: Our model learns an adaptive attention model that automatically determines when to look (**sentinel gate**) and where to look (**spatial attention**) for word generation.

image regions relevant to each generated word.

Most attention models for image captioning and visual question answering attend to the image at every time step, irrespective of which word is going to be emitted next [19, 18]. However, not all words in the caption have corresponding visual signals. Consider the example in Fig. 5.1 that shows an image and its generated caption "A white bird perched on top of a red stop sign". The words "a" and "of" do not have corresponding canonical visual signals. Moreover, language correlations make the visual signal unnecessary when generating words like "on" and "top" following "perched", and "sign" following "a red stop". In fact, gradients from non-visual words could mislead and diminish the overall effectiveness of the visual signal in guiding the caption generation process.

In this section, we introduce an adaptive attention encoder-decoder framework which can automatically decide when to rely on visual signals and when to just rely on the language model. Of course, when relying on visual signals, the model also decides where – which image region – it should attend to. We first propose a novel spatial attention model for extracting spatial image features. Then as our proposed adaptive attention mechanism, we introduce a new Long Short Term Memory (LSTM) extension, which produces an additional "**visual sentinel**" vector instead of a single hidden state. The "visual sentinel", an additional latent representation of the decoder's memory, provides a fallback option to the decoder. We further design a new sentinel gate, which decides how much new information the decoder wants to get from the image as opposed to relying on the visual sentinel when generating the next word. For example, as illustrated in Fig. 5.1, our model learns to attend to the image more when generating words "white", "bird", "red" and "stop", and relies more on the visual sentinel when generating words "top", "of" and "sign".

5.1.1 Approach

Spatial Attention Model

First, we propose a spatial attention model for computing the context vector $\boldsymbol{c}_t$ which is defined as:

$$\boldsymbol{c}_t = g(\boldsymbol{V}, \boldsymbol{h}_t) \tag{5.1}$$

where g is the attention function, $\boldsymbol{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k], \boldsymbol{v}_i \in \mathcal{R}^d$ is the spatial image features, each of which is a d dimensional representation corresponding to a part of the image. $\boldsymbol{h}_t$ is the hidden state of RNN at time t.

Given the spatial image feature $\boldsymbol{V} \in \mathcal{R}^{d \times k}$ and hidden state $\boldsymbol{h}_t \in \mathcal{R}^d$ of the LSTM, we feed them through a single layer neural network followed by a softmax function to generate the attention distribution over the k regions of the image:

$$\begin{aligned} \boldsymbol{z}_t &= \boldsymbol{w}_h^T \tanh(\boldsymbol{W}_v \boldsymbol{V} + (\boldsymbol{W}_g \boldsymbol{h}_t)\mathbb{1}^T) & (5.2) \\ \boldsymbol{\alpha}_t &= \mathrm{softmax}(\boldsymbol{z}_t) & (5.3) \end{aligned}$$

where $\mathbb{1} \in \mathcal{R}^k$ is a vector with all elements set to 1. $\boldsymbol{W}_v, \boldsymbol{W}_g \in \mathcal{R}^{k \times d}$ and $\boldsymbol{w}_h \in \mathcal{R}^k$ are

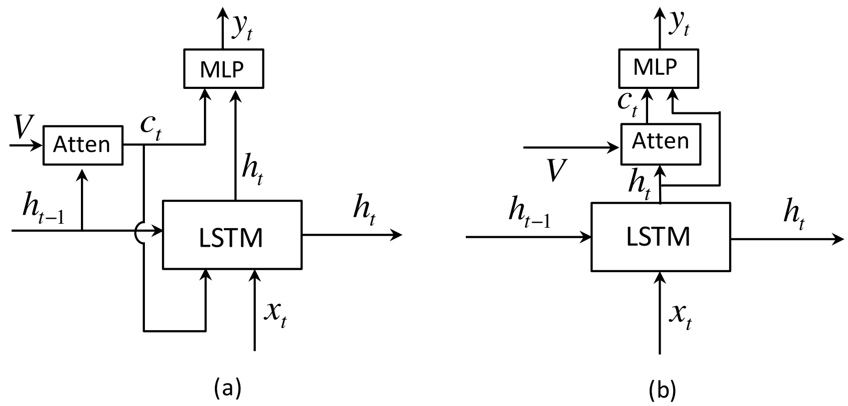


Figure 5.2: A illustration of soft attention model from [18] (a) and our proposed spatial attention model (b).

parameters to be learnt. $\boldsymbol{\alpha} \in \mathcal{R}^k$ is the attention weight over features in $\boldsymbol{V}$. Based on the attention distribution, the context vector $\boldsymbol{c}_t$ can be obtained by:

$$\boldsymbol{c}_t = \sum_{i=1}^{k} \boldsymbol{\alpha}_{ti} \boldsymbol{v}_{ti} \tag{5.4}$$

where $\boldsymbol{c}_t$ and $\boldsymbol{h}_t$ are combined to predict next word y_{t+1}.

Different from [18], shown in Fig. 5.2, we use the current hidden state $\boldsymbol{h}_t$ to analyze where to look (i.e., generating the context vector $\boldsymbol{c}_t$), then combine both sources of information to predict the next word. Our motivation stems from the superior performance of residual network [144]. The generated context vector $\boldsymbol{c}_t$ could be considered as the residual visual information of current hidden state $\boldsymbol{h}_t$, which diminishes the uncertainty or complements the informativeness of the current hidden state for next word prediction. We also empirically find our spatial attention model performs better, as illustrated in Table 5.2 and Table 5.2.

Adaptive Attention Model

While spatial attention based decoders have proven to be effective for image captioning, they cannot determine when to rely on visual signal and when to rely on the language model. In this section, motivated from Merity *et al.* [147], we introduce a new concept – "visual sentinel", which is a latent representation of what the decoder already knows. With the "visual sentinel", we extend our spatial attention model, and propose an adaptive model that is able to determine whether it needs to attend the image to predict next word.

What is visual sentinel? The decoder's memory stores both long and short term visual and linguistic information. Our model learns to extract a new component from this that the model can fall back on when it chooses to not attend to the image. This new component is called the visual sentinel. And the gate that decides whether to attend to the image or to the visual sentinel is the sentinel gate. When the decoder RNN is an LSTM, we consider those information preserved in its memory cell. Therefore, we extend the LSTM to obtain the "visual sentinel" vector $\boldsymbol{s}_t$ by:

$$\boldsymbol{g}_t = \sigma\left(\boldsymbol{W}_x \boldsymbol{x}_t + \boldsymbol{W}_h \boldsymbol{h}_{t-1}\right) \tag{5.5}$$

$$\boldsymbol{s}_t = \boldsymbol{g}_t \odot \tanh\left(\boldsymbol{m}_t\right) \tag{5.6}$$

where $\boldsymbol{W}_x$ and $\boldsymbol{W}_h$ are weight parameters to be learned, $\boldsymbol{x}_t$ is the input to the LSTM at time step t, and $\boldsymbol{g}_t$ is the gate applied on the memory cell $\boldsymbol{m}_t$. $\odot$ represents the element-wise product and σ is the logistic sigmoid activation.

Based on the visual sentinel, we propose an adaptive attention model to compute the context vector. In our proposed architecture (see Fig. 5.3), our new adaptive context vector is defined as $\hat{\boldsymbol{c}}_t$, which is modeled as a mixture of the spatially attended image features (i.e. context vector of spatial attention model) and the visual sentinel vector. This trades off how much new information the network is considering from the image with what it already knows in the decoder memory (i.e., the visual sentinel

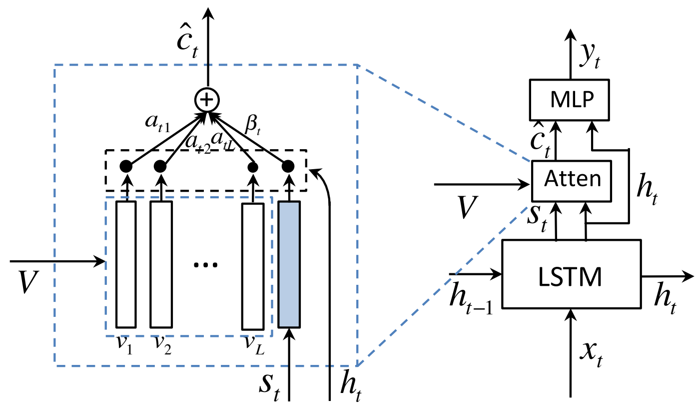


Figure 5.3: An illustration of the proposed model generating the t-th target word y_t given the image.

). The mixture model is defined as follows:

$$\hat{\boldsymbol{c}}_t = \beta_t \boldsymbol{s}_t + (1 - \beta_t)\boldsymbol{c}_t \tag{5.7}$$

where β_t is the new sentinel gate at time t. In our mixture model, β_t produces a scalar in the range $[0, 1]$. A value of 1 implies that only the visual sentinel information is used and 0 means only spatial image information is used when generating the next word.

To compute the new sentinel gate β_t, we modified the spatial attention component. In particular, we add an additional element to z, the vector containing attention scores as defined in Equation 5.2. This element indicates how much "attention" the network is placing on the sentinel (as opposed to the image features). The addition of this extra element is summarized by converting Equation 5.3 to:

$$\hat{\boldsymbol{\alpha}}_t = \mathrm{softmax}([\boldsymbol{z}_t; \boldsymbol{w}_h^T \tanh(\boldsymbol{W}_s \boldsymbol{s}_t + (\boldsymbol{W}_g \boldsymbol{h}_t))]) \tag{5.8}$$

where $[\cdot;\cdot]$ indicates concatenation. $\boldsymbol{W}_s$ and $\boldsymbol{W}_g$ are weight parameters. Notably, $\boldsymbol{W}_g$ is the same weight parameter as in Equation 5.2. $\hat{\boldsymbol{\alpha}}_t \in \mathcal{R}^{k+1}$ is the attention distribution over both the spatial image feature as well as the visual sentinel vector. We interpret the last element of this vector to be the gate value: $\beta_t = \boldsymbol{\alpha}_t[k+1]$.

The probability over a vocabulary of possible words at time t can be calculated as:

$$\boldsymbol{p}_t = \mathrm{softmax}\left(\boldsymbol{W}_p(\hat{\boldsymbol{c}}_t + \boldsymbol{h}_t)\right) \tag{5.9}$$

where $\boldsymbol{W}_p$ is the weight parameters to be learnt.

This formulation encourages the model to adaptively attend to the image vs. the visual sentinel when generating the next word. The sentinel vector is updated at each time step. With this adaptive attention model, we call our framework the adaptive encoder-decoder image captioning framework.

5.1.2 Implementation Details

In this section, we describe the implementation details of our model and how we train our network.

Encoder-CNN. The encoder uses a CNN to get the representation of images. Specifically, the spatial feature outputs of the last convolutional layer of ResNet [144] are used, which have a dimension of $2048 \times 7 \times 7$. We use $\boldsymbol{A} = \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_k\}, \boldsymbol{a}_i \in \mathcal{R}^{2048}$ to represent the spatial CNN features at each of the k grid locations. Following [144], the global image feature can be obtained by:

$$\boldsymbol{a}^g = \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{a}_i \tag{5.10}$$

where $\boldsymbol{a}^g$ is the global image feature. For modeling convenience, we use a single layer perceptron with rectifier activation function to transform the image feature vector into new vectors with dimension d:

$$\begin{aligned} \boldsymbol{v}_i &= \mathrm{ReLU}(\boldsymbol{W}_a \boldsymbol{a}_i) \qquad (5.11) \\ \boldsymbol{v}^g &= \mathrm{ReLU}(\boldsymbol{W}_b \boldsymbol{a}^g) \qquad (5.12) \end{aligned}$$

where $\boldsymbol{W}_a$ and $\boldsymbol{W}_g$ are the weight parameters. The transformed spatial image feature form $\boldsymbol{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k]$.

Decoder-RNN. We concatenate the word embedding vector $\boldsymbol{w}_t$ and global image feature vector $\boldsymbol{v}^g$ to get the input vector $\boldsymbol{x}_t = [\boldsymbol{w}_t; \boldsymbol{v}^g]$. We use a single layer neural network to transform the visual sentinel vector $\boldsymbol{s}_t$ and LSTM output vector $\boldsymbol{h}_t$ into new vectors that have the dimension d.

Training details. In our experiments, we use a single layer LSTM with hidden size of 512. We use the Adam optimizer with base learning rate of 5e-4 for the language model and 1e-5 for the CNN. The momentum and weight-decay are 0.8 and 0.999 respectively. We finetune the CNN network after 20 epochs. We set the batch size to be 80 and train for up to 50 epochs with early stopping if the validation CIDEr [33] score had not improved over the last 6 epochs. Our model can be trained within 30 hours on a single Titan X GPU. We use beam size of 3 when sampling the caption for both COCO and Flickr30k datasets.

5.1.3 Results

Experiment Setting

Flickr30k contains 31,783 images collected from Flickr. Most of these images depict humans performing various activities. Each image is paired with 5 crowd-sourced captions. We use the publicly available splits[1] containing 1,000 images for validation and test each.

[1] https://github.com/karpathy/neuraltalk

Table 5.1: Performance on Flickr30k test splits. † indicates ensemble models. **B-n** is BLEU score that uses up to n-grams. Higher is better in all columns. For future comparisons, our ROUGE-L/SPICE Flickr30k scores are 0.467/0.145

Method	B-1	B-2	B-3	B-4	METEOR	CIDEr
DeepVS [19]	0.573	0.369	0.240	0.157	0.153	0.247
Hard-Attention [18]	0.669	0.439	0.296	0.199	0.185	-
ATT-FCN† [21]	0.647	0.460	0.324	0.230	0.189	-
Ours-Spatial	0.644	0.462	0.327	0.231	0.202	0.493
Ours-Adaptive	**0.677**	**0.494**	**0.354**	**0.251**	**0.204**	**0.531**

Table 5.2: Performance on COCO test splits. † indicates ensemble models. **B-n** is BLEU score that uses up to n-grams. Higher is better in all columns. For future comparisons, our ROUGE-L/SPICE COCO scores are 0.549/0.194

Method	B-1	B-2	B-3	B-4	METEOR	CIDEr
DeepVS [19]	0.625	0.450	0.321	0.230	0.195	0.660
Hard-Attention [18]	0.718	0.504	0.357	0.250	0.230	-
ATT-FCN† [21]	0.709	0.537	0.402	0.304	0.243	-
ERD [89]	-	-	-	0.298	0.240	0.895
MSM† [148]	0.730	0.565	0.429	0.325	0.251	0.986
Ours-Spatial	0.734	0.566	0.418	0.304	0.257	1.029
Ours-Adaptive	**0.742**	**0.580**	**0.439**	**0.332**	**0.266**	**1.085**

COCO is the largest image captioning dataset, containing 82,783, 40,504 and 40,775 images for training, validation and test respectively. This dataset is more challenging, since most images contain multiple objects in the context of complex scenes. Each image has 5 human annotated captions. For offline evaluation, we use the same data split as in [19, 18, 21] containing 5000 images for validation and test each. For online evaluation on the COCO evaluation server, we reserve 2000 images from validation for development and the rest for training.

Pre-processing. We truncate captions longer than 18 words for COCO and 22 for Flickr30k. We then build a vocabulary of words that occur at least 5 and 3 times in the training set, resulting in 9567 and 7649 words for COCO and Flickr30k respectively.

Compared Approaches: For offline evaluation on Flickr30k and COCO, we first compare our full model (**Ours-Adaptive**) with an ablated version (**Ours-Spatial**), which only performs the spatial attention. The goal of this comparison is to verify that our improvements are not the result of orthogonal contributions (e.g. better CNN features or better optimization). We further compare our method with **DeepVS** [19], **Hard-Attention** [18] and recently proposed **ATT** [21], **ERD** [89] and best performed method (LSTM-A_5) of **MSM** [148]. For online evaluation, we compare our method with **Google NIC** [51], **MS Captivator** [22], **m-RNN** [52], **LRCN** [149], **Hard-Attention** [18], **ATT-FCN** [21], **ERD** [89] and **MSM** [148].

Quantitative Analysis

We report results using the COCO captioning evaluation tool [5], which reports the following metrics: BLEU [31], Meteor [150], Rouge-L [32] and CIDEr [33]. We also report results using the new metric SPICE [151], which was found to better correlate with human judgments.

Table 5.1 and Table 5.2 shows results on the Flickr30k and COCO datasets respectively. Comparing the full model w.r.t ablated versions without visual sentinel verifies the effectiveness of the proposed framework. Our adaptive attention model significantly outperforms spatial attention model, which improves the CIDEr score from 0.493/1.029 to 0.531/1.085 on Flickr30k and COCO respectively. When comparing with previous methods, we can see that our single model significantly outperforms all previous methods in all metrics. On COCO, our approach improves the state-of-the-art on BLEU-4 from 0.325 (MSM†) to 0.332, METEOR from 0.251 (MSM†) to 0.266, and CIDEr from 0.986 (MSM†) to 1.085. Similarly, on Flickr30k, our model improves the state-of-the-art with a large margin. We also report scores on ROUGE-L and SPICE for future comparisons.

We compare our model to state-of-the-art systems on the COCO evaluation server

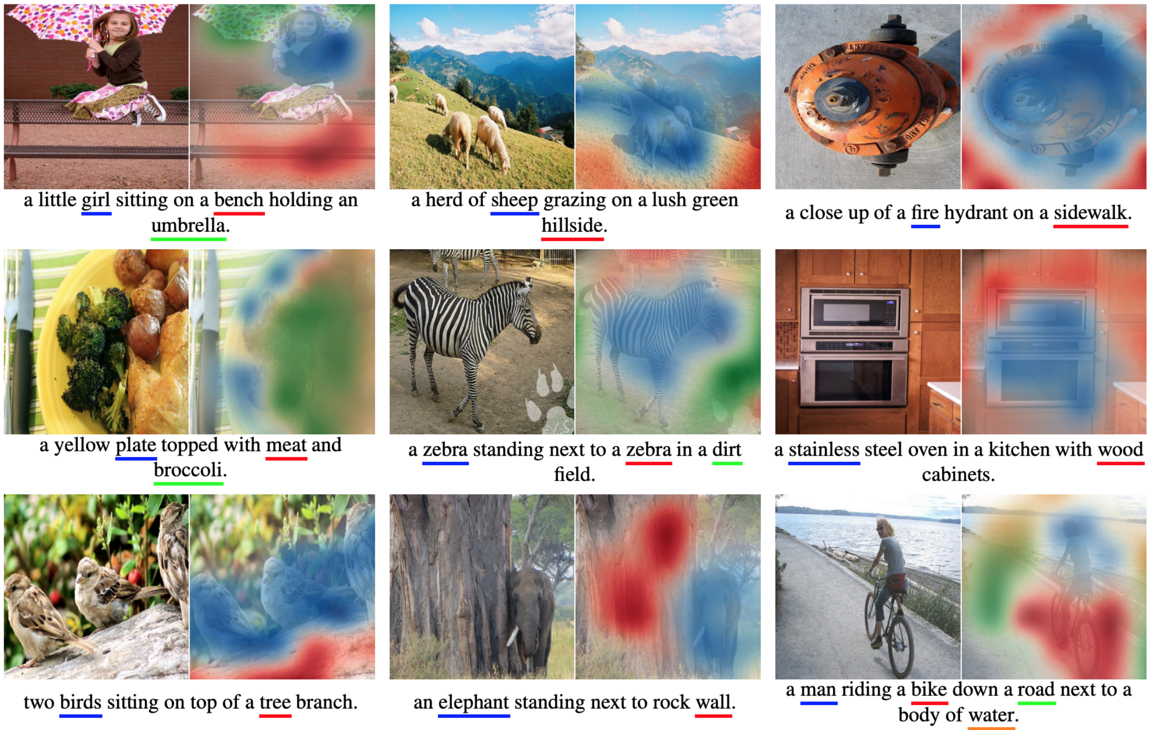


Figure 5.4: Visualization of generated captions and image attention maps on the COCO dataset. Different colors show a correspondence between attended regions and underlined words. First 2 rows are success cases, last row are failure examples. Best viewed in color.

in appendix. We can see that our approach achieves the best performance on all metrics among the published systems. Notably, Google NIC, ERD and MSM use Inception-v3 [152] as the encoder, which has similar or better classification performance compared to ResNet [144] (which is what our model uses).

Qualitative Analysis

To better understand our model, we first visualize the spatial attention weight $\boldsymbol{\alpha}$ for different words in the generated caption. We simply upsample the attention weight to the image size (224×224) using bilinear interpolation. Fig. 5.4 shows generated captions and the spatial attention maps for specific words in the caption. First two columns are success examples and the last one column shows failure examples. We see that our model learns alignments that correspond strongly with human intuition.

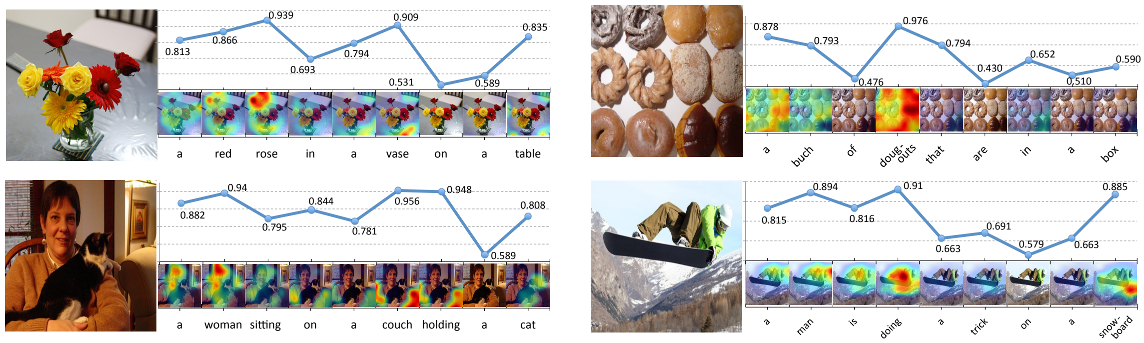


Figure 5.5: Visualization of generated captions, visual grounding probabilities of each generated word, and corresponding spatial attention maps produced by our model.

Note that even in cases where the model produces inaccurate captions, we see that our model does look at reasonable regions in the image – it just seems to not be able to count or recognize texture and fine-grained categories. We provide a more extensive list of visualizations in supplementary material.

We further visualize the sentinel gate as a caption is generated. For each word, we use $1 - \beta$ as its visual grounding probability. In Fig. 5.5, we visualize the generated caption, the visual grounding probability and the spatial attention map generated by our model for each word. Our model successfully learns to attend to the image less when generating non-visual words such as "of" and "a". For visual words like "red", "rose", "doughnuts", "woman" and "snowboard", our model assigns a high visual grounding probabilities (over 0.9). Note that the same word may be assigned different visual grounding probabilities when generated in different contexts. For example, the word "a" usually has a high visual grounding probability at the beginning of a sentence, since without any language context, the model needs the visual information to determine plurality (or not). On the other hand, the visual grounding probability of "a" in the phrase "on a table" is much lower. Since it is unlikely for something to be on more than one table.

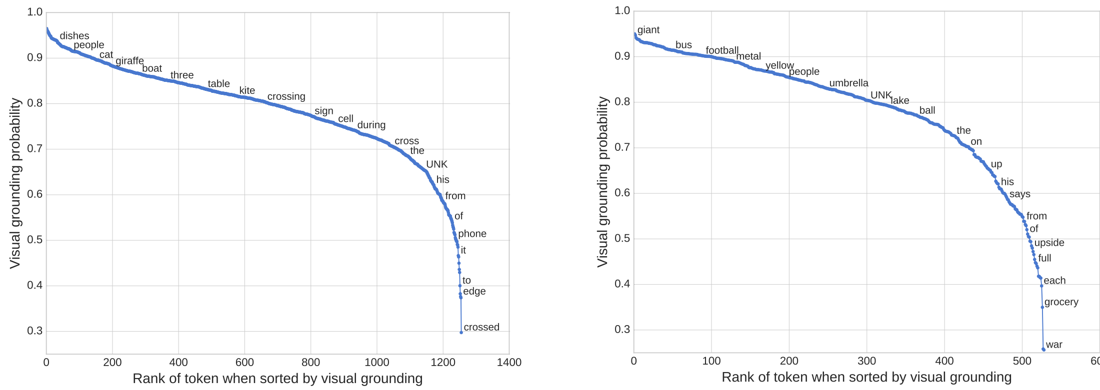


Figure 5.6: Rank-probability plots on COCO (left) and Flickr30k (right) indicating how likely a word is to be visually grounded when it is generated in a caption.

Adaptive Attention Analysis

In this section, we analysis the adaptive attention generated by our methods. We visualize the sentinel gate to understand "when" our model attends to the image as a caption is generated. We also perform a weakly-supervised localization on COCO categories by using the generated attention maps. This can help us to get an intuition of "where" our model attends, and whether it attends to the correct regions.

Learning "when" to attend In order to assess whether our model learns to separate visual words in captions from non-visual words, we visualize the visual grounding probability. For each word in the vocabulary, we average the visual grounding probability over all the generated captions containing that word. Fig. 5.6 shows the rank-probability plot on COCO and Flickr30k.

We find that our model attends to the image more when generating object words like "dishes", "people", "cat", "boat"; attribute words like "giant", "metal", "yellow" and number words like "three". When the word is non-visual, our model learns to not attend to the image such as for "the", "of", "to" etc. For more abstract notions such as "crossing", "during" etc., our model leans to attend less than the visual words and attend more than the non-visual words. Note that our model does not rely on any

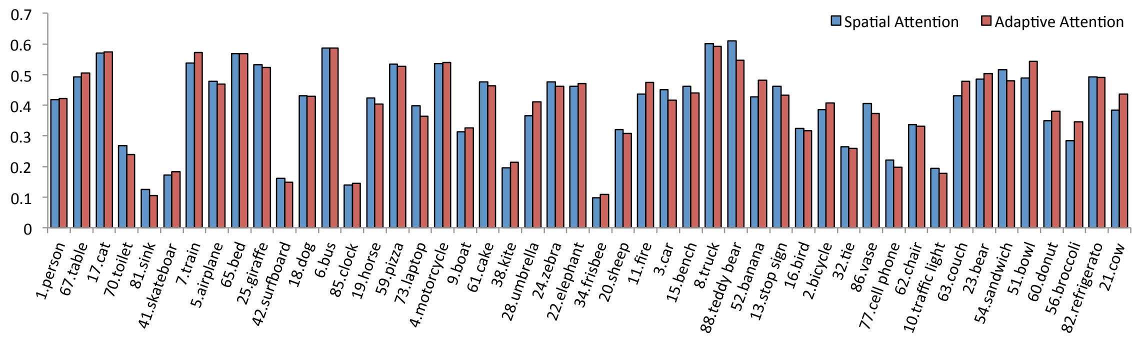


Figure 5.7: Localization accuracy over generated captions for top 45 most frequent COCO object categories. "Spatial Attention" and "Adaptive Attention" are our proposed spatial attention model and adaptive attention model, respectively. The COCO categories are ranked based on the align results of our adaptive attention, which cover 93.8% and 94.0% of total matched regions for spatial attention and adaptive attention, respectively.

syntactic features or external knowledge. It discovers these trends automatically.

Our model cannot distinguish between words that are truly non-visual from the ones that are technically visual but have a high correlation with other words and hence chooses to not rely on the visual signal. For example, words such as "phone" get a relatively low visual grounding probability in our model. This is because it has a large language correlation with the word "cell". We can also observe some interesting trends in what the model learns on different datasets. For example, when generating "UNK" words, our model learns to attend less to the image on COCO, but more on Flickr30k. Same words with different forms can also results in different visual grounding probabilities. For example, "crossing", "cross" and "crossed" are cognate words which have similar meaning. However, in terms of the visual grounding probability learnt by our model, there is a large variance. Our model learns to attend to images more when generating "crossing", followed by "cross" and attend least on image when generating "crossed".

Learning "where" to attend We now assess whether our model attends to the correct spatial image regions. We perform weakly-supervised localization [153, 154] using the generated attention maps. To the best of our best knowledge, no previous works have used weakly supervised localization to evaluate spatial attention for image captioning. Given the word w_t and attention map $\boldsymbol{\alpha}_t$, we first segment the regions of of the image with attention values larger than th (after map is normalized to have the largest value be 1), where th is a per-class threshold estimated using the COCO validation split. Then we take the bounding box that covers the largest connected component in the segmentation map. We use intersection over union (IOU) of the generated and ground truth bounding box as the localization accuracy.

For each of the COCO object categories, we do a word-by-word match to align the generated words with the ground truth bounding box[2]. For the object categories which has multiple words, such as "teddy bear", we take the maximum IOU score over the multiple words as its localization accuracy. We are able to align 5981 and 5924 regions for captions generated by the spatial and adaptive attention models respectively. The average localization accuracy for our spatial attention model is **0.362**, and **0.373** for our adaptive attention model. This demonstrates that as a byproduct, knowing when to attend also helps where to attend.

Fig. 5.7 shows the localization accuracy over the generated captions for top 45 most frequent COCO object categories. We can see that our spatial attention and adaptive attention models share similar trends. We observe that both models perform well on categories such as "cat", "bed", "bus" and "truck". On smaller objects, such as "sink", "surfboard", "clock" and "frisbee", both models perform relatively poorly. This is because our spatial attention maps are directly rescaled from a coarse 7×7 feature map, which looses a lot of spatial resolution and detail. Using a larger feature map may improve the performance.

[2]Since one object category can have multiple words corresponding to it, we manually create a mapping in order to cover more samples. The list can be found in supplementary material.

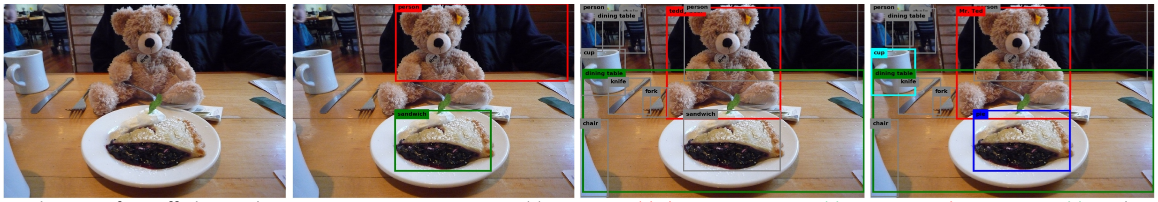
5.1.4 Discussion

We present a novel adaptive attention encoder-decoder framework, which provides a fallback option to the decoder. To realize the adaptive attention mechanism, we introduce a new LSTM extension, which produces an additional "visual sentinel" vector instead of the single hidden state. Our model achieves state-of-the-art performance across standard benchmarks on image captioning. We also perform extensive attention evaluation to analysis our adaptive attention. Through visualization, we can see our model adaptive attends to interpretable regions when generating the captions. Though our model is evaluated on image captioning, it can be potentially applied to a more general attention encoder-decoder framework.

5.2 Neural Baby Talk

Next, I will discuss an approach that can produce natural language explicitly grounded in entities that object detectors find in the image. Our approach is motivated by some of the first attempts at image captioning [25, 24] – before the deep learning "revolution" – relied heavily on outputs of object detectors and attribute classifiers to describe images. For instance, consider the output of Baby Talk [24] in Fig. 1.3 from introduction, that used a slot filling approach to talk about all the objects and attributes found in the scene via a templated caption. The language is unnatural but the caption is very much grounded in what the model sees in the image. Today's approaches fall at the other extreme on the spectrum – the language generated by modern neural image captioning approaches is much more natural but tends to be much less grounded in the image.

In this section, we introduce Neural Baby Talk that reconciles these methodologies. It produces natural language *explicitly* grounded in entities found by object detectors. It is a neural approach that generates a sentence "template" with slot



A close up of a stuffed animal on a plate. | A person is sitting at a table with a sandwich. | A teddy bear sitting on a table with a plate of food. | A Mr. Ted sitting at a table with a pie and a cup of coffee.

Figure 5.8: From left to right is the generated caption using the same captioning model but with different detectors: 1) No detector; 2) A weak detector that only detects “person” and “sandwich”; 3) A detector trained on COCO [5] categories (including “teddy bear”). 4) A detector that can detect novel concepts (e.g. “Mr. Ted” and “pie” that never occurred in the captioning training data). Different colors show a correspondence between the visual word and grounding regions.

locations explicitly tied to image regions. These slots are then filled by object recognizers with concepts found in the regions. The entire approach is trained end-to-end. This results in natural sounding and grounded captions.

Our main technical contribution is a novel neural decoder for grounded image captioning. Specifically, at each time step, the model decides whether to generate a word from the textual vocabulary or generate a “visual” word. The visual word is essentially a token that will hold the slot for a word that is to describe a specific region in the image. For instance, for the image in Fig. 1.3, the generated sequence may be “A <region−17> is sitting at a <region−123> with a <region−3>.” The visual words (<region−[.]>’s) are then filled in during a second stage that classifies each of the indicated regions (e.g., <region−17>→puppy, <region−123>→table), resulting in a final description of “A puppy is sitting at a table with a cake.” – a free-form natural language description that is grounded in the image. One nice feature of our model is that it allows for different object detectors to be plugged in easily. As a result, a variety of captions can be produced for the same image using different detection backends. See Fig. 5.8 for an illustration.

5.2.1 Approach

Given an image $\boldsymbol{I}$, the goal of our method is to generate visually grounded descriptions $\boldsymbol{y} = \{y_1, \ldots, y_T\}$. Let $\boldsymbol{r_I} = \{r_1, ..., r_N\}$ be the set of N images regions extracted from $\boldsymbol{I}$. When generating an entity word in the caption, we want to ground it in a specific image region $r \in \boldsymbol{r_I}$. Following the standard supervised learning paradigm, we learn parameters $\boldsymbol{\theta}$ of our model by maximizing the likelihood of the correct caption:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \sum_{(\boldsymbol{I},\boldsymbol{y})} \log p(\boldsymbol{y}|\boldsymbol{I};\boldsymbol{\theta}) \tag{5.13}$$

Using chain rule, the joint probability distribution can be decomposed over a sequence of tokens:

$$p(\boldsymbol{y}|\boldsymbol{I}) = \prod_{t=1}^{T} p(y_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{I}) \tag{5.14}$$

where we drop the dependency on model parameters to avoid notational clutter. We introduce a latent variable r_t to denote a specific image region so that y_t can explicitly ground in it. Thus the probability of y_t is decomposed to:

$$p(y_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{I}) = p(y_t|r_t, \boldsymbol{y}_{1:t-1}, \boldsymbol{I})p(r_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{I}) \tag{5.15}$$

In our framework, y_t can be of one of two types: a visual word or a textual word, denoted as y^{vis} and y^{txt} respectively. A visual word y^{vis} is a type of word that is grounded in a specific image region drawn from $\boldsymbol{r_I}$. A textual word y^{txt} is a word from the remainder of the caption. It is drawn from the language model , which is associated with a "default" sentinel "region" $\tilde{r}$ obtained from the language model [155] (discussed in Sec. 5.2.1). For example, as illustrated in Fig. 1.3, "puppy" and "cake" grounded in the bounding box of category "dog" and "cake" respectively, are visual words. While "with" and "sitting" are not associated with any image regions and thus are textual words.

With this, Eq. 5.13 can be decomposed into two cascaded objectives. First, maximizing the probability of generating the sentence "template". A sequence of grounding regions associated with the visual words interspersed with the textual words can be viewed as a sentence "template", where the grounding regions are slots to be filled in with visual words.[3] An example template (Fig. 5.9) is "A <region−2> is laying on the <region−4> near a <region−7>. Second, maximizing the probability of visual words y_t^{vis} conditioned on the grounding regions and object detection information, e.g., categories recognized by detector. In the template example above, the model will fill the slots with 'cat', 'laptop' and 'chair' respectively.

In the following, we first describe how we generate the slotted caption template (Sec. 5.2.1), and then how the slots are filled in to obtain the final image description (Sec. 5.2.1). The overall objective function is described in Sec. 5.2.1 and the implementation details in Sec. 5.2.2.

"Slotted" Caption Template Generation

Given an image $\boldsymbol{I}$, and the corresponding caption $\boldsymbol{y}$, the candidate grounding regions are obtained by using a pre-trained Faster-RCNN network [156]. To generate the caption "template", we use a recurrent neural network, which is commonly used as the decoder for image captioning [52, 51]. At each time step, we compute the RNN hidden state $\boldsymbol{h}_t$ according to the previous hidden state $\boldsymbol{h}_{t-1}$ and the input $\boldsymbol{x}_t$ such that $\boldsymbol{h}_t = \text{RNN}(\boldsymbol{x}_t, \boldsymbol{h}_{t-1})$. At training time, x_t is the ground truth token (teacher forcing) and at test time is the sampled token y_{t-1}. Our decoder consists of an attention based LSTM layer [157] that takes convolution feature maps as input. Details can be found in Sec. 5.2.2. To generate the "slot" for visual words, we use a pointer network [158]

[3]Our approach is not limited to any pre-specified bank of templates. Rather, our approach automatically generates a template (with placeholders – slots – for visually grounded words), which may be any one of the exponentially many possible templates.

