UNIFYING CROSS-MODAL CONCEPTS IN VISION AND LANGUAGE

BY

SPENCER WHITEHEAD

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois Urbana-Champaign, 2021

Urbana, Illinois

Doctoral Committee:

Professor Heng Ji, Chair
Professor Alexander Schwing
Professor ChengXiang Zhai
Professor Shih-Fu Chang, Columbia University
Professor Kate Saenko, Boston University

ABSTRACT

Enabling computers to demonstrate a proficient understanding of the physical world is an exceedingly challenging task that necessitates the ability to perceive, through vision or other senses, and communicate through natural language. Key to this endeavor is the representation of concepts present in the world within and across different modalities (*e.g.*, vision and language). To an extent, models can capture concepts implicitly through using large quantities of training data. However, the complementary inter-modal and intra-modal connections between concepts are often not captured, which leads to issues such as difficulty generalizing a concept to new contexts or different appearances and an inability to integrate concepts from different sources. The focus of this dissertation is developing ways to represent concepts within models in a unified fashion across vision and language. In particular, there are three challenges that we address:

1) Linking instances of concepts across modalities without strong supervision or large amounts of data external to the target task. In visual question answering, models tend to rely on contextual cues or learned priors instead of actually recognizing and linking concepts across modalities. Consequently, when a concept appears in a new context, models often fail to adapt. We learn to ground concept mentions in text to image regions in the context of visual question answering using self-supervision. We also demonstrate that learning concept grounding helps facilitate the disentanglement of the skills required to answer questions and concept mentions, which can improve generalization to novel compositions of skills and concepts.

2) Consistency towards different mentions of the same concept. An instance of a concept can take many different forms, such as the appearance of a concept in different images or the use of synonyms in text, and it can be difficult for models to infer these relationships from the training data alone. We show that existing visual question answering models have difficulty handling even straightforward changes in concept mentions and the wordings of the questions. We enforce consistency for related questions in these models not only of the answers, but also of the computed intermediate representations, which improves robustness to such variations.

3) Modeling associations between related concepts in complex domains. In scenarios where multiple related sources of information need to be considered, models must be able to connect concepts found within and across these different sources. We introduce the task of knowledge-aware video captioning for news videos, where models must generate descriptions of videos that leverage interconnected background knowledge pertaining to concepts involved in the videos. We build models that learn to associate patterns of concepts found in related news articles, such as entities and events, with video content in order to generate these knowledge-rich descriptions.

To my family, friends, and mentors for their kindness, support, and wisdom.

ACKNOWLEDGMENTS

I am so grateful to the individuals who have helped me during this Ph.D. process. Without their support, I would not have been able to complete this dissertation.

First, I would like to express my gratitude to my advisor, Professor Heng Ji, for her support, flexibility, and guidance. She is an erudite and tremendously passionate professor, who has taught me what it takes to do research. I will always appreciate the things I learned from her as well as the memories along the way, like our times working late on papers/evaluations or chatting about life while on the train/subway to visit Columbia. Thank you, Heng.

I would also like to extend my deepest gratitude to Professor Alexander Schwing, Professor ChengXiang Zhai, Professor Shih-Fu Chang, and Professor Kate Saenko for being on my committee. I have a deep admiration and respect for all of my committee members for their excellence in research and their contributions to computer vision, natural language processing, and machine learning. I am truly honored and proud to have them on my committee. They have provided me with constructive advice and valuable insights for which I am incredibly grateful. I would like to specifically thank Kate and Prof. Chang for their wonderful guidance and expertise as well as our fruitful collaborations, which have been crucial to the work in this dissertation.

I owe a lot to my other collaborators and mentors as well. I sincerely thank Hui Wu and Rogerio Feris for their knowledge, advice, and help throughout our collaborations. In particular, Hui's advice and the experience working with her has helped shape my development as a researcher. I thank my mentors who had guided me early on in my education and introduced me to research in mathematics and AI: Professor Donald Schwendeman, Professor Selmer Bringsjord, Professor John Licato, and David Goldschmidt. I thank Clare Voss, Achille Fokoue, Ibrahim Abdelaziz, and Pavan Kapanipathi for their support and collaborations during my internships.

Thank you to all the members of the Blender Lab, who I have enjoyed working alongside, including Tongtao Zhang, Lifu Huang, Boliang Zhang, Di Lu, Xiaoman Pan, Ying Lin, Manling Li, Qingyun Wang, Ananya Subburathinam, Pengfei Yu, and Yi Fung. The lab is replete with extraordinary individuals that I have learned a lot from. I am also fortunate to have such wonderful friends, a few of which are: Andy Fiumara, Nick Ambery, Jack Mata, Zach White, Mike Beloff, Nick Barnoski, Thibeault, Michael Morretto, Sean Sutherland, Adam Konja, Miles Humphrey, Wesley Lanigan, Akul Goyal, Marc Canby, Joao Marcos Correia Marques, Mike Shea, Robert Irvin, Julius Alexander, Jeremy Washington, Bryan Murtha, Dan Critz, Louis Chapman, Tamar Rogoszinski, Olivia Fryt, Courtney Murphy, and Beth Lotterer. I also thank the Fiumara, Ambery, and Beloff families for the remarkable kindness they have shown me throughout my degrees.

Most of all, I would like to thank my parents, my brother, and my girlfriend. Words cannot quite capture my gratitude and appreciation for their resolute love, encouragement, and understanding. I am forever grateful to have them in my life. To my girlfriend, you have been key to this journey, and I truly appreciate the closeness we share. To my brother, thank you for always being there to take my mind off things and always believing in me. Finally, Mom and Dad, you have always done your best to give me everything I need to succeed. Whether it was taking me to hockey practice at 6:00AM as a kid or helping me move across the country, you have always been there for me, shown me love, and allowed me to be the best that I can be as well as enjoy life. I feel blessed to have such a wonderful family. My determination to requite your efforts and make you as proud as possible helps drive my ambitions. It is a debt that I can never repay, but this dissertation is a token of my gratitude for all the time, energy, love, and compassion that you have devoted to making me who I am today.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 OVERVIEW

Great strides in computer vision and natural language processing have been made in recent years. The progress in these fields is evidenced by the dramatic performance increases that neural networks have made on tasks like object recognition or machine translation, where large-scale neural networks offer incredibly strong performance. As a result, models designed to perform lower-level tasks like object detection or named entity recognition can serve as part of the inputs to systems performing higher-level tasks. Consequently, exciting new tasks that go beyond the more standard recognition or detection of concepts in a single modality have been explored, many of which are aimed at combining the capabilities of vision and language. For example, being able to generate a description for an image or video, such as "*A girl is throwing a ball.*", or answer natural language questions about the same scene, such as "*What color is the girl's shirt?*", requires that models be able to recognize/detect objects as well as understand how different objects may be related to one another. Furthermore, such tasks also necessitate that models be able to draw connections between natural language and the visual content. To perform these tasks, models must possess rich representations of concepts that capture their complexities.

In this work, we develop techniques for representing concepts within models in a unified fashion, meaning that the models have the ability to link different instances of a concept within and across modalities as well as understand the connections or associations between concepts. To do so, we leverage different forms of structured knowledge about these concepts and inductive biases to link and enforce consistency between concepts. We particularly study these techniques within the context of two important vision and language tasks: Visual Question Answering (VQA) [1] and video captioning [2]. In the following, we outline these motivations and challenges in more detail.

1.2 MOTIVATIONS, CHALLENGES, AND SOLUTIONS

1.2.1 Linking Instances of Concepts Across Modalities

The ability to link instances of a concept in text to image regions (also called grounding) is essential for vision and language tasks. For instance, in VQA, to answer a question like "*What color is the girl's shirt?*", models should locate the girl and her shirt in the image, followed by identifying the color of the girls shirt. However, models can leverage language priors or contextual clues to answer questions, meaning models do not truly recognize and link concepts across modalities.

When a concept appears in a new context, these models often fail to adapt and lean on learned priors to render predictions.

In Chapter 4, we study the ability of VQA models to generalize to compositions of skills, which are vision tasks necessary to answer questions (*e.g.,* color recognition or counting), and concepts. To generalize to novel compositions, models must learn to separate skills and concepts in the question, particularly by grounding concepts to the image. Within this setting, we propose a novel approach to learn grounding and separate skills within the VQA tasks without requiring strong supervision (*e.g.,* annotated concept mention to image region correspondences) or large-scale data external to VQA (*e.g.,* image-caption pairs). For grounding, our approach forces the model to reconstruct masked out concept mentions by contrasting the multimodal representation of the masked concept with multimodal representations of other words in different examples. By doing so, the model learns to leverage the visual information to leverage the visual information common to the different input examples. We present a skill matching loss the trains the model to group questions together with the same skill, regardless of the concepts mentioned. With the combination of these objectives the model learns grounded concept representations and concept-agnostic representations of skills.

1.2.2 Linking Different Mentions of Concepts

Existing vision and language models often try to learn representations of concepts in isolation and do not always have an underlying, unified concept representation. For example, when given a question about an image in VQA, concepts may have many different mentions in the questions (*e.g.,* "*car*", "*sedan*", "*hatchback*") and models can struggle to link these mentions to a single, unified concept (*e.g.,* "*car*"), making them brittle and inconsistent to questions containing different mentions. This inconsistency extends to different phrasings of questions as well (*e.g.,* "*What sport are they participating in?*" versus "*What sport are they taking part in?*"). Such phenomena hinder the generalization capabilities and robustness of these models. We explore learning to link different mentions of concepts and phrasings. We choose to study this in the context of VQA since the task has text inputs that commonly use different mentions of concepts or wordings (*i.e.,* paraphrases of the same question).

We demonstrate that models have difficulty handling different mentions of the same concept and phrasings of questions. To precisely evaluate these capabilities, we introduce a novel benchmark called VQA Perturbed Pairings (VQA P2), where each question is a *perturbed* version of a human-written question. The perturbations that are applied to the questions preserve the coherence and semantics of the original questions, while introducing non-trivial variations to the questions that examine the ability to link different mentions of concepts as well as handle different phrasings of

questions. We utilize large-scale linguistic knowledge bases to mine linguistic perturbations that we apply to existing questions, which allows us to precisely control what kinds of perturbations are applied. Further, we design a new regularization framework to improve the consistency of compositional VQA models based upon a novel inductive bias that questions which are related to one another should have similar activations within the network. Our method is a multi-task approach that treats each type of perturbation as a task and optimizes for the different types of perturbations individually. We show the effectiveness of our method on a variety of compositional architectures and demonstrate that our framework can help models be more consistent to different kinds of linguistic variations.

1.2.3 Modeling Associations Between Concepts in Complex Domains

Most existing work for vision and language problems target settings from everyday life, where background knowledge is not necessarily required. Consequently, datasets and approaches primarily examine or enhance models ability to describe the visual content in simple and concise ways. While this is certainly a challenging, multimodal task, more work can be done towards enhancing the utility of captioning systems by applying them to more real-world scenarios.