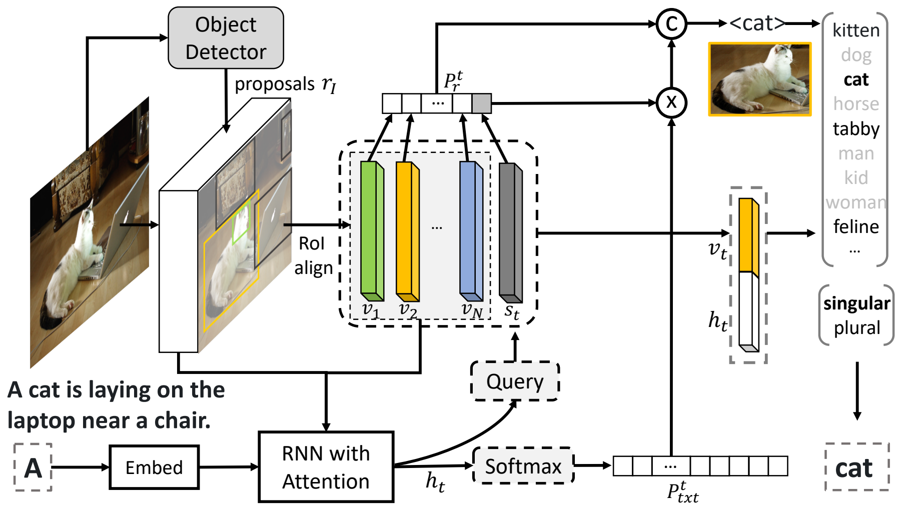


Figure 5.9: One block of the proposed approach. Given an image, proposals from any object detector and current word "A", the figure shows the process to predict the next visual word "cat".

that modulates a content-based attention mechanism over the grounding regions. Let $\boldsymbol{v_t} \in \mathcal{R}^{d\times 1}$ be the region feature of r_t, which is calculated based on Faster R-CNN. We compute the pointing vector with:

$$u_i^t = \boldsymbol{w}_h^T \tanh(\boldsymbol{W}_v \boldsymbol{v}_t + \boldsymbol{W}_z \boldsymbol{h}_t) \tag{5.16}$$

$$\boldsymbol{P}_{\boldsymbol{r_I}}^t = \text{softmax}(\boldsymbol{u}^t) \tag{5.17}$$

where $\boldsymbol{W}_v \in \mathbb{R}^{m\times d}$, $\boldsymbol{W}_z \in \mathbb{R}^{d\times d}$ and $\boldsymbol{w}_h \in \mathbb{R}^{d\times 1}$ are parameters to be learned. The softmax normalizes the vector $\boldsymbol{u}^t$ to be a distribution over grounding regions $\boldsymbol{r_I}$.

Since textual words y_t^{txt} are not tied to specific regions in the image, inspired by [155], we add a "visual sentinel" $\tilde{r}$ as a latent variable to serve as dummy grounding for the textual word. The visual sentinel can be thought of as a latent representation of what the decoder already knows about the image. The probability of a textual word y_t^{txt} then is:

$$p(y_t^{txt}|\boldsymbol{y}_{1:t-1}) = p(y_t^{txt}|\tilde{r}, \boldsymbol{y}_{1:t-1})p(\tilde{r}|\boldsymbol{y}_{1:t-1}) \tag{5.18}$$

where we drop the dependency on $\boldsymbol{I}$ to avoid clutter.

We first describe how the visual sentinel is computed, and then how the textual words are determined based on the visual sentinel. Following [155], when the decoder RNN is an LSTM [20], the representation for visual sentinel $\boldsymbol{s}_t$ can be obtained by:

$$\boldsymbol{g}_t = \sigma\left(\boldsymbol{W}_x \boldsymbol{x}_t + \boldsymbol{W}_h \boldsymbol{h}_{t-1}\right) \tag{5.19}$$

$$\boldsymbol{s}_t = \boldsymbol{g}_t \odot \tanh\left(\boldsymbol{c}_t\right) \tag{5.20}$$

where $\boldsymbol{W}_x \in \mathbb{R}^{d\times d}$, $\boldsymbol{W}_h \in \mathbb{R}^{d\times d}$. $\boldsymbol{x}_t$ is the LSTM input at time step t, and $\boldsymbol{g}_t$ is the gate applied on the cell state $\boldsymbol{c}_t$. $\odot$ represents element-wise product, σ the logistic sigmoid activation. Modifying Eq. 5.17, the probability over the grounding regions including the visual sentinel is:

$$\boldsymbol{P}_{\boldsymbol{r}}^t = \mathrm{softmax}([\boldsymbol{u}^t; \boldsymbol{w}_h^T \tanh(\boldsymbol{W}_s \boldsymbol{s}_t + \boldsymbol{W}_z \boldsymbol{h}_t)]) \tag{5.21}$$

where $\boldsymbol{W}_s \in \mathbb{R}^{d\times d}$ and $\boldsymbol{W}_z \in \mathbb{R}^{d\times d}$ are the parameters. Notably, $\boldsymbol{W}_z$ and $\boldsymbol{w}_h$ are the same parameters as in Eq. 5.16. $\boldsymbol{P}_{\boldsymbol{r}}^t$ is the probability distribution over grounding regions $\boldsymbol{r}_{\boldsymbol{I}}$ and visual sentinel $\tilde{r}$. The last element of the vector in Eq. 5.21 captures $p(\tilde{r}|\boldsymbol{y}_{1:t-1})$.

We feed the hidden state $\boldsymbol{h}_t$ into a softmax layer to obtain the probability over textual words conditioned on the image, all previous words, and the visual sentinel:

$$\boldsymbol{P}_{txt}^t = \mathrm{softmax}\left(\boldsymbol{W}_q \boldsymbol{h}_t\right) \tag{5.22}$$

where $\boldsymbol{W}_q \in \mathbb{R}^{V\times d}$, d is hidden state size, and V is textual vocabulary size. Plugging in Eq. 5.22 and $p(\tilde{r}|\boldsymbol{y}_{1:t-1})$ from the last element of the vector in Eq. 5.21 into Eq. 5.18 gives us the probability of generating a textual word in the template.

Caption Refinement: Filling in The Slots

To fill the slots in the generated template with visual words grounded in image regions, we leverage the outputs of an object detection network. Given a grounding region, the category can be obtained through any detection framework [156]. But outputs of detection networks are typically singular coarse labels *e.g.* "dog". Captions often refer to these entities in a fine-grained fashion *e.g.* "puppy" or in the plural form "dogs". In order to accommodate for these linguistic variations, the visual word y^{vis} in our model is a refinement of the category name by considering the following two factors: First, determine the plurality – whether it should be singular or plural. Second, determine the fine-grained class (if any). Using two single layer MLPs with ReLU activation $f(\cdot)$, we compute them with:

$$\boldsymbol{P}_b^t = \mathrm{softmax}\left(\boldsymbol{W}_b f_b\left([\boldsymbol{v}_t; \boldsymbol{h}_t]\right)\right) \tag{5.23}$$

$$\boldsymbol{P}_g^t = \mathrm{softmax}\left(\boldsymbol{U}^T \boldsymbol{W}_g f_g\left([\boldsymbol{v}_t; \boldsymbol{h}_t]\right)\right) \tag{5.24}$$

$\boldsymbol{W}_b \in \mathbb{R}^{2\times d}$, $\boldsymbol{W}_g \in \mathbb{R}^{300\times d}$ are the weight parameters. $\boldsymbol{U} \in \mathbb{R}^{300\times k}$ is the glove vector embeddings [159] for k fine-grained words associated with the category name. The visual word y_t^{vis} is then determined by plurality and fine-grained class (*e.g.*, if plurality is plural, and the fine-grained class is "puppy", the visual word will be "puppies").

Objective

Most standard image captioning datasets (*e.g.* COCO [5]) do not contain phrase grounding annotations, while some datasets do (*e.g.* Flickr30k [160]). Our training objective (presented next) can incorporate different kinds of supervision – be it strong annotations indicating which words in the caption are grounded in which boxes in the image, or weak supervision where objects are annotated in the image but are not aligned to words in the caption. Given the target ground truth caption $\boldsymbol{y}^*_{1:T}$ and a image captioning model with parameters $\boldsymbol{\theta}$, we minimize the cross entropy loss:

$$L(\boldsymbol{\theta}) = -\sum_{t=1}^{T} \log \Bigg(\overbrace{p(y_t^*|\tilde{r}, \boldsymbol{y}^*_{1:t-1}) p(\tilde{r}|\boldsymbol{y}^*_{1:t-1}) \mathbb{1}_{(y_t^*=y^{\text{txt}})}}^{\text{Textual word probability}} + \\ \underbrace{p\left(b_t^*, s_t^* | \boldsymbol{r}_t, \boldsymbol{y}^*_{1:t-1}\right)}_{\text{Caption refinement}} \underbrace{\left(\frac{1}{m} \sum_{i=1}^{m} p\left(r_t^i | \boldsymbol{y}^*_{1:t-1}\right)\right) \mathbb{1}_{(y_t^*=y^{\text{vis}})}}_{\text{Averaged target region probability}} \Bigg) \tag{5.25}$$

where y_t^* is the word from the ground truth caption at time t. $\mathbb{1}_{(y_t^*=y^{\text{txt}})}$ is the indicator function which equals to 1 if y_t^* is textual word and 0 otherwise. b_t^* and s_t^* are the target ground truth plurality and find-grained class. $\{r_t^i\}_{i=1}^m \in \boldsymbol{r_I}$ are the target grounding regions of the visual word at time t. We maximize the averaged log probability of the target grounding regions.

Visual word extraction. During training, visual words in a caption are dynamically identified by matching the base form of each word (using the Stanford lemmatization toolbox [161]) against a vocabulary of visual words (details of how to get visual word can be found in dataset Sec. 5.2.3). The grounding regions $\{r_t^i\}_{i=1}^m$ for a visual word y_t is identified by computing the IoU of all boxes detected by the object detection network with the ground truth bounding box associated with the category corresponding to y_t. If the score exceeds a threshold of 0.5 and the grounding region label matches the visual word, the bounding boxes are selected as the grounding regions. E.g., given a target visual word "cat", if there are no proposals that match the target bounding box, the model predicts the textual word "cat" instead.

5.2.2 Implementation Details

Detection model. We use Faster R-CNN [156] with ResNet-101 [144] to obtain region proposals for the image. We use an IoU threshold of 0.7 for region proposal suppression and 0.3 for class suppressions. A class detection confidence threshold of

0.5 is used to select regions.

Region feature. We use a pre-trained ResNet-101 [144] in our model. The image is first resized to 576×576 and we random crop 512×512 as the input to the CNN network. Given proposals from the pre-trained detection model, the feature $\boldsymbol{v}_i$ for region i is a concatenation of 3 different features $\boldsymbol{v}_i = [\boldsymbol{v}_i^p; \boldsymbol{v}_i^l; \boldsymbol{v}_i^g]$ where $\boldsymbol{v}_i^p$ is the pooling feature of RoI align layer [162] given the proposal coordinates, $\boldsymbol{v}_i^l$ is the location feature and $\boldsymbol{v}_i^g$ is the glove vector embedding of the class label for region i. Let $x_{\min}, y_{\min}, x_{\max}, y_{\max}$ be the bounding box coordinates of the region b; W_I and H_I be the width and height of the image I. Then the location feature $\boldsymbol{v}_i^l$ can be obtained by projecting the normalized location $[\frac{x_{\min}}{W_I}, \frac{y_{\min}}{H_I}, \frac{x_{\max}}{W_I}, \frac{y_{\max}}{H_I}]$ into another embedding space.

Language model. We use an attention model with two LSTM layers [163] as our base attention model. Given N region features from detection proposals $\boldsymbol{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N\}$ and CNN features from the last convolution layer at K grids $\hat{\boldsymbol{V}} = \{\hat{\boldsymbol{v}}_1, \ldots, \hat{\boldsymbol{v}}_K\}$, the language model has two separate attention layers shown in Fig 5.10. The attention distribution over the image features for detection proposals is:

$$
\begin{aligned}
\boldsymbol{z}_t &= \boldsymbol{w}_z^T \tanh\left(\boldsymbol{W}_v \boldsymbol{V} + (\boldsymbol{W_g h_t})\mathbb{1}^T\right) \\
\boldsymbol{\alpha}_t &= \mathrm{softmax}(\boldsymbol{z}_t)
\end{aligned}
\tag{5.26}
$$

where $\boldsymbol{W}_v \in \mathbb{R}^{m \times d}$, $\boldsymbol{W}_g \in \mathbb{R}^{d \times d}$ and $\boldsymbol{w} \in \mathbb{R}^{d \times 1}$. $\mathbb{1} \in \mathbb{R}^N$ is a vector with all elements set to 1. $\boldsymbol{\alpha}_t$ is the attention weight over N image location features.

Training details. In our experiments, we use a two layer LSTM with hidden size 1024. The number of hidden units in the attention layer and the size of the input word embedding are 512. We use the Adam [164] optimizer with an initial learning rate of 5×10^{-4} and anneal the learning rate by a factor of 0.8 every three epochs. We train the model up to 50 epochs with early stopping. Note that we do not finetune the CNN network during training. We set the batch size to be 100 for COCO [5] and

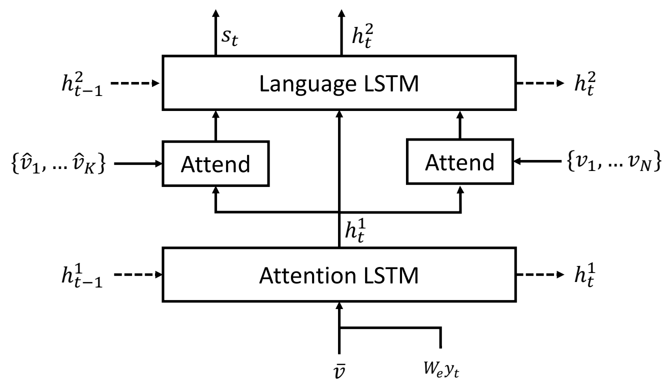


Figure 5.10: Language model used in our approach.

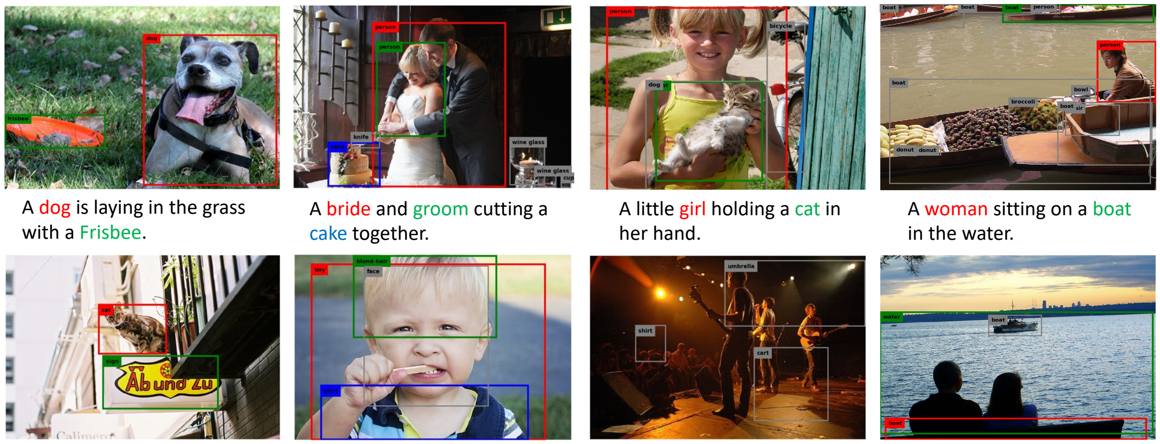
50 for Flickr30k [160].

5.2.3 Experimental Results

Datasets. We experiment with two datasets. Flickr30k Entities [160] contains 275,755 bounding boxes from 31,783 images associated with natural language phrases. Each image is annotated with 5 crowdsourced captions. For each annotated phrase in the caption, we identify visual words by selecting the inner most NP (noun phrase) tag from the Stanford part-of-speech tagger [165]. We use Stanford Lemmatization Toolbox [161] to get the base form of the entity words resulting in 2,567 unique words.

COCO [5] contains 82,783, 40,504 and 40,775 images for training, validation and testing respectively. Each image has around 5 crowdsourced captions. Unlike Flickr30k Entities, COCO does not have bounding box annotations associated with specific phrases or entities in the caption. To identify visual words, we manually constructed an object category to word mapping that maps object categories like <person> to a list of potential fine-grained labels like ["child", "baker", ...]. This results in 80 categories with a total of 413 fine-grained classes. See supp. for details.

Detector pre-training. We use open an source implementation [166] of Faster-RCNN [156] to train the detector. For Flickr30K Entities, we use visual words that occur at least 100 times as detection labels, resulting in a total of 460 detection



A dog is laying in the grass with a Frisbee.

A bride and groom cutting a cake together.

A little girl holding a cat in her hand.

A woman sitting on a boat in the water.

A cat is standing on a sign that says "UNK".

A young boy with blond-hair and a blue shirt is eating a chocolate

A band is performing on a stage.

Two people are sitting on a boat in the water.

Figure 5.11: Generated captions and corresponding visual grounding regions on the standard image captioning task (Top: COCO, Bottom: Flickr30k). Different colors show a correspondence between the visual words and grounding regions. Grey regions are the proposals not selected in the caption. First 3 columns show success and last column shows failure cases (words are grounded in the wrong region).

labels. Since detection labels and visual words have a one-to-one mapping, we do not have fine-grained classes for the Flickr30K Entities dataset – the caption refinement process only determines the plurality of detection labels. For COCO, ground truth detection annotations are used to train the object detector.

Caption pre-processing. We truncate captions longer than 16 words for both COCO and Flickr30k Entities dataset. We then build a vocabulary of words that occur at least 5 times in the training set, resulting in 9,587 and 6,864 words for COCO and Flickr30k Entities, respectively.

Standard Image Captioning

For standard image captioning, we use splits from Karpathy *et al.* [19] on COCO/Flickr30k. We report results using the COCO captioning evaluation toolkit [5], which reports the widely used automatic evaluation metrics, BLEU [31], METEOR [150], CIDEr [33] and SPICE [151].

We present our methods trained on different object detectors: Flickr and COCO.

Table 5.3: Performance on the test portion of Karpathy *et al.* [19]'s splits on Flickr30k Entities dataset.

Method	BLEU1	BLEU4	METEOR	CIDEr	SPICE
Hard-Attention [18]	66.9	19.9	18.5	-	-
ATT-FCN [21]	64.7	23.0	18.9	-	-
Adaptive [155]	67.7	25.1	20.4	53.1	14.5
NBT	**69.0**	**27.1**	**21.7**	**57.5**	**15.6**
NBT^{oracle}	72.0	28.5	23.1	64.8	19.6

Table 5.4: Performance on the test portion of Karpathy *et al.* [19]'s splits on COCO dataset. $*$ directly optimizes the CIDEr Metric, † uses better image features, and are thus not directly comparable.

Method	BLEU1	BLEU4	METEOR	CIDEr	SPICE
Adaptive [155]	74.2	32.5	26.6	**108.5**	19.5
Att2in [157]	-	31.3	26.0	101.3	-
Up-Down [163]	74.5	33.4	26.1	105.4	19.2
Att2in* [157]	-	33.3	26.3	111.4	-
Up-Down† [163]	79.8	36.3	27.7	120.1	21.4
NBT	**75.5**	**34.7**	**27.1**	107.2	**20.1**
NBT^{oracle}	75.9	34.9	27.4	108.9	20.4

We compare our approach (referred to as NBT) to recently proposed Hard-Attention [18], ATT-FCN [21] and Adaptive [155] on Flickr30k, and Att2in [157], Up-Down [163] on COCO. Since object detectors have not yet achieved near-perfect accuracies on these datasets, we also report the performance of our model under an oracle setting, where the ground truth object region and category is also provided during test time. (referred to as NBT^{oracle}) This can be viewed as the upper bound of our method when we have perfect object detectors.

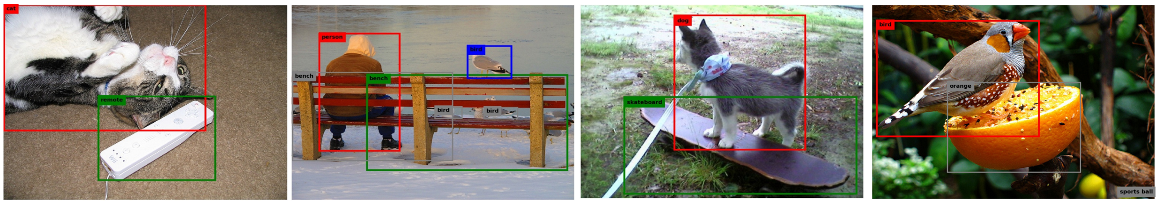
Table 5.3 shows results on the Flickr30k dataset. We see that our method achieves state of the art on all automatic evaluation metrics, outperforming the previous state-of-art model Adaptive [155] by 2.0 and 4.4 on BLEU4 and CIDEr. When using ground truth proposals, NBT^{oracle} significantly outperforms previous methods, improving 5.1 on SPICE, which implies that our method could further benefit from improved object detectors.

Table 5.4 shows results on the COCO dataset. Our method outperforms 4 out of 5 automatic evaluation metrics compared to the state of the art [157, 155, 163] without using better visual features or directly optimizing the CIDEr metric. Interestingly, the $\text{NBT}^{\text{oracle}}$ has little improvement over NBT. We suspect the reason is that explicit ground truth annotation is absent for visual words. Our model can be further improved with explicit co-reference supervision where the ground truth location annotation of the visual word is provided. Fig. 5.11 shows qualitative results on both datasets. We see that our model learns to correctly identify the visual word, and ground it in image regions even under weak supervision (COCO). Our model is also robust to erroneous detections and produces correct captions (3rd column).

Robust Image Captioning

To quantitatively evaluate image captioning models for novel scene compositions, we present a new split of the COCO dataset, called the robust-COCO split. This new split is created by re-organizing the train and val splits of the COCO dataset such that the distribution of co-occurring objects in train is different from test. We also present a new metric to evaluate grounding.

Robust split. To create the new split, we first identify entity words that belong to the 80 COCO object categories by following the same pre-processing procedure. For each image, we get a list of object categories that are mentioned in the caption. We then calculate the co-occurrence statistics for these 80 object categories. Starting from the least co-occurring category pairs, we greedily add them to the test set and ensure that for each category, at least half the instances of each category are in the train set. As a result, there are sufficient examples from each category in train, but at test time we see novel compositions (pairs) of categories. Remaining images are assigned to the training set. The final split has 110,234/3,915/9,138 images in



A cat laying on the floor next to a remote control. | A man sitting on a bench next to a bird. | A dog is standing on a skateboard in the grass. | A bird sitting on a branch in a tree.

Figure 5.12: Generated captions and corresponding visual grounding regions for the robust image captioning task. “cat-remote”, “man-bird”, “dog-skateboard” and “orange-bird” are co-occurring categories excluded in the training split. First 3 columns show success and last column shows failure case (orange was not mentioned).

Table 5.5: Performance on the test portion of the robust image captioning split on COCO dataset.

Method	BLEU4	METEOR	CIDEr	SPICE	Accuracy
Att2in [157]	31.5	24.6	90.6	17.7	39.0
Up-Down [163]	31.6	25.0	92.0	18.1	39.7
NBT	**31.7**	**25.2**	**94.1**	**18.3**	**42.4**
$\text{NBT}^{\text{oracle}}$	31.9	25.5	95.5	18.7	45.7

train/val/test respectively.

Evaluation metric. To evaluate visual grounding on the robust-COCO split, we want a metric that indicates whether or not a generated caption includes the new object combination. Common automatic evaluation metrics such as BLEU [31] and CIDEr [33] measure the overall sentence fluency. We also measure whether the generated caption contains the novel co-occurring categories that exist in the ground truth caption. A generated caption is deemed 100% accurate if it contains at least one mention of the *compositionally novel* category-pairs in any ground truth annotation that describe the image.

Results and analysis. We compare our method with state of the art Att2in [157] and Up-Down [163]. These are implemented using the open source implementation from [167] that can replicate results on Karpathy’s split. We follow the experimental setting from [157] and train the model using the robust-COCO train set. Table 5.5 shows the results on the robust-COCO split. As we can see, all models perform worse

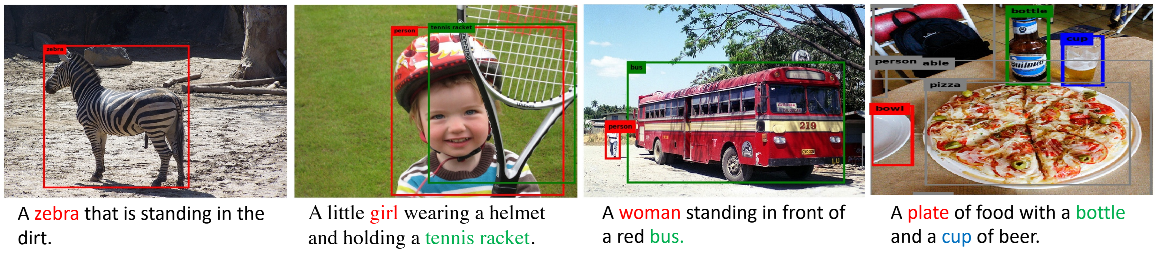


Figure 5.13: Generated captions and corresponding visual grounding regions for the novel object captioning task. “zebra”, “tennis racket”, “bus” and “pizza” are categories excluded in the training split. First 3 columns show success and last column shows a failure case.

on the robust-COCO split than the Karpathy’s split by 2$\sim$3 points in general. Our method outperforms the previous state of the art methods on all metrics, outperforming Up-Down [163] by 2.7 on the proposed metric. The oracle setting (NBT^{oracle}) has consistent improvements on all metrics, improving 3.3 on the proposed metric.

Fig. 5.12 shows qualitative results on the robust image captioning task. Our model successfully produces a caption with novel compositions, such as “cat-remote”, “man-bird” and “dog-skateboard” to describe the image. The last column shows failure cases where our model didn’t select “orange” in the caption. We can force our model to produce a caption containing “orange” and “bird” using constrained beam search [168], further illustrated in Sec. 5.2.3.

Novel Object Captioning

Since our model directly fills the “slotted” caption template with the concept, it can seamlessly generate descriptions for out-of-domain images. We replicated an existing experimental design [169] on COCO which excludes all the image-sentence pairs that contain at least one of eight objects in COCO. The excluded objects are ‘bottle’, “bus”, “couch”, “microwave”, “pizza”, “racket”, “suitcase” and “zebra”. We follow the same splits for training, validation, and testing as in prior work [169]. We use Faster R-CNN in conjunction with ResNet-101 which is pre-trained on COCO train split as the detection model. Note that we do not pre-train the language model using

Table 5.6: Evaluation of captions generated using the proposed method. G means greedy decoding, and T1−2 means using constrained beam search [168] with 1−2 top detected concepts. $*$ is the result using VGG-16 [172] and † is the result using ResNet-101.

	Out-of-Domain Test Data				In-Domain Test Data		
Method	F1	SPICE	METEOR	CIDEr	SPICE	METEOR	CIDER
DCC [169]	39.8	13.4	21.0	59.1	15.9	23.0	77.2
NOC [170]	49.1	-	21.4	-	-	-	-
C-LSTM [171]	55.7	-	23.0	-	-	-	-
Base+T4 [168]	54.0	15.9	23.3	77.9	18.0	24.5	86.3
NBT*+G	48.5	15.7	22.8	77.0	17.5	24.3	87.4
NBT†+G	53.2	16.6	23.9	84.0	**18.4**	25.3	94.0
NBT†+T1	57.3	16.7	23.9	85.7	**18.4**	**25.5**	**95.2**
NBT†+T2	**70.3**	**17.4**	**24.1**	**86.0**	18.0	25.0	92.1

COCO captions as in [169, 170, 171], and simply replace the novel object's word embedding with an existing one which belongs to the same super-category in COCO (e.g., bus ← car).

Following [168], the test set is split into in-domain and out-of-domain subsets. We report F1 as in [169], which checks if the specific excluded object is mentioned in the generated caption. To evaluate the quality of the generated caption, we use SPICE, METEOR and CIDEr metrics and the scores on out-of-domain test data are macro-averaged across eight excluded categories. For consistency with previous work [163], the inverse document frequency statistics used by CIDEr are determined across the entire test set.

As illustrated in Table 5.6, simply using greedy decoding, our model (NBT*+G) can successfully caption novel concepts with minimum changes to the model. When using ResNet-101 and constrained beam search [168], our model significantly outperforms prior works under F1 scores, SPICE, METEOR, and CIDEr, across both out-of-domain and in-domain test data. Specifically, NBT†+T2 outperforms the previous state-of-art model C-LSTM by 14.6% on average F1 scores. From the category F1 scores, we can see that our model is less likely to select small objects, e.g. "bottle", "racket" when only using the greedy decoding. Since the visual words are grounded at the object-level, by using [168], our model was able to significantly boost the captioning performance on out-of-domain images. Fig. 5.13 shows qualitative novel object captioning results.

5.2.4 Discussion

In this section, we introduce Neural Baby Talk, a novel image captioning framework that produces natural language explicitly grounded in entities object detectors find in images. Our approach is a two-stage approach that first generates a hybrid template that contains a mix of words from a text vocabulary as well as slots corresponding to image regions. It then fills the slots based on categories recognized by object detectors in the image regions. We also introduce a robust image captioning split by re-organizing the train and val splits of the COCO dataset. Experimental results on standard, robust, and novel object image captioning tasks validate the effectiveness of our proposed approach.

CHAPTER 6
VISUAL DIALOG

In this chapter, our goal will be to study the novel training paradigms for generating perceptual grounded questions and responses in the context of the visual dialog. Apart from the model architectures which are explored in the previous section, training paradigms are also important to learn better grounding and improve multi-modal AI capabilities. Specifically, we first study the standard training paradigm for neural dialog models – maximum likelihood estimation (MLE) of a 'ground-truth' human response. Across a variety of domains, we find out a recurring problem with MLE trained neural dialog models is that they tend to produce 'safe' generic responses. Inspired by the success of adversarial training, we introduce a discriminant perceptual loss to transfer knowledge from the discriminative model to the generative model and achieves state of the art performance on visual dialog tasks.

In the second section, we study the training paradigms of goal-oriented dialog generation in the context of an image guessing game. A popular approach to these tasks has been to observe humans engaging in dialogs and let the agent mimic human dialogs to generate human interpretable language (*i.e.*, meaningful English, not gibberish). However, this requires to collect new human dialogs for each new task, which is laborious and costly. A pragmatic alternative is to use goal completion as supervision signals (Discriminant perceptual loss can be viewed as a special case) to adapt agents to new tasks. To solve this task, we propose a novel model that decomposes generating question intent from the words used to express that intent. It does this by introducing a discrete latent representation that is the only input to the language decoder. We also develop an incremental learning curriculum that first learns "how to speak" by pretraining with a conditional variational auto-encoders

(CVAE), and then learns "what to speak" by finetuning with task-specific rewards with discrete latent space. To verify the effectiveness of our approach, we pair our agent with human and find our agent learns a strategy for this task that is amenable to human-AI collaboration. This is in contrast to prior work [110] that showed that improvements captured by task-trained models for similar image-retrieval tasks did not transfer when paired with human partners.

6.1 Best of Both Worlds: Transferring Knowledge from Discriminative Learning to a Generative Visual Dialog Model

The standard training paradigm for neural dialog models is maximum likelihood estimation (MLE) or equivalently, minimizing the cross-entropy (under the model) of a 'ground-truth' human response. Across a variety of domains, a recurring problem with MLE trained neural dialog models is that they tend to produce 'safe', generic responses, such as *'Not sure'* or *'I don't know'* in text-only dialog [28], and *'I can't see'* or *'I can't tell'* in visual dialog [4, 26].

One reason for this emergent behavior is that the space of possible next utterances in a dialog is *highly* multi-modal (there are many possible paths a dialog may take in the future). In the face of such highly multi-modal output distributions, models 'game' MLE by latching on to the head of the distribution or the frequent responses, which by nature tend to be generic and widely applicable. Such safe generic responses break the flow of a dialog and tend to disengage the human conversing with the agent, ultimately rendering the agent useless. It is clear that novel training paradigms are needed; that is the focus of this paper.

One promising alternative to MLE training proposed by recent work [29, 30] is *sequence-level training* of neural sequence models, specifically, using reinforcement learning to optimize task-specific sequence metrics such as BLEU [31], ROUGE [32], CIDEr [33]. Unfortunately, in the case of dialog, *all existing* automatic metrics correlate poorly with human judgment [34], which renders this alternative infeasible for dialog models.

In this section, inspired by the success of adversarial training [35], we propose to train a *generative* visual dialog model (G) to produce sequences that score highly under a *discriminative* visual dialog model (D). A discriminative dialog model receives as input a candidate list of possible responses and learns to sort this list from the training dataset. The generative dialog model (G) aims to produce a sequence that D will rank the highest in the list.

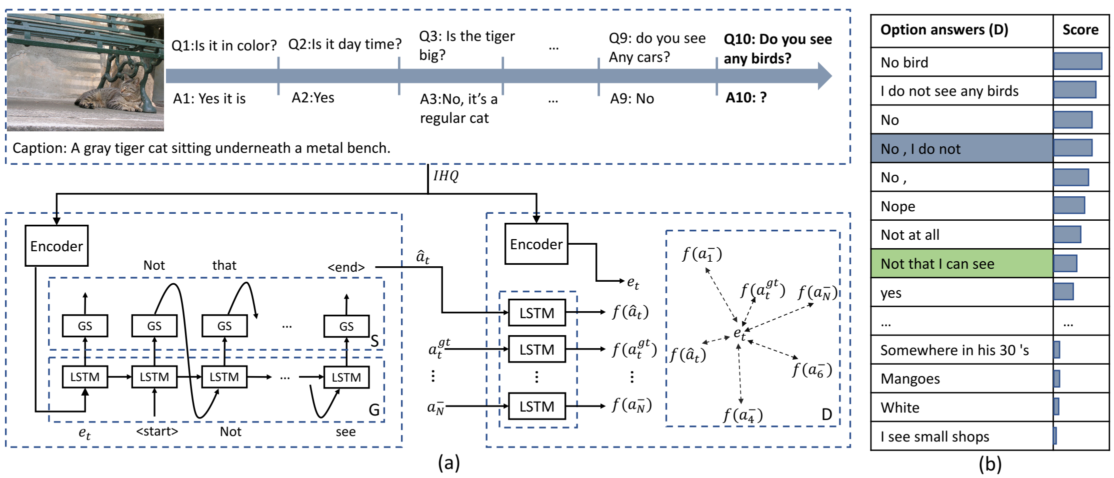


Figure 6.1: (a): Model architecture (b): Given the image, history, and question, D's score for different candidate answers. Note that the multiple plausible responses all score high. The candidate in the blue box is the true response and in green is the response generated by G.

Note that while our proposed approach is inspired by adversarial training, there are a number of subtle but crucial differences over generative adversarial networks (GANs). Unlike traditional GANs, one novelty in our setup is that our discriminator has *access to more information* than G – specifically, D receives a list of candidate responses and explicitly learns to reason about similarities and differences across candidates. In this process, D learns a task-dependent perceptual similarity [173, 174, 175] and learns to recognize multiple correct responses in the feature space. For example, as shown in Fig. 6.1 (b), given the image, dialog history, and question *'Do* *you see any bird?'*, besides the ground-truth answer *'No, I do not'*, D can also assign high scores to other options that are valid responses to the question, including the one generated by G: *'Not that I can see'*. In that sense, our proposed approach may be viewed as an instance of 'knowledge transfer' [176, 177] from D to G. We employ a metric-learning loss function and a self-attention answer encoding mechanism for D that makes it particularly conducive to this knowledge transfer by encouraging perceptually meaningful similarities to emerge. This is especially fruitful since prior work has demonstrated that discriminative dialog models significantly outperform their generative counterparts, but are not as useful since they necessarily need a list of candidate responses to rank, which is only available in a dialog dataset, not in real conversations with a user. In that context, our work aims to achieve the best of both worlds – the practical usefulness of G and the strong performance of D – via this knowledge transfer.

Our primary technical contribution is an end-to-end trainable generative visual dialog model, where the generator receives gradients from the discriminator loss of the sequence sampled from G. Note that this is challenging because the output of G is a sequence of discrete symbols, which naïvely is not amenable to gradient-based training. We propose to leverage the recently proposed Gumbel-Softmax (GS) approximation to the discrete distribution [178, 179] – specifically, a Recurrent Neural Network (RNN) augmented with a sequence of GS samplers, which when coupled with the straight-through gradient estimator [180, 178] enables end-to-end differentiability.

6.1.1 Approach: Backprop Through Discriminative Losses for Generative Training

In this section, we describe our approach to transfer knowledge from a discriminative visual dialog model (D) to generative visual dialog model (G). Fig. 6.1 (a) shows the overview of our approach. Given the input image $\boldsymbol{I}$, dialog history $\boldsymbol{H}$, and question $\boldsymbol{q}_t$, the encoder converts the inputs into a joint representation $\boldsymbol{e}_t$. The generator G

takes $\boldsymbol{e}_t$ as input, and produces a distribution over answer sequences via a recurrent neural network (specifically an LSTM). At each word in the answer sequence, we use a Gumbel-Softmax sampler S to sample the answer token from that distribution. The discriminator D in it's standard form takes $\boldsymbol{e}_t$, ground-truth answer $\boldsymbol{a}_t^{gt}$ and $N-1$ "negative" answers $\{\boldsymbol{a}_{t,i}^{-}\}_{i=1}^{N-1}$ as input, and learns an embedding space such that similarity$(\boldsymbol{e}_t, f(\boldsymbol{a}_t^{gt})) >$ similarity$(\boldsymbol{e}_t, f(\boldsymbol{a}_{t,\cdot}^{-}))$, where $f(\cdot)$ is the embedding function. When we enable the communication between D and G, we feed the sampled answer $\hat{\boldsymbol{a}}_t$ into discriminator, and optimize the generator G to produce samples that get higher scores in D's metric space. We now describe each component of our approach in detail.

History-Conditioned Image Attentive Encoder (HCIAE)

An important characteristic in dialogs is the use of co-reference to avoid repeating entities that can be contextually resolved. In fact, in the VisDial dataset [4] nearly all (98%) dialogs involve at least one pronoun. This means that for a model to correctly answer a question, it would require a reliable mechanism for co-reference resolution.

A common approach is to use an encoder architecture with an attention mechanism that implicitly performs co-reference resolution by identifying the portion of the dialog history that can help in answering the current question [4, 181, 182]. However, previous encoders used for this task use a holistic representation for the image without an attention mechanism. Intuitively, the answer to the question is likely to be localized to regions in the image that are consistent with attended history.

With this motivation, we propose a novel encoder architecture (called HCIAE) shown in Fig. 6.2. Our encoder first uses the current question to attend to the exchanges in the history, and then use the question and attended history to attend to the image, so as to obtain the final encoding.

Specifically, we use the spatial image features $\boldsymbol{V} \in \mathcal{R}^{d\times k}$ from a convolution layer

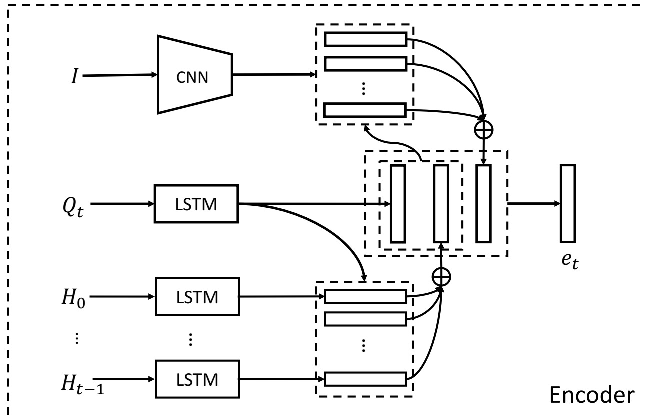


Figure 6.2: Structure of the proposed encoder.

of a CNN. $\boldsymbol{q}_t$ is encoded with an LSTM to get a vector $\boldsymbol{m}_t^q \in \mathcal{R}^d$. Simultaneously, each previous round of history $(H_0, \ldots, H_{t-1})$ is encoded separately with another LSTM as $\boldsymbol{M}_t^h \in \mathcal{R}^{d \times t}$. Conditioned on the question embedding, the model attends to the history. The attended representation of the history and the question embedding are concatenated, and used as input to attend to the image:

$$\begin{aligned} \boldsymbol{z}_t^h &= \boldsymbol{w}_a^T \tanh(\boldsymbol{W}_h \boldsymbol{M}_t^h + (\boldsymbol{W}_q \boldsymbol{m}_t^q)\mathbb{1}^T) && (6.1) \\ \boldsymbol{\alpha}_t^h &= \mathrm{softmax}(\boldsymbol{z}_t^h) && (6.2) \end{aligned}$$

where $\mathbb{1} \in \mathcal{R}^t$ is a vector with all elements set to 1. $\boldsymbol{W}_h, \boldsymbol{W}_q \in \mathcal{R}^{t \times d}$ and $\boldsymbol{w}_a \in \mathcal{R}^k$ are parameters to be learned. $\boldsymbol{\alpha} \in \mathcal{R}^k$ is the attention weight over history. The attended history feature $\hat{\boldsymbol{m}}_t^h$ is a convex combination of columns of $\boldsymbol{M}_t$, weighted appropriately by the elements of α_t^h. We further concatenate $\boldsymbol{m}_t^q$ and $\hat{\boldsymbol{m}}_t^h$ as the query vector and get the attended image feature $\hat{\boldsymbol{v}}_t$ in the similar manner. Subsequently, all three components are used to obtain the final embedding $\boldsymbol{e}_t$:

$$\boldsymbol{e}_t = \tanh(\boldsymbol{W}_e[\boldsymbol{m}_t^q, \hat{\boldsymbol{m}}_t^h, \hat{\boldsymbol{v}}_t]) \tag{6.3}$$

where $\boldsymbol{W}_e \in \mathcal{R}^{d\times 3d}$ is weight parameters and $[\cdot]$ is the concatenation operation.

Discriminator Loss

Discriminative visual dialog models produce a distribution over the candidate answer list $\mathcal{A}_t$ and maximize the log-likelihood of the correct option $\boldsymbol{a}_t^{gt}$. The loss function for D needs to be conducive for knowledge transfer. In particular, it needs to encourage perceptually meaningful similarities. Therefore, we use a metric-learning multi-class N-pair loss [183] defined as:

$$\mathcal{L}_D = \mathcal{L}_{n-pair}\left(\{\boldsymbol{e}_t, \boldsymbol{a}_t^{gt}, \{\boldsymbol{a}_{t,i}^{-}\}_{i=1}^{N-1}\}, f\right) = \overbrace{\log\left(1 + \sum_{i=1}^{N} \exp\left(\underbrace{\boldsymbol{e}_t^{\top} f(\boldsymbol{a}_{t,i}^{-}) - \boldsymbol{e}_t^{\top} f(\boldsymbol{a}_t^{gt})}_{\text{score margin}}\right)\right)}^{\text{logistic loss}} \tag{6.4}$$

where f is an attention based LSTM encoder for the answer. This attention can help the discriminator better deal with paraphrases across answers. The attention weight is learnt through a 1-layer MLP over LSTM output at each time step. The N-pair loss objective encourages learning a space in which the ground truth answer is scored higher than other options, and at the same time, encourages options similar to ground truth answers to score better than dissimilar ones. This means that, unlike the multiclass logistic loss, the options that are correct but different from the correct option may not be overly penalized, and thus can be useful in providing a reliable signal to the generator. See Fig. 6.1 for an example. An added benefit of the n-pair loss is its computational efficiency: Batches can be constructed such that incorrect options for each example in the batch can be assigned on-the-fly. This leads to repeated use of the same options across the batch, and hence lesser computation.

Discriminant Perceptual Loss and Knowledge Transfer from D to G

At a high-level, our approach for transferring knowledge from D to G is as follows: G repeatedly queries D with answers $\hat{\boldsymbol{a}}_t$ that it generates for an input embedding $\boldsymbol{e}_t$ to get feedback and update itself. In each such update, G's goal is to update its parameters to try and have $\hat{\boldsymbol{a}}_t$ score higher than the correct answer, $\boldsymbol{a}_t^{gt}$, under D's learned embedding and scoring function. Formally, the perceptual loss that G aims to optimize is given by:

$$\mathcal{L}_G = \mathcal{L}_{1-pair}\Big(\{\boldsymbol{e}_t, \hat{\boldsymbol{a}}_t, \boldsymbol{a}_t^{gt}\}, f\Big) = \log\Bigg(1 + \exp\Big(\boldsymbol{e}_t^\top f(\boldsymbol{a}_t^{gt}) - \boldsymbol{e}_t^\top f(\hat{\boldsymbol{a}}_t)\Big)\Bigg) \tag{6.5}$$

where f is the embedding function learned by the discriminator as in (6.4). Intuitively, updating generator parameters to minimize $\mathcal{L}_G$ can be interpreted as learning to produce an answer sequence $\hat{\boldsymbol{a}}_t$ that 'fools' the discriminator into believing that this answer should score higher than the human response $\boldsymbol{a}_t^{gt}$ under the discriminator's learned embedding $f(\cdot)$ and scoring function.

While it is straightforward to sample an answer $\hat{\boldsymbol{a}}_t$ from the generator and perform a forward pass through the discriminator, naïvely, it is not possible to backpropagate the gradients to the generator parameters since sampling discrete symbols results in zero gradients w.r.t. the generator parameters. To overcome this, we leverage the recently introduced continuous relaxation of the categorical distribution – the Gumbel-softmax distribution or the Concrete distribution [178, 179].

At an intuitive level, the Gumbel-Softmax (GS) approximation uses the so called 'Gumbel-Max trick' to reparametrize sampling from a categorical distribution and replaces argmax with softmax to obtain a continuous relaxation of the discrete random variable. Formally, let $\boldsymbol{x}$ denote a K-ary categorical random variable with parameters denoted by $(p_1, \ldots p_K)$, or $\boldsymbol{x} \sim Cat(\boldsymbol{p})$. Let $\Big(g_i\Big)_1^K$ denote K IID samples from the standard Gumbel distribution, $g_i \sim F(g) = e^{-e^{-g}}$. Now, a sample from the Concrete distribution can be produced via the following transformation:

$$y_i = \frac{e^{(\log p_i + g_i)/\tau}}{\sum_{j=1}^{K} e^{(\log p_j + g_j)/\tau}} \qquad \forall i \in \{1, \ldots, K\} \tag{6.6}$$

where τ is a temperature parameter that control how close samples $\boldsymbol{y}$ from this Concrete distribution approximate the one-hot encoding of the categorical variable $\boldsymbol{x}$.

As illustrated in Fig. 6.1, we augment the LSTM in G with a sequence of GS samplers. Specifically, at each position in the answer sequence, we use a GS sampler to sample an answer token from that conditional distribution. When coupled with the straight-through gradient estimator [180, 178] this enables end-to-end differentiability. Specifically, during the forward pass we discretize the GS samples into discrete samples, and in the backward pass use the continuous relaxation to compute gradients. In our experiment, we set the temperature parameter consistently to 0.5, and do not perform any temperature annealing.

6.1.2 Experiments

Dataset and Setup. We evaluate our proposed approach on the VisDial dataset [4], which was collected by Das *et al.* by pairing two subjects on Amazon Mechanical Turk to chat about an image. One person was assigned the role of a 'questioner' and the other of 'answerer'. One worker (the questioner) sees only a single line of text describing an image (caption from COCO [5]); the image remains hidden to the questioner. Their task is to ask questions about this hidden image to "imagine the scene better". The second worker (the answerer) sees the image and caption and answers the questions. The two workers take turns asking and answering questions for 10 rounds. We perform experiments on VisDial v0.9 (the latest available release) containing 83k dialogs on COCO-train and 40k on COCO-val images, for a total of

1.2M dialog question-answer pairs. We split the 83k into 82k for `train`, 1k for `val`, and use the 40k as `test`, in a manner consistent with [4]. The caption is considered to be the first round in the dialog history.

Evaluation Protocol. Following the evaluation protocol established in [4], we use a retrieval setting to evaluate the responses at each round in the dialog. Specifically, every question in VisDial is coupled with a list of 100 candidate answer options, which the models are asked to sort for evaluation purposes. D uses its score to rank these answer options, and G use the log-likelihood of these options for ranking. Models are evaluated on standard retrieval metrics – (1) mean rank, (2) recall @k, and (3) mean reciprocal rank (MRR) – of the human response in the returned sorted list.

Pre-processing. We truncate captions/questions/answers longer than 24/16/8 words respectively. We then build a vocabulary of words that occur at least 5 times in `train`, resulting in 8964 words.

Training Details In our experiments, all 3 LSTMs are single layer with 512d hidden state. We use the Adam optimizer with a base learning rate of 4e-4. We pre-train G using standard MLE for 20 epochs, and D with supervised training based on Eq (6.4) for 30 epochs. Following [183], we regularize the L^2 norm of the embedding vectors to be small. Subsequently, we train G with $\mathcal{L}_G + \alpha\mathcal{L}_{MLE}$, which is a combination of discriminative perceptual loss and MLE loss. We set α to be 0.5. We found that including $\mathcal{L}_{MLE}$ (with teacher-forcing) is important for encouraging G to generate grammatically correct responses.

Results and Analysis

Baselines. We compare our proposed techniques to the current state-of-art generative and discriminative models developed in [4]. Specifically, [4] introduced 3 encoding architectures – Late Fusion (**LF**), Hierarchical Recurrent Encoder (**HRE**), Memory Network (**MN**) – each trained with a generative (**-G**) and discriminative (**-D**) decoder. We compare to all 6 models.

Our approaches. We present a few variants of our approach to systematically study the individual contributions of our training procedure, novel encoder (HCIAE), self-attentive answer encoding (ATT), and metric-loss (NP).

- **HCIAE-G-MLE** is a generative model with our proposed encoder trained under the MLE objective. Comparing this variant to the generative baselines from [4] establishes the improvement due to our encoder (HCIAE).

- **HCIAE-G-DIS** is a generative model with our proposed encoder trained under the mixed MLE and discriminator loss (knowledge transfer). This forms our best generative model. Comparing this model to **HCIAE-G-MLE** establishes the improvement due to our discriminative training.

- **HCIAE-D-MLE** is a discriminative model with our proposed encoder, trained under the standard discriminative cross-entropy loss. The answer candidates are encoded using an LSTM (no attention). Comparing this variant to the discriminative baselines from [4] establishes the improvement due to our encoder (HCIAE) in the discriminative setting.

- **HCIAE-D-NP** is a discriminative model with our proposed encoder, trained under the n-pair discriminative loss (as described in Section 6.1.1). The answer candidates are encoded using an LSTM (no attention). Comparing this variant to **HCIAE-D-MLE** establishes the improvement due to the n-pair loss.

- **HCIAE-D-NP-ATT** is a discriminative model with our proposed encoder, trained under the n-pair discriminative loss (as described in Section 6.1.1), and using the self-attentive answer encoding. Comparing this variant to **HCIAE-D-NP** establishes the improvement due to the self-attention mechanism while encoding the answers.

Results. Tables 6.1 present results for all our models and baselines in generative and discriminative settings. The key observations are:

1. **Main Results for HCIAE-G-DIS:** Our final generative model with all 'bells and whistles', **HCIAE-G-DIS**, uniformly performs the best under all the metrics, outperforming the previous state-of-art model **MN-G** by 2.43% on R@5. This shows the importance of the knowledge transfer from the discriminator and the benefit from our encoder architecture.

2. **Knowledge transfer *vs.* encoder for G:** To understand the relative importance of the proposed history conditioned image attentive encoder (HCIAE) and the knowledge transfer, we compared the performance of **HCIAE-G-DIS** with **HCIAE-G-MLE**, which uses our proposed encoder but without any feedback from the discriminator. This comparison highlights two points: first, **HCIAE-G-MLE** improves R@5 by 0.7% over the current state-of-art method (**MN-D**) confirming the benefits of our encoder. Secondly, and importantly, its performance is lower than **HCIAE-G-DIS** by 1.7% on R@5, confirming that the modifications to encoder alone will not be sufficient to gain improvements in answer generation; knowledge transfer from D greatly improves G.

3. **Metric loss** *vs.* **self-attentive answer encoding**: In the purely discriminative setting, our final discriminative model (**HCIAE-D-NP-ATT**) also beats the performance of the corresponding state-of-art models [4] by 2.53% on R@5. The n-pair loss used in the discriminator is not only helpful for knowledge transfer but it also improves the performance of the discriminator by 0.85% on R@5 (compare **HCIAE-D-NP** to **HCIAE-D-MLE**). The improvements obtained by using the answer attention mechanism leads to an additional, albeit small, gains of 0.4% on R@5 to the discriminator performance (compare **HCIAE-D-NP** to **HCIAE-D-NP-ATT**).

Table 6.1: Results (generative) on VisDial dataset.

Model	MRR	R@1	R@5	R@10	Mean
LF-G [4]	0.5199	41.83	61.78	67.59	17.07
HREA-G [4]	0.5242	42.28	62.33	68.17	16.79
MN-G [4]	0.5259	42.29	62.85	68.88	17.06
HCIAE-G-MLE	0.5386	44.06	63.55	69.24	16.01
HCIAE-G-DIS	**0.5467**	**44.35**	**65.28**	**71.55**	**14.23**

Table 6.2: Results (discriminative) on VisDial dataset.

Model	MRR	R@1	R@5	R@10	Mean
LF-D [4]	0.5807	43.82	74.68	84.07	5.78
HREA-D [4]	0.5868	44.82	74.81	84.36	5.66
MN-D [4]	0.5965	45.55	76.22	85.37	5.46
HCIAE-D-MLE	0.6140	47.73	77.50	86.35	5.15
HCIAE-D-NP	0.6182	47.98	78.35	87.16	4.92
HCIAE-D-NP-ATT	**0.6222**	**48.48**	**78.75**	**87.59**	**4.81**

Does updating discriminator help?

Recall that our model training happens as follows: we independently train the generative model **HCIAE-G-MLE** and the discriminative model **HCIAE-D-NP-ATT**. With **HCIAE-G-MLE** as the initialization, the generative model is updated based on the feedback from **HCIAE-D-NP-ATT** and this results in our final **HCIAE-G-DIS**.

We performed two further experiments to answer the following questions:

- What happens if we continue training **HCIAE-D-NP-ATT** in an adversarial setting? In particular, we continue training by maximizing the score of the ground truth answer $\boldsymbol{a}_t^{gt}$ and minimizing the score of the generated answer $\hat{\boldsymbol{a}}_t$, effectively setting up an adversarial training regime $\mathcal{L}_D = -\mathcal{L}_G$. The resulting discriminator **HCIAE-GAN1** has significant drop in performance, as can be seen in Table. 6.3 (45.78% R@5). This is perhaps expected because **HCIAE-GAN1** updates its parameters based on only two answers, the ground truth and the generated sample (which is likely to be similar to ground truth). This wrecks the structure that **HCIAE-D-NP-ATT** had previously learned by leveraging additional incorrect options.

- What happens if we continue structure-preserving training of **HCIAE-D-NP-ATT**? In addition to providing **HCIAE-D-NP-ATT** samples from G as fake answers, we also include incorrect options as negative answers so that the structure learned by the discriminator is preserved. **HCIAE-D-NP-ATT** continues to train under loss $\mathcal{L}_D$. In this case (**HCIAE-GAN2** in Table. 6.3), we find that there is a small improvement in the performance of G. The additional computational overhead to training the discriminator supersedes the performance improvement. Also note that **HCIAE-D-NP-ATT** itself gets worse at the dialog task.

One might wonder, why not train a GAN for visual dialog? Formulating the task in a GAN setting would involve G and D training in tandem with D providing feedback as to whether a response that G generates is real or fake. We found this to be a particularly unstable setting, for two main reasons: First, consider the case when the ground truth answer and the generated answers are the same. This happens for answers that are typically short or 'cryptic' (e.g. *'yes'*). In this case, D can not train itself or provide feedback, as the answer is labeled both positive and negative. Second, in cases where the ground truth answer is descriptive but the generator provides a short answer, D can quickly become powerful enough to discard generated samples as fake. In this case, D is not able to provide any information to G to get better at the task. Our experience suggests that the discriminator, if one were to consider a 'GANs for visual dialog' setting, can not merely be focused on differentiating fake from real. It needs to be able to score similarity between the ground truth and other answers. Such a scoring mechanism provides a more reliable feedback to G. In fact, as we show in the previous two results, a pre-trained D that captures this structure

Table 6.3: Adversarial training results on VisDial dataset.

	Discriminative					Generative				
Model	MRR	R@1	R@5	R@10	Mean	MRR	R@1	R@5	R@10	Mean
HCIAE-D-NP-ATT	0.6222	48.48	78.75	87.59	4.81	-	-	-	-	-
HCIAE-G-DIS	-	-	-	-	-	0.5467	44.35	65.28	71.55	14.23
HCIAE-GAN1	0.2177	8.82	32.97	52.14	18.53	0.5298	43.12	62.74	68.58	16.25
HCIAE-GAN2	0.6050	46.20	77.92	87.20	4.97	0.5459	44.33	65.05	71.40	14.34

is the key ingredient in sharing knowledge with G. The adversarial training of D is not central.

Qualitative Comparison

In Fig 6.3 we present a couple of qualitative examples that compares the responses generated by G-MLE and G-DIS. G-MLE predominantly produces 'safe' and less informative answers, such as *'Yes'* and or *'I can't tell'*. In contrast, our proposed model G-DIS does so less frequently, and often generates more diverse yet informative responses (see 2nd example in particular).

Figure 6.3: Qualitative comparison. "Ours" are samples from G-DIS model with different gumbel noise z. Images from the COCO dataset

6.1.3 Discussion

Generative models for (visual) dialog are typically trained with an MLE objective. As a result, they tend to latch on to safe and generic responses. Discriminative (or retrieval) models on the other hand have been shown to significantly outperform their generative counterparts. However, discriminative models can not be deployed as dialog agents with a real user where canned candidate responses are not available. In this work, we propose transferring knowledge from a powerful discriminative visual dialog model to a generative model. We leverage the Gumbel-Softmax (GS) approximation to the discrete distribution – specifically, an RNN augmented with a sequence of GS samplers, which coupled with the straight-through gradient estimator enables end-to-end differentiability. We also propose a novel visual dialog encoder that reasons about image-attention informed by the history of the dialog; and employ a metric learning loss along with a self-attentive answer encoding to enable the discriminator to learn meaningful structure in dialog responses. The result is a generative visual dialog model that significantly outperforms state-of-the-art.

6.2 Dialog without Dialog: Learning Image-Discriminative Dialog Policies from Single-Shot Question Answering Data

A popular approach to these tasks has been to observe humans engaging in dialogs like the ones we would like to automate and then train agents to mimic these human dialogs [26, 36]. Mimicking human dialogs allows agents to generate interpretable language (*i.e.*, meaningful English, not gibberish). However, these models are typically fragile and generalize poorly to new tasks. As such, each new task requires collecting new human dialogs, which is a laborious and costly process often requiring many iterations before high quality dialogs are elicited.

A promising pragmatic alternative is to use goal completion as a supervisory signal to adapt agents to new tasks. That is, after training dialog agents to mimic human dialogs for one task, fine-tune them on a new task by simply rewarding the agents for solving the task regardless of the dialog's content. This approach can indeed improve task performance, but language quality suffers even for similar tasks. It tends to drifts from human language, becoming ungrammatical and loosing human interpretable semantics – sometimes even turning into unintelligible code. Though bots might understand it, humans cannot, so humans will not be able to use it either. Both effects have been observed in prior dialog work [26, 36].

In this section, we consider an image guessing game as a test-bed for balancing task performance and language drift. Our Dialog without Dialog (DwD) task requires agents to generalize from single round visual question generation with full supervision to a multi-round dialog based image guessing game without direct language supervision. Specifically, as illustrated in Fig. 1.4 (top), agents are trained to mimic human-generated, visually-grounded questions that when answered can discern which of two images is secretly indicated to the answerer. We then develop techniques to transfer these agents to a multi-round, QA-based image guessing game over pools of various sizes, difficulties, and even image domains.

To solve this task we propose a an architecture for the questioner agent, Q-bot, that decomposes generating question intent from the words used to express that intent. It does this by introducing a discrete latent representation that is the only input to the language decoder. We pair this with an incremental learning curriculum that adapts the single round Q-bot to dialog in stages – first learning simply to follow the dialog and then to influence question intention.

We show that our model can be fine-tuned to increase task performance while maintaining human interpretable language. To measure interpretability we take a two pronged approach, getting humans to evaluate the fluency and relevance of questions generated by our model on one hand and using automatic measures of fluency, relevance, and diversity to help scale our analysis.

6.2.1 Dialog-based Image Guessing Game

Our objective is to examine how to transfer grounded language models from one task to another by training agents only to maximize task success. We consider an image-guessing communication game as the context for our experiments. In this section, we introduce this game and a model for this multi-round dialog task. In the following sections, we will discuss how to train such a model using non-dialog data.

Game Definition

We consider a conceptually simple image guessing game demonstrated in Fig. 1.4. In each episode, one agent (A-bot in red) secretly selects an image y (starred) from an image pool (in the dashed green box). The other agent (Q-bot in green) must identify this image by executing a multi-round question-answer based dialog with A-bot. To succeed, Q-bot will need to understand the image pool, generate discriminative questions, and interpret the answers A-bot provides to identify A-bot's selected image.

At a high-level functional view, we can consider the dialog as following a simple structure. At each round r, Q-bot observes the pool $\mathcal{I} = \{I_1, \dots, I_P\}$ and dialog history $q_0, a_0, \dots q_{r-1}, a_{r-1}$ and produces a question

$$q_r = \texttt{QBot.Ask}(\mathcal{I}, q_0, a_0, \dots q_{r-1}, a_{r-1}). \tag{6.7}$$

Given this question q_r, A-bot provides an answer a_r based on its selected image I_y:

$$a_r = \texttt{ABot.Answer}(I_y, q_r) \tag{6.8}$$

Once Q-bot receives the answer from A-bot, it makes a prediction $\hat{y}_{r+1}$ about the target image:

$$\hat{y}_r = \texttt{QBot.Predict}(\mathcal{I}, q_0, a_0, \dots, q_r, a_r) \tag{6.9}$$

where the task performance of Q-bot can be calculated by comparing $\hat{y}_r$ and y.

Comparison to GuessWhich. [26] presented a similar dialog-based guessing game called GuessWhich. In GuessWhich, Q-bot initially observes a caption describing A-bot's selected image and must predict the selected image's features to retrieve it from a large, fixed pool of images. The inclusion of the caption leaves little room for the dialog to add information [184] and the fixed-pool would not enable us to inspect how Q-bot's behavior generalizes to different pools. As described above, we drop both these assumptions to enable our analysis.

Modelling A-bot

In this work, we focus primarily on Q-bot agent rather than A-bot. We set A-bot to be a standard visual question answering agent, specifically the Bottom-up Top-down [163] model; however, we do make one modification. Q-bot may generate questions that are not well grounded in A-bot's selected image (though they may be grounded in other pool images) – e.g. asking about a surfer when none exists. To enable A-bot to respond appropriately, we augment A-bot's answer space with a `Not Relevant` token. We augment every image with an additional, randomly-sampled question and set `Not Relevant` as its target answer. A-bot is trained independently from Q-bot on the VQAv2 dataset and then frozen.

Modelling Q-bot

We conceptualize Q-bot as having three major tasks: encoding the state of the game to decide what to ask about, actually formulating this intent in language, and making predictions about A-bot's selection. Respectively, these correspond to planner, speaker, and predictor modules. As we focus on language transfer across tasks, we make fairly standard design choices here.

Pool & Image Encoding We represent the pth image I_p of the pool as a set of B

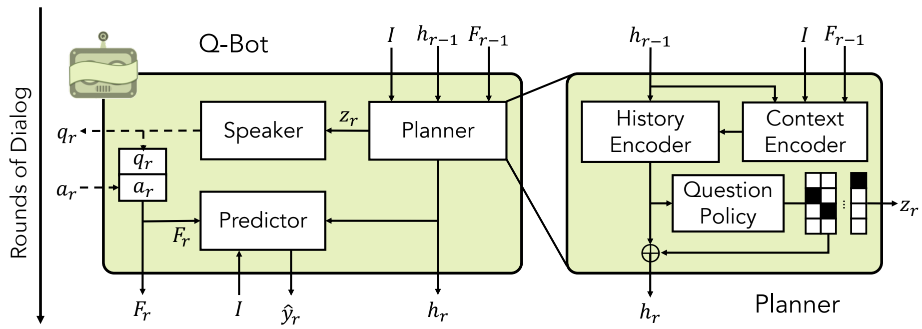


Figure 6.4: A single round of our q-bot which decomposes into the modules described in Section 6.2.1. This factorization allows us to fine-tune just the intention of the model for task performance, limiting language drift.

bounding boxes such that I_p^b is the embedding of the b-th box following [163]. Note that we do not assume prior knowledge about the size or composition of the pool.

Planner

The planner's role is to encode the dialog context (image pool and dialog history) and decide what to ask about in each round. To limit clutter, we denote the QA pair at round r as a 'fact' $F_r = [q_r, a_r]$.

Context Encoder Given the prior dialog state h_{r-1}, F_{r-1}, and image pool $\mathcal{I}$, the context encoder performs hierarchical attention to identify image regions in the pool that are most relevant for generating the next question. As we describe in the appendix, F_{r-1} and h_{r-1} to query the image to compute an attention distribution over both set of images (α_j) and P distributions over the bounding boxes in each image (β_j^i). The overall image encoding $\hat{v}_r$ at round r is computed as

$$\hat{v}_r = \sum_{j=1}^{P} \sum_{i=1}^{B} \alpha_j \beta_j^i v_j^i \tag{6.10}$$

where both image and region attentions are combined. We leave the details on computing these attention distributions to the appendix to conserve space. We note that this mechanism is agnostic to the pool size.

History Encoder. To track the state of the game, the planner applies an LSTM-based history encoder that takes $\hat{v}_r$ and F_r as input and produces an intermediate hidden state h_{r+1}. Here h_{r+1} includes a compact representation of question intent and dialog history, providing a differentiable connection between the intent and final predictions through the dialog state.

Question Policy. The question policy transforms h_{r+1} to a question representation z_r that will be passed to the speaker model to generate the actual question text. In some sense, z_r corresponds to the "intent" of the question (e.g. checking the existence of surfers) that triggers the speaker to produce corresponding text (e.g. "Is anyone surfing?"). A default choice for z_r is identity function (*i.e.*, $z_r = h_{r+1}$). Later we explore choices where z_r is a random variable (continuous or discrete) parameterized by h_{r+1}.

Speaker

Given an intent z_r, the speaker generates a natural language question. We model the speaker as a standard LSTM-based decoder with an initial hidden state equal to z_r (or an embedding of z_r for discrete z_r).

Predictor

The predictor uses the planner's hidden state to guess which image A-bothas selected. The predictor takes a concatenation $F = [F_1, \ldots, F_{r+1}]$ of fact embeddings and the dialog state h_{r+1} and computes an attention pooled feature $\hat{F}$ using h_{r+1} as attention context. A score is then computed for each image in the pool based on the image features, the pooled representation, and the dialog state (see appendix for full model details). These scores are normalized via a softmax to predict the target image. The model can then be trained end-to-end to minimize a cross-entropy loss on this prediction. Note the model is agnostic to the pool size.

6.2.2 Dialog without Dialog

Aside from some abstracted details, the game setting and model presented in the previous section could be trained without any further information – a pool of images could be generated, A-bot could be assigned an image, the game could be rolled out for arbitrarily many rounds, and Q-bot could be trained to predict the correct image given A-bot's answers. While conceptually possible, there is an obvious shortcoming – it would be nigh impossible for Q-bot to learn to produce interpretable questions. Nobody discovers French. They have to learn it.

At the other extreme (and representing standard practice in dialog problems), human's could be recruited to perform this image guessing game and provide dense supervision for what questions Q-bot should ask to perform well at this specific task. However, this suggests a machine learning paradigm that requires collecting language data for every new task. Aside from being costly, it is intellectually dissatisfying for agents' knowledge of natural language to be so inseparably intertwined with individual tasks. After all, one of the greatest powers of language is the ability to use it to communicate about many different problems.

In this section, we consider a middle-ground – training our agents with single-shot question answering data and then learning an agent that can carry on our task-driven dialog without further supervision.

Stage 1: Language Pre-training

We want Q-bot's language to be interpretable – in this paper we take that to mean it should be understandable by and semantically meaningful to humans, so it has to be something like a meaningful subset of a known human language. To pre-train the model to use interpretable human language, we design a supervised learning task for a single-round version of our game.

We leverage the VQAv2 [101] dataset as our language source to learn how to ask interpretable questions. By construction, for each question in VQAv2 there exists at least one image pair which are visually similar but have different ground truth answers to the question. This somewhat mirrors our dialog game – the image pair is the pool, the question is guaranteed to be discriminative, and we can provide an answer depending on A-bot's selected image. We can view this as a special case of our game that is fully supervised but contains only a single round of dialog. We can then train our Q-bot to mimic the human question (e.g. via cross-entropy teacher forcing) and to predict the correct image given the ground-truth answer.

Stage 2: Transferring to Dialog

The VQA dataset contains simple questions about images, but they are not aimed at accomplishing our image guessing task. Consequently, the goal of Dialog without Dialog is to transfer this learned language understanding to new tasks and demonstrate generalization in terms of interpretability *and* task performance across many task variations (e.g. multiple rounds of conversation and new pools of images).

As an initial setting, we could take the pre-trained weights from Stage 1 and simply fine-tune for our full image guessing task. However, this agent would face a number of challenges. It has never had to model multiple steps of a dialog. Further, while following the task objective of predicting A-bot's selected image, there is little to encourages Q-bot to continue producing interpretable language. We consider a number of modifications to address these problems.

Discrete Intention z Representation. Rather than a continuous vector passing from the question policy to the speaker, we consider a discrete random variable. Specifically, we consider a representation composed of N K-way Concrete variables [179] so $z_n \in [0,1]^K$ is a distribution over K objects.

We learn a linear transformation from the intermediate dialog state $\bar{h}_r$ to a set of logits $l^z_{Kn:K(n+1)-1}$ for each variable n in z:

$$l^z_{Kn:K(n+1)-1} = \text{LogSoftmax}(h_{Kn:K(n+1)-1}) \forall n \tag{6.11}$$

This parameterizes encoder distribution $p(z_r)$.

To provide input to the speaker, z_r is embedded using a learned dictionary of embeddings. In our case each variable in z has a dictionary of K learned embeddings. The value of z_n ($\in \{1, \ldots, K\}$) picks one of the embeddings for each variable and the final representation simply sums over all variables:

$$e_z = \sum_{n=0}^{N-1} E^z_n(z_n). \tag{6.12}$$

VAE Pre-training When using this representation for the intent, we train Stage 1 by replacing the likelihood with an ELBO loss to restrict information flow through z. This requires an encoder and a decoder. The decoder is the speaker and the encoder is a new module $q(z|q_0, \mathcal{I})$ that forms a conditional distribution over z. For the encoder we use a version of the previously described context encoder that uses just the question q_0 as attention query and parameterizes this Concrete distribution with a linear transformation of the resulting hidden state. The resulting ELBO loss is like the Full ELBO described (but not implemented) in [107]:

$$\begin{aligned}\mathcal{L} =& E_{z\sim q(z|q_0,\mathcal{I})}\left[\log p(\texttt{speaker}(z))\right] & (6.13)\\ &+ \frac{1}{N}\sum_{n=0}^{N-1} D_{KL}\left[q(z_n|q_0,\mathcal{I})||\mathcal{U}(K)\right] & (6.14)\end{aligned}$$

The first term encourages the encoder to mimic the VQA question. The second term pushes the distribution of z close to a K-way uniform prior, which forces z to only carry relevant information. Combined, the first two terms form an ELBO on the question likelihood given the image pool [178, 107].

Fixed Speaker Since the speaker contains only lower level information about how to generate language, we freeze it during task transfer. We want only the high level ideas represented by z and the predictor which receives direct feedback to adapt to the new task. If we updated the speaker then it could overfit its language to the sparse feedback available in each new setting.