When operating in complex domains, such as medicine or news, vision and language models must not only consider the concepts, but also the associations or connections between concepts. Models must understand the concepts necessary to yield correct predictions, which may be found in the inputs as well as background knowledge, as well as how they are related to one another. For example, in healthcare, one may want to utilize a captioning system to create medical image reports [3, 4]. In this setting, models must leverage concepts identified in the image, retrieve relevant concepts from background knowledge, and amalgamate the information conveyed by these two sources to produce factually accurate predictions.

In Chapter 6, our work expands the video captioning task to the news domain. For this task, models must generate natural language descriptions of news videos. Here, one of the foremost challenges is the need to include important concepts in the descriptions, such as named entities. To facilitate the training and evaluation of models for knowledge-aware video captioning, we collect a novel news video captioning dataset, where each example contains a video, meta-data about the video, and a natural language description that discusses the content of the video. We develop an approach that uses video meta-data to retrieve topically related news documents for a video and extracts the events and named entities from these documents. Then, given the video as well as the extracted events and entities, we generate a description using a novel model called the Knowledge-aware Video Description network. The model learns to incorporate entities found in the topically related documents into the description and the generation procedure is guided by the

event and entity types from the same documents. Our model effectively learns the complementary connections between the retrieved knowledge elements and the visual information.

1.3 CONTRIBUTIONS

- We develop a contrastive learning method that trains VQA models to ground concepts mentioned in a given question to image regions. Our approach does not require strong supervision (*e.g.,* word-region correspondence annotations) or data external to the VQA task (*e.g.,* image captioning data). We propose a new evaluation setting for VQA models that examines the ability to compose skills, which are visual tasks required to answer questions, and concepts mentioned in questions.

- We present a novel benchmark to evaluate the consistency of VQA models towards different expressions of the same concept as well as other variations in questions wordings. Our benchmark utilizes controlled linguistic perturbations to examine consistency to different kinds of variations. We introduce a regularization framework for compositional models that enforces consistency of the answers as well as of the intermediate representations of the models.

- To the best of our knowledge, our knowledge-aware video captioning task is the first of its kind and the only captioning work to target real-world, news videos. Not only is this a promising application, but it is also exemplifies the complementary nature of utilizing knowledge of concepts in multiple modalities. We also present an end-to-end trainable model for this task that leverages the video as well as relevant background knowledge.

1.4 DISSERTATION STRUCTURE

This dissertation is structured as follows:

- Chapter 2 provides some basic information about concepts in vision and language as well as the task definitions with which we operate.

- Chapter 3 gives a literature survey of representation learning for vision and language, both unimodal as well as multimodal. It also provides background information related to image/video captioning and VQA approaches.

- Chapter 4 introduces our work on learning concept grounding within VQA and our skill-concept composition setting that examines the generalization capabilities of VQA models.

It elaborates our contrastive learning approach for separating skills and concepts within the representations of models.

- Chapter 5 expounds our work towards improving the consistency of VQA models to different concept mentions and question phrasings. It describes the creation of a new benchmark for measuring consistency as well as our approach to improving consistency by regularizing the internal representations of modular architectures.

- Chapter 6 presents our knowledge-aware video captioning efforts. It discusses our creation of a news video captioning dataset as well as an end-to-end trainable method that can learn associations between the video content and concepts in background knowledge to generate captions/descriptions.

- Chapter 7 summarizes the conclusions and contributions of this dissertation. Further, it discusses some remaining limitations and offers some directions for future research.

CHAPTER 2: BACKGROUND

2.1 DEFINING CONCEPTS IN VISION AND LANGUAGE

Broadly speaking, "*concepts*" are interconnected artifacts that we use to understand the world around us [5]. For instance, there is the concept of "*car*", which can appear in images as a bright red Ferrari or a beige Ford Pinto. Similarly, in text, this concept can be referred to by a variety of terms, such as the make/model (*e.g.*, "*Ferrari*"), "*automobile*", etc. These concepts are interrelated, sharing connections between one another: a "*car*" is a "*vehicle*" with "*wheels*" and can be an artifact used in a "`transport`" event. Although the specific semantics of how concepts are related can differ from one domain to the next, these relationships hold important information that should be represented in vision and language models in order to address complex problems like VQA, captioning, or knowledge-aware tasks. Concepts and their relationships can be represented or encoded in a variety of ways. External knowledge [6, 7], co-occurrences [8], or inductive biases within the model or training procedure [9] can all be used to encode these elements.

2.2 TASKS

Vision and language covers a broad range of tasks that differ from one another in sometimes subtle, yet important, ways. We briefly describe some of the prominent vision and language tasks:

- **Visual Description Generation (Captioning) [10, 11]:** Given an input image (image captioning) or video (video captioning), the goal of this is to generate a natural language description of the visual content.
- **Visual Question Answering (VQA) [1, 12]:** The inputs for this task are an image/video and a question about the image/video. Approaches to this are expected to produce the correct answer to the question based on the image/video. There are a number of different variations of this task that query specific information, such as TextVQA [13] where models must read text in the image to answer the questions.
- **Phrase Grounding [14]:** Models are given an input textual phrase and an image. The output should be the location in the image that the textual phrase refers to. While this can be treated as a task of its own, phrase grounding is typically an integral part of other tasks that use language to query an image or video.
- **Visual Dialog [15]:** This task aims to simulate an agent (model) conducting a conversation with a human. Models are given an image, a dialog history, and a question as input, and

must ground the question to the image, consider the dialog context, and produce the correct answer. In a sense, this task can be viewed as requiring a fusion of the capabilities of captioning and VQA.

- **Vision-and-Language Navigation [16]:** Here, an agent must maneuver through an environment based on a set of natural language instructions. Agents must interpret the instructions based on their egocentric view of their position in the environment and move accordingly. Ultimately, the agent must reach a goal point in the environment.

In this dissertation, there are two tasks that we focus on VQA and video captioning.

2.3 DATASETS AND SETTINGS

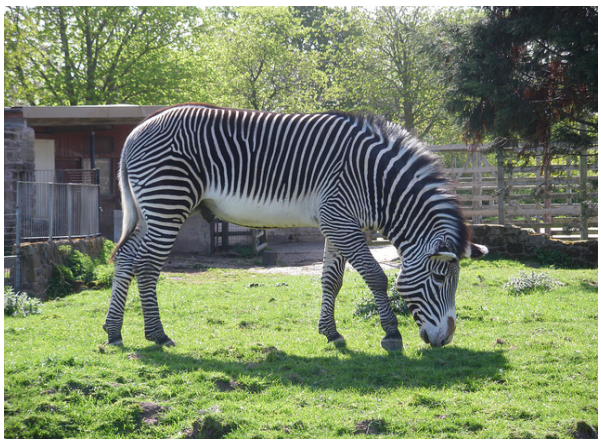
In this section, we elaborate some of the details of the main datasets for VQA and video captioning as well as some of the particulars of how these tasks are performed with the different datasets.

2.3.1 Visual Question Answering (VQA)

Although many datasets for VQA exist, there are two datasets in popular use that form the basis of many current VQA efforts, which we discuss in this section. Furthermore, we will also discuss some of the common settings used for approaches and evaluation.

VQA [1, 12] This dataset has two version VQA v1 [1] and VQA v2 [12]. Both versions have open-ended questions about images, where each question has ten answers and each answer can be a single word to a short phrase. The images contained in the both versions of the dataset fall under two settings: real images and abstract scenes. We focus on the real image data as this is most relevant to the work in this dissertation. The images are collected from MS COCO [10] and contain a variety of objects. Statistics for VQA v1 are given in Table 2.1. In VQA v1, for each image, the authors collect three questions per image using human annotators, where each question has free form answers. As shown in Figure 2.1, the questions include ones that query more standard vision tasks (*e.g.*, "*What is this animal?*") as well as ones that require some amount of commonsense knowledge (*e.g.*, "*Is this a recent photograph?*"). The result of the collection process of VQA v1 is a large set of image-question-answer tuples that can be used to train models. However, it has been discovered that the language component of some vision and language datasets, including VQA v1, offer a strong prior that can result in apparently strong performance without truly considering the

Is this a recent photograph? | no



What is this animal? | zebra

Figure 2.1: Examples from VQA v1 that require commonsense (left) and query more standard vision capabilities, such as subcategory recognition (right).

Dataset	Split	# Images	# Questions	# Answers	# Q/I	# A/Q
VQA v1	Train	82,783	248,349	2,483,490	3.0	10
	Val	40,504	121,512	1,215,120	3.0	10
	Test	81,434	244,302	2,443,020	3.0	10
VQA v2	Train	82,783	443,757	4,437,570	5.4	10
	Val	40,504	214,354	2,143,540	5.3	10
	Test	81,434	447,793	4,477,930	5.5	10

Table 2.1: Statistics for the real image data in VQA v1 and VQA v2. Q/I = Questions per Image, A/Q = Answers per Question.

visual content [17, 18, 19, 20, 21, 22]. For example, in VQA v1, 39% of questions beginning with "*How many*" have the answer "2" [12].

VQA v2 is designed to mitigate some of these issues with language priors by creating a balanced dataset that elevates the role of the visual content. To do so, for each image-question-answer tuple, the authors have humans identify a different image that is similar to the original image, but yields a different answer to the question. The idea behind this being that the image will be a primary determining factor of the answer. The pairs of images are called complementary pairs and each new image-question-answer tuple containing a complementary image and answer are added to the dataset. This results in an additional 195k training, 93k validation, and 203k testing examples. Table 2.1 shows statistics of the resulting dataset.

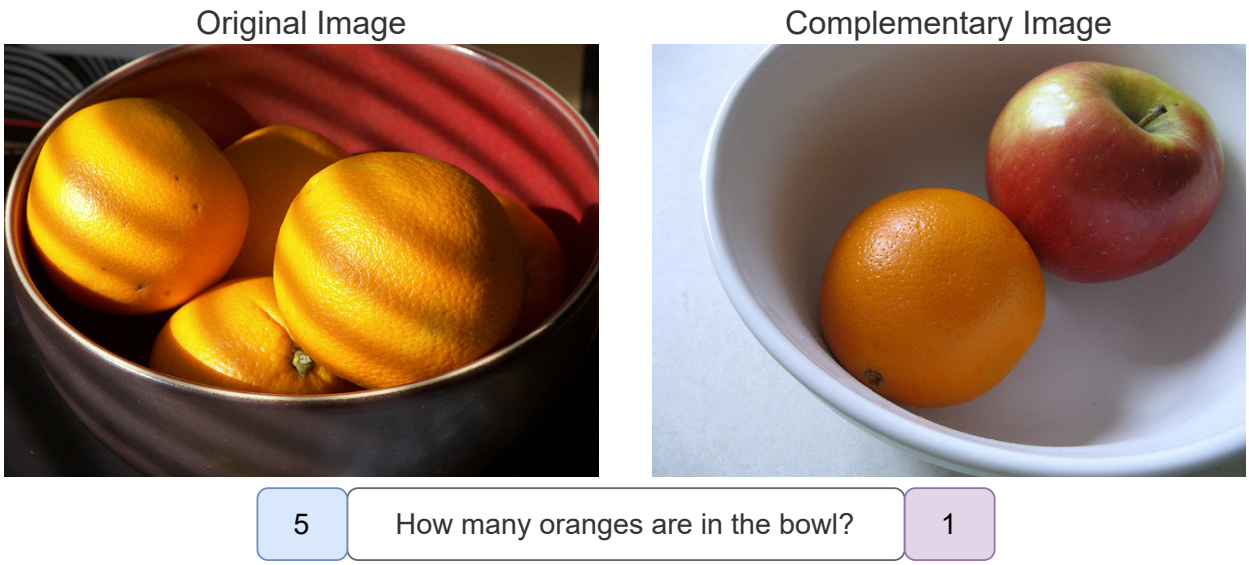


Figure 2.2: Examples of complementary pairs that appear in VQA v2.

CLEVR [23] CLEVR is designed to be a diagnostic dataset that probes the reasoning capabilities of VQA models, de-emphasizing the recognition and commonsense aspects of the VQA v1/v2 dataset. It elevates the evaluation of the actual reasoning capabilities of VQA models and minimizes biases that may be present in other datasets. All images, questions, and answers in CLEVR are automatically generated. The images are generated by randomly sampling a scene graph [24], which specifies the objects as well as their attributes and their spatial relationships relative to other objects, and using 3D rendering software to generate an image based on the sampled graph. The objects span three shapes (cube, sphere, and cylinder) with two absolute sizes (small, large), two materials (metal, rubber), and eight colors. There are four spatial relationships (left, right, behind, in front) that can be used in the scene graphs. Meanwhile, a template-based approach is used to generate questions. The authors define 90 question families, where each family has one functional program that can be executed on a scene graph and 4 question templates. To generate a question for a given image and scene graph, a question family is first selected and values are chosen from the scene graph to fill the parameters of the functional program. Next, the functional program is executed on the scene graph with the chosen parameters, yielding an answer. Finally, the same parameters are used to fill on of the text templates, which produces a question. Other post processing methods are employed to ensure the quality of the dataset, such as rejection sampling to ensure an almost uniform answer distribution. Examples of the images, questions, programs, and answers are shown in Figure 2.3 and statistics are provided in Table 2.2.

The CLEVR dataset has facilitated the development of a number of modeling techniques that emphasize interpretability. Namely, Neural Module Networks (NMNs) [25, 26, 27, 28]. NMNs

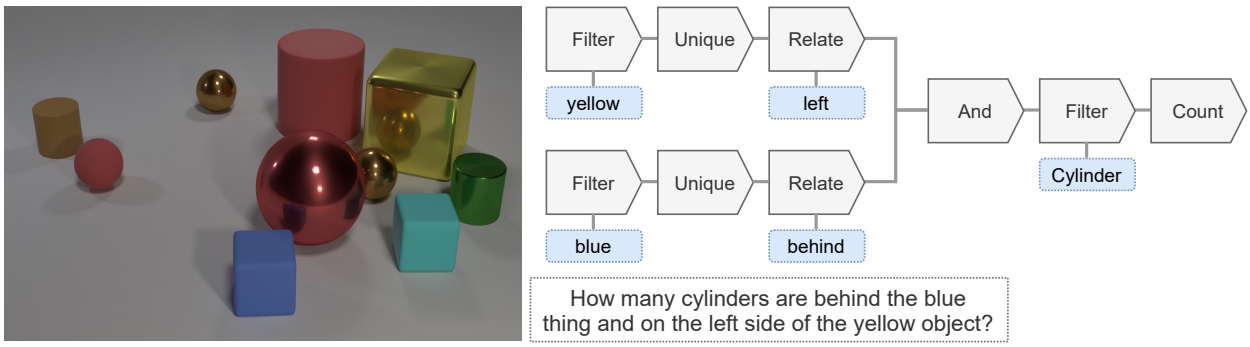


Figure 2.3: Illustration of the data in CLEVR, inlcuding an image, functional program, and question.

Split	# Images	# Questions	# Unique Questions	Overlap with Train
Train	70,000	699,989	608,607	-
Val	15,000	149,991	140,448	17,338
Test	15,000	149,988	140,352	17,335

Table 2.2: Statistics for the CLEVR dataset.

are models comprised of a set of reusable modules, each of which is responsible for an elementary operation (*e.g.*, find an object, logical operations). Given a question, these networks decompose the reasoning process into a program of intermediate operations and then execute the program using modules the modules. These models offer interpretability since every reasoning step is explicitly modeled. These networks are very effective for CLEVR and perform well on real image VQA data [27, 28]. We do not experiment with CLEVR in this dissertation, but we elaborate the details here to elucidate part of the motivation and background of NMNs.

Evaluations and General Approaches For both of these datasets, VQA v2 and CLEVR, the problem is typically modeled as a classification task [29]. The set of possible answers is heuristically determined based on the frequencies of each unique answer [30], which can yield a relatively large set (*e.g.*, ∼3,000 for VQA v2). VQA performance is usually measured using accuracy. For CLEVR, this amounts to a standard accuracy calculation. However, for VQA v1 and VQA v2, this calculation is slightly different due to the open-ended nature of the dataset and the fact that each question has ten answers. For a question, Q, let A' be the answer predicted by the model. Accuracy on VQA v1 and VQA v2 is given by:

$$\text{accuracy} = \min\left(\frac{\text{\# humans who answered } A' \text{ for } Q}{3}, 1\right). \tag{2.1}$$

This accuracy is averaged over all $\binom{10}{9}$ combinations of answers. Equation 2.1 implies that, for a given combination, an answer A' is considered 100% correct if at least three human annotators provided the same answer. This also awards partial credit for answers that agree with some, but not all, of the annotators.

In general, VQA approaches follow a pipeline of, first, encoding the question, Q, and image, I, with unimodal encoders ϕ_l and ϕ_v, respectively. Once representations are obtained for each input modality, the rest of the model, $f(\phi_l(Q), \phi_v(I))$, is used to combine the visual and textual inputs and predict an answer. The answer prediction is typically posed as a classification task over the set of possible answers, so the model outputs $\hat{a} \in \mathbb{R}^{N_a}$ dimensional vector, where N_a is the number of possible answers. The VQA loss/objective is then computed as a binary cross-entropy loss [30] given by

$$\mathcal{L}_{CE}(\hat{a}, a|I, Q) = -\sum_{i=1}^{N_a} a_i \log(\hat{a}_i) + (1 - a_i)\log(1 - \hat{a}_i), \tag{2.2}$$

where a is a vector encoding of the ground truth answer(s). When there are multiple possible answers, this acts as a soft loss that accounts for the multiple correct answers. This is done by creating a ground truth vector where each slot represents the VQA accuracy (Equation 2.1) of the corresponding answer [30]. Otherwise, a standard multi-class classification loss can be used. The VQA work in this dissertation (Chapter 4 and Chapter 5) utilizes the soft loss in Equation 2.2, which is referred to as the VQA loss or VQA objective. Innovations at each step in this process have advanced the state-of-the-art and continue to do so.

2.3.2 Video Captioning

Similar to the previous section, we describe two popular video captioning datasets as well as common settings used for the task. We very briefly describe the datasets and emphasize the general approaches since the background of the approaches is more relevant to this dissertation.

MSVD [31], MSR-VTT [11] MSVD is one of the earliest video captioning datasets and is still used in many captioning efforts [32, 33, 34]. MSR-VTT is a more recent popular dataset that goes further to expand the diversity of the content and scale of video captioning datasets. Both datasets are collected using Amazon Mechanical Turk, where workers are instructed to create a sentence describing the video content. Table 2.3 gives some statistics for both datasets. For MSVD, there is an average of 35 descriptions per video. Meanwhile, MSR-VTT has 20 descriptions per video. These datasets cover broader categories than other datasets, such as the TACoS cooking video

Dataset	# Videos	# Clips	# Sentences	# Words
MSVD [31]	-	1,970	70,028	607,339
MSR-VTT [11]	7,180	10,000	200,000	1,856,523

Table 2.3: Statistics for the MSVD and MSR-VTT datasets.

datasets [35, 36, 37].

Evaluations and General Approaches When evaluating video captioning approaches, a suite of text generation scores are used. Each score essentially compares a machine generated sequence to one or more human-written sequences and computes a similarity score based on different criteria. The criteria used for computing the similarity is largely where these different metrics differ from one another. There are five particularly prevalent evaluation metrics that are used:

- **BLEU [38]** is a n-gram precision based metric for machine translation. Meaning that it mainly takes into account the proportion of the matched n-grams and the total number of n-grams in the machine output. This is typically calculated for $n = 1, 2, 3, 4$ and then combined using a geometric mean. n-gram precision alone does not penalize very short outputs, so a brevity penalty is often added for machine outputs that are much shorter than the human ones.

- **METEOR [39]** is a metric designed to improve upon some of the shortcomings of BLEU for evaluating machine translation approaches. It is computed as the harmonic mean of unigram precision and recall. Within the harmonic mean, recall is weighted higher than precision. One of the shortcomings of BLEU that METEOR addresses is the lack of recall in the metric. Recall can be important to evaluating the quality of generated outputs because it represents the coverage that the generated outputs have of the ground truth. METEOR also matches morphological variants and synonymous, which can make it particularly useful for settings where only one ground truth output is available.

- **ROUGE-L [40]** is intended to evaluate generated summaries, but can serve as a general scoring metric. This metric is based on a F1 score, where the computed precision and recall are based upon the ratio of the length of the longest common subsequence to the length of the generated and ground truth outputs, respectively.

- **CIDEr [41]** is based on human consensus and utilizes TF-IDF statistics to compute a score. CIDEr is meant to evaluate the output of image and video captioning models. For a generated output, all n-grams are computed and a weight is assigned to each n-gram based on TF-IDF

statistics. A vector is then formed that contains the scores of all n-grams that appear in the generated output. Another vector is formed for the ground truth output in the exact same way. The cosine similarity is computed to measure the similarity between the generated and ground truth outputs.