Adaptation Curriculum As the pre-trained model has never had to keep track of dialog contexts beyond the first round, we fine-tune in two stages. In **Stage 2.A** we fix the Context Encoder and Question Policy parts of the Planner so the model can learn to track dialog effectively without trying to generate better dialog at the same time. This stage takes 20 epochs to train. Once Q-bot learns how to track dialog we update the entire planner in **Stage 2.B** for 5 epochs.[1]

6.2.3 Experiments

Settings

We consider experimental settings which test generalization along four dimensions: dialog round, pool type, pool size, and image domain. We can control the difficulties of the proposed DwD task by setting the number of dialog round, number of type of images in the pool and whether the task is operate on a different image domain. We consider three image sources – COCO [5], CUB [185], and AWA [186]. We vary pool size to be either 2 or 9 images either randomly selected or a contrasting pair (the synthetic VQA pools from Stage 1, only defined for VQA pool size 2). Unless specified, performance is reported for Q-bot's final guess at the last round.

[1] We find that 5 epochs stops training early enough to avoid the significant overfitting that can otherwise occur.

Metrics

We consider metrics addressing both **Task** performance and **Language** quality. While task performance is straightforward, language quality is harder to measure. We use multiple metrics including human evaluations reported in Section 6.2.3.

Task - Guessing Game Accuracy via A-bot. The point of transfer is to improve task performance so we report the accuracy of Q-bot's guess at the final round of dialog.

Language - Question Relevance via A-bot. To be human understandable, the generated questions should be relevant to at least one image in the pool. We measure question relevance as the maximum question-image relevance across the pool as measured by A-bot, i.e. $1 - p(\texttt{Not Relevant})$. We note that this is only a proxy for actual question relevance as A-bot may report `Not Relevant` erroneously if it fails to understand Q-bot's question; however, in practice we find A-bot does a fair job in determining relevance. We also provide human relevance judgements in Section 6.2.3.

Language - Fluency via Perplexity To evaluate Q-bot's fluency, we train an LSTM-based language model on the entire corpus of questions in VQA. This allows us to evaluate the perplexity of the questions generated by Q-bot for dialogs on its new tasks. Lower perplexity indicates the generated questions are similar to VQA questions in terms of syntax and content. Questions generated for the new tasks could have lower perplexity because they have drifted from English or because different things must be asked for the new task, so lower perplexity is not always better [187].

Language - Diversity via Distinct *n*-grams This considers the set of all questions generated by Q-bot across all rounds of dialog on the val set. It counts the number of n-grams in this set, N_n, and the number of distinct n-grams in this set, D_n, then reports $\frac{N_n}{D_n}$ for each value of $n \in \{1, 2, 3, 4\}$. Note that instead of normalizing by the number of words as in previous work [188, 189], we normalize by the number of n-grams so that the metric represents a percentage for values of n other than

Table 6.4: Performance of our models and baselines in different experimental settings. From setting `A` to setting `F`, agents are tasked with generalizing further from the source data. Our method strikes a balance between guessing game performance and interpretability.

			Accuracy ↑	Perplexity↓	A-bot Relevance ↑	Diversity↑
VQA 2 Contrast 1 Round	A1	Stage 1	0.73	2.62	0.87	0.50
	A2	Non-Var Cont	0.71	10.62	0.66	**5.55**
	A3	Ours	**0.82**	**2.6**	**0.88**	0.54
VQA 2 Contrast 5 Rounds	B1	Stage 1	0.67	2.62	0.87	0.50
	B2	Non-Var Cont	0.74	10.62	0.66	**5.55**
	B3	Ours	**0.87**	**2.60**	**0.88**	0.54
VQA 2 Random 5 Rounds	C1	Stage 1	0.64	**2.64**	0.75	1.73
	C2	Non-Var Cont	0.86	16.95	0.62	**8.13**
	C3	Ours	**0.95**	2.69	**0.77**	2.34
VQA 9 Random 9 Rounds	D1	Stage 1	0.18	2.72	**0.77**	1.11
	D2	Non-Var Cont	**0.78**	40.66	**0.77**	**2.57**
	D3	Ours	0.53	**2.55**	0.75	0.95
AWA 9 Random 9 Rounds	E1	Stage 1	0.47	2.49	**0.96**	0.24
	E2	Non-Var Cont	0.48	12.56	0.64	**2.21**
	E3	Ours	**0.74**	**2.41**	**0.96**	0.28
CUB 9 Random 9 Rounds	F1	Stage 1	0.36	2.56	**1.00**	0.04
	F2	Non-Var Cont	0.38	20.92	0.47	**2.16**
	F3	Ours	**0.74**	**2.47**	**1.00**	0.04

$n = 1$. Generative language models frequently produce safe standard outputs [188], so diversity is a sign this problem is decreasing, but diversity by itself does not make language meaningful or useful.

Results

Baselines. We compare our proposed approach to two baselines – **Stage 1** and **Non-Var Cont** – each ablating some aspects of our design choices. The **Stage 1** baseline is our model after the single-round fully-supervised pretraining. Improvements over this model represent gains made from task-based fine-tuning. The **Non-Var Cont** baseline is our model under standard encoder-decoder dialog model design choices – i.e. a continuous latent variable, maximum-likelihood pre-training, and fine-tuning the speaker model.

Results. Table. 6.4 presents results for our model and baselines in different settings. Starting from the first setting and moving downward, agents are tasked with generalizing further and further from their source data – from setting `A` which mimics the human data pretraining to setting `F` where agents must carry on a nine round dialog about 9 images containing only different bird species. Our final model uniformly performs well on both task performance and language fluency across different settings in terms of the automatic evaluation metrics (see bolded results). Other key findings are:

Ours *vs.* Stage 1: To understand the relative importance of the proposed stage 2 training which transferring to dialog for DwD task, we compared the task accuracy performance of our model with Stage 1. For setting `A` which matches the training regime, our model outperforms Stage 1 by 9% on task performance. As the task differs, we see further gains with our model consistently outperforming Stage 1 by 20-38%. Despite these gains, our model maintains similar language perplexity, A-bot relevance, and diversity.

Ours *vs.* Non-Var Cont: Our discrete latent variable, variational pre-training objective, and fixed speaker also play a important roles in avoiding language drift. Compared to the Non-Var Cont model without these techniques, our model achieves over 4x lower perplexity and 10-53% better A-bot Relevance. Our model also improves the averaged accuracy over the Non-Var Cont model, which means more interpretable language also improves the task performance. Note that Non-Var Cont has 2-100x higher diversity compared to our model, since the language is shifted away from English (and towards gibberish).

Table 6.5: Human evaluation of language quality – question fluency (top) and relevance (bottom). Each row compares a pair of agent-generated questions, asking users which (or possibly neither) is more fluent/relevant. The values report the percentage of times the option represented by that column was chosen.

	Neither	Stage 1	Non-Var Cont	Ours
Stage 1 vs Non-Var Cont	31.7%	48.1%	20.2%	–
Stage 1 vs Ours	49.0%	26.2%	–	24.8%
Non-Var Cont vs Ours	32.7%	–	17.9%	49.4%
Stage 1 vs Non-Var Cont	19.6%	48.8%	31.7%	–
Stage 1 vs Ours	25.0%	38.4%	–	36.6%
Non-Var Cont vs Ours	22.0%	–	30.2%	47.8%

Game Variations:

– **Dialog Rounds:** Longer dialogs (more rounds) achieve better accuracy (A3 vs B3).

– **Pool Type:** Random pools are easier compared to contrast pool (B3 vs C3 accuracy), however, language fluency and relevance drop on the random pools (B3 vs C3 perplexity and a-bot relevance).

– **Image Source:** CUB and AWA pools are harder compared to COCO image domain (D3 vs E3 vs F3). Surprisingly, our models maintains similar perplexity and high a-bot relevance even on these out-of-domain image pools. The Stage 1 and Non-Var Cont baselines generalize poorly to these different image domains – reporting task accuracies nearly half our model performance.

Human Studies

In addition to the automatic metrics, we also evaluate our models through human studies. Specifically, we use workers (turkers) on Amazon Mechanical Turk to evaluate the relevance, fluency, and task performance of our models. We discuss each study below.

Human Study for Question Relevance. To get a more accurate measure of question relevance, we asked humans to evaluate questions generated by our model and the baselines (Stage 1 & Non-Var Cont). We curated 300 random, size 4 pools where all three models predicted the target correctly at round 5. For a random round, we show turker's the questions from a pair of models and ask ”Which question is most relevant to the images?” Answering the question is a forced choice between three options: either of the pair of models or an “equally relevant” option. More details including an example of the interface can be found in appendix C. Table. 6.5 (bottom) shows the frequency with which each option was chosen for each model pair. Our model is considered more relevant than the Non-Var Cont model (47.8% vs. 30.2% preference) and about the same as the Stage 1 model (36.6% vs. 38.4% preference).

Human Study for Fluency. We also evaluate fluency by asking humans to compare questions. In particular, we presented the same pairs of questions to turkers as in the relevance study, but this time we did *not* present the pool of images and asked them ”Which question is more understandable?” As before, there was a forced choice between two models and an “equally understandable” option. This captures fluency because humans are more likely to understand grammatically correct. Table. 6.5 (top) shows the frequency with which each option was chosen for each model pair. Our model is considered more fluent than the Non-Var Cont model (49.4% vs. 17.9% preference) and about the same as the Stage 1 model (49.0% neither question more fluent).

Human Study for Task Performance. What we really want is for humans to be able to collaborate with bots to solve tasks. Therefore, the most direct evaluation of our the DwD task is to have humans interact dynamically with Q-bot. We implemented an interface that allowed turkers to interact with Q-bot in real time. Q-bot asks a question. A human answers it. Q-bot asks a new question in response to the human answer and the human responds to that question. After the 4th answer

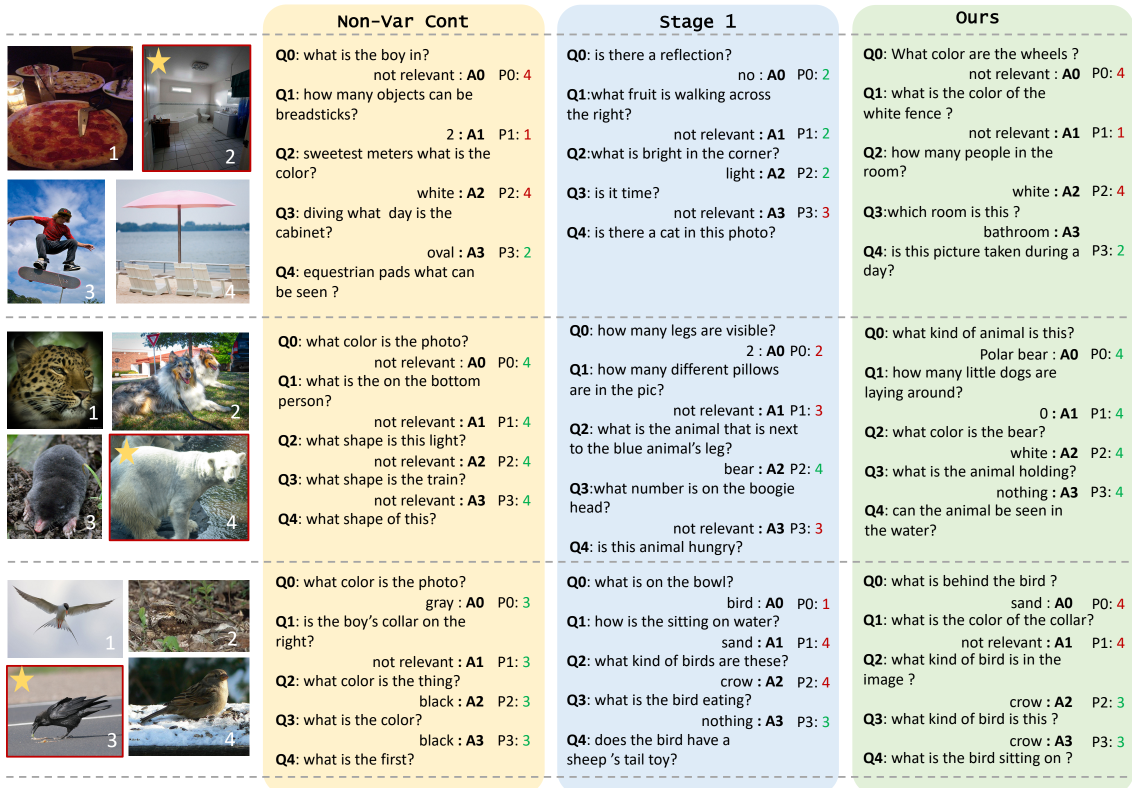


Figure 6.5: Qualitative comparison of dialogs generated by our model with those generated by Non-Var Cont and Stage 1 baselines. Top / middle /bottom rows are image pool from COCO / AWA / CUB images respectively. Our model pretrained on VQA (COCO image) generates more interpretable questions for the DwD task which is semantic meaning and generalize well to out-of-domain images.

Q-bot makes a guess about which target image the human was answering based on. Our interface is described in section C of the supplement. We perform this study for the same pools for each model and find our approach achieves an accuracy of 69.39% – significantly higher than Non-Var Cont at 44.90% and Stage 1 at 22.92%. This study shows that our model learns a strategy for this task that is amenable to human-AI collaboration. This is in contrast to prior work [110] that showed that improvements captured by task-trained models for similar image-retrieval tasks did not transfer when paired with human partners.

Qualitative Results

Figure 6.5 shows example outputs of Non-Var Cont baseline, Stage 1 model and our proposed models on three different image sources – COCO, AWA and CUB datasets. We can see that COCO images contains varieties of concepts while AWA images contains on different animals and CUB images contains on different species of birds. The A-bot is not accurate, which introduces noisy signals for Q-bot to learn the DWD tasks. Compared with the baselines, our approach asks more relevant and interpretable questions in the dialog.

Model Ablations

We investigate the impact of our modelling choices from Section 6.2.2. In Table. 6.6 we report the mean of all four automated metrics averaged over pool sizes, pool sampling strategies, and datasets.[2]Next we explain how we vary each of these model dimensions

– Our 128 4-way Concrete variables require 512 logits (**Discrete**). Thus we compare to the standard Gaussian random variable common throughout VAEs with 512 dimensions (**Continuous**). This just removes the KL term ((6.14)).
– In both discrete and continuous cases we train with an ELBO loss (**ELBO**), so we compare to a maximum likelihood only model (**MLE**) that uses an identity function as in the default option for the Question Policy (see Section 6.2.1).
– We consider checkpoints after each step of our training curriculum: **Stage 1**, **Stage 2.A**, and **Stage 2.B**. For some approaches we skip Stage 2.A and go straight to fine-tuning everything except the speaker as in Stage 2.B. This is denoted by **Stage 2**.
– We consider 3 variations on how the speaker is fine-tuned. The first is our proposed

[2]This includes 10 settings: {random 2, 4, 9 pools }$\times$ {VQA, AWA, CUB} and 2 contrats pools on VQA

Table 6.6: Various ablations of our training curriculum.

	z Structure	Loss	Curriculum	Speaker	Accuracy	Perplexity	Relevance	Diversity
1	Discrete	ELBO	Stage 2.B	Fixed (Ours)	0.81	2.57	0.89	0.86
2	Discrete	ELBO	Stage 2	Fine-tuned	0.82	2.54	0.85	0.59
3	Discrete	ELBO	Stage 2	Parallel	0.78	2.60	0.88	0.73
4	Discrete	ELBO	Stage 1	Fixed	0.72	2.60	0.91	0.48
5	Discrete	ELBO	Stage 2.A	Fixed	0.80	2.59	0.89	0.81
6	Discrete	ELBO	Stage 2	Fixed	0.80	2.53	0.85	0.62
7	Continuous	ELBO	Stage 2.B	Fixed	0.75	2.45	0.66	0.23
8	Continous	MLE	Stage 2.B	Fixed	0.78	4.27	0.83	4.33

approach of fixing the speaker (**Fixed**). The next fine-tunes the speaker (**Fine-tuned**). To evaluate the impact of fine-tuning we also consider a version of the speaker which can not learn to ask better questions by using a parallel version of the same model (**Parallel**). This last version will be described more below.

Discrete Outperforms Continuous z. By comparing our model in row 1 of Table. 6.6 to row 7 we see that our discrete model outperforms the corresponding continuous model in terms of task performance (higher Accuracy) and about matches it in interpretability (similar Perplexity and higher Relevance). This may be a result of discreteness constraining the optimization problem to prevent over-fitting and is consistent with previous work that used a discrete latent variable to model dialog.

Stage 2.B Less Important than Stage 2.A Comparing rows 4, 5, and 1 of Table. 6.6, we can see that each additional step, Stage 2.A (row 4 `->` 5) and Stage 2.B (row 5 `->` 1), increases task performance and stays about the same in terms of interpretability. However, most gains in task performance happen between Stage 1 and Stage 2. This indicates that improvements in task performance are mainly from learning to incorporate information over multiple rounds of dialog.

Better Predictions, Slightly Better Questions To further investigate whether Q-bot is asking better questions or just understanding dialog context for prediction better we considered the **Parallel** speaker model. This model loaded two copies of Q-bot, A and B both starting at Stage 1. Copy A was fine-tuned for task performance, but every z it generated was ignored and replaced with the z generated by copy B, which was not updated at all. The result was that copy A of the model could not incorporate dialog context into its questions any better than the Stage 1 model, so all it could do was track the dialog better for prediction purposes. By comparing the performance of copy A (row 3 of Table. 6.6) to our model (row 1) we can see a 3 point different in accuracy, so the question content of our model has improved after fine-tuning, but not by a lot. Most improvements are from dialog tracking for prediction (row 3 accuracy is much higher than row 4 accuracy).

Fine-tuned Speaker During both Stage 2.A and Stage 2.B we fix the Speaker module because it is intended to capture low level language details and we do not want it to change its understanding of English. Row 2 of Table. 6.6 does not fix the Speaker during Stage 2 fine-tuning. Instead, it uses each softmax at each step of the LSTM decoder to parameterize one Concrete variable [178] per word. This allows gradients to flow through the decoder during fine-tuning, allowing the model to tune low-level signals. This is similar to previous approaches which either used this technique [190] or REINFORCE [26] This model is competitive with DWD in terms of task performance. When we inspect its output we see somewhat less interpretable language. We favor our model because it is slightly better in terms o

Variational Prior Helps Interpretability We found the most important factor for maintaining interpretability to be the ELBO loss we applied during pre-training. Comparing the continuous Gaussian variable (row 7) to a similar hidden state (row 8) trained without the prior term (6.14) we see drastically different perplexity and diversity. Perplexity and diversity drop because the model has drifted far from English. This is similar to the effect in the Non-Var Cont, which is the model from row 8 with a fine-tuned speaker.

6.2.4 Discussion

In this section we proposed the Dialog without Dialog (DwD) task along with a model designed to solve this task and an evaluation scheme that takes its goals into account. The task is to build a dialog agent that generates meaningful and useful dialogs with language supervision only from , *i.e.*, without dialog. This balance is hard to strike, but our proposed model manages to strike it. We find it helps to represent dialogs with a discrete latent variable and carefully transfer language information via multi-stage training. While baseline models either perform well at new tasks through fine-tuning or maintain interpretability, our model achieves the goal of DwD by doing both. We hope both our task and our model help inspire useful dialog agents that can also interact with humans.

CHAPTER 7

VISION AND LANGUAGE PRETRAINING

In this chapter, our goal is to build a general multi-modal AI model and training paradigm that has a unified structure, utilize large external dataset and handle multiple tasks at the same time.

In the first section, motivated by recently proposed BERT [15], which can effectively learn the textual representations through large-scale pretraining. We want a model that can learn visual grounding – associations between textual phrase and visual representations through pretraining. We extend the popular BERT architecture to a multi-modal two-stream model, processing both visual and textual inputs in separate streams that interact through co-attentional transformer layers. We pretrain our model through two proxy tasks on the large, automatically collected Conceptual Captions dataset and then transfer it to multiple established vision-and-language tasks – visual question answering, visual commonsense reasoning, referring expressions, and caption-based image retrieval – by making only minor additions to the base architecture. Our work represents a shift away from learning groundings between vision and language only as part of task training and towards treating visual grounding as a pretrainable and transferable capability.

Next, we investigate these relationships between vision-and-language tasks by developing a large-scale, multi-task training regime. Our approach culminates in a single model on 12 datasets from four broad categories of task including visual question answering, caption-based image retrieval, grounding referring expressions, and multi-modal verification. Compared to independently trained single-task models, this represents a reduction from approximately 3 billion parameters to 270 million while simultaneously improving performance by 2.05 points on average across tasks. We use our multi-task framework to perform an in-depth analysis of the effect of joint training on diverse tasks. Further, we show that finetuning task-specific models from our single multi-task model can lead to further improvements, achieving performance at or above the state-of-the-art.

7.1 ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks

We present a joint model for learning task-agnostic visual grounding from paired visiolinguistic data which we call Vision & Language BERT (ViLBERT for short). Our approach extends the recently developed BERT [15] language model to jointly reason about text and images. Our key technical innovation is introducing separate streams for vision and language processing that communicate through co-attentional transformer layers. This structure can accommodate the differing processing needs of each modality and provides interaction between modalities at varying representation depths. We demonstrate that this structure outperforms a single-stream unified model in our experiments.

In analogy to the training tasks in [15], we train our model on Conceptual Captions on two proxy tasks: predicting the semantics of masked words and image regions given the unmasked inputs, and predicting whether an image and text segment correspond. We apply our pretrained model as a base for four established vision-and-language tasks – visual question answering [39], visual commonsense reasoning [40], referring expressions [2], and caption-based image retrieval [41] – setting state-of-the-art on all four tasks. We find improvements of 2 to 10 percentage points across these tasks when compared to state-of-the-art task-specific baselines using separately pretrained vision and language models. Furthermore, our structure is simple to modify for each of these tasks – serving as a common foundation for visual grounding across multiple vision-and-language tasks.

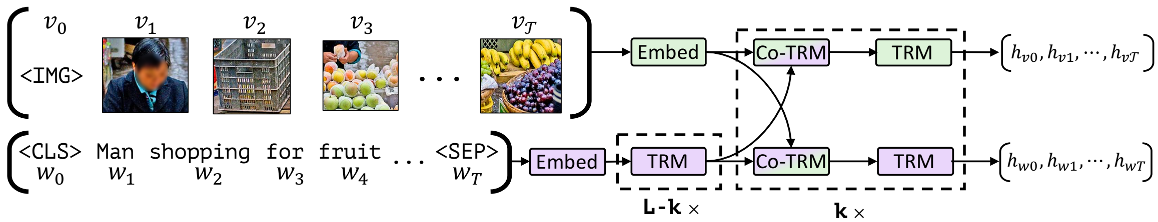


Figure 7.1: Our ViLBERT model consists of two parallel streams for visual (green) and linguistic (purple) processing that interact through novel co-attentional transformer layers. This structure allows for variable depths for each modality and enables sparse interaction through co-attention. Dashed boxes with multiplier subscripts denote repeated blocks of layers.

7.1.1 ViLBERT

Inspired by BERT's success at language modeling, we would like to develop analogous models and training tasks to learn joint representations of language and visual content from paired data. Specifically, we consider jointly representing static images and corresponding descriptive text.

One straightforward approach is to make minimal changes to BERT – simply discretizing the space of visual inputs via clustering, treat these visual 'tokens' exactly like text inputs, and start from a pretrained BERT model. This architecture suffers from a number of drawbacks. First, initial clustering may result in discretization error and lose important visual details. Second, it treats inputs from both modalities identically, ignoring that they may need different levels of processing due to either their inherent complexity or the initial level of abstraction of their input representations. For instance, image regions may have weaker relations than words in a sentence and visual features are themselves often already the output of a very deep network. Finally, forcing the pretrained weights to accommodate the large set of additional visual 'tokens' may damage the learned BERT language model. Instead, we develop a two-stream architecture modelling each modality separately and then fusing them through a small set of attention-based interactions. This approach allows for variable network depth for each modality and enables cross-modal connections at different depths.

Our model which we call ViLBERT is shown in Fig. 7.1 and consists of two parallel BERT-style models operating over image regions and text segments. Each stream is a series of transformer blocks (TRM) and novel co-attentional transformer layers (Co-TRM) which we introduce to enable information exchange between modalities. Given an image I represented as a set of region features $v_1, \ldots, v_{\mathcal{T}}$ and a text input $w_0, \ldots, w_T$, our model outputs final representations $h_{v0}, \ldots, h_{v\mathcal{T}}$ and $h_{w0}, \ldots, h_{wT}$. Notice that exchange between the two streams is restricted to be between specific layers and that the text stream has significantly more processing before interacting with visual features – matching our intuitions that our chosen visual features are already fairly high-level and require limited context-aggregation compared to words in a sentence.

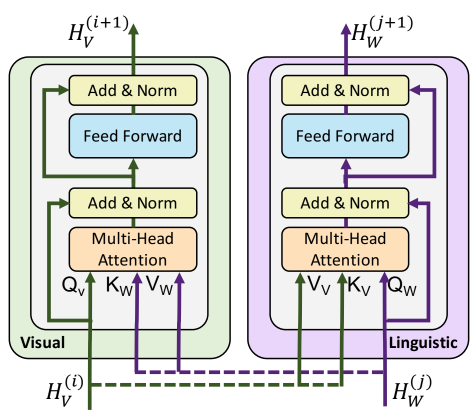
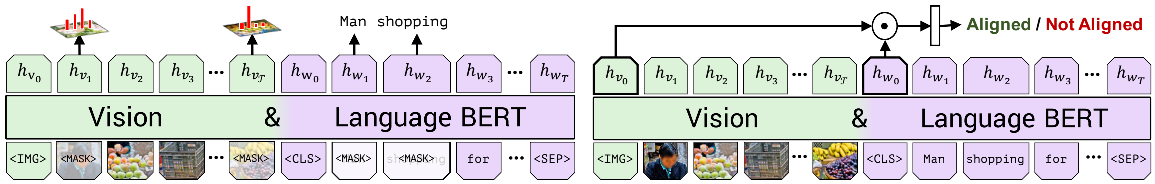


Figure 7.2: Our co-attention transformer layer

Co-Attentional Transformer Layers. We introduce a co-attentional transformer layer shown in Fig. 7.2. Given intermediate visual and linguistic representations $H_V^{(i)}$ and $H_W^{(j)}$, the module computes query, key, and value matrices as in a standard transformer block. However, the keys and values from each modality are passed as input to the other modality's multi-headed attention block. Consequentially, the attention block produces attention-pooled features for each modality conditioned on the other – in effect performing image-conditioned language attention in the visual stream and



(a) Masked multi-modal learning (b) Multi-modal alignment prediction

Figure 7.3: We train ViLBERT on the Conceptual Captions [38] dataset under two training tasks to learn visual grounding. In masked multi-modal learning, the model must reconstruct image region categories or words for masked inputs given the observed inputs. In multi-modal alignment prediction, the model must predict whether or not the caption describes the image content.

language-conditioned image attention in the linguistic stream. The latter mimics common attention mechanisms found in vision-and-language models [163]. The rest of the transformer block proceeds as before, including a residual add with the initial representations – resulting in a multi-modal feature. In general, co-attention for vision-and-language is not a new idea (being first proposed in [191]) and concurrent work yu2019deep, peng2018dynamic has shown the effectiveness of similar co-attentional transformer structures on the visual question answering task.

Image Representations. We generate image region features by extracting bounding boxes and their visual features from a pre-trained object detection network (see Sec. 7.1.2). Unlike words in text, image regions lack a natural ordering. we encode spatial location instead, constructing a 5-d vector from region position (normalized top-left and bottom-right coordinates) and the fraction of image area covered. This is then projected to match the dimension of the visual feature and they are summed.

We mark the beginning of an image region sequence with a special `IMG` token representing the entire image (i.e. mean-pooled visual features with a spatial encoding corresponding to the entire image).

Training Tasks and Objectives. In analogy to those described in the previous section, we consider two pretraining tasks: *masked multi-modal modelling* and *multi-modal alignment prediction*.

The masked multi-modal modelling task (shown in Fig. 7.3a) follows from the masked language modelling task in standard BERT – masking approximately 15% of both words and image region inputs and tasking the model with reconstructing them given the remaining inputs. Masked image regions have their image features zeroed out 90% of the time and are unaltered 10%. Masked text inputs are handled as in BERT. Rather than directly regressing the masked feature values, the model instead predicts a distribution over semantic classes for the corresponding image region. To supervise this, we take the output distribution for the region from the same pretrained detection model used in feature extraction. We train the model to minimize the KL divergence between these two distributions. This choice reflects the notion that language often only identifies high-level semantics of visual content and is unlikely to be able to reconstruct exact image features. Further, applying a regression loss could make it difficult to balance losses incurred by masked image and text inputs.

In the multi-modal alignment task (shown in Fig. 7.3b), the model is presented an image-text pair as $\{\texttt{IMG}, v_1, \ldots, v_\mathcal{T}, \texttt{CLS}, w_1, \ldots, w_T, \texttt{SEP}\}$ and must predict whether the image and text are aligned, *i.e.* whether the text describes the image. We take the outputs $h_{\texttt{IMG}}$ and $h_{\texttt{CLS}}$ as holistic representations of the visual and linguistic inputs. Borrowing another common structure from vision-and-language models, we compute the overall representation as an element-wise product between $h_{\texttt{IMG}}$ and $h_{\texttt{CLS}}$ and learn a linear layer to make the binary prediction whether the image and text are aligned. However, the Conceptual Captions [38] dataset only includes aligned image-caption pairs. To generate negatives for an image-caption pair, we randomly replace either the image or caption with another.

7.1.2 Experimental Settings

In this section, we describe how we train our model and provide overviews of the vision-and-language tasks to which we transfer the trained model.

Training ViLBERT

To train our full ViLBERT model, we apply the training tasks presented in Sec. 7.1.1 to the Conceptual Captions dataset [38]. Conceptual Captions is a collection of 3.3 million image-caption pairs automatically scraped from alt-text enabled web images. The automatic collection and sanitation process leaves some noise and the 'captions' are sometimes not human-like or short on details (*e.g.* "actors attend the premiere at festival"). However, it presents a huge diversity of visual content and serves as an excellent dataset for our purposes. Since some links had become broken by the time we downloaded the data, our model is trained with around 3.1 million image-caption pairs.

Implementation Details. We initialize the linguistic stream of our ViLBERT model with a BERT language model pretrained on the BookCorpus [192] and English Wikipedia. Specifically, we use the $\texttt{BERT}_{\texttt{BASE}}$ model [15] which has 12 layers of transformer blocks with each block having a hidden state size of 762 and 12 attention heads. We choose to use the `BASE` model due to concerns over training time but find it likely the more powerful $\texttt{BERT}_{\texttt{LARGE}}$ model could further boost performance.

We use Faster R-CNN [156] (with ResNet-101 [] backbone) pretrained on the Visual Genome dataset [120] (see [163] for details) to extract region features. We select regions where class detection probability exceeds a confidence threshold and keep between 10 to 36 high-scoring boxes. For each selected region i, v_i is defined as the mean-pooled convolutional feature from that region. Transformer and co-attentional transformer blocks in the visual stream have hidden state size of 1024 and 8 attention heads.

We train on 8 TitanX GPUs with a total batch size of 512 for 10 epochs. We use the Adam optimizer with initial learning rates of 1e-4. We use a linear decay learning rate schedule with warm up to train the model. Both training task losses are weighed equally.

Figure 7.4: Examples for each vision-and-language task we transfer ViLBERT to in our experiments.

Vision-and-Language Transfer Tasks

We transfer our pretrained ViLBERT model to a set of four established vision-and-language tasks and one diagnostic task. We follow a fine-tuning strategy where we modify the pretrained base model to perform the new task and then train the entire model end-to-end. In all cases, the modification is trivial – typically amounting to learning a classification layer. This is in stark contrast to the significant efforts made within the community to develop specialized models for each of these tasks. We describe the problem, dataset, model modifications, and training objective for each task below.

Visual Question Answering (VQA). The VQA task requires answering natural language questions about images. We train and evaluate on the VQA 2.0 dataset [39] consisting of 1.1 million questions about COCO images [49] each with 10 answers. To fine-tune ViLBERT on VQA, we learn a two layer MLP on top of the element-wise product of the image and text representations $h_{\texttt{IMG}}$ and $h_{\texttt{CLS}}$, mapping this representation to 3,129 possible answers. As in [163], we treat VQA as a multi-label classification task – assigning a soft target score to each answer based on its relevancy to the 10 human answer responses. We then train with a binary cross-entropy loss on the soft target scores using a batch size of 256 over a maximum of 20 epochs. We use the Adam optimizer with an initial learning rate of 4e-5. At inference, we simply take a softmax.

Visual Commonsense Reasoning (VCR). Given an image, the VCR task presents two problems – visual question answering (Q→A) and answer justification (QA→R) – both being posed as multiple-choice problems. The holistic setting (Q→AR) requires both the chosen answer and then the chosen rationale to be correct. The Visual Commonsense Reasoning (VCR) dataset consists of 290k multiple choice QA problems derived from 110k movie scenes. Different from the VQA dataset, VCR integrates object tags into the language providing direct grounding supervision and explicitly excludes referring expressions. To finetune on this task, we concatenate the question and each possible response to form four different text inputs and pass each through ViLBERT along with the image. We learn a linear layer on top of the post-elementwise product representation to predict a score for each pair. The final prediction is a softmax over these four scores and is trained under a cross-entropy loss over 20 epochs with a batch size of 64 and initial learning rate of 2e-5.

Grounding Referring Expressions. The referring expression task is to localize an image region given a natural language reference. We train and evaluate on the RefCOCO+ dataset [193]. A common approach to this task is to rerank a set of image region proposals given the referring expression. Thus we directly use the bounding box proposals provided by [194], which use a Mask R-CNN [162] pretrained on the COCO dataset. For fine-tuning, we pass the final representation h_{v_i} for each image region i into a learned linear layer to predict a matching score. We label each proposal box by computing the IoU with the ground truth box and thresholding at 0.5. We train with a binary cross-entropy loss for a maximum of 20 epochs with a batch size of 256 and an initial learning rate of 4e-5. At inference, we use the highest scoring region as the prediction.

Caption-Based Image Retrieval. Caption-based image retrieval is the task of identifying an image from a pool given a caption describing its content. We train and evaluate on the Flickr30k dataset [41] consisting of 31,000 images from Flickr with five captions each. Following the splits in [195], we use 1,000 images for validation and test each and train on the rest. These captions are well-grounded in and descriptive of the visual content and are qualitatively different than the automatically collected Conceptual Captions. We train in a 4-way multiple-choice setting by randomly sampling three distractors for each image-caption pair – substituting a random caption, a random image, or a hard negative from among the 100 nearest neighbors of the target image. We compute the alignment score (as in alignment prediction pretraining) for each and apply a softmax. We train this model under a cross-entropy loss to select the true image-caption pair for 20 epochs with a batch size of 64 and an initial learning rate of 2e-5. At inference, we score each caption-image pair in the test set and then sort. For efficiency, we cache the linguistic stream representation before the first Co-TRM layer – effectively freezing the linguistic representation before fusion.

'Zero-shot' Caption-Based Image Retrieval. The previous tasks are all transfer tasks that include dataset specific fine-tuning. In this 'zero-shot' task, we directly apply the pretrained the multi-modal alignment prediction mechanism to caption-based image retrieval in Flickr30k [41] *without fine-tuning* (thus the description as 'zero-shot'). The goal of this task is to demonstrate that the pretraining has developed the ability to ground text and that this can generalize to visual and linguistic variation without any task specific fine-tuning. We directly use the ViLBERT model trained on Conceptual Captions dataset described in Sec. 7.1.2. We use the alignment prediction objective as a scoring function and test on the same split as the caption-based image retrieval task described above.

7.1.3 Results and Analysis

Baselines. We compare our pretrained ViLBERT model against two ablative baselines:

– **Single-Stream** consisting of a single BERT architecture that processes both modality inputs through the same set of transformer blocks – sharing parameters and processing stacks for both visual and linguistic inputs. Like [119], this model avoids making changes to the BERT architecture, resulting in significantly deeper visual processing and earlier interaction between modalities than in our model. The model is initialized with $\text{BERT}_{\texttt{BASE}}$ and trained identically to our full model. We compare to this baseline to establish the impact of our two-stream architecture. As both streams interact throughout, we cannot cache any representations for efficiency. As such, we do not evaluate this baseline on image retrieval and zero-shot image retrieval due to high computational cost.