- **SPICE [42]** is a metric designed to overcome some of the limitations of the previously discussed n-gram overlap metrics. Specifically, n-gram metrics can award high scores even if the semantic meaning of the generated and ground truth output are different. For example, for a given image, if generated output is "*A old man sitting on top of a stool.*" and ground truth output is " *A cat sitting on top of a car.*", then the generated output will have relatively high n-gram overlap scores because of the matching n-gram "*sitting on top of a*". However, if, for a different image, generated output is "*A pizza sitting on a white plate next to eating utensils.*" and ground truth output is "*A meal and a beverage on a table with utensils.*", then the generated output would obtain worse scores even though the two captions/descriptions have the same meaning and describe the same image. To address this issue, SPICE uses semantic parsing to parse the generated and ground truth outputs into scene graphs [24], and then computes F1 scores based on the overlaps between the scene graphs.

These metrics capture different perspectives of the generated output and, in combination with one another, offer a more complete view of how well the generated text match the expected output. While suitable in some ways, measuring performance of natural language generation approaches is difficult in general [43]. This is particularly true for open-ended generation tasks. For knowledge-aware generation tasks, as presented in this dissertation, these metrics must be complemented with other knolwedge-centric metrics to measure the veracity of the output text.

A standard approach to video captioning is a sequence-to-sequence (seq2seq) approach, where an encoder-decoder model [44, 45] is used to encode the video sequence and then decode the hidden representation(s) into a text sequence. The model is trained to estimate the conditional probability of an output sequence, given an input sequence: $p(y_1, \ldots, y_M | x_1, \ldots, x_N)$. One frequent architecture choice is to utilize a recurrent neural network (RNN), such as a GRU [44] or a LSTM [46] cell, as both the encoder and decoder. Given an input sequence $X = (x_1, \ldots, x_N)$ and an output sequence $Y = (y_1, \ldots, y_M)$, the model encodes X into a sequence of hidden states $H = (h_1, \ldots, h_N)$. The model defines a distribution over the output sequence given by

$$p(y_1, \ldots, y_M | x_1, \ldots, x_N) = \prod_{t=1}^{M} p(y_t | s_t, H), \tag{2.3}$$

where s_t is the decoder state at step t. In video captioning, the input sequence would be a se-

quence of frame representations and the output sequence would be a sentence. Let θ be the model parameters and let $\mathcal{D}$ be the dataset of video-caption pairs. To train the model, we typically try to maximize the probability of the correct sentence [45, 47] by maximizing

$$\theta^* = \arg\max_{\theta} \sum_{(X,Y)\in\mathcal{D}} \log p(Y|X;\theta). \tag{2.4}$$

This log-likelihood (or cross-entropy) formulation is also used in other seq2seq problems, like machine translation [44, 48]. When training video captioning models, if multiple descriptions for each video are available, then each video-caption pair is treated as an individual training sample. Our video captioning work in this dissertation (Chapter 6) employs the objective in Equation 2.4 for training.

CHAPTER 3: LITERATURE REVIEW

3.1 REPRESENTATIONS OF VISION AND LANGUAGE

3.1.1 Unimodal Representations

Unimodal representations of vision and language underpin the approaches to downstream vision and language tasks. The proper input representations of each modality has a noticeable influence on the performance [49, 50, 51]. Throughout this dissertation, we employ different variants of these unimodal encoding strategies to obtain useful representations that capture the semantics of the concepts and context in the inputs.

Vision Convolution Neural Network (CNN) models are often used for encoding visual information. CNNs, pre-trained on ImageNet [52] or large-scale video datasets [53], can be used to extract image or spatio-temporal features that can be easily utilized in downstream tasks, such as captioning [7, 47, 54, 55] or retrieval [56, 57]. Additionally, initializing the layers of a network with these pre-trained weights can offer performance benefits, as in object detection [58]. As of late, for a number of vision and language tasks, in particular image captioning and VQA, object features [49] have become a go-to option since they can provide rich, localized information about the objects in a scene. Moreover, when trained using fine-grained annotations like those in the Visual Genome [24], performance can be even further boosted for tasks like VQA because these features are able to capture more of the nuances of the objects and attributes that are frequently necessary for answering questions. However, recent work has shown that these benefits may not be inherent to object features as much as they are a result from the fine-grained training [51]. For videos, most approaches either use CNNs to encode each frame and Recurrent Neural Networks (RNNs), such as Long Short-Term Memory networks (LSTMs) [59], to encode temporal information into the frame representations or 3D convolutions [60, 61] to obtain spatio-temporal features directly. Video encoding can also leverage weakly supervised objectives to enhance the learned representations for tasks like action recognition [62].

Language Recent methods of representing words rely heavily on the distributional hypothesis [63] in order to induce low-dimensional vector representations of words. Classical approaches to representing words often focus on different factorizations of co-occurrence matrices [64], while more recent approaches utilize neural networks to learn representations of words that can be used to predict the surrounding context or vice versa [65, 66]. Learning standalone semantic representa-

tions of words is important, but representing words in specific contexts is also of vital importance for a variety of tasks [67]. More modern approaches to doing so employ RNNs or transformers [68] to encode entire sentences or phrases, which has important applications in vision and language tasks, like image retrieval, where an entire sentence can be given as input and the full meaning of the sentence must be considered. These encoders along with input word embeddings, pre-trained or otherwise, can be learned jointly and/or fine-tuned with the target task. Currently, large-scale pre-training of contextualized encoders has gained popularity [67, 69]. These contextual encoders can be trained on prodigious amounts of text in an unsupervised manner, and can significantly boost a variety of downstream tasks. These models are trained via masked language modeling where the model must predict the correct word given a specific context. Transformers only rely on attention mechanisms [48, 70], fully connected layers, and residual connections [71], without requiring a recurrent state in order to perform auto-regressive tasks.

3.1.2 Multimodal Representations

An important step for many vision and language tasks is to link corresponding appearances of a concept across modalities (*e.g.,* the word "*dog*" to an image region of the dog), which is commonly known as grounding [14]. When working on grounding as a task in itself, apart from another downstream task, the problem is framed as an embedding space learning task and the goal is to learn a multimodal embedding space where distance metrics can be used to indicate which words and regions correspond to each other [9, 72, 73, 74]. Key to many of these approaches is the use of weak supervision from aligned image-text pairs to distill correspondences. One approach to learning multimodal embedding spaces is to utilize multiple instance learning [57], contrastive learning [72, 75], or other unsupervised/weakly supervised objectives to train encoders that learn multimodal representations [8, 9, 73, 76, 77]. This can be done via a two-stream approach [72, 73, 75], where visual and textual inputs are input into distinct unimodal models and then projected into a multimodal embedding space, or a single stream approach [8, 9, 77], where both visual and textual inputs are given to a single multimodal encoder. Either approach, though typically the single stream approach, may be utilized as encoders for other downstream tasks, such as VQA, and can be scaled up to pre-train on large quantities of data [8, 9, 73, 76, 77]. Alternatively, learning correspondences between image regions and text can be induced while learning target tasks, like VQA [27, 78], captioning [6, 7, 54], or multimedia event extraction [74]. For example, in multimedia event extraction [74], an alignment step is performed to align the semantic graphs of the vision and language inputs. The work elaborated in this dissertation mainly focuses on learning grounding or other cross-modal associations jointly within specific tasks, namely captioning and VQA.

3.2 IMAGE AND VIDEO CAPTIONING

In general, captioning is the task of taking an image or video as input and generating a textual description of the visual content. Approaches to this problem often use an encoder-decoder setup, where an encoder produces a representation of the visual inputs, which is then given to the decoder to generate the text [45, 47, 49, 54, 79, 80, 81, 82, 83]. In image captioning, images are either represented as a single vector from a full-connected layer of a CNN [47] or a set of vectors from a CNN or object detector [58] that can be used via an attention mechanism on the decoder [49, 54]. For video captioning, the temporal dimension of the inputs adds another layer of complexity to handling the problem. Many efforts go towards learning and effectively utilizing spatio-temporal representations within captioning models using 3D CNNs and/or spatio-temporal attentions [45, 55, 79, 82, 84, 85, 86, 87].

Most work on captioning focuses on general domain images and videos that contain sets of everyday objects and actions. Large datasets have been constructed to investigate captioning in these general settings, including MS COCO [10], Flickr30k [88], Visual Genome [24], MSVD [31], MSR-VTT [11], and ActivityNet Captions [89], where each dataset generally contains a large number of image/video-caption pairs. Beyond the general domain, there are efforts in image captioning to personalize captions [90], incorporate novel objects into captions [91, 92], and perform open domain captioning [83, 93] where the objects, attributes, and actions are not from a fixed set.

Incorporating background knowledge in captioning, known as knowledge-aware or entity-aware captioning, is another promising line of work [6, 7, 94, 95]. In the knowledge-aware setting, models must learn the connections between the visual information and not only the generated text, but also the relevant background knowledge. This knowledge is obtained from sources relevant to the visual inputs, such as either retrieved documents or documents that contain the input image/video. The additional context can be used by the model as a source of specific knowledge like named entities and events that would be otherwise difficult, or intractable, to identify from the visual inputs alone.

3.3 VISUAL QUESTION ANSWERING

In Visual Question Answering (VQA) [1], given an image and a question about the image, models must predict the correct answer. VQA can be posed as an open-ended task, where the answer must be generated, or a multiple choice task, where the answer must be selected from a given set of answers. Most often, and for the purposes of this dissertation, open-ended VQA is the selected setting. Although open-ended VQA is a generative task, most approaches to VQA treat it as a classification problem where an answer is predicted from a large bank of possible answers to

any question [30]. Since the introduction of the task and first dataset [1] (VQA v1), follow up work has been done to reduce unimodal biases (VQA v2 [12]) and out-of-distribution answers (VQA-CP [96]), so that the model must consider the visual content more in order to answer the question correctly. In addition to issues of biases in the datasets, robustness of VQA models to variations in the inputs is also an issue [97, 98, 99], where models yield different answers depending upon, for example, how the question is worded. Lastly, other efforts present datasets that test the reasoning capabilities of models by creating questions that require the model to follow complex referential chains, such as CLEVR [23] and GQA [100], which has lead to a wealth of work on interpretable methods for visual reasoning [25, 26, 27, 28, 101, 102].

Effective VQA approaches must understand the question and then reason about the visual content. Most approaches to VQA first obtain an encoding of the question and images, followed by an attention of some kind over the visual information and then a fusion step. The encoding of the image is typically done using either a CNN [51, 71] or object/attribute detection [49], both of which yield a set of image region features. Question representations are computed using word embeddings [65, 66] and often followed by a recurrent or transformer layer to obtain contextualized representations of each word. Much of the design of VQA approaches lies in the attention and fusion steps. Bilinear [103, 104] and co-attention [78, 105] methods to these steps have proven to be effective ways to learn high-capacity multimodal representations. More recently, transformer models, pre-trained or otherwise, are popular architecture choices for VQA models due to their flexibility and representational power [8, 9, 73, 76, 77, 78]. Another important class of VQA models are compositional and interpretable architectures, such as Neural Module Networks (NMNs) [25, 26, 27, 28, 101] or neural state machines [102]. These light-weight models reason about the image and question by predicting and executing a sequence of re-usable modules, where each module is responsible for an elementary operation (*e.g.*, find an object or compare two objects). Once the sequence of modules is executed, the result is combined with the question representation to predict an answer. In this dissertation, we utilize transformer-based models and NMNs at different points in our methods.