- **ViLBERT†** which is a ViLBERT architecture that has *not undergone our pre-training tasks.* Notably, it does still have BERT initilization for the linguistic stream and represents image regions with the same Faster R-CNN model as the full ViLBERT model. We compare to this baseline to isolate gains over task-specific baseline models that might be due to our architecture, language initialization, or visual features as opposed to our pretraining process on Conceptual Captions .

For both baselines and our model, we finetune the transfer tasks as described in the previous section.

Task-Specific Baselines. To put our results in context, we present published results of problem-specific methods that are to our knowledge state-of-the-art in each task: DFAF [196] for VQA, R2C [40] for VCR, MAttNet [194] for RefCOCO+, and SCAN [195] for caption-based image retrieval.

Results. Tab. 7.1 shows results across all transfer tasks and we highlight key findings below:

- **Our architecture improves performance over a single-stream model.** We observe improvements across tasks for ViLBERT over the single-stream baseline for both pretrained (Single-Stream vs. ViLBERT) and non-pretrained (Single-Stream† vs. ViLBERT†). Most significant gains are observed for VQA and RefCOCO+.

Table 7.1: Transfer task results for our ViLBERT model compared with existing state-of-the-art and sensible architectural ablations. † indicates models without pretraining on Conceptual Captions. For VCR and VQA which have private test sets, we report test results (in parentheses) only for our full model. Our full ViLBERT model outperforms task-specific state-of-the-art models across all tasks.

		VQA [39]	VCR [40]			RefCOCO+ [193]			Image Retrieval [41]			ZS Image Retrieval		
	Method	test-dev (test-std)	Q→A	QA→R	Q→AR	val	testA	testB	R1	R5	R10	R1	R5	R10
SOTA	DFAF [196]	70.22 (70.34)	-	-	-	-	-	-	-	-	-	-	-	-
	R2C [40]	-	63.8 (65.1)	67.2 (67.3)	43.1 (44.0)	-	-	-	-	-	-	-	-	-
	MAttNet [194]	-	-	-	-	65.33	71.62	56.02	-	-	-	-	-	-
	SCAN [195]	-	-	-	-	-	-	-	48.60	77.70	85.20	-	-	-
Ours	Single-Stream†	65.90	68.15	68.89	47.27	65.64	72.02	56.04	-	-	-	-	-	-
	Single-Stream	68.85	71.09	73.93	52.73	69.21	75.32	61.02	-	-	-	-	-	-
	ViLBERT†	68.93	69.26	71.01	49.48	68.61	75.97	58.44	45.50	76.78	85.02	0.00	0.00	0.00
	ViLBERT	**70.55 (70.92)**	**72.42 (73.3)**	**74.47 (74.6)**	**54.04 (54.8)**	**72.34**	**78.52**	**62.61**	**58.20**	**84.90**	**91.52**	**31.86**	**61.12**	**72.80**

– **Our pretraining tasks result in improved visiolinguistic representations.** Our models further improve by between 2% and 13% across tasks when using a ViLBERT model that has been pretrained under our proxy tasks (ViLBERT vs ViLBERT†). We also observe improvements on Single-Stream which verifies our proxy tasks can generalize to different model architectures.

– **Finetuning from ViLBERT is a powerful strategy for vision-and-language tasks.** With a single base architecture, our transfer task performance exceeds state-of-the-art task-specific models for all four established tasks. We set state-of-the-art for VCR, RefCOCO+ and image retrieval by significant margins (7-10 percentage points improvement). Further, extending to these tasks was simple – requiring the addition of a single classifier for each task.

Overall, these results demonstrate that our ViLBERT model is able to learn important visual-linguistic relationships that can be exploited by downstream tasks.

Effect of Visual Stream Depth. In Tab. 7.2 we compare the results transferring from ViLBERT models of varying depths. We consider depth with respect to the number of repeated CO-TRM→TRM blocks (shown in a dashed box in Fig. 7.1) in our model. We find that VQA and Image Retrieval tasks benefit from greater depth - performance increases monotonically until a layer depth of 6. Likewise, zero-shot

Table 7.2: Ablation study of the depth of our model with respect to the number of Co-TRM→TRM blocks (shown in a dashed box in Fig. 7.1). We find that different tasks perform better at different network depths – implying they may need more or less context aggregation.

	VQA [39]	VCR [40]			RefCOCO+ [193]			Image Retrieval [41]			ZS Image Retrieval [41]		
Method	test-dev	Q→A	QA→R	Q→AR	val	testA	testB	R1	R5	R10	R1	R5	R10
ViLBERT (2-layer)	69.92	72.44	**74.80**	**54.40**	71.74	**78.61**	62.28	55.68	84.26	90.56	26.14	56.04	68.80
ViLBERT (4-layer)	70.22	**72.45**	74.00	53.82	72.07	78.53	**63.14**	55.38	84.10	90.62	26.28	54.34	66.08
ViLBERT (6-layer)	**70.55**	72.42	74.47	54.04	**72.34**	78.52	62.61	58.20	84.90	**91.52**	31.86	61.12	72.80
ViLBERT (8-layer)	70.47	72.33	74.15	53.79	71.66	78.29	62.43	**58.78**	**85.60**	91.42	**32.80**	**63.38**	**74.62**

image retrieval continues making significant gains as depth increases. In contrast, VCR and RefCOCO+ seem to benefit from shallower models.

Benefits of Large Training Sets. We also studied the impact of the size of the pretraining dataset. For this experiment, we take random subsets of 25% and 50% from the conceptual caption dataset, and pretrain and finetune ViLBERT using the same setup as above. We can see that the accuracy grows monotonically as the amount of data increases, which suggests that ViLBERT may benefit from even more pretraining data.

Table 7.3: Transfer task results for ViLBERT as a function of the percentage of the Conceptual Captions dataset used during pre-training. We see monotonic gains as the pretraining dataset size grows.

	VQA [39]	VCR [40]			RefCOCO+ [193]			Image Retrieval [41]			ZS Image Retrieval [41]		
Method	test-dev	Q→A	QA→R	Q→AR	val	testA	testB	R1	R5	R10	R1	R5	R10
ViLBERT (0 %)	68.93	69.26	71.01	49.48	68.61	75.97	58.44	45.50	76.78	85.02	0.00	0.00	0.00
ViLBERT (25 %)	69.82	71.61	73.00	52.66	69.90	76.83	60.99	53.08	80.80	88.52	20.40	48.54	62.06
ViLBERT (50 %)	70.30	71.88	73.60	53.03	71.16	77.35	61.57	54.84	83.62	90.10	26.76	56.26	68.80
ViLBERT (100 %)	**70.55**	**72.42**	**74.47**	**54.04**	**72.34**	**78.52**	**62.61**	**58.20**	**84.90**	**91.52**	**31.86**	**61.12**	**72.80**

What does ViLBERT learn during pretraining? To get a sense for what ViLBERT learns during Conceptual Caption pretraining, we look at zero-shot caption-based image retreival and some qualitative examples. While zero-shot performance (Tab. 7.1, right) is significantly lower than the fine-tuned model (31.86 vs 58.20 R1) it performs reasonably without having seen a Flickr30k image or caption (31.86 vs 48.60 R1 for prior SOTA) – indicating that ViLBERT has learned a semantically meaningful alignment between vision and language during pretraining.

7.1.4 Discussion

We develop a joint model for image content and text and pretrain it on a large, automatically-collected dataset to learn visual grounding. Our ViLBERT model introduces a novel two-stream architecture with co-attentional transformer blocks that outperforms sensible ablations and exceeds state-of-the-art when transferred to multiple established vision-and-language tasks. Furthermore, transferring our model to these tasks is simple and easy to implement – requiring only the addition of a classifier for each task we examined here. We consider extensions of our model to other vision-and-language tasks (including those requiring generation) as well as multi-task learning as exciting future work.

7.2 12-in-1: Multi-Task Vision and Language Representation Learning

In this work, we develop a multi-task model for discriminative vision-and-language tasks based on the recently proposed ViLBERT[42] model. We consider four categories of tasks – training jointly on a total of 12 different datasets. Our results not only show that a single model can perform all these tasks, but also that joint training can lead to improvements on task metrics compared to single-task training with the same architecture. Before undertaking this effort, it was not obvious to us that this would be the case – multitask training is notorious challenging and vision-and-language datasets vary greatly in size, interface, and difficulty. Our model attains improvements of 0.25 to 4.19 absolute points from multi-task training – improving over corresponding single-task models for 11 out of 12 tasks. Further, we demonstrate that multi-task training is an effective pretraining step for single-task models

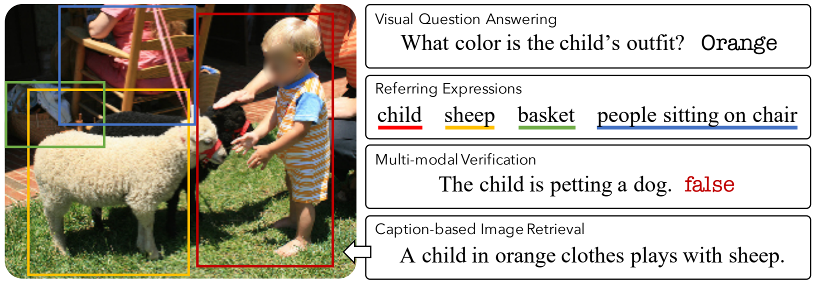


Figure 7.5: We introduce an approach for effective multi-task learning, training a single model on 12 popular vision-and-language datasets. This single model performs at par or even better than independent task-specific state-of-the-art approaches for many tasks.

– leading to further gains and setting a new state-of-the-art for 7 out of 12 tasks.

Large-scale multi-task learning is challenging as datasets can vary in size and difficulty. To address these issues, we introduce a dynamic stop-and-go training scheduler, task-dependent input tokens, and simple hyper-parameter heuristics. Using our proposed pipeline, we were able to train many multi-task models with varying datasets – assessing the relationships between different vision-and-language tasks in terms of their performance when trained together.

To summarize, we make the following contributions:

– We systematically analyze the joint training relationships between different of vision-and-language datasets and tasks and present a *Clean V&L Multi-Task setup*, which ensures no train-test leaks across task.

– We develop a single multi-task model trained on **12** popular V&L datasets. Compared to a set of independent models, this represents a reduction from ∼3 billion parameters to ∼270 million while simultaneously *improving* average performance by 2.05 points.

– We demonstrate that multi-task training is useful even in cases where single-task performance is paramount. On average, fine-tuning from our multi-task model for single tasks resulted in an average improvement of 2.98 points over baseline single-task trained models.

7.2.1 Vision-and-Language Tasks

Task-Groups and Datasets

We consider 12 popular vision and language datasets. These datasets cover a wide range of tasks and require diverse grounding granularity and reasoning skills. We group related datasets into four groups to facilitate our analysis:

Vocab-based VQA. Given an image and a natural-language question, select an answer from a fixed vocabulary. We consider three popular datasets for this group – VQAv2[101], GQA [197], and Visual Genome (VG) QA [120].

Image Retrieval. Given a caption and a pool of images, retrieve the target image that is best-described by the caption. We consider COCO[49] and Flickr30K[160] captioning datasets for this task-group.

Referring Expressions. Given a natural language expression and an image, identify the target region that is referred to by expression. The expression can vary greatly across datasets from simple noun phrases to multi-round dialogs. We consider phrase grounding in RefCOCO(+/g) [193, 198], Pointing questions in Visual7W [199], and dialog sequences in the GuessWhat [109]. We note that these language inputs vary significantly in terms of detail and structure.

Multi-modal Verification. Given one or more images and a natural language statement, judge the correctness or predict their semantic relationship. We consider NLVR^2 [200] and SNLI-VE [201]. In NLVR^2, two images are given and the statement must be true for both to be true. In SNLI-VE, image-statement pairs are classified as representing an entailment, contradiction, or neutral. That is, whether the content of the image confirms, refutes, or is insufficient to comment on the truth of the corresponding statement.

Table 7.4: Percentage of row-task test images that are present in column-tasks train/val images.

	% Row-Task Test Images in Column-Task Train/Val Set											
	[A]	[B]	[C]	[D]	[E]	[F]	[G]	[H]	[I]	[J]	[K]	[L]
[A] VQA2.0[101]	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
[B] VG QA[120]	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
[C] GQA[197]	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
[D] COCO[49]	100%	43%	33%	0%	0%	0%	0%	0%	7%	46%	0%	0%
[E] Flickr30k[160]	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	98%	0%
[F] RefCOCO[193]	100%	36%	27%	100%	0%	0%	0%	66%	8%	62%	0%	0%
[G] RefCOCO+[193]	100%	38%	27%	100%	0%	0%	0%	66%	8%	62%	0%	0%
[H] RefCOCOG [198]	100%	41%	31%	100%	0%	53%	53%	0%	8%	63%	0%	0%
[I] Visual 7W [199]	50%	100%	79%	48%	0%	8%	8%	10%	0%	24%	0%	0%
[J] GuessWhat[109]	100%	40%	31%	96%	0%	20%	20%	26%	7%	0%	0%	0%
[K] SNLI-VE[201]	0%	0%	0%	0%	94%	0%	0%	0%	0%	0%	0%	0%
[L] NLVR^2 [200]	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

A Clean V&L Multi-Task Setup

Many V&L tasks are built on top of each other and share significant overlap in terms of individual images. However, as each task is often examined in isolation, there does not exist an in-depth analysis of this overlap across different V&L tasks. Table. 7.4 shows the percentage of test images for the target tasks which are present in other tasks' train/val sets. As we can see, there exists significant overlap across tasks. Even though different tasks require different inputs and outputs, other task annotations will provide clues about the visual grounding – for example, a referring expression for a "blue striped ball" at training could unfairly improve a VQA model's ability to answer "What color is the striped ball?" for the same image at test time. To avoid information leakage from the annotations of other tasks, we propose a *cleaned* multi-task split for V&L tasks where test images are removed from train/val for all the tasks. We stress that the test sets are not modified in any way such that our results are comparable to prior work. Cleaning results in about 11% reduction in training data on average across datasets. Full details of this process and statistics regarding cleaned dataset size are available in the supplement.

7.2.2 Approach

Base Architecture

Our base architecture is ViLBERT, which is introduced in previous section. We make two important modifications to this pretraining process. First, when masking visual regions we also mask other regions with significant overlap (large than 0.4 IoU) to avoid leaking visual information. This forces the model to rely more heavily on language to predict image content. Second, we do not enforce the masked multi-modal modelling loss when sampling a negative (unmatching) caption for multi-modal alignment prediction. This will effectively remove the noise introduced by negative samples. While orthogonal to our primary contribution of multi-task learning, we found these modifications to make the baseline model more effective. For further discussion, see the supplemental material. All models we present are first pretrained in this manner.

Multi-Task Learning

We consider a simple multi-task model where each task has a task-specific 'head' network that branches off a common, shared 'trunk' ViLBERT model.As such, we learn shared trunk parameters θ_s and a set of task-specific layers $\{\theta_t\}_{t=1}^{\mathcal{T}}$ for $\mathcal{T}$ tasks. Our goal is to learn parameters $\theta_s \cup \{\theta_t\}_{t=1}^{\mathcal{T}}$ that minimize loss across all tasks. Details on heads and other modifications follow.

Task Token. While relying on the same groundings, different tasks may still require the model to process inputs differently – *e.g.* referring expressions just require grounding while VQA must follow grounding with additional reasoning. To enable this, we augment the query with a task token $\texttt{TASK}_t$ such that the new input format is $\{\texttt{IMG}, v_1, \ldots, v_n, \texttt{CLS}, \texttt{TASK}_t, w_1, \ldots, w_m, \texttt{SEP}\}$. The architecture can then leverage this task information in a bottom-up manner. In what follows, we describe the task-specific heads by task groups.

Vocab-Based VQA Output: We compute a overall image-query representation as an element-wise product between the holistic $h_{\texttt{IMG}}$ and $h_{\texttt{CLS}}$ representations. As in [163, 197], we treat vocab-based VQA as a multi-label classification task – assigning a soft target score to each answer based on its relevancy to the ground truth answer. We compute scores for a set of the pre-defined answers A by using a two-layer MLP on top of the overall representation:

$$P_v(A|I,Q) = \sigma(\texttt{MLP}(h_{\texttt{IMG}} \odot h_{\texttt{CLS}})) \tag{7.1}$$

where σ is the sigmoid function. Due to the answer vocabulary differences, VQA and VG QA share the MLP and answer vocabulary while GQA learns a separate one.

Image Retrieval Output: Using the same overall representation, we compute an alignment score between image-caption pairs as:

$$\texttt{Rel}(I,Q) = W_i(h_{\texttt{IMG}} \odot h_{\texttt{CLS}}) \tag{7.2}$$

where $W_i \in \mathbb{R}^{d\times 1}$ is shared across COCO and Flickr30k image retrieval tasks As in [42], we train a 4-way multiple-choice against hard-negatives selected off-line and then fixed. Recent work has used online hard-negative mining [202, 46] but this is costly to compute.

Referring Expressions Output: We rerank a set of region proposals [194] given the referring expression. We pass the final representation h_{v_i} for each image region i into a learned projection $W_r \in \mathbb{R}^{d\times 1}$ to predict a matching score.

$$\texttt{Rel}(v_i,Q) = W_r h_{vi} \tag{7.3}$$

Note that Q may be either a phrase, question or dialog based on different tasks (Ref-

COCO+/g, Visual7W, GuessWhat). W_r is shared across all the referring expression tasks.

Multi-modal Verification Output: Taking NLVR^2 as an example, the input is a concatenation of two images (I_0 and I_1) and a statement Q, that the model must judge the validity of the statement given the images. We consider this a classification problem given an embedding that encodes the two image-statement pairs (I_0, Q) and (I_1, Q). The output probability is predicted by a 2-layer MLP with softmax:

$$P_v(C|I_0, I_1, Q) = \texttt{softmax}\left(\texttt{MLP}\left(\begin{bmatrix} h^0_{\texttt{IMG}} \odot h^0_{\texttt{CLS}} \\ h^1_{\texttt{IMG}} \odot h^1_{\texttt{CLS}} \end{bmatrix}\right)\right) \tag{7.4}$$

where [ ] is concatenation. For SNLI-VE, the input is a single image and statement. We thus learn a separate classifier of the same form that predicts the sentiment (entailment, neutral, contradiction) from the inputs.

Large-Scale Multitask Training

With 6 task heads, 12 datasets, and over 4.4 million individual training instances – training our multi-task ViLBERT model is a daunting proposition. Multi-task learning (especially at this scale) poses significant challenges as learning objectives have complex and unknown dynamics and may compete [134]. Further, vision-and-language datasets vary significantly in size and difficulty. For instance, a single epoch of VG (our largest dataset) corresponds to 19.8 epochs of RefCOCOg (our smallest). Likewise, when trained in isolation RefCOCOg converges in 5K iterations whereas VQA takes 84K iterations (over 16 times more). Below, we describe the details of our multi-task training approach and techniques to overcome these challenges.

Pretraining. All our models are pretrained on Conceptual Caption dataset [38] including our self-supervised task modifications.

Round-Robin Batch-Level Sampling. We consider a round-robin batch-level

Algorithm 1: `DSG` for Multi-Task Learning

```
n_t ← number of iterations per epoch for task t
Δ ← size of gap between iterations in stop mode
DSG_t ← go
for i ← 1 to MaxIter :
    for t ∈ Tasks :
        if DSG_t = go or (DSG_t = stop and i mod Δ = 0) :
            Compute task loss L_t(θ) and gradient ∇_t(θ)
            Update θ ← θ − ϵ∇_t(θ), where θ = θ_s ∪ θ_t
        if i mod n_t = 0 :
            Compute validation score s_t on task t
            if DSG_t = go and Converged (s_t) :
                DSG_t ← stop
            else if DSG_t = stop and Diverged (s_t) :
                DSG_t ← go
    end
end
```

sampling regime that cycles through each task from the beginning of multi-task training. As such, one multi-task iteration consists of each task forwarding a batch and updating parameters in sequence.

Dynamic Stop-and-Go. As noted earlier, different tasks have different difficulties and dataset sizes. Consequentially, simply cycling through all tasks may drastically over-train smaller tasks leading to overfitting. Typically early-stopping provides a strong defense to this phenomenon; however, stopping a task in multi-task training introduces problems with *catastrophic forgetting* as the base network drifts over time due to other tasks. We introduce an intuitive but effective dynamic stop and go (`DSG`) mechanism to avoid these problems. We monitor the validation loss s_t of each task t, computing it once per task epoch. If performance improvement is less than 0.1% over 2 epochs, we consider it `Converged` and shift it into `stop` mode. In `DSG stop` mode, a task only updates every iter-gap (Δ) iterations. If validation performance degrades by 0.5% from the task's best measured performance while in `stop` mode, the task is considered `Diverged` and is returned to `DSG go`. This procedure is shown in Algorithm 1.

Curriculum Learning. Inspired by prior multi-task literature [203] [129], we experimented with both curriculum and anti-curriculum strategies based on task difficulty.

Specifically, for anti-curriculum we first train on the slowest-converging task-group G1 (Vocab-Based VQA) before starting full round-robin multi-task training. Inversely for the curriculum setting we first train on our fastest-converging task-group G3 (Referring Expressions). Different from previous observation [129, 137], we found that using no curriculum lead to superior performance when combined with other strategies proposed in this section.

Table 7.5: Comparison of our multi-task models to single-task performance. We find multi-task training (rows 3-5) provides significant gains over single-task training (rows 1-2) while reducing the parameter count from over 3 billion to 270 million. Further, following multi-task training by task-specific fine-tuning (rows 6-9) further gains can be made at the cost of increased parameters.

		Vocab-based VQA (G1)			Image Retrieval (G2)		Referring Expression (G3)					Verification (G4)			
		VQAv2	GQA	VG QA	COCO	Flickr30k	COCO	COCO+	COCOg	V7W	GW	NLVR^2	SNLI-VE	# params	All Tasks
	Clean	test-dev	test-dev	val	test(R1)	test(R1)	test	test	test	test	test	testP	test	(# models)	Average
1 Single-Task (ST)		71.82	58.19	34.38	65.28	61.14	78.63	71.11	72.24	80.51	62.81	74.25	76.72	3B (12)	67.25
2 Single-Task (ST)	✓	71.24	59.09	34.10	64.80	61.46	78.17	69.47	72.21	80.51	62.53	74.25	76.53	3B (12)	67.03
3 Group-Tasks (GT)	✓	72.03	59.60	36.18	65.06	66.00	80.23	72.79	75.30	81.54	64.78	74.62	76.52	1B (4)	68.72
4 All-Tasks (AT)	✓	72.57	60.12	36.36	63.70	63.52	80.58	73.25	75.96	82.75	65.04	78.44	76.78	**270M (1)**	69.08
5 All-Tasks$_{\text{w/o G4}}$	✓	72.62	59.55	36.76	64.46	64.18	80.43	73.40	76.43	82.99	64.80	-	-	266M (1)	-
6 GT $\xrightarrow{\text{finetune}}$ ST	✓	72.61	59.96	35.81	66.26	66.98	79.94	72.12	75.18	81.57	64.56	74.47	76.34	3B (12)	68.81
7 AT $\xrightarrow{\text{finetune}}$ ST	✓	72.92	60.48	36.56	65.46	65.14	80.86	73.45	76.00	83.01	65.15	**78.87**	76.73	3B (12)	69.55
8 AT $\xrightarrow{\text{finetune}}$ ST		**73.15**	**60.65**	**36.64**	**68.00**	**67.90**	**81.20**	**74.22**	**76.35**	**83.35**	**65.69**	**78.87**	**76.95**	3B (12)	**70.24**

Setting Multi-Task Hyperparameters. We follow a simple design philosophy – identify simple heuristics based on hyper-parameters tuned for each task in single-task training. This significantly reduces the burden of searching for joint-training hyper-parameters.

Batch Size: For multi-task, we keep the batch size tuned for single-task training for each task.

Warm-up Duration: We found it important to set warm-up duration relative to the largest dataset. Specifically, we run linear warm-up over $\eta * N$ iterations where N is the max. number of iterations taken to train any dataset in the single-task setting. We observe significant performance degradation for harder tasks when warm-up was shorter. We set η to 0.1 for our experiments.

Loss Scaling: Our model has shared and task-specific parameters and we found it

Table 7.6: Pair-wise (left) and triple-wise (right) inter-group representative task analysis. Each entry is the relative performance change from single-task training for the row-task when jointly trained with the column-task(s).

		Trained With					Trained With						
		G1	G2	G3	G4	Avg.	G1 & G2	G1& G3	G1 & G4	G2 & G3	G2 & G4	G3 & G4	Avg.
Relative PERF	G1 (VQAv2)	-	0.38%	0.38%	-0.20%	0.19%	-	-	-	0.63%	-0.08%	0.18%	0.24%
	G2 (Flickr30k)	0.46%	-	0.23%	-4.13%	-1.15%	-	1.24%	0.49%	-	-	-4.36%	-0.88%
	G3 (Visual7W)	0.39%	0.78%	-	0.24%	0.47%	0.86%	-	0.19%	-	0.29%	-	0.44%
	G4 (NLVR^2)	2.29%	1.47%	0.67%	-	1.48%	3.69%	3.22%	-	2.73%	-	-	3.21%
	Avg.	1.04%	0.88%	0.43%	-1.36%	-	2.27%	2.23%	0.34%	1.68%	0.10%	-2.09%	-

important to maintain separate learning rates. For the shared base model, we set the the base learning rate to the minimum over all single-task dataset parameters. To accommodate variable learning rates for each dataset, we scale the task loss for each dataset by the ratio of task target learning rate over base learning rate.

Implementation Details. Image features are from a ResNeXT-152 [204] based Faster-RCNN [156] trained on Visual Genome [120] with attribute loss. Our model first initialized from pretrained BERT weights [15]. Our models are trained using AdamW optimizer [205] with a linear warmup and linear decay learning rate scheduler. We train our multi-task model for 40K total iterations (same as number of iterations for VG QA single task) on 8 NVIDIA V100 GPUs for 5 days. See the supplement for a full list of per task learning rates, batch sizes, and hyperparameter settings.

7.2.3 Experiments and Results

Single-Task Performance

To establish baseline performance for the ViLBERT architecture that forms the backbone of our multi-task experiments, we first train single-task models on top of the base ViLBERT architecture for each of our 12 datasets. Rows 1 and 2 in Table. 7.5 show the performance of these models trained on the full and cleaned datasets, respectively. As expected, reducing the training set size through cleaning results in lower performance in most cases. Our improvements over the pretraining objective results in better downstream tasks performance (71.82 vs. 70.55 on VQA and 61.46

vs. 58.20 on Flickr30k Recall@1). See the supplementary for full comparison. Overall, our base architecture is competitive with prior work and a good starting point for multi-task learning.

Intra-Group Multi-task Performance

We begin with the most intuitive multi-task setting – jointly training tasks within the same groups. As grouped tasks are typically highly related, this is akin to some existing data augmentation practices (*e.g.* adding Visual Genome (VG) QA data when training VQA). Note this corresponds to four separate multi-task models – one for each group.

Table. 7.5 row 3 shows the result of intra-group multi-task training. Comparing with single-task models trained on the same data (row 2), we see meaningful improvements of between 0.37% (NLVR^2) and 4.54% (Flickr30k retrieval) points for 11 out of 12 tasks (only SNLI-VE did not improve). Comparing to row 1, we see that intra-group multi-task training overcomes the data-loss from cleaning with an average score of 68.72, outperforming the single-task models trained on the full datasets which have an average score of 67.25. Further, the total number of parameters drops by a factor of 3× – going from 12 full models to only 4.

Inter-Group Multi-task Performance

Representative Task Analysis. We next consider the interplay between different task-groups. For efficiency, we consider multi-task training with representative tasks from each group – specifically VQA (G1), Retrieval Flickr30k (G2), Visual7W (G3), and NLVR^2 (G4). These were selected to maximize diversity in underlying image sources. We examine their relationships by jointly training all pairs and triplets of tasks under our multi-task training approach.

Table. 7.6 (left) shows the results of training each representative task pair. Each entry is the percent change from single-task performance for the row-task when jointly trained with the column-task. As such, the Avg. row (bottom) shows the mean impact each column-task has on other tasks, and likewise the Avg. column (right) shows the mean impact other tasks have on each row-task. For instance, we find that adding VQA (G1) benefits other tasks with an average improvement of +1.04%. Interestingly, adding NLVR^2 (G4) degrades other tasks on average (-1.36%) while making significant gains itself (+1.48%). This is primarily due to a -4.13% interaction with G2. Table 7.6 (right) shows all task triplets. Gains in the paired-experiments are not simply additive. In the pair-wise analysis, G3 gained +0.39% and +0.78% from G1 and G2 respectively. As before, G4 has some strong negative effects on other groups (-4.36% G2 with G3 & G4) but these effects can be regulated by other tasks (+0.49% G2 with G1 & G4).

Full Multi-task Results. We move to our main result – a single model trained on all 12 datasets. The results of this All-Tasks (`AT`) model are shown in Table 7.5 row `4`. This model outperforms independent single-task models trained on the same data (row `2`) for 11 out of 12 tasks and improve the average score by 2.05 points (69.08 vs. 67.03). We reiterate for emphasis, average performance *improves* by 2.05 points while *reducing* the number of parameters from over 3 billion to 270 million (a 12× reduction). This is also true for comparison with single-task models trained on full datasets (row `1`) by a similar margin of 1.83 points.

Our `AT` model also outperforms the Group-Task (`GT`) models (row `3`) despite having 4x fewer parameters (avg. 69.08 vs 68.72). This implies that despite their diversity, tasks across different groups can benefit from joint training.

We observed from the representative task analysis that G4 tends to have a negatively effect other groups during joint training. To validate this observation on all tasks, we train an All-Task model without G4 (row `5`). This model achieves higher avg. score of 67.56 for G1+G2+G3 compared to the full `AT` model's 67.38.

Table 7.7: Comparison to recent SOTA. For image retrieval (IR) COCO and Flickr we report R1 scores on the 1K test set.

Task	Split	SOTA	UNITER [202]		$\text{Ours}_{\texttt{AT}}$	$\text{Ours}_{\texttt{AT->ST}}$
			BERT_{B}	BERT_{L}	BERT_{B}	BERT_{B}
VQA	test-dev	-	72.27	**73.24**	72.57	73.15
VG QA	val	-	-	-	36.36	**36.64**
GQA	test-dev	60.00 [43]	-	-	60.12	**60.65**
IR COCO	test (R1)	**68.50** [46]	-	-	63.70	68.00
IR Flickr30k	test (R1)	-	71.50	**73.66**	63.52	67.90
RefCOCO	test	-	80.21	80.88	80.58	**81.20**
RefCOCO+	test	-	72.90	73.73	73.25	**74.22**
RefCOCOg	test	-	74.41	75.77	75.96	**76.35**
Visual 7W	test	72.53 [206]	-	-	82.75	**83.35**
GuessWhat	test	61.30 [109]	-	-	65.04	**65.69**
NLVR^2	testP	-	77.87	**79.50**	78.44	78.87
SNLI-VE	test	-	78.02	**78.98**	76.78	76.95
# params (# models)			602M (7 x 86M)	2.1B (7 x 303M)	**270M (1 x 270M)**	3B (12 x 250M)

Multi-Task Learning as Pretraining

For some applications, single task performance may be paramount and justify storing a task-specific model. Even then, fine-tuning from a multi-task trained model may allow the model to take advantage of the additional, diverse supervision captured during multi-task training. Following [16], we finetune our trained multi-task models (`GT` and `AT`) on each downstream task and show results in Table 7.5. Rows `6` and `7` show that finetuning from the all-task model (`AT`) outperforms finetuning from the group-task models (`GT`) with an average score of 69.51 vs. 68.81. For comparison with our multi-task models, these are finetuned on the cleaned datasets which are 11% smaller on average. To compare to prior work, we also finetune on the full dataset for individual tasks (Row `8`) and observe further improvements. Recall that our multi-task model was trained on cleaned data so there is no possibility of test leak here. These model outperform single-task models without multi-task pretraining (row `1`) by a large margin (70.23 vs. 67.25 avg. score).

Table 7.8: Comparison with other multi-task models. VQA score is on test-dev and the retrieval tasks on their respective 1K test split. For Flickr Grounding (FG) we report R1 on Flickr30K test.

	VQA	COCO Retrieval			Flickr Retrieval			FG
		R1	R5	R10	R1	R5	R10	R1
OmniNet [136]	55.76	-	-	-	-	-	-	-
HDC [137]	69.28	57.40	88.40	95.60	56.10	82.90	89.40	57.39
Ours	**72.70**	**65.16**	**91.00**	**96.20**	**65.06**	**88.66**	**93.52**	**64.61**

Comparison with Existing Work

In Table 7.7 we compare with existing state-of-the-art. We draw special comparison with the recent UNITER [202] architecture as it is similar to our base ViLBERT model. Like ViLBERT, UNITER is a general BERT-based vision-and-language architecture pretrained through self-supervised tasks and then finetuned for each downstream task. We show two UNITER columns corresponding to their underlying BERT model – either Base B or Large L. Our ViLBERT model uses the smaller $\text{BERT}_{\texttt{B}}$. Our single all-task model ($\text{Ours}_{\texttt{AT}}$) achieves competitive performance to state-of-the-art task-specific models. Our single-task finetuned models ($\text{Ours}_{\texttt{AT->ST}}$) surpass state-of-the-art on 7 out of 12 tasks.

Table 7.8 compares our method with other recently proposed multi-modal, multi-task learning approaches – OmniNet [136] and Hierarchical Dense Co-Attention (HDC) [137]. OmniNet is trained on part-of-speech tagging, image captioning, visual question answering, and video activity recognition, while HDC is trained on image caption retrieval, visual question answering, and visual grounding. We train a multi-task model on the same tasks and cleaned datasets used in HDC [137]. Flickr Grounding is a new task that we include for this comparison. Our multi-task model outperforms these approaches by a large margin.

7.2.4 Analysis and Ablation Study

Ablations on task token and training strategies. To verify our design choices, we perform ablations for different task token granularity and multi-task training strategies. The results are shown in Table 7.9. We report average group and overall average performance. Detailed breakdown for each task can be found in supplement.

For task tokens, our default setting is with a different task token per dataset (12 total,(Row `1`). We compare this with two ablations: one task token per output head (4 total, Row `2`) and no task tokens (Row `3`). We observe that task-specific tokens lead to better performance compared to head-based tokens (avg. 69.08 *vs.* 68.52) and no task tokens (avg. 69.08 *vs.* 68.53). This shows that task-aware feature embedding is useful even within the same output space; *e.g.* per-task tokens may help differentiate noun phrases and pointing questions in Referring Expression.

For multi-task training schedule, we compare our dynamic stop-and-go (`DSG`) (Row `3`) with Curriculum (Row `5`) and Anti-Curriculum (Row `6`) approaches. We consider convergence rate as a measure of task difficulty. For Curriculum, we first train tasks in G4 and then train all tasks together (easier $\longrightarrow$ harder). For Anti-Curriculum, we train G1 tasks first and then train on all tasks together (harder $\longrightarrow$ easier). Table 7.9 shows our dynamic stop-and-go training schedule outperforms anti-curriculum (avg. 68.53 *vs.* 67.98) and curriculum (avg. 68.53 *vs.* 67.24). Row `7` shows results of a 'vanilla', round-robin training scheme with no task tokens or training scheduling. The average score of vanilla multitask is close to anti-curriculum (67.92 *vs.* 67.98). Consistent with prior work [129], performance on harder tasks (G1) is worse compared to anti-curriculum. Our full training regime outperforms this significantly (avg. 69.08 *vs.* 67.92).

Behavior of Dynamic Stop-and-Go training. To characterize our dynamic stop-and-go training scheme, we visualize the dynamic training schedule in Fig. 7.6 (left) – bold lines indicate normal `go` training and thin lines are `stop` states when

Table 7.9: Ablations on our design choices and comparison to curriculum and anti-curriculum learning multi-task approaches.