3.4 NATURAL LANGUAGE GENERATION

Natural Language Generation (NLG) is an important problem in AI as it provides an effective means for machines and humans to communicate. This area of natural language processing has a wide variety of tasks and applications, including captioning. The body of work in NLG is vast so, for the purposes of this dissertation, we do not exhaustively review NLG. Instead, we focus on a few specific aspects that are relevant to our work that are from dialogue systems (or conversational

AI) [106] and abstractive summarization [107], which are two notable areas of NLG.

Dialogue Systems Classical dialogue systems utilize rule-based approaches [108] or n-gram models [109]. However, more recent approaches based on neural networks have shown strong performance [106, 110, 111, 112]. An important aspect of these systems is content control. One mechanism for achieving such control is a gating mechanism on the decoder that takes in a vector encoding of the intended response information, such as a binary or one-hot vector [106, 111]. This conditions the model on the elements that are expected in the output. During the generation process, the model can adjust the values of the vector using the gating mechanism, which can help learn when to incorporate each desired piece of information in the output.

Abstractive Summarization This task seeks to generate a summary of some input document, usually a long text passage, where the summary is a shorter text passage that captures the key information of the input content [107]. An important problem in this area is content incorporation [107, 113], since models are expected to generate summaries that are consistent with the input document. A means for addressing this challenge is utilizing a pointer network [107, 113, 114] (or copy mechanism [115]) to copy sequences of text from the input into the output. This mechanism is quite effect at incorporating sequences in the output and can be very useful for content selection in a variety of problems, such as knowledge base description generation [116].

CHAPTER 4: GROUNDING CONCEPTS FOR SKILL-CONCEPT DISENTANGLEMENT IN VQA

4.1 MOTIVATION

The ability to ground concepts mentioned in questions is very important to generalizable VQA models. Without this ability, VQA models may simply leverage superficial correlations in the training data [17, 18, 19, 20, 21, 22]. In this chapter, we explore a new evaluation setting to examine the compositional capabilities of VQA models. We propose a method to improve grounding as well as the compositional performance in VQA Models without requiring strong supervision or data external to the VQA task.

When humans answer questions, such as in VQA, we first interpret the question, dissecting its content into parts (like concepts, relationships, actions, question types), and then we select and execute the *skill* (or plan/program) necessary to produce an answer based on this information and the relevant knowledge base (*e.g.,* the image) [117, 118, 119]. The skills needed to produce an answer are general and can be applied to (composed with) many types of question-specific content. For example, if one can answer questions about "*colors*" for a variety of objects as well as recognize and answer questions about "*cars*", then questions like "*What color is the car?*" should be straightforward to answer even if this specific composition has yet to be seen. This ability of seamlessly adapting and composing conceptual representations with skills is imperative to demonstrating true understanding of VQA and learning to generalize from less labeled data.

Compositionality is recognized as one of the essential properties of human cognition [120], but more research is still needed on incorporating compositionality into models and developing data-efficient, generalizable systems. While much progress has been made to achieve better performance on standard VQA test benchmarks [12], most state-of-the-art models are still designed without any notion of built-in compositionality and tend to entangle skills and concepts in their learned representations. Some previous work has studied the lack of generalization ability of VQA models, and evaluated models using test splits with different answer distributions from the training data. However, this measurement only indirectly addresses the central issue (lack of compositionality), which manifests itself as poor generalization and over-reliance on language priors [96, 121].

To address these issues, we first propose a new evaluation setting to view VQA compositionality, called *skill-concept composition*, and a new evaluation procedure that directly targets how VQA models can generalize to novel compositions of skills and concepts. This setting is motivated by our observation that, to answer a natural question on real images requires the understanding of two distinct elements: 1) the visual concept referred to by the question; and 2) what information we need to extract from the referred concept. We elucidate this in Section 4.2 and evaluate a number

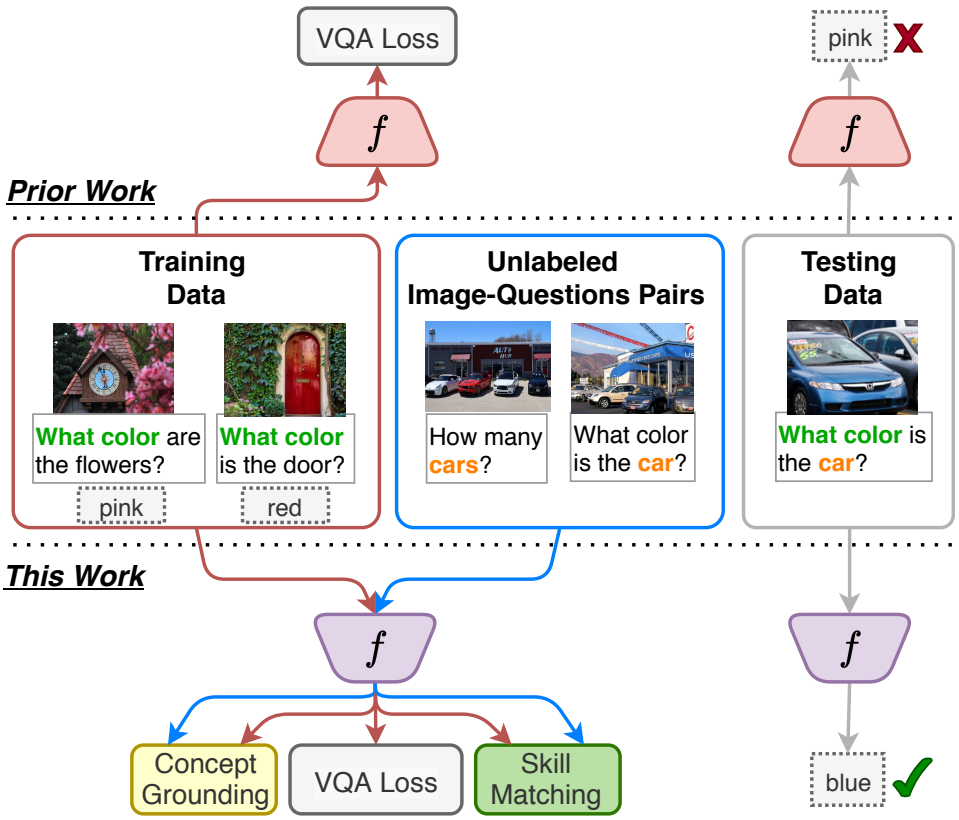


Figure 4.1: We propose a new view of compositionality in VQA that explores the ability to answer questions about unseen compositions of skills (*e.g.*, color) and concepts (*e.g.*, car). We present a method that learns to separate skills and concepts that can utilize both labeled and unlabeled image-question pairs in order to generalize to novel questions with new skill-concept compositions and new concepts.

of VQA architectures using this setting and demonstrate that the existing models have much to improve upon to answer novel questions.

We propose a novel approach to improve generalization that utilizes contrastive learning to separate skills and concepts within the internal representations of a model, while jointly learning to answer questions. We use grounding as a proxy to separate concepts so that the model learns to identify a concept in both the question and image, regardless of the specific context. Akin to weakly supervised grounding [72, 75], we train the model to recover a concept mentioned in a given image-question pair by contrasting the multimodal representation of the masked concept word to the multimodal representations of words in other questions. We utilize a new way to curate positive and negative examples for the contrastive loss so that the model learns to predict the concept based on relevant visual information rather than using superficial contextual cues. Additionally, our approach learns to separate skills from concepts by contrasting question repre-

sentations that have the same or different skills. These properties are learned jointly alongside the VQA objective, on top of state-of-the-art models, and are generalizable to new architectures.

Some advantages of our approach are: 1) We learn grounding in a self-supervised manner using the VQA data *alone*, without external annotations. This is in contrast to previous approaches with similar goals that incur large expenses due to annotation requirements [122, 123]. 2) Our method does not rely on answer labels to learn skill-concept separation, so we are able to use *unlabeled* image-question pairs to learn these properties. Consequently, we are able to acquire new concepts and learn to answer questions about them without having labeled data with these concepts, which is pivotal for generalizing to a new domain or novel instances. Moreover, we focus on data-efficient methods and do not use prodigious amounts of data external to VQA, like pre-training approaches [9, 73, 76], which is expensive to obtain and can require prior knowledge of the domain and/or concepts in order to perform well [124, 125].

4.2 SKILL-CONCEPT COMPOSITION IN VQA

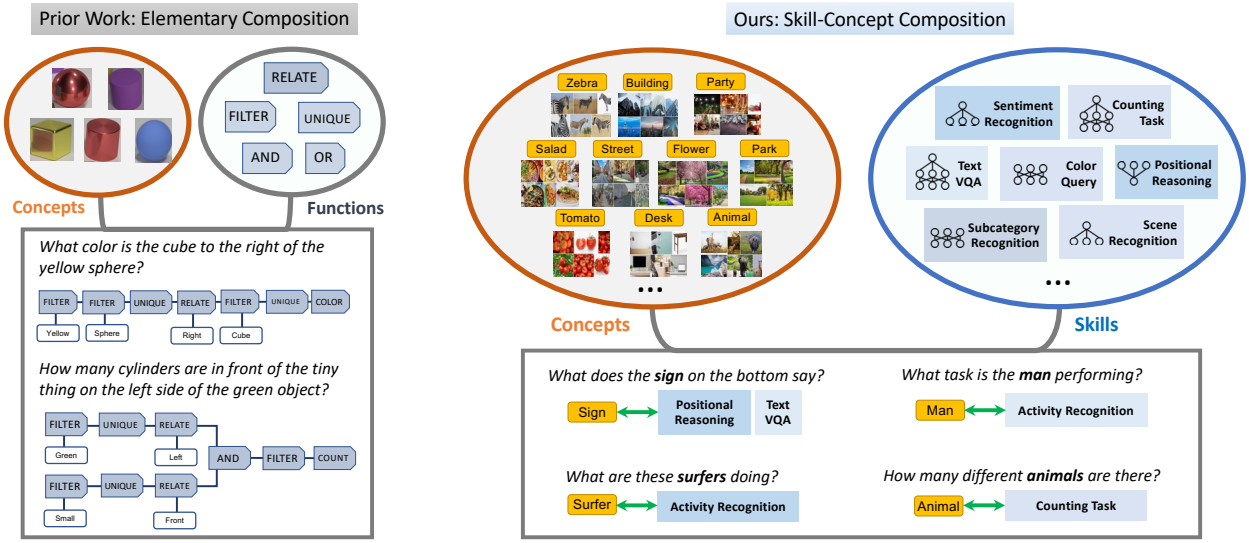


Figure 4.2: Illustration of skill-concept composition, a new view and evaluation setting for compositionality in VQA.

We propose a novel, compositional view of VQA, called *skill-concept composition* (Figure 4.2). *Concepts* are objects and other visually grounded words or phrases. By *skills*, we refer to the collection of high-level vision understanding processes involved in answering common questions about real-world images. These skills operate on concepts and vary in terms of input/output representation complexity and the necessary reasoning processes. Our taxonomy of these skills is ex-

tracted from annotating a subset of the VQA v2 questions as well as taking inspiration from prior work on VQA. Skills are generally standalone from each other and have been studied independently in the VQA literature (*e.g.,* TextVQA [13], positional reasoning [100], or counting [126]).

We make an important yet intuitive observation about these VQA skills: to answer a question, it often requires the application of only a small number of skills (most often one) to one or more concepts in the image (Figure 4.2). This observation provides an interpretable view of a model's generalization ability to out-of-distribution data: a model should learn that a skill is a separable process that can be applied to different concepts, and that the prediction process should not be tied to specific concepts co-occurring with this skill during training. This explicit notion of skill-concept separation underlies the contributions of this paper, including a new novel-VQA evaluation method which we will introduce next, as well as a new framework to weakly learn VQA models that can answer novel questions (Section 4.3).

Novel-VQA Evaluation While conceptually intuitive, this skill-concept view offers natural ways to guide the evaluation of VQA models in terms of out-of-distribution data. In our experiments, we evaluate two novel-VQA settings: 1) answering questions on novel compositions of skills and concepts, called novel skill-concept composition VQA; 2) answering questions about concepts for which the model has not seen any answers before, called novel concept VQA.

Comparison to Existing Evaluations Our evaluation protocol is different from existing VQA benchmarks that also aim to measure VQA models' generalization ability. VQA-CP [96] builds train-test splits from VQA v2/v1 [1, 12] with distinct answer distributions by greedily dividing the questions based on their annotated question types (*i.e.,* first few words in the question: "*how many*", "*is the*",...) and answers, but this does not capture skill-concept compositions because these question types do not necessarily correspond to skills (*e.g.,* "*Is the dog waiting?*" requires action recognition and "*Is the sky blue?*" requires color recognition yet both have question type "*is the*"), and the same skill-concept composition can appear in training and testing, which violates our novel-VQA setting. TDIUC [127] evaluates VQA accuracy on different categories of task types, without regard for the concepts in the questions. [128] creates testing splits such that at least one word of a question is unseen during training, which does not consider skills. None of these benchmarks directly address and evaluate skill-concept compositionality like our evaluation protocol.

Skill-Concept Composition vs. Elementary Composition Existing compositional evaluations primarily define compositions as relational reasoning chains associated with questions [23, 100] (Figure 4.2), which are suited for learning programs of elementary operations to answer questions.

There are two main issues with applying this existing compositional view to real-image question answering. First, the concepts and their attributes are over-simplified and not representative of the diverse visual presentations in the real world. Second, the kinds of compositional questions in these synthetic datasets rarely appear in natural questions about real-world images. Our proposed *skill-concept composition* view is more applicable to real-world VQA and, as a result, can better represent the capabilities that people care about in real-image question answering.

Types of Skills To construct a comprehensive list of common skills required to answer a VQA question, we draw information from three sources: (1) our own annotation on 400 randomly selected VQA questions; (2) user study from [119]; and (3) previous work on question types [100, 127]. The user study in [119] only provides four types of vision skills. Existing work on question types have relevant information, however, the question types are not always directly translatable to our setting of skill and concept composition. For example, concept recognition is considered as a question type in [127] (object presence), but in our setting, it is considered as *concept grounding* rather than as a separate skill. Besides, existing question types are sometimes incomplete [127], or not representative of natural questions typically asked about images [100]. For instance, skills that require comparison or text reading form $\sim$6% of the questions according to our labeling results, but they are not covered in [127]. We consolidate our annotations with groupings in existing work, which results in the following set of skills:

- Color recognition: *What color hair does the woman have? What color is his shirt?*
- Attribute recognition (non-color attributes): *Is the bed made? Is this desk messy?*
- Subcategory recognition: *What kind of car is parked? What kind of animals are shown?*
- Action recognition: *What is the man doing in the street? Are they comparing their phones?*
- Scene recognition: *Is this on a farm? Are they outside?*
- Counting: *How many lights are there? How many zebras are in this picture?*
- Commonsense knowledge: *Is the sun going down? Is this in America?*
- Positional reasoning: *What is on top of the toaster? What is the zebra standing on?*
- Text Recognition: *What number bus is it? What is the store called?*
- Comparison: *Is the tank the same color as the toilet? Are they facing the same direction?*

4.3 APPROACH

We aim to learn separable skills and concepts, such that we can compose them to answer novel questions. To do so, the model should recognize that concepts mentioned in the question are man-

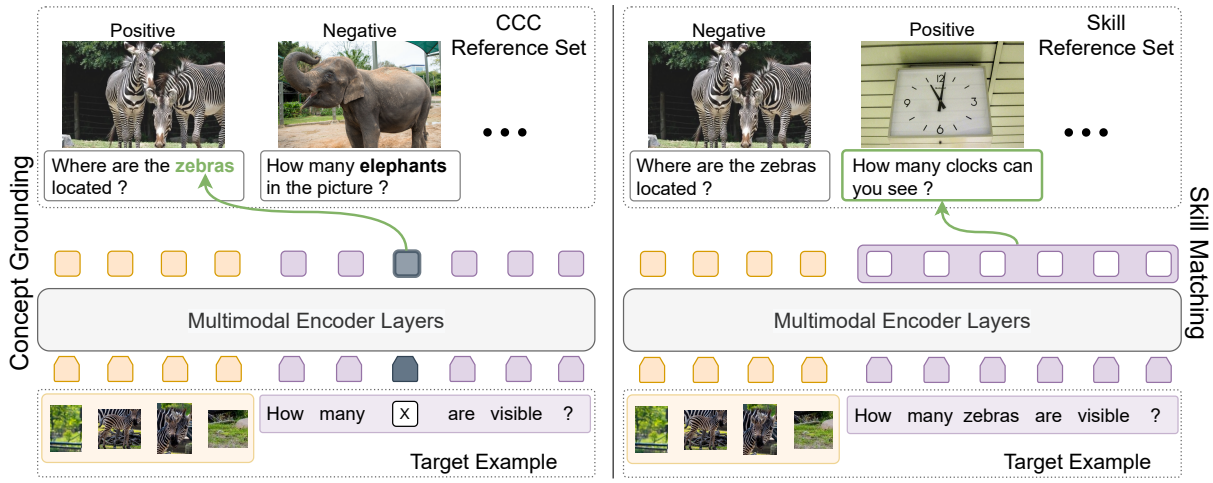


Figure 4.3: Overview of our approach. Left: We learn to ground concept representations by contrasting the multimodal representations of a masked concept token in the target example and words in other questions. Right: We encode skills in the summary representations of the question by contrasting with summary representations of other questions with the same (positive) or different (negative) skills.

ifested by their appearances in the image (*i.e.,* grounding) and that skills should be identifiable regardless of the concepts in the question or image. Gathering supervision for identifying concepts in the question, grounding them in the image, and labeling questions with skills would be very costly. Therefore, we propose to learn skill-concept separation in a self-supervised manner using contrastive learning [129, 130]. Illustrated in Figure 4.3, we train the model with two additional contrastive objectives jointly with the VQA objective: *concept grounding* (Section 4.3.2), which learns grounded concept representations, and *skill matching* (Section 4.3.3), which encodes concept-agnostic representations of skills. For each of our objectives, the model is presented with a *target example* and a *reference set* of positive and negative examples sampled from carefully curated *candidate references*. Each objective trains the model to make the representation from the target example similar to those of the positive ones. We expound our training procedure for learning these objectives jointly with VQA in Section 4.3.4.

4.3.1 Preliminaries

We assume that we are given a partially labeled dataset of tuples with image I, question Q, and answer labels A, where $(I^a, Q^a, A^a) \in \mathcal{D}^a$ has labels and $(I^u, Q^u) \in \mathcal{D}^u$ does not. Typically, models are trained with a VQA loss [30] using the labeled dataset, $\mathcal{D}^a$. Given an example (I, Q), image region features, $g_v(I) = \{v_1, ..., v_M\}$, and question token embeddings, $g_w(Q) = \{x_1, ..., x_N\}$, are

extracted and input to a multimodal encoder to produce multimodal representations of both modalities, $f(g_v(I), g_w(Q)) = (\{z_m\}_{m=1}^{M}, \{h_i\}_{i=1}^{N})$, where z_m and h_i are the image and text multimodal representations, respectively. An answer is predicted by pooling the encoded representations to a single representation (or using a CLS token as input [67]), which is then input to a softmax output layer. We build upon this basic VQA setup to learn skill-concept separation, and uniquely take advantage of both labeled and unlabeled data.

4.3.2 Concept Grounding

To learn grounded representations of concepts, we mask the concept mention from the target question and then train the model to recover this concept mention, using the multimodal contextual information, by pointing to the same concept mention in examples in the reference set (Figure 4.3). This procedure consists of two steps: 1) discovering and locating concept mentions in questions; 2) concept grounding by learning to point to the correct concept mentions in questions, leveraging a novel strategy to construct the reference set that offers effective contrastive learning signal.

Concept Discovery We first identify the concept words that can be grounded in images. While this can be done with different methods [131, 132, 133], we simply use heuristics. We use POS tagging and lemmatization [134] to identify the 400 most frequent nouns in VQA v2 and then we filter out concepts that cannot be grounded (*e.g.,* "*time*"). For a given question, Q, we want to find examples that have co-occurring mentions of a concept and the appearance of that concept in the image. It is likely that if a question about an image mentions a concept, then that concept may appear in the image [135]. Therefore, we identify the set of questions that mention the same concept, c, call it $\tilde{\mathcal{R}}_g^+(I, Q, c)$, which we consider as candidate positive references for Q. The set of questions not mentioning any of the same concepts are considered as candidate negative references, call it $\tilde{\mathcal{R}}_g^-(I, Q, c)$. To increase the likelihood of the concept appearing in the image, we employ a set of NLP-based heuristics to remove questions whose images may not contain the concept, such as counting questions with an answer of "0".