		Task Token	Dynamic Stop-and-Go	G1	G2	G3	G4	All Tasks Average
	AT (our)							
1	token per dataset	✓	✓	**56.35**	**63.61**	**75.52**	**77.61**	**69.08**
2	token per head	✓	✓	55.95	61.48	75.35	77.37	68.52
3	w/o task token		✓	55.67	62.55	75.38	76.73	68.53
4	w/o DSG	✓		55.50	62.92	75.24	76.31	68.52
5	w/ curriculum			54.68	61.21	75.19	76.70	67.24
6	w/ anti-curriculum			55.82	59.58	73.69	75.94	67.98
7	vanilla multitask			54.09	61.45	75.28	76.71	67.92

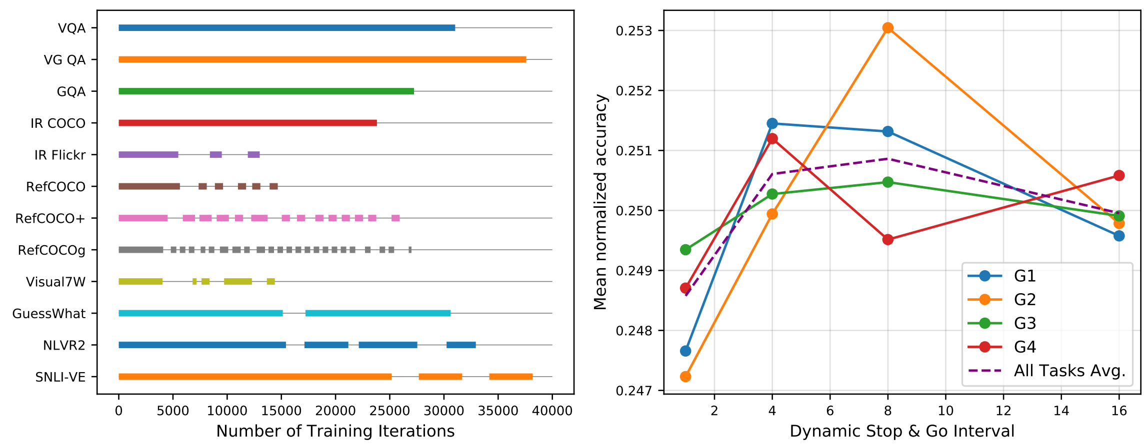


Figure 7.6: Left: Visualization of Dynamic stop-and-go during multi-task training. Solid line indicates in the go mode while thin line indicates stop mode. Right: Mean accuracy (normalized group-wise for easier comparison) for each group with different iter-gap Δ for Dynamic stop-and-go .

datasets receive sparser updates at a fixed iteration gap (every 4th iteration here). We see that smaller datasets quickly converge and enter stop state training early. As the base model drifts over time, they periodically return to full go state training to adjust. Interestingly, after some cycles of this, they enter the stop state and continue with only sparse updates for the rest of training.

Another aspect of dynamic stop-and-go training is the sparsity of updates in the stop state. Fig. 7.6 (right) shows the mean normalized accuracy for each group for multi-task models trained with different iteration gaps (Δ). We observe that raising Δ (*i.e.* updating more sparsely) improves performance initially but degrades for larger values. Absolute and per-task scores are provided in the supplement.

Multi-Task visual grounding consistency. Given the common shared base model, one question is whether multitask models exhibit more consistent visual groundings than independent task-specific models. For example, does a model that correctly answers "What color is the largest dog?" also correctly ground the referring expression "largest dog"? To assess this, we consider 1500 images from the RefCOCO/+ test sets that also have VQA annotations such that for each image I_i there are associated questions $\{q^{(i)}\}$ and referring expressions $\{r^{(i)}\}$. To measure the overlap in visual concepts between a question $q_j^{(i)}$ and reference $r_k^{(i)}$, we count overlapping nouns and adjectives (identified using a part-of-speech tagger) and denote this $d(q_j^{(i)}, r_k^{(i)})$. Armed with this notion of similarity, we consider each question-reference pair for each image (total 111,275 combinations) and compute a weighted accuracy. A pair is considered correct if the question was answered correctly and the referent was localized. Each pair is weighed by their overlap $d(q_j^{(i)}, r_k^{(i)})$. Note that if $q_j^{(i)}$ and $r_k^{(i)}$ do not have any common visual concept $(d(q_j^{(i)}, r_k^{(i)}))$, the correctness of this pair does not affect the overall metric.

We evaluate our Single-Task (`ST`), All-Task (`AT`), and finetuned from All-Task (`AT->ST`) models on the proposed metric. `AT` consistently outperforms `ST` (55.40 % *vs.* 58.30%) and `AT->ST` achieves the best performance (64.64%). This shows our model trained on multiple tasks achieve better visual grounding consistency across different tasks. Further analysis can be found in the supplement.

Regularizing effects of multi-task learning. We find multi-task training to have a regularizing effect on tasks which overfit when trained separately. In Fig. 7.7 we plot the training and validation curves for two tasks (SNLI-VE and Flickr Grounding) where single task training overfits quickly. On the other hand when trained in a multi-task setup with all other tasks, the validation score improves and there is no overfitting.

Qualitative examples. Figure 7.8 shows example outputs of our models. Due to space limitation, we provide extensive visualizations in the supplement.

7.2.5 Discussion

In this work, we develop a training regime and experimental setting for large-scale, multi-modal, multi-task learning. As one part of this, we introduce a novel task scheduling approach to help avoid over- or under-training tasks with differing sizes or difficulties. Using this framework, we explore the relationships between 12 vision-and-language datasets – our single multi-task model outperforms 12 single-task models. We find multi-task training can lead to significant gains over independent task training. Further, we show that multi-task learning is an effective pre-training task for training state-of-the-art single-task models.

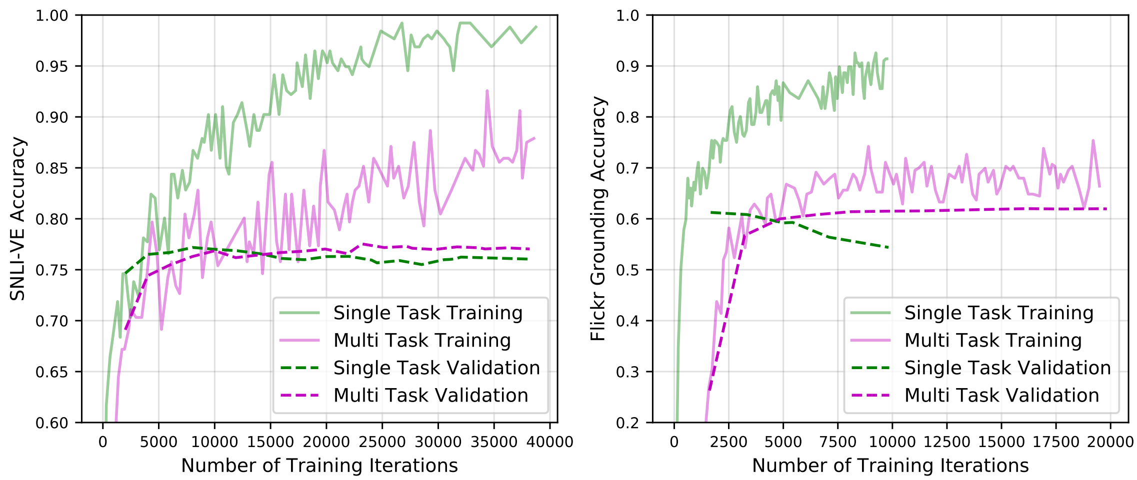


Figure 7.7: Multi-Task training acts as a regularizer.

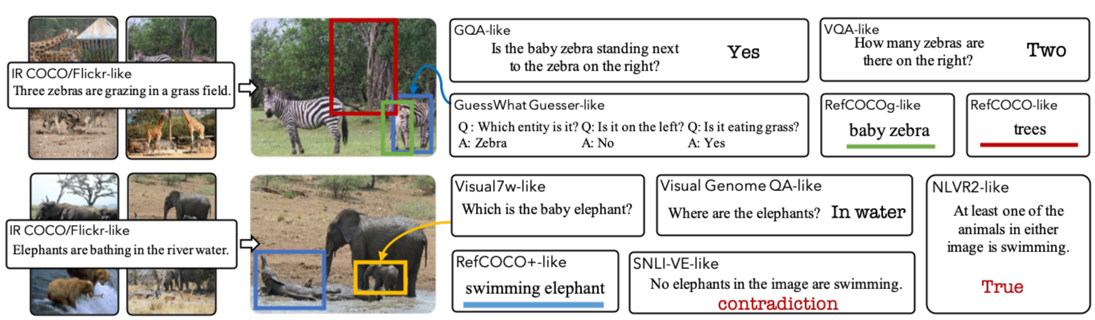


Figure 7.8: Our single model ($\text{Our}_{\texttt{AT}}$) can perform a multitude of V&L tasks: caption and image retrieval, question answering, grounding phrases, guessing image regions based on a dialog, verifying facts about a pair of images, natural language inferences from an image, etc. Here we show outputs of our model for a variety of inputs (that mimic tasks from the 12 datasets it has been trained on).

CHAPTER 8
CONCLUSION

In this thesis, we studied how inducing appropriate grounding in models improves multi-modal AI capabilities. Specifically, we walked through different approaches by different vision and language tasks. We first collected a large scale visual question answering dataset and provided various baselines to benchmark this task. To jointly reason about image and question, we then proposed a co-attention mechanism that can learn fine-grained grounding to answer the questions.

Next, we addressed the model designs for the sequence generation problem in image captioning. We proposed an adaptive attention encoder-decoder framework that decided how much new information the decoder wants to get from the image as opposed to relying on the decoder itself when generating the next word. Even with advanced attention mechanism, the model was still lack of visual grounding – hallucinating objects that do not appear in the image. We thus designed a novel framework that can directly utilize the output of the object detector to generate captions. This approach essentially serves as a 'bridge' between detection and captioning.

Third, we explored novel training paradigms to learn better visual grounding for visual dialog. Compared to VQA and image captioning, visual dialog requires both the ability of 'understanding' and 'generation' in the contexts. We studied both sides of the visual dialog agents – questioner and answerer. For answerer which answers visual questions in dialog, we proposed a novel discriminant perceptual loss that transfers knowledge from a discriminative model to a generative model. For questioner, we considered an image guessing game as a test-bed for balancing task performance and language drift. Our Dialog without Dialog (DwD) task requires agents to generalize from single round visual question generation with full supervision to a multi-round dialog-based image guessing game without direct language supervision. We further proposed a novel training paradigm which first learns "how to speak" by pretraining with a conditional variational auto-encoders and then learns "what to speak" by fine-tuning with task-specific rewards with discrete latent space. I believe this is a solid step to transfer interpretable and grounded language for goal-oriented tasks.

Finally, we studied general multi-modal AI models that can learn visual groundings from massive meta-data on the internet and handle many vision and language tasks at the same time. We thus first explored how to pretraining task-agnostic visiolinguistic representations which is useful for multiple vision and language tasks. Our work represents a shift away from learning groundings between vision and language only as part of task training and towards treating visual grounding as a pretrainable and transferable capability. We further explored multi-task vision and language representation learning. Our results not only show that a single model can perform all these tasks, but also that joint training can lead to improvements on task metrics compared to single-task training with the same architecture.

I believe that the research thrust explored in this thesis has value for the long-term process in AI. The problem of learning *grounding* – the connection between different modalities – is the core to improve multi-modal AI capabilities. From the basic 'late-fusion' algorithms for VQA to the recent multi-task vision and language representation learning that can handle 12 tasks simultaneously. We are witnessing a great process in the vision and language communities. I am excited about the future directions of progress in these areas.

Appendices

APPENDIX A
APPENDIX FOR KNOWING WHEN TO LOOK

A.1 COCO Categories Mapping List for Weakly-Supervised Localization

We first use WordNetLemmatizer from NLTK[1] to lemmatize each word of the caption. Then we map "people", "woman", "women", "boy", "girl", "man", "men", "player","baby" to COCO "**person**" category; "plane", "jetliner", "jet" to COCO "**airplane**" category; "bike" to COCO "**bicycle**" category; "taxi" to COCO "**car**" category. We also change the COCO category name from "**dining table**" to "table" while evaluation. For the rest categories, we keep their original names. We show the visualization of bounding box in Fig. A.1

Figure A.1: Image attention visualization of word "of" on several images. For each image pair, left: output of spatial attention model (no visual sentinel), right: output of our adaptive attention model (with visual sentinel).

A.2 Adaptive attention across different datasets

We show the visual grounding probability for the same words across COCO and Flickr30 datasets in Table A.1. Trends are generally similar between the two datasets.

[1]http://www.nltk.org/

Table A.1: Visual grounding probabilities of the same word on COCO and Flickr30K datasets.

Dataset	giant	people	bus	metal	umbrella	lake	yellow	on	the	UNK	full	says	of	up
COCO	0.921	0.917	0.868	0.856	0.843	0.837	0.827	0.713	0.685	0.654	0.622	0.612	0.541	0.527
Flickr30K	0.947	0.856	0.914	0.889	0.830	0.791	0.869	0.702	0.726	0.803	0.445	0.586	0.510	0.652

To quantify this, we sort all common words between the two datasets by their visual grounding probabilities from both datasets. The rank correlation is 0.483. Words like "sheep" and "railing" have high visual grounding in COCO but not in Flickr30K, while "hair" and "run" are the reverse. Apart from different distributions of visual entities present in the dataset, some differences may be a consequence of different amounts of training data. Will add this to the paper.

A.3 More Visualization of Attention

Fig. A.2 and Fig. A.3 show additional visualization of spatial and temporal attention.

A.3.1 Visualization of Weakly Supervised Localization

Fig. A.4 shows the visualization of weakly supervised localization.

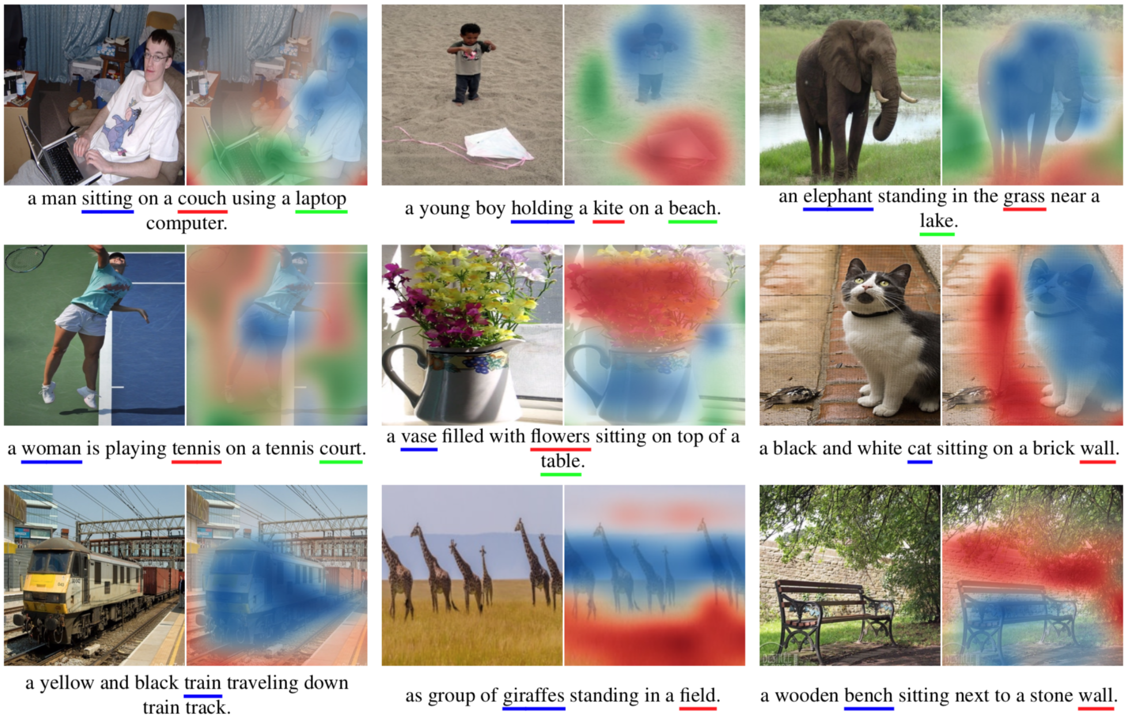


Figure A.2: Visualization of generated captions and image attention maps on the COCO dataset. Different colors show a correspondence between attended regions and underlined words.

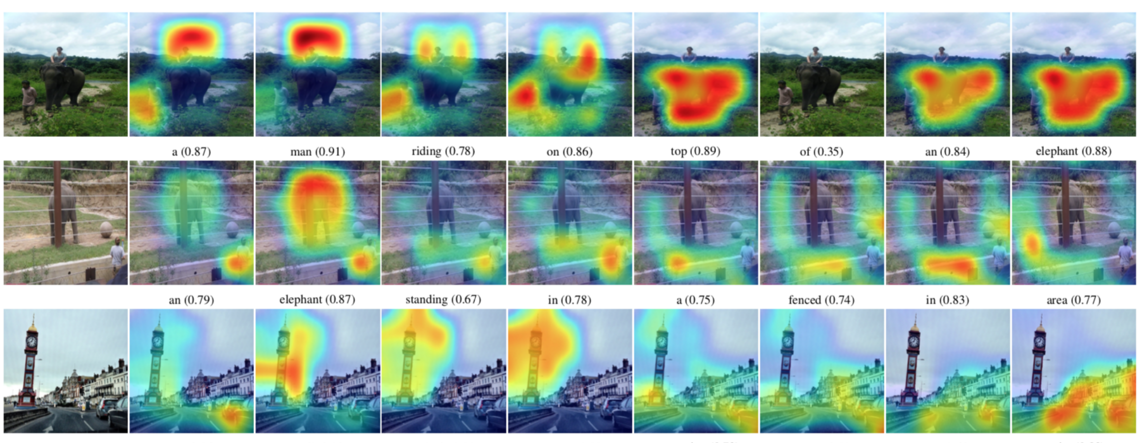


Figure A.3: Example of generated caption, spatial attention and visual grounding probability.

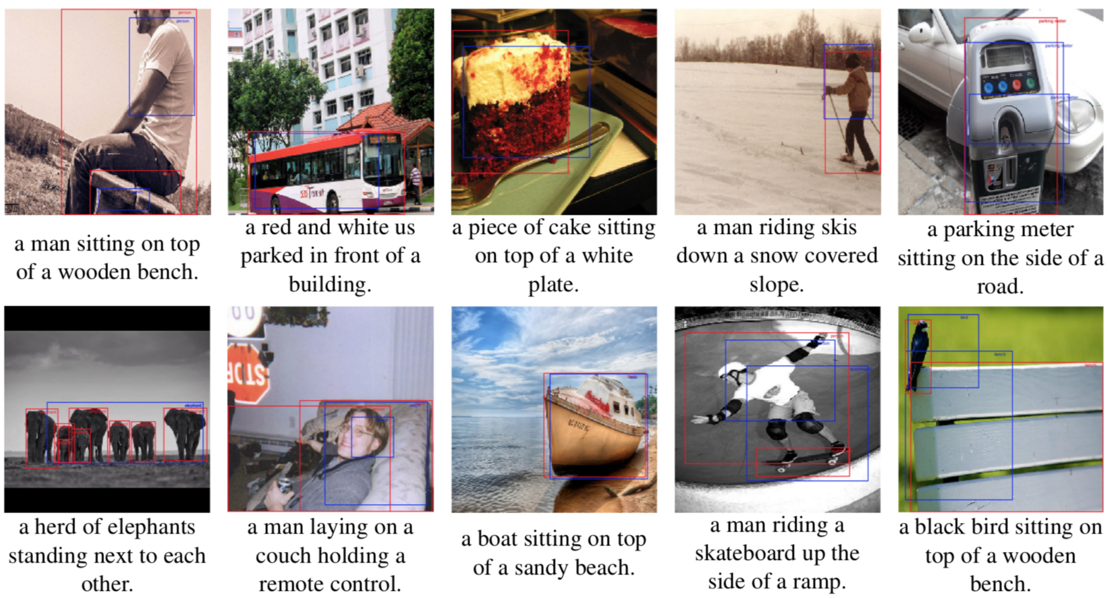


Figure A.4: Visualization of generated captions and weakly supervised localization result. Red bounding box is the ground truth annotation, blue bounding box is the predicted location using spatial attention map.

APPENDIX B

APPENDIX FOR NEURAL BABY TALK

The COCO [5] dataset does not have bounding box annotations associated with specific phrases or entities in the caption. We use category level detection annotations and create a category mapping list that maps the object categories like <Person> to a list of potential fine-grained labels like ["child", "man", "baker",...]. We first use the Stanford lemmatization toolbox [161] to get the base form of the entity words in the caption. For each category class, we retrieve the top 200 similar words in the WordVec [207] space. We then manually verify each word in the list, resulting in 413 fine-grained classes. A complete list of the fine-grained class for each object category can be found in Table B.1 and Table B.3.

Table B.1: COCO category mapping list for visual words.

Object category	**Fine-grained class**
<person>	person, girl, boy, man, woman, kid, child, chef, baker, people, adult, rider, children, baby, worker, passenger, sister, biker, policeman, officer, lady, cowboy, bride, groom, male, female, guy, traveler, mother, father, gentleman, pitcher, player, skier, snowboarder, skater, skateboarder, foreigner, caller, offender, coworker, trespasser, patient, politician, soldier, serviceman, walker, drinker, doctor, bicyclist, thief, buyer, teenager, student, camper, driver, solider, hunter, shopper, villager, cop, grandchild
<bicycle>	bicycle, bike, unicycle, minibike, trike
<car>	car, automobile, van, minivan, sedan, suv, hatchback, cab, jeep, coupe, taxicab, limo, taxi
<motorcycle>	motorcycle, scooter, motor bike, motor cycle, motorbike, moped
<airplane>	airplane, jetliner, plane, air plane, monoplane, aircraft, jet, airbus, biplane, seaplane bus, minibus, trolley
<bus>	bus, minibus, schoolbus, trolley
<train>	train, locomotive, tramway, caboose
<truck>	truck, pickup, lorry, hauler, firetruck
<boat>	boat, ship, liner, sailboat, motorboat, dinghy, powerboat, speedboat, canoe, skiff, yacht, kayak, catamaran, pontoon, houseboat, vessel, rowboat, trawler, ferryboat, watercraft, tugboat, schooner, barge, ferry, sailboard, paddleboat, lifeboat, freighter, steamboat, riverboat, surfboard, battleship, steamship
<traffic light>	traffic light, street light, traffic signal, stop light, streetlight, stoplight
<fire hydrant>	fire hydrant, hydrant
<stop sign>	stop sign, street sign
<parking meter>	parking meter
<bench>	bench, pew
<cat>	cat, kitten, feline, tabby
<dog>	dog, puppy, beagle, pup, chihuahua, schnauzer, dachshund, rottweiler, canine, pitbull, collie, pug, terrier, poodle, labrador, doggie, doberman, mutt, doggy, spaniel, bulldog, sheepdog, weimaraner, corgi, cocker, greyhound, retriever, brindle, hound, whippet, husky
<horse>	horse, colt, pony, racehorse, stallion, equine, mare, foal, palomino, mustang, clydesdale, bronc, bronco
<sheep>	sheep, lamb, goat, ram, cattle, ewe
<cow>	cow, cattle, oxen, ox, calf, ewe, holstein, heifer, buffalo, bull, zebu, bison
<elephant>	elephant
<bear>	bear, panda
<zebra>	zebra
<giraffe>	giraffe
<backpack>	backpack, knapsack
<umbrella>	umbrella
<handbag>	handbag, handbag, wallet, purse, briefcase
<tie>	tie
<suitcase>	suitcase, suit case, luggage
<frisbee>	frisbee
<skis>	skis, ski
<snowboard>	snowboard

Table B.2: COCO category mapping list for visual words (continued).

Table B.3: COCO category mapping list for visual words (continued).

Object category	**Fine-grained class**
<sports ball>	sports ball, baseball, ball, football, soccer, basketball, softball, volleyball, pinball, fastball, racquetball
<kite>	kite
<baseball bat>	baseball bat
<baseball glove>	baseball glove
<skateboard>	skateboard
<surfboard>	surfboard, longboard, skimboard, shortboard, wakeboard
<tennis racket>	tennis racket
<bottle>	bottle
<wine glass>	wine glass
<cup>	cup
<fork>	fork
<knife>	knife, pocketknife, knive
<spoon>	spoon
<bowl>	bowl, container, plate
<banana>	banana
<apple>	apple
<sandwich>	sandwich, burger, sub, cheeseburger, hamburger
<orange>	orange, lemons
<broccoli>	broccoli
<carrot>	carrot
<hot dog>	hot dog
<pizza>	pizza
<donut>	donut, doughnut, bagel
<cake>	cake, cheesecake, cupcake, shortcake, coffeecake, pancake
<bird>	bird, ostrich, owl, seagull, goose, duck, parakeet, falcon, robin, pelican, waterfowl, heron, hummingbird, mallard, finch, pigeon, sparrow, seabird, osprey, blackbird, fowl, shorebird, woodpecker, egret, chickadee, quail, bluebird, kingfisher, buzzard, willet, gull, swan, bluejay, flamingo, cormorant, parrot, loon, gosling, waterbird, pheasant, rooster, sandpiper, crow, raven, turkey, oriole, cowbird, warbler, magpie, peacock, cockatiel, lorikeet, puffin, vulture, condor, macaw, peafowl, cockatoo, songbird
<chair>	chair, seat, recliner, stool
<couch>	couch, sofa, recliner, futon, loveseat, settee, chesterfield
<potted plant>	potted plant, houseplant
<bed>	bed
<dining table>	dining table, table
<toilet>	toilet, urinal, commode, lavatory, potty
<tv>	tv, monitor, televison, television
<laptop>	laptop, computer, notebook, netbook, lenovo, macbook
<mouse>	mouse
<remote>	remote
<keyboard>	keyboard
<cell phone>	cell phone, mobile phone, phone, cellphone, cellphone, telephone, phon, smartphone, iPhone
<sink>	sink

Object category	**Fine-grained class**
<refrigerator>	refrigerator, fridge, refrigerator, fridge, freezer, refridgerator, frig
<book>	book
<clock>	clock
<vase>	vase
<scissors>	scissors
<teddy bear>	teddy bear, teddybear
<hair drier>	hair drier, hairdryer
<toothbrush>	toothbrush

APPENDIX C
APPENDIX FOR DIALOG WITHOUT DIALOG

C.1 Additional Results

Experiments in the main paper considered dialog performance after the first round (top of Table 1) and at the final round of dialog (either 5 or 9 depending on pool size). This does not give much sense for how dialog performance increases over rounds of dialog, so we report Q-Bots guessing game performance at each round of dialog in Fig. C.1. For all fine-tuned models performance goes up over multiple rounds of dialog, though some models benefit more than others. Stage 1 models decrease in performance after round 1 because it is too far from the training data such models have been exposed to.

C.2 Mechanical Turk Studies

In the experiments section we described two studies where we asked humans to compare questions.

In the relevance study turkers were presented with the interface depicted in Fig. C.2. It asked them to compare questions based on their relevance to any image in the image pool. The question with higher relevance should have been picked even if the question was not very grammatical. All model pairs were evaluated for each pool of images. The questions were presented in a random order, though the Equally relevant option was always last.

In the fluency study (Fig. C.3) turkers were presented with the same pairs of questions as in the relevance interface but they were not given image pools with which to associate the questions. We asked them to compare questions based on how

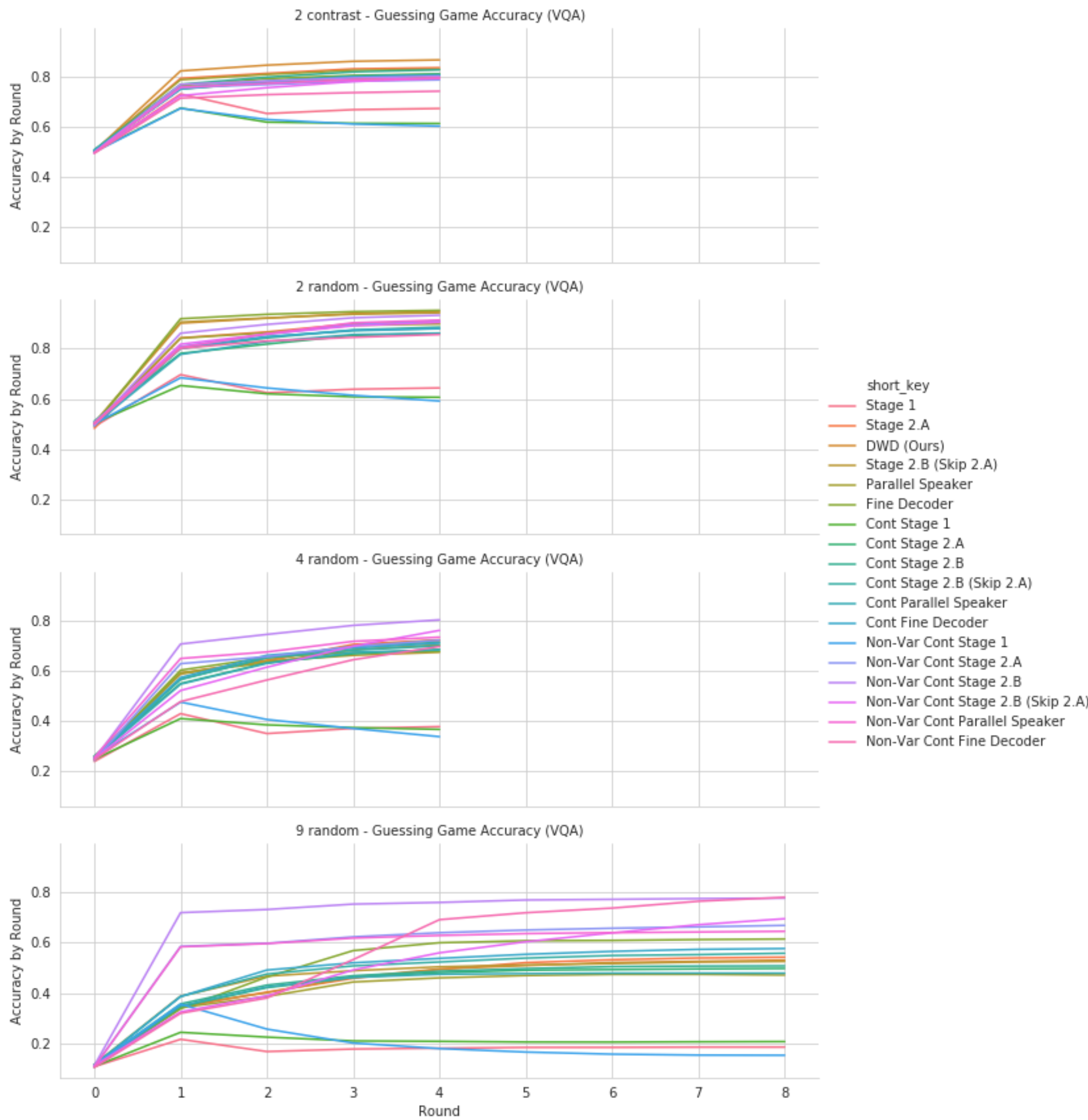


Figure C.1: Task performance (guessing game accuracy) over rounds of dialog. Performance increases over rounds for all models except the Stage 1 models.

well they could be understood. As in the relevance study questions were presented in a random order.

In the figure 4, we display the interface which was used to pair up the QBot with a human in real time. The QBot asks a question in order to guess the target image and a human answers the question by looking at the target image. This sequence of question/answer starts with a random guess from QBot and goes on for 4 Rounds.

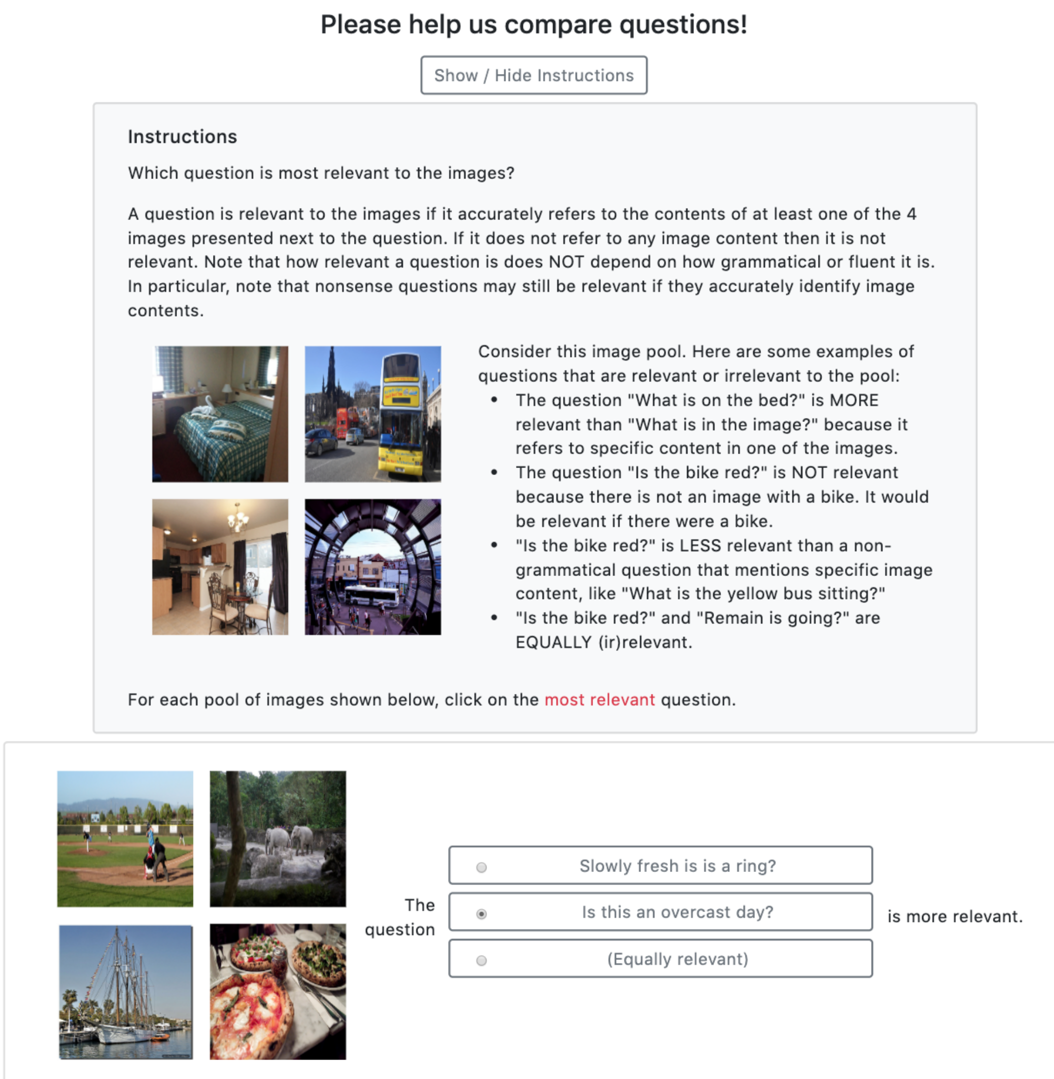

Please help us compare questions!
Show / Hide Instructions
Instructions
Which question is most relevant to the images?
A question is relevant to the images if it accurately refers to the contents of at least one of the 4 images presented next to the question. If it does not refer to any image content then it is not relevant. Note that how relevant a question is does NOT depend on how grammatical or fluent it is. In particular, note that nonsense questions may still be relevant if they accurately identify image contents.
Consider this image pool. Here are some examples of questions that are relevant or irrelevant to the pool:
• The question "What is on the bed?" is MORE relevant than "What is in the image?" because it refers to specific content in one of the images.
• The question "Is the bike red?" is NOT relevant because there is not an image with a bike. It would be relevant if there were a bike.
• "Is the bike red?" is LESS relevant than a non-grammatical question that mentions specific image content, like "What is the yellow bus sitting?"
• "Is the bike red?" and "Remain is going?" are EQUALLY (ir)relevant.
For each pool of images shown below, click on the most relevant question.
The question
Slowly fresh is is a ring?
Is this an overcast day?
(Equally relevant)
is more relevant.