Concept-Context Contrastive (CCC) References Given a target question, Q, and a target concept mention, c, in Q, we could simply create reference sets by randomly sampling positive and negative examples [72, 130] from $\tilde{\mathcal{R}}_g^+(I, Q, c)$ and $\tilde{\mathcal{R}}_g^-(I, Q, c)$, respectively, based solely on whether the question contains c or not. However, we propose a novel reference example filtering strategy to encourage concept grounding. Our motivation is that, during VQA training, a concept often co-occurs with certain types of visual scenes or language priors. So the positive and negative examples should force the model to not rely on superficial cues when contrasted against

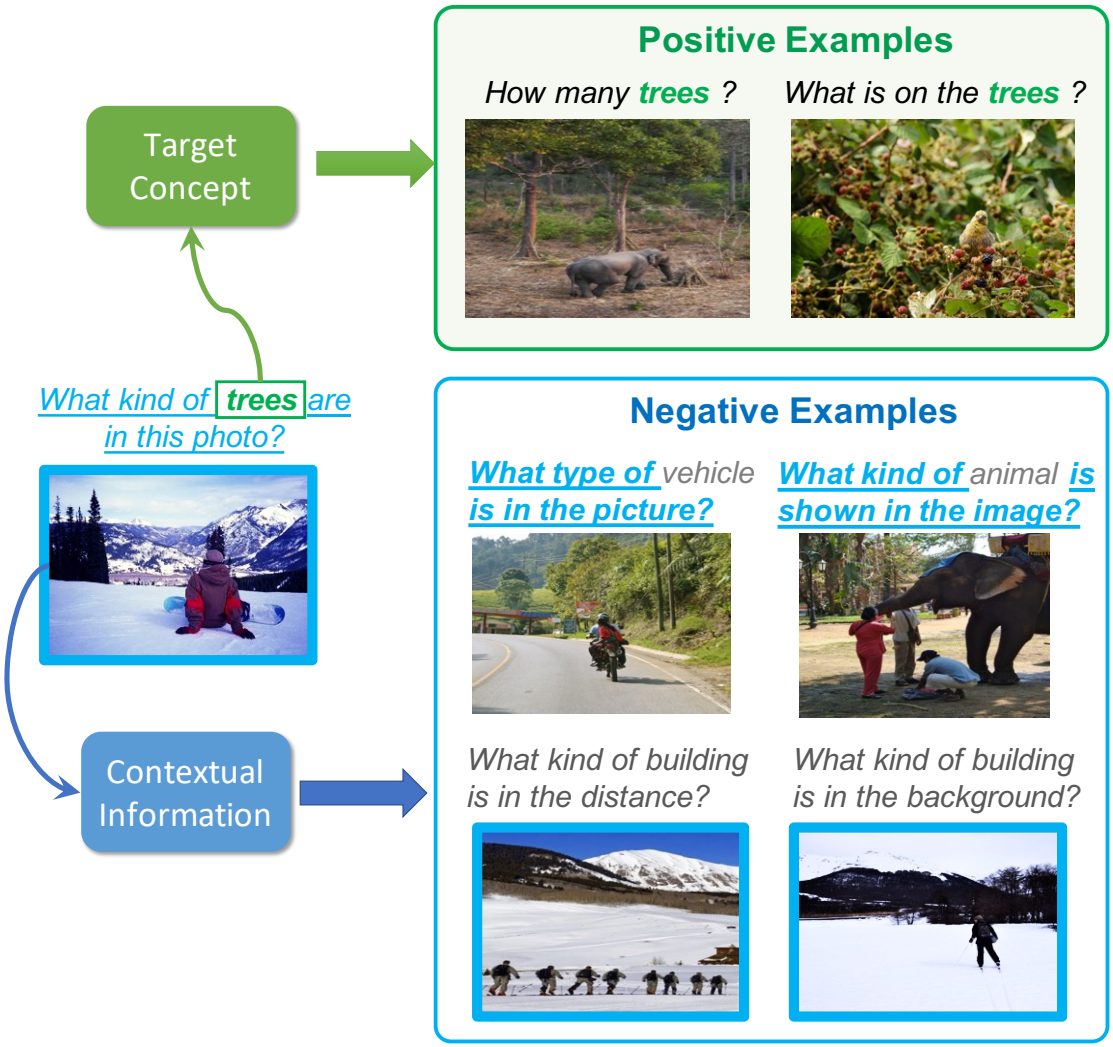


Figure 4.4: Target example (left) and a reference set of positive/negative examples (right) from our CCC references.

the target example and, instead, look at the correct visual regions. Our solution is to build sets of refined reference candidates, $\mathcal{R}_g^+(I, Q, c)$ and $\mathcal{R}_g^-(I, Q, c)$, for each (I, Q, c) tuple to ensure that the co-occurrence factor present in the dataset can be reduced. As shown in Figure 4.4, we want to find positive examples that also contain the concept "*tree*", but with distinct visual scenes and questions from the target. For negative examples, we seek distractors that are similar to the target in terms of the question or visual scene (*e.g.*, mountains with skiers in Figure 4.4), but do not reference "*tree*". To achieve this, we first represent the context of c by masking out c in the question and inputting the masked question and the image into off-the-shelf feature extractors to obtain question context representation q and image representation v.[1] We measure the contextual

[1]We use BERT [67] for questions and ResNet-101 [71] for images.

similarity by

$$\xi = \beta \text{cos}(q, q') + (1 - \beta)\text{cos}(v, v'), \tag{4.1}$$

where β is a scalar and (v, q) and (v', q') are the representations from target and candidate examples, respectively.

To select positive examples from $\tilde{\mathcal{R}}_g^+(I, Q, c)$, we use $\beta = 0.6$ and sample a set, $\mathcal{R}_g^+(I, Q, c)$, of N^+ examples that *minimize* ξ as our candidate positive examples for (I, Q, c). For negatives, we apply two settings of β that *maximize* ξ: $\beta = 0.7$, which favors examples with more textual similarity, and $\beta = 0.3$, which prioritizes images with similar visual context. We select N^- examples from each setting as our candidate negative examples, $\mathcal{R}_g^-(I, Q, c)$. Illustrated in Figure 4.4, when sampling reference sets from these two sets of candidates, the examples encourage the model to learn the specific correspondence between the concept mention in the question and its appearance in the image. Intuitively, the model must to learn to ground the concept mention in the presence of the distractors.

Concept Grounding Loss Let (I, Q) and c be the target example and target concept mention, respectively, and let $\mathcal{X}_g = \{(I_k, Q_k)\}_{k=1}^K$ be a corresponding reference set. Let k^* be the index of the positive example in $\mathcal{X}_g$ sampled from $\mathcal{R}_g^+(I, Q, c)$, while the other $K - 1$ examples are negative examples from $\mathcal{R}_g^-(I, Q, c)$. Let w_i be the token in Q that refers to the concept c. We mask out w_i and input this masked version of the question along with the corresponding image into the model, f, which outputs multimodal representations from which we extract the representation of the masked concept token, h_i. Next, we individually feed the examples from $\mathcal{X}_g$ into the model to obtain each token representation $\hat{h}_{k,j}$, where j is the index of a token in Q_k. Let $\hat{h}_{k^*,j^*}$ be the representation of the concept mention in the positive example's question. Our grounding loss is an NCE objective [129, 130] that requires the model to match the multimodal representation of the masked concept mention to the representation of the same concept mention in the reference set:

$$\mathcal{L}_g = -\log \frac{\exp(\text{sim}(\phi_g(h_i), \phi_g(\hat{h}_{k^*,j^*}))/\tau_g)}{\sum_{k,j} \exp(\text{sim}(\phi_g(h_i), \phi_g(\hat{h}_{k,j}))/\tau_g)}, \tag{4.2}$$

where ϕ_g is a learned projection function, $\text{sim}(\cdot, \cdot)$ is similarity function (*e.g.*, dot product or cosine similarity), and τ_g is a temperature. For this loss, we want to maximize the similarity of the masked target concept token to the correct concept token in the positive reference example. Since we are directly comparing tokens between examples, we model the similarity computation as an attention [48, 68, 70] with which the model must point [114] to the correct concept token. Specifically,

our projection function, $\phi_g(\cdot)$, and similarity function, $\text{sim}(\cdot, \cdot)$, are defined as

$$\phi_g(x) = W_g x + b_g, \tag{4.3}$$

$$\text{sim}(x, y) = x^\intercal y, \tag{4.4}$$

$$\tau_g = \sqrt{d} \tag{4.5}$$

where d is the dimension of x and y, $W_g \in \mathbb{R}^{d \times d}$, and $b_g \in \mathbb{R}^d$. Though this is similar to an attention, our formulation matches more traditional contrastive learning objectives [129, 130], where $\sqrt{d}$ is the temperature and we use a dot product as our similarity measure. To correctly match h_i with $\hat{h}_{k^*,j^*}$, the model must encode the visual features that match between the images of these examples in both token representations. Our CCC references encourage these representations of the concept mention in the positive example and the masked concept mention to be grounded to the right visual regions as the model cannot rely on superficial textual or visual co-occurrences.

4.3.3 Skill Matching

Contrary to concepts, the essential skill needed to answer a certain question is largely independent of image appearances and mentions of concepts in the question. For example, *counting* questions should share a similar process to produce an answer: image areas associated with the subjects of counting are summarized to make the count prediction. This process should be independent of the type of objects being asked about. In other words, we seek to learn summary representations of questions that share the essential steps to infer the answer and are invariant to concepts.

Skill References A straightforward approach to learn skills is to annotate questions which explicitly require the same reasoning steps. This annotation can be readily available on synthetic datasets [23, 100], but not available on datasets involving real-world images and questions. Instead, we propose to mine sets of contrasting examples to learn which questions require the same/different skills, matching questions with the same skills. Since the skills required for the question are typically indicated by the words of the question, we identify questions that are semantically similar. Essentially, questions that require the same skill (*e.g.*, "*What color ...*") should be related to one another, regardless of the specific concept mentions in the question. So, for each question, we mask out the concept words and we compute their BERT [67] representations. For a given (I, Q), the set of positive reference examples, $\mathcal{R}_s^+(I, Q)$, are sampled from the top-200 most similar questions using BERT representation, and the set of negative examples, $\mathcal{R}_s^-(I, Q)$, are randomly chosen from the rest of the dataset.