Figure C.2

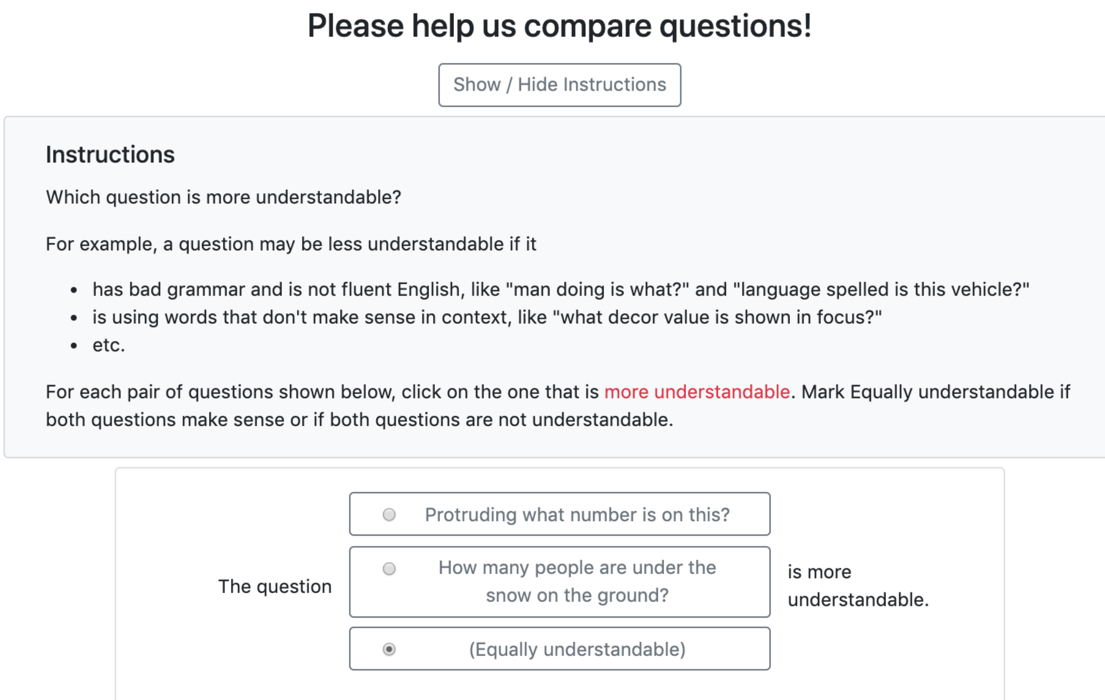


Figure C.3

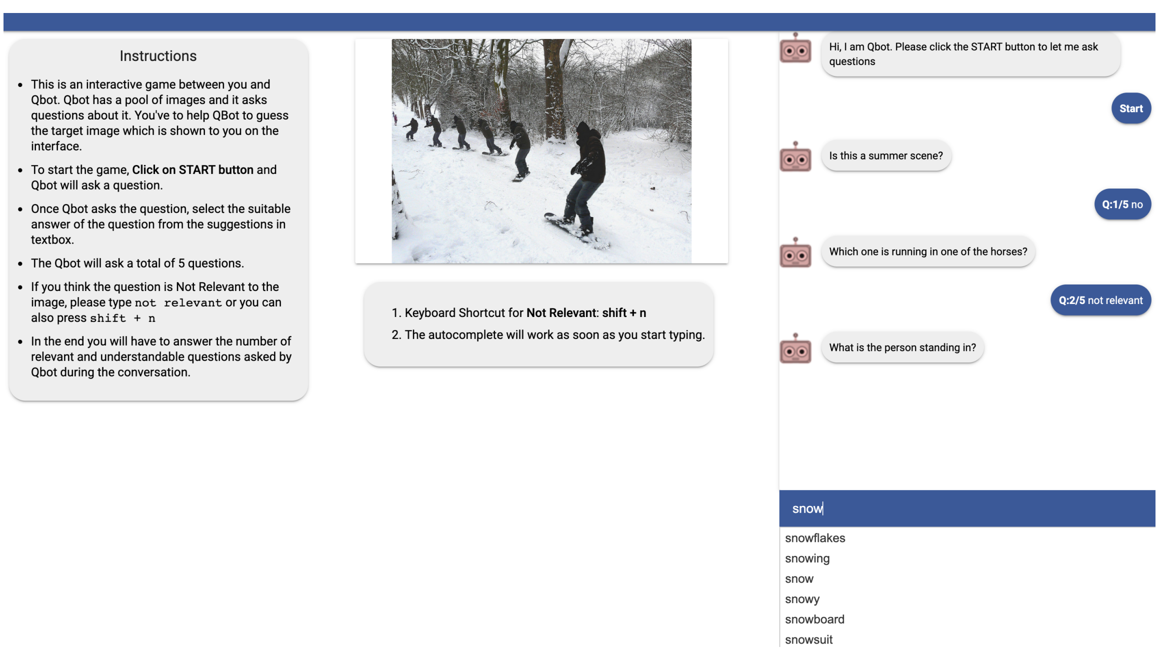


Figure C.4

APPENDIX D

APPENDIX FOR MULTI-TASK VISION AND LANGUAGE REPRESENTATION LEARNING

we first show the full details of the cleaned dataset in Sec. D.1. We further discuss the modifications in pretraining, show our multi-task model architecture and describe the implementation details in Sec. D.2, Sec. D.3 and Sec. D.4 respectively. The rest of the section provides extensive experiment results to fully analyze our proposed model.

D.1 Datasets

Table D.1 shows the number of images in the train+val and test sets before and after cleaning. Our cleaning process removes 13.02% of the total number of images on average. It is important to note that here we show the number of images per dataset and not number of actual training samples. Different tasks have different number of training samples for each image. For details on training samples please refer Table D.2. We collect the union of all dataset test sets and remove any occurrence of these images from all training and validation sets; in this way we arrive at the *Clean* training and validation sets. With this strategy, the test sets of the original datasets are not modified in any way.

D.2 Improvements over ViLBERT Pretraining

In this section, we discuss in detail the modification we made to the base ViLBERT pretraining approach.

Masked prediction with mislaigned pairs. In the original ViLBERT pretraining procedure, the model observes an image and caption as inputs. The caption is either

Table D.1: Number of images in the train+val and test sets before and after cleaning. We use the training part of the cleaned dataset in the multi-task experiments. Note that this is not the number of training samples but the number of images in the dataset.

	Train+Val	Test	Cleaned Train+Val	% Removed
[A] VQA2.0[101]	123,287	81,434	98,861	19.81
[B] VG QA[120]	108,249	-	92,147	14.87
[C] GQA[197]	82,374	2,987	69,868	15.18
[D] COCO Retrieval[49]	118,287	5,000	99,435	15.93
[E] Flickr30k Retrieval [160]	30,014	1,000	29,077	3.12
[F] RefCOCO[193]	18,494	1,500	14,481	21.69
[F] RefCOCO+[193]	18,492	1,500	14,479	21.70
[H] RefCOCOG [198]	23,199	2,600	17,903	22.82
[I] Visual 7W [199]	17,953	7,780	16,415	8.56
[J] GuessWhat[109]	56,638	9,899	51,291	9.44
[K] SNLI-VE[201]	30,783	1,000	29,808	3.16
[L] NLVR^2 [200]	95,522	8,056	95,522	0
Average	-	-	-	13.02

obtained from the paired caption (with $p = 0.5$) or a randomly sampled misaligned caption from the dataset. The *multi-modal alignment prediction* task, which predicts whether the image and caption are aligned, is crucial for image retrieval tasks [42, 43, 46]. Recent work [47] has questioned the necessity of the *multi-modal alignment prediction* task and observed better performance on non-image retrieval tasks without this pretraining objective. Similar observations are also found in the natural language understanding tasks [208, 118, 209, 210]. Digging further into this, we find that both the alignment and prediction tasks are typically done together. For misaligned image-caption pairs, this amounts to forcing the model to predict missing image or text regions based on incorrect paired data! We find the model will learn worse context representations in this setup. Instead of removing the *multi-modal alignment prediction* task, we only perform the *mask multi-modal modelling* task on **aligned image-caption pairs**. This will effectively remove the noise introduced by negative samples.

Masking overlapping regions. Different from words embedding in the caption, visual feature embeddings (extracted from a pretrained Faster-RCNN [156]) have a lot of repetitions due to overlapped image regions. To avoid visual clue leakage from the visual embedding of other elements, VL-BERT [47] sets the pixels laid in the masked RoI to zeros before applying Faster R-CNN. However, overlapped image patches with boundary information may still leak the visual clues for the masked RoI. We mask the overlapped image regions in a more aggressive manner – any visual embedding that overlaps a masked region by 40% IOU or more is also masked. We observe significant improvements over the ViLBERT model as shown in Table D.3 when comparing column ViLBERT with Ours_{ST}.

D.3 Model Architecture

Fig. D.1 shows the architecture of the our model for V&L multi-task learning, which is described in Sec. 7.2.2. We use ViLBERT as our base model shared across different tasks. For the task-specific heads, our model jointly train with four different task group – Vocab-Based VQA; Image Retrieval, Refer Expression and Multimodal Verification.

D.4 Implementation Details

Image features are extracted from a ResNeXT-152 Faster-RCNN model trained on Visual Genome(VG) with attribute loss. We use AdamW optimizer and warmup linear schedule. Hyperparameters like learning rate and batch sizes used for each task are listed in Table D.2. We also report the number of training samples used in various settings in our experiments.

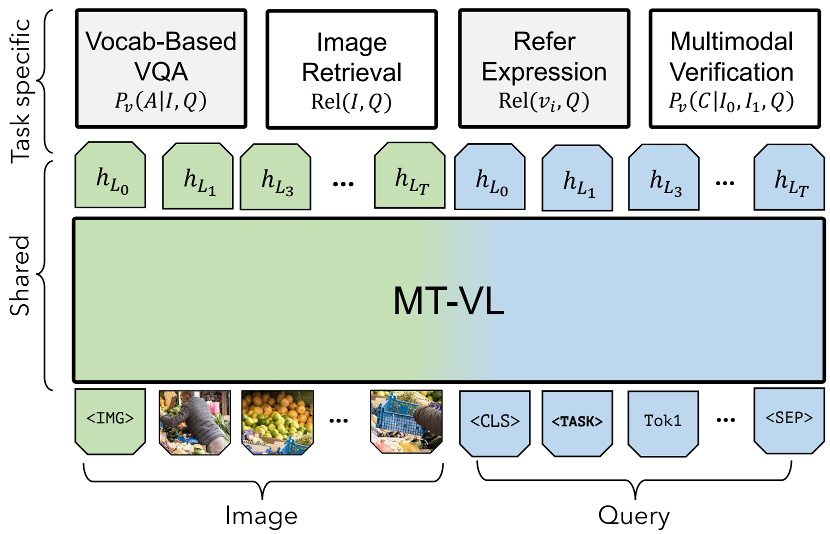


Figure D.1: Architecture of the our model for V&L multi-task learning. We augment the input query with a task token to learn the task-aware feature embedding.

D.5 Multi-Task Training

To further illustrate the multi-task training process, in Fig. D.2 we show the training curves for single-task *vs*. multi-task for all the 12 tasks in our setup. Green lines show single-task training and blue lines show multi-task training. Since we train the model with maximum iterations across different datasets for multi-task training, for some smaller datasets (*e.g*. RefCOCO, Visual7W *etc*.), the number of iterations for single task is much smaller compared to the multi-task setting. By comparing the training curves of single-tasks and multi-tasks, we can see that most of the tasks have similar training curves. However, the tasks in the vocab-based VQA group benefit from the multi-task training with faster convergence within first 10000 iterations.

D.6 Comparison with other SOTA

Table D.3 shows the detailed comparison of Ours$_{\texttt{ST}}$ (also shown in Table 7.5, line 1) and Ours$_{\texttt{AT->ST}}$ (also shown in Table 7.5, line 8) with the recent SOTA approaches, inlcuding ViLBERT [42], Unicoder-VL [46], VisualBERT [44], LXMERT [43] and

Table D.2: Training details including sample sizes, testing metric and hyperparameters for single task and multi-task training.

	Samples				Hyperparams	
	Full Train	Cleaned Train	Test	Metric	BS	LR
[A] VQA2.0[101]	655,111	542,104	447,793	VQA Accuracy	128	4e-5
[B] VG QA[120]	1,437,931	1,294,255	5,000	VQA Accuracy	128	4e-5
[C] GQA[197]	1,072,062	962,928	12,578	VQA Accuracy	128	4e-5
[D] IR COCO [49]	566,747	487,600	1,000	Recall @ 1, 5, 10	128	2e-5
[E] IR Flickr30k [160]	145,000	140,485	1,000	Recall @ 1, 5, 10	128	2e-5
[F] RefCOCO[193]	120,624	96,221	10,752	Accuracy	256	2e-5
[F] RefCOCO+[193]	120,191	95,852	10,615	Accuracy	256	2e-5
[H] RefCOCOG [198]	80,512	65,514	9,602	Accuracy	256	2e-5
[I] Visual 7W [199]	93,813	93,813	57,265	Accuracy	256	2e-5
[J] GuessWhat[109]	113,221	100,398	23,785	Accuracy	64	2e-5
[K] NLVR^2 [200]	86,373	86,373	6,967	Accuracy	64	2e-5
[L] SNLI-VE[201]	529,527	512,396	17,901	Accuracy	256	2e-5
Total	5,021,112	4,477,939	604,258	-	-	-

UNITER [202]. Most of the recent proposed methods follows the pretrain-then-finetune scheme, usually pretraining on out-of-domain data or in-domain data. The out-of-domain data contains Conceptual Caption Dataset (CC) [38] and SBU dataset [211] while in-domain data contains COCO [49] and Visual Genome [120]. Pre-training on the in-domain datasets usually leads to better downstream performance, since there is less domain transfer from pretraining to finetuning. Similar to ViLBERT, we pretrain our model on CC, which is different from VLBERT (CC + Wiki Corpus), VisualBERT (CC + COCO), LXMERT (COCO + VG) and UNITER (CC + SUB + COCO + VG). We achieve comparable performance with less pretrained data. The table also shows the improvements in Sec D.2 result in better performance for ViLBERT model.

D.7 Full Breakdown of Ablation Study

Table D.4 shows the full breakdown of paper's ablation results and Fig. 7.6 per task in the main paper. RC refers to Retrieval COCO and RF refers to Retrieval Flickr30k.

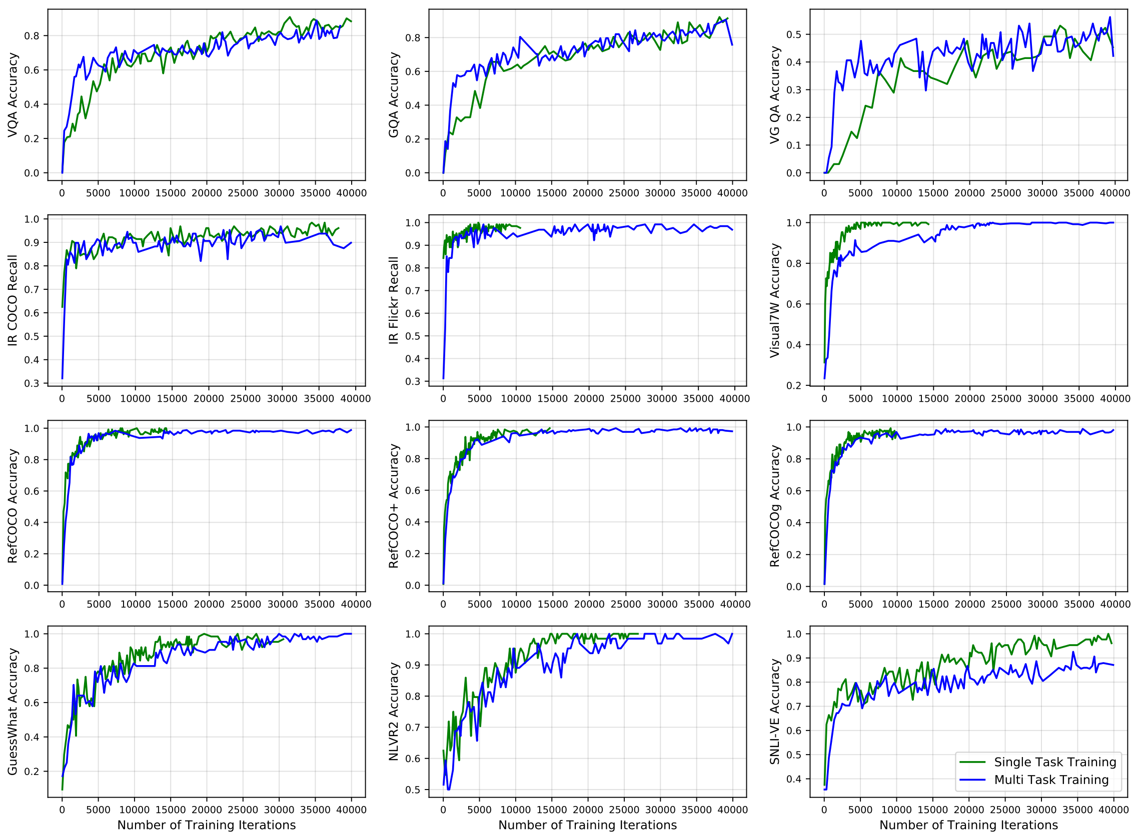


Figure D.2: Training curves on *train* set for Ours$_{\texttt{ST}}$ (Table 7.5 `Row 2`) vs Ours$_{\texttt{AT}}$ (Table 7.5 `Row 4`) models for all the 12 tasks in our experiments. Green lines show single-task training(Ours$_{\texttt{ST}}$) and blue lines show multi-task training(Ours$_{\texttt{AT}}$). Note that all these training are with the *Clean V&L* setup. We can observe that for some of the tasks the training for Ours$_{\texttt{ST}}$ are shorter as they have fewer number of iterations when trained alone. Please refer to Sec. D.5 for more details.

VQA and GQA are evaluated on `test-dev` splits. Retrieval COCO and Flickr30k are evaluated on their respective 1K test split. NLVR^2 is evaluated on `testP` split. All other datasets are evaluated on their respective test splits. Table D.5 shows the full scores for each task for different `DSG` iteration gap (Δ).

D.8 Multi-task visual grounding consistency

In Sec. 7.2.4, we propose the multi-task visual grounding consistency. We explain the proposed metric in more details. Given N images with RefCOCO/+ refer expression and VQA questions, we want to test that whether multi-task models exhibit

Table D.3: Comparison of $\text{Ours}_{\texttt{ST}}$ (Table. 7.5 `Row 1`) and $\text{Ours}_{\texttt{AT->ST}}$ (Table. 7.5 `Row 8`) models on full dataset with other SOTA methods. Results for RefCOCO and RefCOCO+ are reported on the full test split (testA + testB). Refer to Sec D.6 for more details.

Tasks		SOTA	ViLBERT	VLBERT	Unicoder-VL	VisualBERT	LXMERT	UNITER BASE	UNITER LARGE	$\text{Ours}_{\texttt{ST}}$	$\text{Ours}_{\texttt{AT->ST}}$
	Pretraining Data		CC	CC + Wiki Corpus	CC	CC + COCO	COCO + VG	CC+SUB+COCO+VG		CC	CC
VQA	test-dev	70.63	70.55	70.50	-	70.80	72.42	72.27	**73.24**	71.82	73.15
VG QA	val	-	-	-	-	-	-	-	-	34.38	**36.64**
GQA	test-dev	-	-	-	-	-	60.00	-	-	58.19	**60.65**
IR COCO	R1	61.60	-	-	**68.50**	-	-	-	-	65.28	68.00
	R5	89.6	-	-	**92.70**	-	-	-	-	91.02	92.38
	R10	95.2	-	-	**96.90**	-	-	-	-	96.18	96.52
IR Flickr	R1	48.60	58.20	-	68.30	-	-	71.50	**73.66**	61.14	67.90
	R5	77.70	84.90	-	90.30	-	-	91.16	**93.06**	87.16	89.60
	R10	85.20	91.52	-	94.60	-	-	95.20	**95.98**	92.30	94.18
Visual 7W	test	72.53	-	-	-	-	-	-	-	80.51	**83.35**
Ref-COCO	test	77.12	-	-	-	-	-	80.48	80.88	78.63	**81.20**
Ref-COCO+	test	67.17	70.93	69.47	-	-	-	73.26	73.73	71.11	**74.22**
Ref-COCOg	test	69.46	-	-	-	-	-	74.51	75.77	72.24	**76.35**
GuessWhat	test	61.30	-	-	-	-	-	-	-	62.81	**65.69**
NLVR^2	test-P	53.50	-	-	-	67.00	74.50	77.87	**79.50**	74.25	78.87
SNLI-VE	test	71.16	-	-	-	-	-	78.02	**78.98**	76.72	76.95

	VQA	VG QA	GQA	Mean G1	RC R@1	RC R@5	RC R@10	RF R@1	RF R@5	RF R@10	Mean G2 (R1)	RefCOCO	RefCOCO+	RefCOCOG	Visual 7W	GuessWhat	Mean G3	NLVR^2	SNLI-VE	Mean G4	MT Score
token per dataset	**72.57**	**36.36**	60.12	**56.35**	**63.70**	**90.84**	**96.16**	**63.52**	**87.48**	**93.16**	**63.61**	80.58	73.25	**75.96**	82.75	**65.04**	**75.52**	**78.44**	76.78	**77.61**	**69.08**
token per head	72.11	35.84	59.91	55.95	60.66	88.96	94.86	62.30	86.20	92.00	61.48	**80.67**	73.10	75.82	**82.92**	64.24	75.35	77.65	**77.08**	77.37	68.52
w/o task token	72.00	35.09	59.92	55.67	63.16	90.48	95.44	61.94	86.96	92.88	62.55	80.32	73.04	75.94	82.72	64.89	75.38	76.99	76.46	76.73	68.53
w/o `DSG`	71.99	35.59	58.93	55.50	62.54	90.08	95.42	63.30	86.98	92.86	62.92	79.99	73.09	75.94	82.68	64.52	75.24	77.37	76.31	76.84	68.52
w/ curriculum	70.59	35.54	57.91	54.68	61.14	89.74	95.04	61.28	86.58	92.56	61.21	80.11	73.35	75.62	82.38	64.51	75.19	77.20	76.19	76.69	67.98
w/ anti-curriculum	71.53	35.54	**60.39**	55.82	61.04	88.78	94.96	58.12	84.66	90.84	59.58	78.99	71.34	74.24	80.80	63.08	73.69	76.14	75.74	75.94	67.24
vanilla multitask	70.39	33.31	58.57	54.09	61.50	89.72	95.42	61.40	87.04	92.74	61.45	80.42	**73.51**	75.53	82.48	64.50	75.28	77.09	76.34	76.71	67.92

Table D.4: Full per task accuracy for the different ablation studies. RC is Retrieval COCO and RF is Retrieval Flickr30k. Mean of G2 is taken over the Recall@1 scores. We can see that with task token per dataset and `DSG` achieve the best performance.

more consistent visual groundings than independent task-specific models. For each image I_i, there are associated VQA question $\{q^{(i)}\}$ and referring expression $\{r^{(i)}\}$. To measure the overlap in visual concepts between a question $q_j^{(i)}$ and reference $r_k^{(i)}$, we count the the number of overlapped noun / adj as $d(q_j^{(i)}, r_k^{(i)})$, the multi-task visaul grounding consistency can be calculated as:

$$\texttt{MT-VGC} = \frac{\sum_{k=0}^{N} |\sum_j \sum_k d(q_j^{(i)}, r_k^{(i)}) \mathbb{1}_{\{y(q_j^{(i)})=1 \& y(r_k^{(i)})=1\}}|}{\sum_{i=0}^{N} |\sum_j \sum_k d(q_j^{(i)}, r_k^{(i)}) \mathbb{1}|} \tag{D.1}$$

	VQA	VG QA	GQA	Mean G1	RC R@1	RC R@5	RC R@10	RF R@1	RF R@5	RF R@10	Mean G2 (R1)	RefCOCO	RefCOCO+	RefCOCOG	Visual 7W	GuessWhat	Mean G3	NLVR^2	SNLI-VE	Mean G4	MT Score
DSG $\Delta 1$	71.99	35.59	58.93	55.50	62.54	90.08	95.42	63.30	86.98	92.86	62.92	79.99	73.09	75.94	82.68	64.52	75.24	77.37	76.31	76.84	68.52
DSG $\Delta 4$	72.57	36.36	**60.12**	**56.35**	63.70	90.84	**96.16**	63.52	87.48	**93.16**	63.61	80.58	73.25	**75.96**	82.75	65.04	75.52	**78.44**	**76.78**	**77.61**	69.08
DSG $\Delta 8$	72.61	**36.65**	59.69	56.32	**65.24**	90.86	96.02	**63.56**	87.60	93.08	**64.40**	80.32	**73.56**	75.88	**82.79**	**65.33**	**75.58**	77.43	76.75	77.09	**69.15**
DSG $\Delta 16$	**72.74**	35.34	59.70	55.93	64.78	**91.04**	95.86	62.36	**87.66**	92.92	63.57	**80.59**	73.17	75.88	82.61	64.79	75.41	78.18	76.66	77.42	68.90

Table D.5: Full per task accuracy for Fig. 7.6 showing different Dynamic Stop-and-Go Iteration Gaps (Δ). Mean of G2 is taken over the Recall@1 scores.

where $y(q_k^{(i)}) = 1$ means the model correctly answer the question $q_k^{(i)}$ based on VQA accuracy metric and $y(r_k^{(i)}) = 1$ means the model correctly locate the image regions (IoU ≤ 0.5) given the reference $r_k^{(i)}$.

D.9 Qualitative Results

Fig. D.3 shows more qualitative examples of our single model Our$_{\texttt{AT}}$ on different vision and language tasks and Fig. D.4 shows some failure cases. The examples in Fig. D.3 show that the `AT` model works well for these wide range of tasks consistently. It can perform well in both short as well as long reasoning questions, image retrieval, pointing tasks, referring expressions and multi-modal validation. Failure cases mostly occur when the model encounters `counting` questions or difficult referring expressions and phrases for fine grained recognition.

D.10 Attention Visualizations

To examine the visual groundings learned by the techniques we presented in Sec. D.2. We verify this by visualizing the attentions of our pretrained model, which is trained on the Conceptual Caption dataset. Given a test image, and corresponding caption "The boy and his mom pet the black and white sheep", we feed the image-caption pair as input and take the image to question co-attention for visualization. For each image patch, we use the most attended word to represent its semantic meaning, and show the patches corresponding to the visual words ('boy', 'mom', 'pet', 'white', 'sheep').

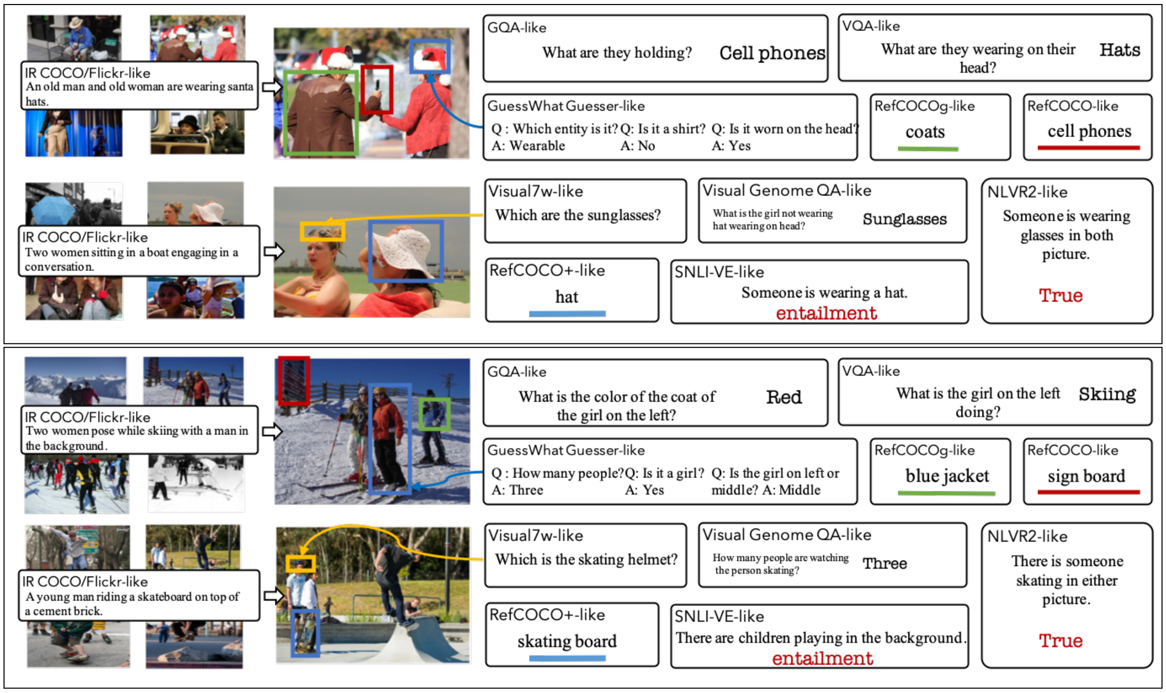


Figure D.3: **Our single multi-task model can solve multiple task consistently and correctly.** Additional qualitative examples of our single model Our$_{\texttt{AT}}$ on multitude of V&L tasks: caption and image retrieval, question answering, grounding phrases, guessing image regions based on a dialog, verifying facts about a pair of images, natural language inferences from an image, etc. Here we show outputs of our model for a variety of inputs (that mimic tasks from the 12 datasets it has been trained on).

Fig. D.5 shows the correspondence between attended regions and underlined words. We can see that the pretrained model learns meaningful visual grounding for the concept 'boy', 'sheep', 'white' and 'pet'.

To visualize the attention for our multi-task trained model (Ours$_{\texttt{AT}}$), we use BertVis[1] to visualization the attention distribution on the sentence to sentence self-attention $S{\rightarrow}S$, sentence to image co-attention $S{\rightarrow}I$, image to sentence co-attention $I{\rightarrow}S$ and image to image self attention $I{\rightarrow}I$. Fig. D.6 shows an example of the sentence to sentence attention for all layers and all heads (middle) and a specific layer and head (right). We can see that our model learns the previous words attention pattern, bag of words attention pattern (Layer 1 Head 1) and next words attention pattern

[1] https://github.com/jessevig/bertviz

Figure D.4: Failure cases of our single `AT` model on multitude of V&L tasks. Failure cases mostly occur when the model encounters `counting` questions or difficult referring expressions and phrases for fine grained recognition.

(Layer 2 Head 0). This shows that model is able to generate position-aware queries and keys to calculate the attentions. To get a sense of the difference of attention distribution across different tasks, Fig. D.7 and Fig. D.8 show the attention distribution. We can see for different tasks, the model learns to use significant different sentence to sentence self-attention pattern.

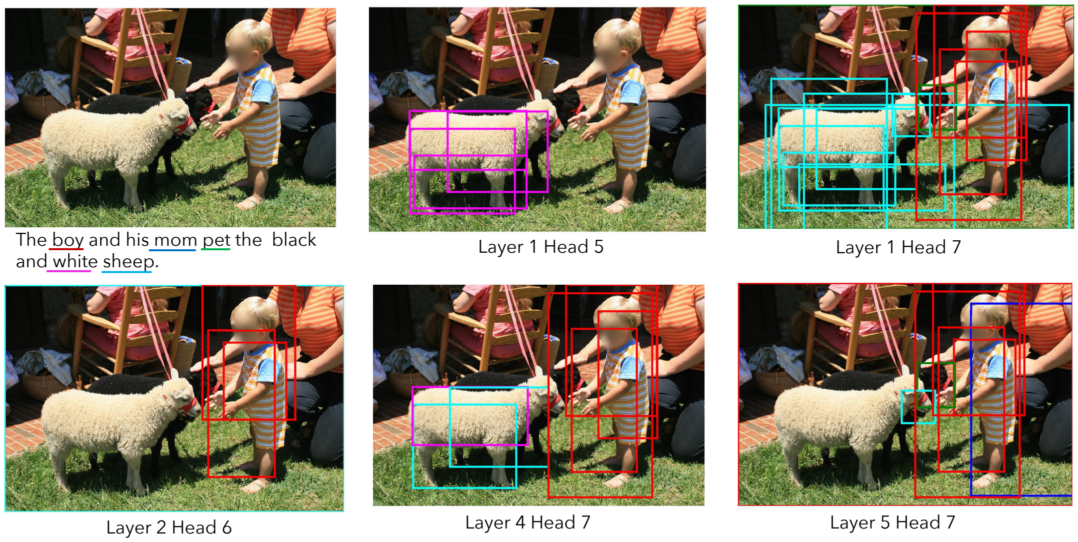


Figure D.5: Visualizations of image to sentence attention for the pretrained model on conceptual caption dataset. Given the image to sentence co-attention, we use the most attended word to represent its semantic meaning, and show the patches corresponding to the visual words ('boy', 'mom', 'pet', 'white', 'sheep'). Different colors show a correspondence between attended regions and underlined words. We can see that the model learns meaningful concept through pretraining.

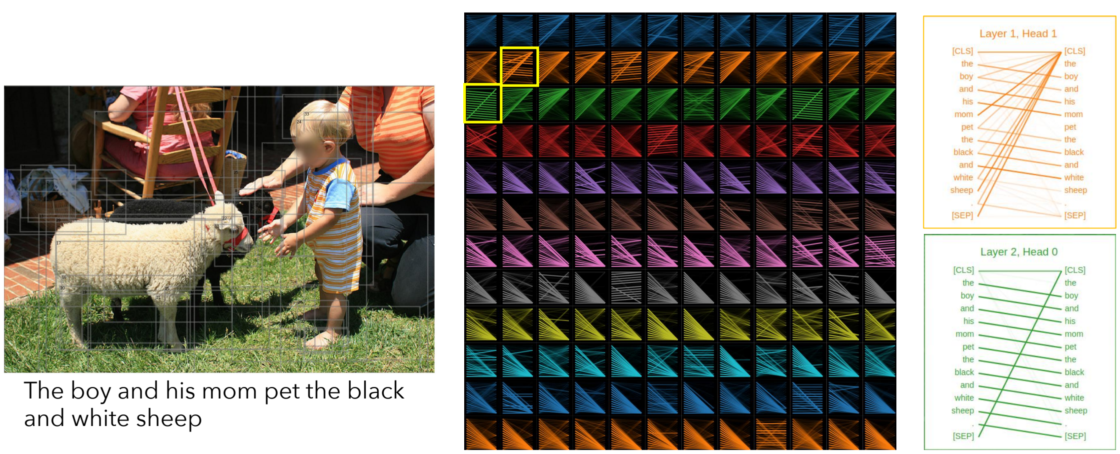


Figure D.6: Visualizations of the attentions of the pretrained model on conceptual caption dataset using BertVis toolbox. From left to right: Image and associate caption, sentence to sentence self-attention for all layers and all heads, sentence to sentence self-attention for Layer 1 Head 1 and Layer 2 Head 0. Our model learns the previous words attention pattern, bag of words attention pattern and next words attention pattern.

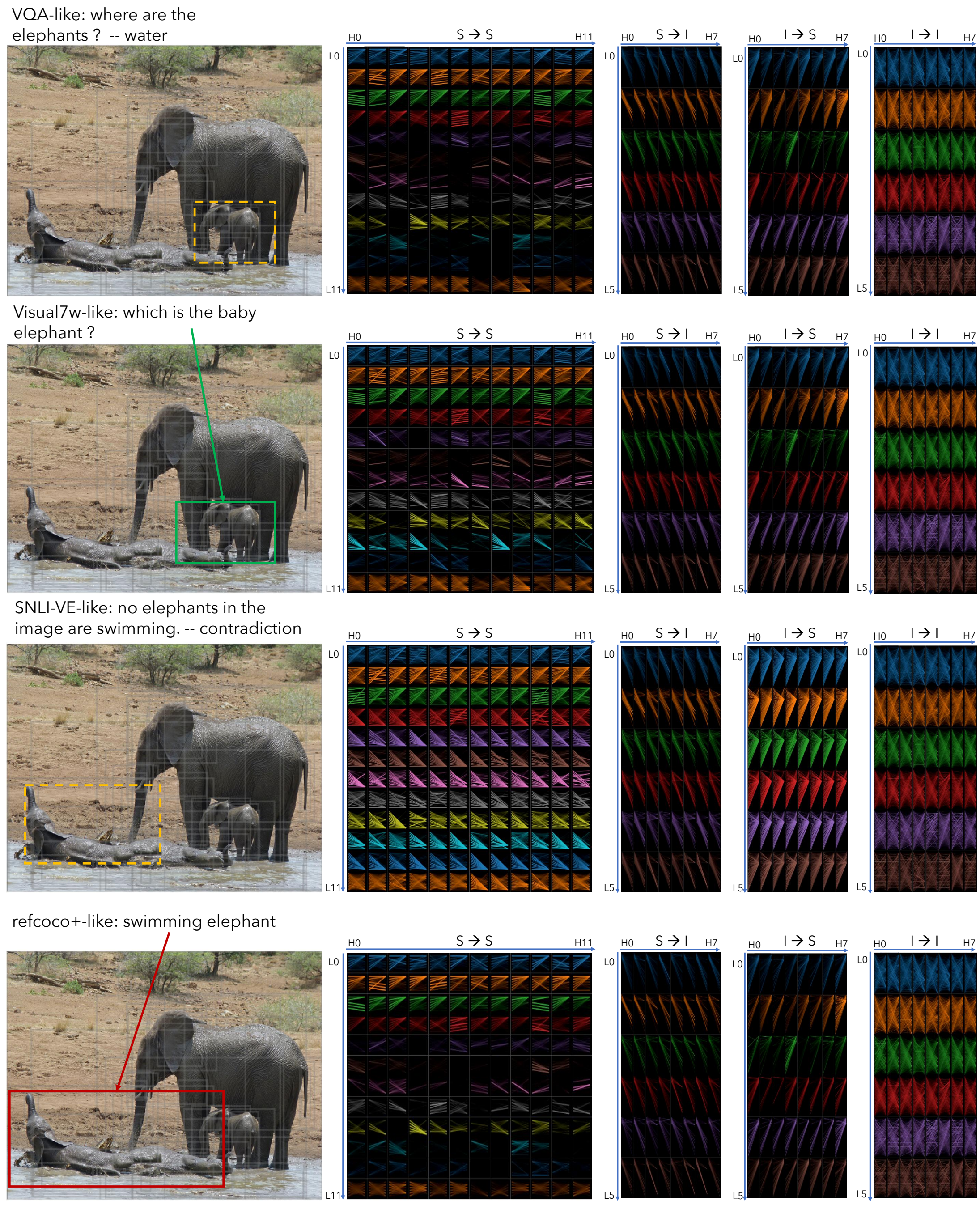


Figure D.7: Visualizations of the attentions of Our$_{\mathtt{AT}}$ model using BertVis toolbox on each tasks. From left to right are image and associate sentence, sentence to sentence self-attention, sentence to image co-attention image to sentence co-attention image to image self-attention. Dashed orange bounding boxes in the image are the referring expression outputs regardless of tasks. The model learns to use significant different sentence to sentence self-attention pattern for different tasks.

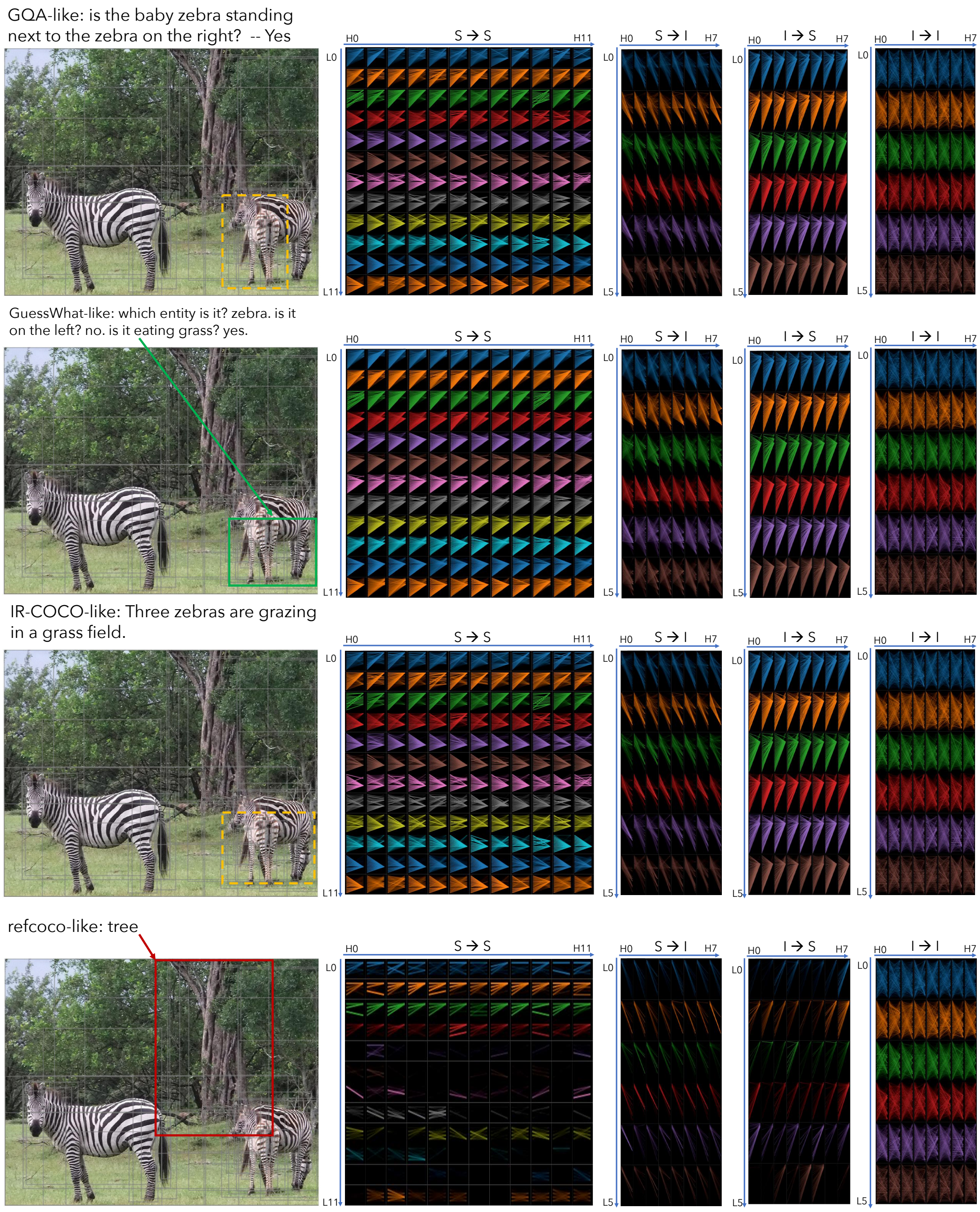


Figure D.8: Visualizations of the attentions of Our$_{\mathtt{AT}}$ model using BertVis toolbox on each tasks. From left to right are image and associate sentence, sentence to sentence self-attention, sentence to image co-attention image to sentence co-attention image to image self-attention. Dashed orange bounding boxes in the image are the referring expression outputs regardless of tasks. The model learns to use significant different sentence to sentence self-attention pattern for different tasks.

REFERENCES

[1] M. A. Boden, *Mind as Machine: A History of Cognitive Science*. Oxford University Press, 2008.

[2] S. Kazemzadeh, V. Ordonez, M. Matten, and T. L. Berg, "ReferItGame: Referring to Objects in Photographs of Natural Scenes," in *EMNLP*, 2014.

[3] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "VQA: Visual Question Answering," in *ICCV*, 2015.

[4] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh, and D. Batra, "Visual Dialog," in *CVPR*, 2017.

[5] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*, 2014.

[6] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra, "Embodied question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 5, p. 6.

[7] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. van den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *CVPR)*, 2018.

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[9] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *IJCV*, 2015.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR*, 2009.

[11] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[12] *English wikipedia*, 2019.

[13] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. of NAACL*, 2018.

[14] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding with unsupervised learning," Technical report, OpenAI, Tech. Rep., 2018.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[16] X. Liu, P. He, W. Chen, and J. Gao, "Multi-task deep neural networks for natural language understanding," *arXiv preprint arXiv:1901.11504*, 2019.

[17] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," *arXiv preprint arXiv:1804.07461*, 2018.

[18] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," *CoRR*, vol. abs/1502.03044, 2015.

[19] A. Karpathy and L. Fei-Fei, "Deep Visual-Semantic Alignments for Generating Image Descriptions," in *CVPR*, 2015.

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," in *Neural computation*, 1997.

[21] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in *CVPR*, 2016.

[22] H. Fang, S. Gupta, F. N. Iandola, R. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, C. L. Zitnick, and G. Zweig, "From Captions to Visual Concepts and Back," *CoRR*, vol. abs/1411.4952, 2014.

[23] A. Das, H. Agrawal, C. L. Zitnick, D. Parikh, and D. Batra, "Human Attention in Visual Question Answering: Do Humans and Deep Networks Look at the Same Regions?" In *EMNLP*, 2016.

[24] G. Kulkarni, V. Premraj, S. L. Sagnik Dhar and, Y. Choi, A. C. Berg, and T. L. Berg, "Baby Talk: Understanding and Generating Simple Image Descriptions," in *CVPR*, 2011.

[25] A. Farhadi, M. Hejrati, A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth, "Every Picture Tells a Story: Generating Sentences for Images," in *ECCV*, 2010.

[26] A. Das, S. Kottur, J. M. Moura, S. Lee, and D. Batra, "Learning cooperative visual dialog agents with deep reinforcement learning," *arXiv preprint arXiv:1703.06585*, 2017.

[27] D. Yarats and M. Lewis, "Hierarchical text generation and planning for strategic dialogue," *arXiv preprint arXiv:1712.05846*, 2017.

[28] J. Li, W. Monroe, T. Shi, A. Ritter, and D. Jurafsky, "Adversarial learning for neural dialogue generation," *arXiv preprint arXiv:1701.06547*, 2017.

[29] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, "Sequence level training with recurrent neural networks," *arXiv preprint arXiv:1511.06732*, 2015.

[30] S. Liu, Z. Zhu, N. Ye, S. Guadarrama, and K. Murphy, "Optimization of image description metrics using policy gradient methods," *arXiv preprint arXiv:1612.00370*, 2016.

[31] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ser. ACL '02, Association for Computational Linguistics, 2002, pp. 311–318.

[32] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *ACL 2004 Workshop*, 2004.

[33] R. Vedantam, C Lawrence Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *CVPR*, 2015.

[34] C.-W. Liu, R. Lowe, I. V. Serban, M. Noseworthy, L. Charlin, and J. Pineau, "How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation," *arXiv preprint arXiv:1603.08023*, 2016.

[35] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[36] M. Lewis, D. Yarats, Y. N. Dauphin, D. Parikh, and D. Batra, "Deal or no deal? end-to-end learning for negotiation dialogues," *arXiv preprint arXiv:1706.05125*, 2017.

[37] G. Larsson, M. Maire, and G. Shakhnarovich, “Colorization as a proxy task for visual understanding,” in *CVPR*, 2017, pp. 6874–6883.

[38] P. Sharma, N. Ding, S. Goodman, and R. Soricut, “Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning,” in *ACL*, 2018.

[39] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, “VQA: Visual question answering,” in *ICCV*, 2015.

[40] R. Zellers, Y. Bisk, A. Farhadi, and Y. Choi, “From recognition to cognition: Visual commonsense reasoning,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[41] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, “From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions,” *TACL*, 2014.

[42] J. Lu, D. Batra, D. Parikh, and S. Lee, “Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks,” *arXiv preprint arXiv:1908.02265*, 2019.

[43] H. Tan and M. Bansal, “Lxmert: Learning cross-modality encoder representations from transformers,” *arXiv preprint arXiv:1908.07490*, 2019.

[44] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, “Visualbert: A simple and performant baseline for vision and language,” *arXiv preprint arXiv:1908.03557*, 2019.

[45] C. Alberti, J. Ling, M. Collins, and D. Reitter, “Fusion of detected objects in text for visual question answering,” *arXiv preprint arXiv:1908.05054*, 2019.

[46] G. Li, N. Duan, Y. Fang, D. Jiang, and M. Zhou, “Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training,” *arXiv preprint arXiv:1908.06066*, 2019.

[47] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, “Vl-bert: Pre-training of generic visual-linguistic representations,” *arXiv preprint arXiv:1908.08530*, 2019.

[48] L. Zhou, H. Palangi, L. Zhang, H. Hu, J. J. Corso, and J. Gao, “Unified vision-language pre-training for image captioning and vqa,” *arXiv preprint arXiv:1909.11059*, 2019.

[49] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick, “Microsoft COCO captions: Data collection and evaluation server,” *CoRR*, vol. abs/1504.00325, 2015.

[50] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed Representations of Words and Phrases and their Compositionality,” in *NIPS*, 2013.

[51] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, “Show and Tell: A Neural Image Caption Generator,” in *CVPR*, 2015.

[52] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille, “Deep captioning with multimodal recurrent neural networks (m-rnn),” in *ICLR*, 2015.

[53] P. Chattopadhyay, D. Yadav, V. Prabhu, A. Chandrasekaran, A. Das, S. Lee, D. Batra, and D. Parikh, “Evaluating visual conversational agents via cooperative human-ai games,” *arXiv preprint arXiv:1708.05122*, 2017.

[54] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *NeurIPS*, 2017.

[55] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, *et al.*, “Google’s neural machine translation system: Bridging the gap between human and machine translation,” *arXiv preprint arXiv:1609.08144*, 2016.

[56] D. Geman, S. Geman, N. Hallonquist, and L. Younes, “A Visual Turing Test for Computer Vision Systems,” in *PNAS*, 2014.

[57] M. Malinowski and M. Fritz, “A Multi-World Approach to Question Answering about Real-World Scenes based on Uncertain Input,” in *NIPS*, 2014.

[58] K. Tu, M. Meng, M. W. Lee, T. E. Choe, and S. C. Zhu, “Joint Video and Text Parsing for Understanding Events and Answering Queries,” *IEEE MultiMedia*, 2014.

[59] J. P. Bigham, C. Jayant, H. Ji, G. Little, A. Miller, R. C. Miller, R. Miller, A. Tatarowicz, B. White, S. White, and T. Yeh, “VizWiz: Nearly Real-time Answers to Visual Questions,” in *User Interface Software and Technology*, 2010.

[60] M. Malinowski, M. Rohrbach, and M. Fritz, “Ask your neurons: A neural-based approach to answering questions about images,” in *ICCV*, 2015.

[61] X. Lin and D. Parikh, “Don’t just listen, use your imagination: Leveraging visual common sense for non-visual tasks,” in *CVPR*, 2015.

[62] F. Sadeghi, S. K. Kumar Divvala, and A. Farhadi, “Viske: Visual knowledge extraction and question answering by visual verification of relation phrases,” in *CVPR*, 2015.

[63] R. Vendantam, X. Lin, T. Batra, C. L. Zitnick, and D. Parikh, “Learning common sense through visual abstraction,” in *ICCV*, 2015.

[64] L. Yu, E. Park, A. C. Berg, and T. L. Berg, “Visual madlibs: Fill-in-the-blank description generation and question answering,” in *ICCV*, 2015.

[65] H. Gao, J. Mao, J. Zhou, Z. Huang, and A. Yuille, “Are you talking to a machine? dataset and methods for multilingual image question answering,” in *NIPS*, 2015.

[66] M. Ren, R. Kiros, and R. Zemel, “Exploring models and data for image question answering,” in *NIPS*, 2015.

[67] L. Ma, Z. Lu, and H. Li, “Learning to answer questions from image using convolutional neural network,” in *AAAI*, 2016.

[68] M. Ren, R. Kiros, and R. Zemel, “Image question answering: A visual semantic embedding model and a new dataset,” in *NIPS*, 2015.

[69] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei, “Visual7w: Grounded question answering in images,” in *CVPR*, 2016.

[70] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, “Deep compositional question answering with neural module networks,” in *CVPR*, 2016.

[71] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, “Stacked attention networks for image question answering,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 21–29.

[72] H. Xu and K. Saenko, “Ask, attend and answer: Exploring question-guided spatial attention for visual question answering,” in *European Conference on Computer Vision*, Springer, 2016, pp. 451–466.

[73] K. J. Shih, S. Singh, and D. Hoiem, “Where to look: Focus regions for visual question answering,” in *CVPR*, 2016.

[74] C. Xiong, S. Merity, and R. Socher, “Dynamic memory networks for visual and textual question answering,” in *ICML*, 2016.

[75] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.

[76] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, “Teaching machines to read and comprehend,” in *NIPS*, 2015.

[77] T. Rocktäschel, E. Grefenstette, K. M. Hermann, T. Kočiskỳ, and P. Blunsom, “Reasoning about entailment with neural attention,” in *ICLR*, 2016.

[78] W. Yin, H. Schütze, B. Xiang, and B. Zhou, “Abcnn: Attention-based convolutional neural network for modeling sentence pairs,” in *ACL*, 2016.

[79] C. d. Santos, M. Tan, B. Xiang, and B. Zhou, “Attentive pooling networks,” *arXiv preprint arXiv:1602.03609*, 2016.

[80] P. Kuznetsova, V. Ordonez, A. C. Berg, T. L. Berg, and Y. Choi, “Collective generation of natural image descriptions,” in *ACL*, 2012.

[81] R. Kiros, R. Salakhutdinov, and R. S. Zemel, “Multimodal neural language models,” in *ICML*, 2014.

[82] J. Devlin, H. Cheng, H. Fang, S. Gupta, L. Deng, X. He, G. Zweig, and M. Mitchell, “Language models for image captioning: The quirks and what works,” *arXiv preprint arXiv:1505.01809*, 2015.

[83] R. Kiros, R. Salakhutdinov, and R. S. Zemel, “Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models,” *TACL*, 2015.

[84] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.

[85] I. Sutskever, O. Vinyals, and Q. Le, “Sequence to Sequence Learning with Neural Networks,” in *Neural Information Processing Systems (NIPS)*, 2014.

[86] X. Chen and C. L. Zitnick, “Mind’s Eye: A Recurrent Visual Representation for Image Caption Generation,” in *CVPR*, 2015.

[87] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille, “Explain Images with Multimodal Recurrent Neural Networks,” *CoRR*, vol. abs/1410.1090, 2014.

[88] Q. Wu, C. Shen, L. Liu, A. Dick, and A. v. d. Hengel, “What value do explicit high level concepts have in vision to language problems?” *arXiv preprint arXiv:1506.01144*, 2015.

[89] Z. Yang, Y. Yuan, Y. Wu, R. Salakhutdinov, and W. W. Cohen, "Encode, review, and decode: Reviewer module for caption generation," in *NIPS*, 2016.

[90] E. L. Denton, S. Chintala, A. Szlam, and R. Fergus, "Deep generative image models using a laplacian pyramid of adversarial networks," *Neural Information Processing Systems*, 2015.

[91] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *CoRR*, vol. abs/1511.06434, 2015.

[92] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," *CoRR*, vol. abs/1609.04802, 2016.

[93] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *arXiv preprint arXiv:1703.10593*, 2017.

[94] B. Dai, D. Lin, R. Urtasun, and S. Fidler, "Towards diverse and natural image descriptions via a conditional gan," *arXiv preprint arXiv:1703.06029*, 2017.

[95] R. Shetty, M. Rohrbach, L. A. Hendricks, M. Fritz, and B. Schiele, "Speaking the same language: Matching machine to human captions by adversarial training," *CoRR*, vol. abs/1703.10476, 2017.

[96] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient," *AAAI Conference on Artificial Intelligence*, 2017.

[97] M. J. Kusner and J. M. Hernández-Lobato, "Gans for sequences of discrete elements with the gumbel-softmax distribution," *CoRR*, vol. abs/1611.04051, 2016.

[98] J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based generative adversarial network," *arXiv preprint arXiv:1609.03126*, 2016.

[99] I. Misra, R. B. Girshick, R. Fergus, M. Hebert, A. Gupta, and L. van der Maaten, "Learning by asking questions," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11–20, 2017.

[100] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, "Visual curiosity: Learning to ask questions to learn visual recognition," in *CoRL*, 2018.

[101] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, "Making the v in vqa matter: Elevating the role of image understanding in visual question answering," in *CVPR*, 2017.

[102] T. Zhao, K. Lee, and M. Eskénazi, "Unsupervised discrete sentence representation learning for interpretable neural dialog generation," in *ACL*, 2018.

[103] T. Zhao and M. Eskénazi, "Zero-shot dialog generation with cross-domain latent actions," in *SIGDIAL Conference*, 2018.

[104] T.-H. Wen, Y. Miao, P. Blunsom, and S. Young, "Latent intention dialogue models," *arXiv preprint arXiv:1705.10229*, 2017.

[105] I. V. Serban, I. Ororbia, G Alexander, J. Pineau, and A. Courville, "Piecewise latent variables for neural variational text processing," *arXiv preprint arXiv:1612.00377*, 2016.

[106] H. Hu, D. Yarats, Q. Gong, Y. Tian, and M. Lewis, "Hierarchical decision making by generating and following natural language instructions," *arXiv preprint arXiv:1906.00744*, 2019.

[107] T. Zhao, K. Xie, and M. Eskénazi, "Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models," in *NAACL-HLT*, 2019.

[108] D. Lewis, "Convention: A philosophical study," 1969.

[109] H. De Vries, F. Strub, S. Chandar, O. Pietquin, H. Larochelle, and A. Courville, "Guesswhat?! visual object discovery through multi-modal dialogue," in *CVPR*, 2017.

[110] P. Chattopadhyay, D. Yadav, V. Prabhu, A. Chandrasekaran, A. Das, S. Lee, D. Batra, and D. Parikh, "Evaluating visual conversational agents via cooperative human-ai games," in *Proceedings of the Fifth AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*, 2017.

[111] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *ICCV*, 2015, pp. 1422–1430.

[112] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *ECCV*, Springer, 2016, pp. 649–666.

[113] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with exemplar convolutional neural networks," *IEEE PAMI*, vol. 38, no. 9, pp. 1734–1747, 2015.

[114] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, “Context encoders: Feature learning by inpainting,” in *CVPR*, 2016, pp. 2536–2544.

[115] D. Jayaraman and K. Grauman, “Learning image representations tied to ego-motion,” in *CVPR*, 2015, pp. 1413–1421.

[116] I. Misra, C. L. Zitnick, and M. Hebert, “Shuffle and learn: Unsupervised learning using temporal order verification,” in *ECCV*, Springer, 2016, pp. 527–544.

[117] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” in *NACCL*, 2018.

[118] G. Lample and A. Conneau, “Cross-lingual language model pretraining,” *arXiv preprint arXiv:1901.07291*, 2019.

[119] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid, “Videobert: A joint model for video and language representation learning,” *arXiv preprint arXiv:1904.01766*, 2019.

[120] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, *et al.*, “Visual genome: Connecting language and vision using crowdsourced dense image annotations,” *arXiv preprint arXiv:1602.07332*, 2016.

[121] R. Caruana, “Multitask learning,” *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[122] S. Ruder, “An overview of multi-task learning in deep neural networks,” *arXiv preprint arXiv:1706.05098*, 2017.

[123] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, “Robust visual tracking via structured multi-task sparse learning,” *International journal of computer vision*, vol. 101, no. 2, pp. 367–383, 2013.

[124] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, “Facial landmark detection by deep multi-task learning,” in *European conference on computer vision*, Springer, 2014, pp. 94–108.

[125] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, “Cross-stitch networks for multi-task learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3994–4003.

[126] I. Kokkinos, “Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory,” in

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6129–6138.

[127] R. Collobert and J. Weston, “A unified architecture for natural language processing: Deep neural networks with multitask learning,” in *Proceedings of the 25th international conference on Machine learning*, ACM, 2008, pp. 160–167.

[128] X. Liu, J. Gao, X. He, L. Deng, K. Duh, and Y.-Y. Wang, “Representation learning using multi-task deep neural networks for semantic classification and information retrieval,” 2015.

[129] B. McCann, N. S. Keskar, C. Xiong, and R. Socher, “The natural language decathlon: Multitask learning as question answering,” *arXiv preprint arXiv:1806.08730*, 2018.

[130] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *arXiv preprint arXiv:1910.10683*, 2019.

[131] E. Parisotto, J. L. Ba, and R. Salakhutdinov, “Actor-mimic: Deep multitask and transfer reinforcement learning,” *arXiv preprint arXiv:1511.06342*, 2015.

[132] M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Z. Leibo, D. Silver, and K. Kavukcuoglu, “Reinforcement learning with unsupervised auxiliary tasks,” *arXiv preprint arXiv:1611.05397*, 2016.

[133] Y. Teh, V. Bapst, W. M. Czarnecki, J. Quan, J. Kirkpatrick, R. Hadsell, N. Heess, and R. Pascanu, “Distral: Robust multitask reinforcement learning,” in *Advances in Neural Information Processing Systems*, 2017, pp. 4496–4506.

[134] T. Standley, A. R. Zamir, D. Chen, L. Guibas, J. Malik, and S. Savarese, “Which tasks should be learned together in multi-task learning?” *arXiv preprint arXiv:1905.07553*, 2019.

[135] K. Clark, M.-T. Luong, U. Khandelwal, C. D. Manning, and Q. V. Le, “Bam! born-again multi-task networks for natural language understanding,” *arXiv preprint arXiv:1907.04829*, 2019.

[136] S. Pramanik, P. Agrawal, and A. Hussain, “Omninet: A unified architecture for multi-modal multi-task learning,” *arXiv preprint arXiv:1907.07804*, 2019.

[137] D.-K. Nguyen and T. Okatani, “Multi-task learning of hierarchical vision-language representation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 492–10 501.

[138] K. Shuster, D. Ju, S. Roller, E. Dinan, Y.-L. Boureau, and J. Weston, *The dialogue dodecathlon: Open-domain knowledge and image grounded conversational agents*, 2019. arXiv: `1911.03768 [cs.CL]`.

[139] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[140] P. Zhang, Y. Goyal, D. Summers-Stay, D. Batra, and D. Parikh, "Yin and yang: Balancing and answering binary visual questions," *arXiv preprint arXiv:1511.05099*, 2015.

[141] H. Agrawal, C. S. Mathialagan, Y. Goyal, N. Chavali, P. Banik, A. Mohapatra, A. Osman, and D. Batra, "Cloudcv: Large-scale distributed computer vision as a cloud service," in *Mobile Cloud Visual Media Computing*, Springer International Publishing, 2015, pp. 265–290.

[142] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," in *NIPS*, 2014.

[143] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," in *BigLearn, NIPS Workshop*, 2011.

[144] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[145] I. Ilievski, S. Yan, and J. Feng, "A focused dynamic attention model for visual question answering," *arXiv:1604.01485*, 2016.

[146] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015.

[147] S. Merity, C. Xiong, J. Bradbury, and R. Socher, "Pointer sentinel mixture models," *arXiv preprint arXiv:1609.07843*, 2016.

[148] T. Yao, Y. Pan, Y. Li, Z. Qiu, and T. Mei, "Boosting image captioning with attributes," *arXiv preprint arXiv:1611.01646*, 2015.

[149] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term Recurrent Convolutional Networks for Visual Recognition and Description," in *CVPR*, 2015.

[150] M. Denkowski and A. Lavie, "Meteor universal: Language specific translation evaluation for any target language," in *EACL 2014 Workshop on Statistical Machine Translation*, 2014.

[151] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "Spice: Semantic propositional image caption evaluation," in *ECCV*, 2016.

[152] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *arXiv preprint arXiv:1512.00567*, 2015.

[153] R. R.Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization," *arXiv:1611.01646*, 2016.

[154] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," *arXiv preprint arXiv:1512.04150*, 2015.

[155] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing when to look: Adaptive attention via A visual sentinel for image captioning," *CoRR*, vol. abs/1612.01887, 2016.

[156] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NuerIPS*, 2015, pp. 91–99.

[157] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, "Self-critical sequence training for image captioning," in *CVPR*, 2017.

[158] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *NIPS*, 2015.

[159] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *EMNLP*, 2014.

[160] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik, "Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models," in *ICCV*, 2015.

[161] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *ACL*, 2014.

[162] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017, pp. 2961–2969.

[163] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *CVPR*, 2018.

[164] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[165] D. Chen and C. Manning, “A fast and accurate dependency parser using neural networks,” in *EMNLP*, 2014.

[166] J. Yang, J. Lu, D. Batra, and D. Parikh, *A faster pytorch implementation of faster r-cnn*, `https://github.com/jwyang/faster-rcnn.pytorch`, 2017.

[167] R. Luo, *Unofficial pytorch implementation for self-critical sequence training for image captioning*, `https://github.com/ruotianluo/self-critical.pytorch`, 2017.

[168] P. Anderson, B. Fernando, M. Johnson, and S. Gould, “Guided open vocabulary image captioning with constrained beam search,” *EMNLP*, 2017.

[169] L. Anne Hendricks, S. Venugopalan, M. Rohrbach, R. Mooney, K. Saenko, and T. Darrell, “Deep compositional captioning: Describing novel object categories without paired training data,” in *CVPR*, 2016.

[170] S. Venugopalan, L. A. Hendricks, M. Rohrbach, R. Mooney, T. Darrell, and K. Saenko, “Captioning images with diverse objects,” in *CVPR*, 2017.

[171] T. Yao, Y. Pan, Y. Li, and T. Mei, “Incorporating copying mechanism in image captioning for learning novel objects,” in *CVPR*, 2017.

[172] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[173] A. Dosovitskiy and T. Brox, “Generating images with perceptual similarity metrics based on deep networks,” in *Advances in Neural Information Processing Systems*, 2016, pp. 658–666.

[174] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in *European Conference on Computer Vision*, Springer, 2016, pp. 694–711.

[175] L. A. Gatys, A. S. Ecker, and M. Bethge, “A neural algorithm of artistic style,” *arXiv preprint arXiv:1508.06576*, 2015.

[176] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *arXiv preprint arXiv:1503.02531*, 2015.

[177] T. Chen, I. Goodfellow, and J. Shlens, “Net2net: Accelerating learning via knowledge transfer,” *arXiv preprint arXiv:1511.05641*, 2015.

[178] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[179] C. J. Maddison, A. Mnih, and Y. W. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," *arXiv preprint arXiv:1611.00712*, 2016.

[180] Y. Bengio, N. Léonard, and A. C. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *CoRR*, vol. abs/1308.3432, 2013.

[181] I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," *arXiv preprint arXiv:1507.04808*, 2015.

[182] H. Mei, M. Bansal, and M. R. Walter, "Coherent dialogue with attention-based language models," *arXiv preprint arXiv:1611.06997*, 2016.

[183] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Advances in Neural Information Processing Systems*, 2016, pp. 1849–1857.

[184] M. Mironenco, D. Kianfar, K. Tran, E. Kanoulas, and E. Gavves, "Examining cooperation in visual dialog models," *arXiv preprint arXiv:1712.01329*, 2017.

[185] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011.

[186] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[187] L. Theis, A. van den Oord, and M. Bethge, "A note on the evaluation of generative models," *CoRR*, vol. abs/1511.01844, 2015.

[188] A. K. Vijayakumar, M. Cogswell, R. R. Selvaraju, Q. Sun, S. Lee, D. J. Crandall, and D. Batra, "Diverse beam search: Decoding diverse solutions from neural sequence models," *AAAI*, 2018.

[189] J. Li, M. Galley, C. Brockett, J. Gao, and W. B. Dolan, "A diversity-promoting objective function for neural conversation models," in *HLT-NAACL*, 2015.

[190] J. Lu, A. Kannan, J. Yang, D. Parikh, and D. Batra, "Best of both worlds: Transferring knowledge from discriminative learning to a generative visual

dialog model," in *Advances in Neural Information Processing Systems*, 2017, pp. 314–324.

[191] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," in *Advances In Neural Information Processing Systems*, 2016, pp. 289–297.

[192] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *ICCV*, 2015.

[193] S. Kazemzadeh, V. Ordonez, M. Matten, and T. Berg, "Referitgame: Referring to objects in photographs of natural scenes," in *EMNLP*, 2014.

[194] L. Yu, Z. Lin, X. Shen, J. Yang, X. Lu, M. Bansal, and T. L. Berg, "Mattnet: Modular attention network for referring expression comprehension," in *CVPR*, 2018.

[195] K.-H. Lee, X. Chen, G. Hua, H. Hu, and X. He, "Stacked cross attention for image-text matching," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 201–216.

[196] G. Peng, H. Li, H. You, Z. Jiang, P. Lu, S. Hoi, and X. Wang, "Dynamic fusion with intra-and inter-modality attention flow for visual question answering," *arXiv preprint arXiv:1812.05252*, 2018.

[197] D. A. Hudson and C. D. Manning, "Gqa: A new dataset for compositional question answering over real-world images," *arXiv preprint arXiv:1902.09506*, 2019.

[198] J. Mao, J. Huang, A. Toshev, O. Camburu, A. L. Yuille, and K. Murphy, "Generation and comprehension of unambiguous object descriptions," in *CVPR*, 2016.

[199] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei, "Visual7w: Grounded question answering in images," in *CVPR*, 2016.

[200] A. Suhr, S. Zhou, A. Zhang, I. Zhang, H. Bai, and Y. Artzi, "A corpus for reasoning about natural language grounded in photographs," in *ACL*, 2019.

[201] N. Xie, F. Lai, D. Doran, and A. Kadav, "Visual entailment task for visually-grounded language learning," *arXiv preprint arXiv:1811.10582*, 2018.

[202] Y.-C. Chen, L. Li, L. Yu, A. E. Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu, "Uniter: Learning universal image-text representations," *arXiv preprint arXiv:1909.11740*, 2019.

[203] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, ACM, 2009, pp. 41–48.

[204] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[205] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam," *arXiv preprint arXiv:1711.05101*, 2017.

[206] R. Hu, M. Rohrbach, J. Andreas, T. Darrell, and K. Saenko, "Modeling relationships in referential expressions with compositional modular networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[207] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[208] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[209] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *arXiv preprint arXiv:1906.08237*, 2019.

[210] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "Spanbert: Improving pre-training by representing and predicting spans," *arXiv preprint arXiv:1907.10529*, 2019.

[211] V. Ordonez, G. Kulkarni, and T. Berg, "Im2text: Describing images using 1 million captioned photographs," in *NeurIPS*, 2011.

LIST OF PUBLICATIONS

1. **Dialog without Dialog: Learning Image-Discriminative Dialog Policies from Single-Shot Question Answering Data**: Michael Cogswell*, Jiasen Lu*, Devi Parikh, Stefan Lee, Dhruv Batra. *In Submission, 2019*

2. **12-in-1: Multi-Task Vision and Language Representation Learning**: Jiasen Lu*, Vedanuj Goswami*, Marcus Rohrbach, Devi Parikh, Stefan Lee. *arXiv:1912.02315, 2019*

3. **ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks**: Jiasen Lu, Dhruv Batra, Devi Parikh, Stefan Lee. *Neural Inforamtion Processing Systems (NeurIPS) 2019*

4. **Emergence of Compositional Language with Deep Generational Transmission**: Michael Cogswell, Jiasen Lu, Stefan Lee, Devi Parikh, Dhruv Batra. *arXiv:1904.09067, 2019*

5. **Self-Monitoring Navigation Agent via Auxiliary Progress Estimation**: Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, Caiming Xiong. *International Conference on Learning Representations (ICLR), 2019*

6. **Visual Curiosity: Learning to Ask Questions to Learn Visual Recognition**: Jianwei Yang*, Jiasen Lu*, Stefan Lee, Dhruv Batra, Devi Parikh. *Conference on Robot Learning (CoRL) 2018* ***(Oral)***

7. **Graph R-CNN for Scene Graph Generation**: Jianwei Yang*, Jiasen Lu*, Stefan Lee, Dhruv Batra, Devi Parikh. *European Conference on Computer Vision (ECCV) 2018*

8. **Neural Baby Talk**: Jiasen Lu*, Jianwei Yang*, Dhruv Batra, Devi Parikh. *Computer Vision and Pattern Recognition (CVPR), 2018* ***(Spotlight)***

9. **Best of Both Worlds: Transferring Knowledge from Discriminative Learning to a Generative Visual Dialog Model**: Jiasen Lu, Anitha Kannan, Jianwei Yang, Devi Parikh, Dhruv Batra. *Neural Inforamtion Processing Systems (NIPS) 2017*

10. **ParlAI: A Dialog Research Software Platform**: Alexander H. Miller, Will Feng, Adam Fisch, Jiasen Lu, Dhruv Batra, Antoine Bordes, Devi Parikh, Jason Weston. *Conference on Empirical Methods on Natural Language Processing (EMNLP), 2017*

11. **Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning**: Jiasen Lu*, Caiming Xiong*, Devi Parikh, Richard Socher. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017* ***(Spotlight)***

12. **VQA: Visual Question Answering**: Aishwarya Agrawal*, Jiasen Lu*, Stanislaw Antol*, Margaret Mitchell, C. Lawrence Zitnick, Devi Parikh, Dhruv Batra. *International Journel of Computer Vision (IJCV), 2016*

13. **Hierarchical Question-Image Co-Attention for Visual Question Answering**: Jiasen Lu, Jianwei Yang, Dhruv Batra, Devi Parikh. *Neural Inforamtion Processing Systems (NIPS) 2016*

14. **VQA: Visual Question Answering** : Stanislaw Antol*, Aishwarya Agrawal*, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, Devi Parikh. *International Conference on Computer Vision (ICCV), 2015*

15. **Human Action Segmentation with Hierarchical Supervoxel Consistency**: Jiasen Lu, Ran Xu and Jason J. Corso. *IEEE Conference on Computer*

Vision and Pattern Recognition (CVPR), 2015

16. **Improving Word Representations via Global Visual Context**: Ran Xu, Jiasen Lu, Caiming Xiong, Zhi Yang, Jason J. Corso. *NIPS workshop on Learning Semantics, 2014*

* Equal Contribution