Skill Matching Loss For a given target example, (I, Q), let h be a summary representation of the target question. This can be computed using a special input token like BERT [67] or via a pooling operation on all question token representations output from the encoder. We sample a reference set of image-question pairs, $\mathcal{X}_s = \{(I_l, Q_l)\}_{l=1}^{L}$, where the positive example, Q_{l^*} from $\mathcal{R}_s^+(I, Q)$, shares the same skill as the target question, and the rest of the reference set are negative examples from $\mathcal{R}_s^-(I, Q)$. Let $\hat{h}_l$ be a summary representation for a question in the reference set. Shown in Figure 4.3, our skill matching loss is defined as

$$\mathcal{L}_s = -\log \frac{\exp(\text{sim}(\phi_s(h), \phi_s(\hat{h}_{l^*}))/\tau_s)}{\sum_l \exp(\text{sim}(\phi_s(h), \phi_s(\hat{h}_l))/\tau_s)}, \tag{4.6}$$

where $\hat{h}_{l^*}$ is the summary representation of the positive example, ϕ_s is another learned projection function, $\text{sim}(\cdot, \cdot)$ is a similarity function, and τ_s is the temperature ($\tau_s = 0.5$ in our experiments). This loss seeks to maximize the similarity of the summary representation of the target question with the summary representations of other questions with the same skill, regardless of the concepts mentioned. To obtain summary representations of questions, we simply use mean pooling over the question token representations. Our projection, $\phi_s(\cdot)$, and similarity, $\text{sim}(\cdot, \cdot)$, functions are

$$\phi_s(x) = W_s^{(2)} \psi(W_s^{(1)} x + b_s^{(1)}) + b_s^{(2)}, \tag{4.7}$$

$$\text{sim}(x, y) = \cos(x, y), \tag{4.8}$$

where ψ is a ReLU nonlinearity. Since we are not directly comparing token representations, we use the more standard contrastive objective [129] as opposed to the attention-based formulation used for concept grounding.

4.3.4 Training Procedure

With our losses, we use a multi-task learning procedure [136, 137], where at each step we employ our objectives with probability p_{sep} or not with probability $1 - p_{\text{sep}}$. During training, we always first sample an instance from the labeled data, $\mathcal{D}^a$, and update the model by minimizing the VQA objective. If at the current iteration we do not use our skill and concept objectives, then we only use the VQA objective. Otherwise, we first use the VQA objective and then apply our other objectives. Both objectives are computed in the same fashion: for $\mathcal{L}_g$ (or $\mathcal{L}_s$), we sample a target example from $\mathcal{D}^a \cup \mathcal{D}^u$ along with N_r^+ positive examples from $\mathcal{R}_g^+$ (or $\mathcal{R}_s^+$) as well as N_r^- negative examples from $\mathcal{R}_g^-$ (or $\mathcal{R}_s^-$), combine the sampled references to form the current reference set, and compute the loss term. We then sum $\mathcal{L}_g$ and $\mathcal{L}_s$, and update the model to minimize the negative

Algorithm 4.1: Our Skill-Concept Separation Training Algorithm

input: training steps T; labeled data $\mathcal{D}^a$; unlabeled data $\mathcal{D}^u$; candidate references $\mathcal{R}_g^+, \mathcal{R}_g^-$ and $\mathcal{R}_s^+, \mathcal{R}_s^-$; model f

for $i \in \{1, ..., T\}$ **do**

  sample (I_i, Q_i, A_i) from $\mathcal{D}^a$

  compute $\mathcal{L}_{CE}$ w.r.t. (I_i, Q_i, A_i)

  update f to minimize $\mathcal{L}_{CE}$

  $u \sim \text{Bernoulli}(p_{\text{sep}})$

  if $u = 1$ **then**

    sample (I_j, Q_j) from $\mathcal{D}^a \cup \mathcal{D}^u$

    sample concept c, where c is in Q_j

    $\mathcal{X}_g \leftarrow$ sample N_r^+ examples from $\mathcal{R}_g^+(I_j, Q_j, c)$ and N_r^- from $\mathcal{R}_g^-(I_j, Q_j, c)$

    compute $\mathcal{L}_g$ w.r.t. (I_j, Q_j) and $\mathcal{X}_g$

    ---

    sample (I_n, Q_n) from $\mathcal{D}^a \cup \mathcal{D}^u$

    $\mathcal{X}_s \leftarrow$ sample N_r^+ examples from $\mathcal{R}_s^+(I_n, Q_n)$ and N_r^- from $\mathcal{R}_s^-(I_n, Q_n)$

    compute $\mathcal{L}_s$ w.r.t. (I_n, Q_n) and $\mathcal{X}_s$

    ---

    update f to minimize $\mathcal{L}_g + \mathcal{L}_s$

  end

end

sum. This procedure is detailed in Algorithm 4.1, where $\mathcal{L}_{CE}$ is the VQA loss [30] (see Chapter 2).

4.4 EXPERIMENTS

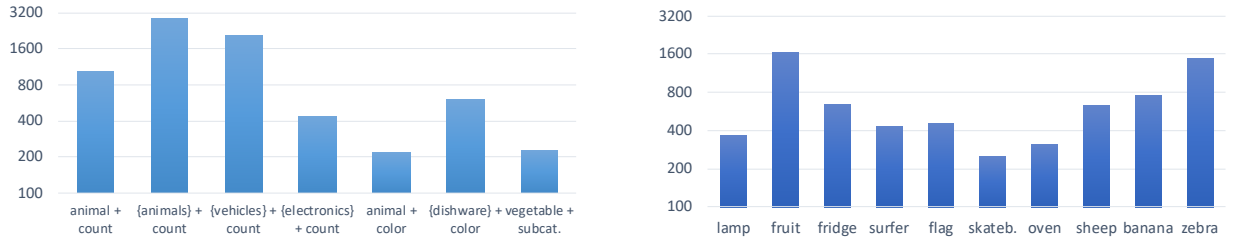
4.4.1 Data

We run our experiments on VQA v2 [12], which contains real images, human-written questions, and a variety of skills required to answer the questions. Since the goal of this work is to examine a model's performance on different types of novel questions, it requires the availability of answer annotations for the test data. Because the annotations of test-dev and test-std sets of VQA v2 are not publicly available, we use questions from the validation set for testing. This practice is also used by VQA-CP [96], VQA Rephrasings [138]. We do not train or tune hyperparameters with the validation set, it is strictly used for model evaluation. We measure and compare different models using the VQA accuracy [1] on different splits of novel questions.

We select three prevalent and common skills present in VQA v2: counting, color querying and subcategory recognition. For each skill, we remove the data labels for its co-occurring questions with one concept or a set of multiple concepts which can form a distinct category from training,

and then test on these compositions. The concepts (or concept groups) are sampled from the dataset, with two criteria: each skill-concept composition contains reliable amount of test data to measure accuracy (*i.e.*, at least 400 training and 200 testing questions) and the compositions have diverse coverage across the dataset. To increase coverage and ensure the minimally required size, we create some concept groups where the concepts in a group all fall under a broader category (*e.g.*, {animals} = {*giraffe*, *zebra*,...}). Statistics for the compositions and concepts are shown in Figure 4.5. The list of concepts within each concept group is:

- {animals}: *giraffe*, *zebra*, *bird*, *sheep*, *horse*, *elephant*, *cow*, *dog*, *cat*
- {vehicles}: *motorcycle*, *airplane*, *plane*, *jet*, *bus*, *car*, *truck*, *bike*, *bicycle*
- {electronics}: *computer*, *monitor*, *laptop*, *phone*, *cellphone*
- {dishware}: *plate*, *bowl*



(a) Novel Skill-Concept Composition

(b) Novel Concept

Figure 4.5: Statistics of novel skill-concept compositions and concept evaluation questions.

4.4.2 Model Comparisons

We select a set of recent VQA models to benchmark their novel-VQA performance. The first category is compositional models [25, 26, 27, 28]. We use **StackNMN** [27] and **XNM** [28], which are designed to handle compositional questions, like those in CLEVR [23], and have state-of-the-art performance on these datasets while also being applicable to real images without supervision from functional programs or image scene graphs. For XNM, we use the implementation provided by the authors as well as the recommended settings. To ensure consistency between the two compositional models, we implement StackNMN within the same code base as XNM. Specifically, we match the controller and the modules of StackNMN to the original paper. We use hidden dimension sizes of 512 for StackNMN and 1024 for XNM. We use the recommended number of reasoning steps, $T = 3$, for XNM and use the same for StackNMN. Both these models are trained with the Adam optimizer [139] and have the same learning rate of 0.0008 and batch size of 256.

The second type of model we experiment with is transformer-based [8, 9, 67, 68, 73, 76, 77, 78]. We use two top-performing transformer architectures from this model family: 1) a two-stream, cross-attention model [73] (**X-Att**), which has modality specific branches and cross-attentions in early layers followed by multimodal layers later in the network; and 2) a vision-and-language transformer model [9] (**X-BERT**) that acts as multimodal encoder throughout the entire network. For fair comparison, we do not use pre-training, same as our model, since we are specifically interested in the generalization ability of data-efficient models without requiring large-scale (*e.g.*, 9M+ image-text pairs), in-domain data external to VQA [124, 125].

Lastly, our base encoder model is a variant of the standard multimodal transformer [9, 67]. As is standard with transformers, we input visual regions, question tokens, and a special CLS that is appended to the beginning of the inputs, which we use to predict answers via a softmax output layer. There are two minor differences between a standard transformer and our model. First, before inputting the question into the transformer layers, we encode sequential information in the question tokens using a LSTM, yielding a slight improvement than positional embeddings [68]. Second, in each layer, the CLS token and visual regions can attend to all inputs, including themselves, and the question tokens only directly attend to themselves and the class token. The change allows the CLS token to act as a bottleneck through which text information interacts with the visual information. When the base encoder is trained without the proposed skill-concept contrastive losses, it serves as a baseline model, which we denote by **Base**.

For both X-Att and X-BERT, we use the original model source code. X-Att uses the recommended settings with a hidden size of 768, 12 layers, 12 attention heads, learning rate of 0.0001, batch size of 64, 20 training epochs, and the Adam optimizer. Due to their similarities in architecture, we use the same settings for Base, X-Bert and our framework for a more head-to-head comparison. Specifically, we use a hidden size of 512, 6 layers, and 8 attention heads. We match the training settings as well: a learning rate of 0.0001, batch size of 64, 13 training epochs, step learning rate decay with a rate of 0.2, and the Adam optimizer.

When forming our CCC candidate references from which we sample our reference sets, we use $N^+ = 20$ and $N^- = 40$ (since we have two settings for negative examples), so there are N^+ positive and N^- negative examples that can be selected from to form a reference set for a given target example. Meanwhile, we use $N^+ = 200$ and $N^- = 200$ for our skill matching candidate references. Then, in our training procedure, we use $p_{\text{sep}} = 0.1$ as the probability of applying our framework at each training step. Additionally, we simply use $N_r^+ = 1$ and $N_r^- = 2$ for both concept grounding and skill matching, so the model will contrast between a single positive example and two distractor negative examples. For our concept grounding loss, we sample one negative example from both of our settings as our negative examples.

Model	Counting				Color		Subcat.	Overall
	animal	{animals}	{vehicles}	{electronics}	animal	{dishware}	vegetable	
XNM	56.02	48.32	44.35	51.94	77.22	65.73	57.33	57.27
StackNMN	54.22	47.56	46.10	52.83	76.57	69.22	57.17	57.67
X-Att	58.94	56.28	46.30	57.05	73.15	67.29	57.25	59.47
X-BERT	63.58	54.58	42.34	56.84	75.88	70.31	58.96	60.36
Base	62.57	59.19	48.33	61.84	76.57	72.91	58.33	62.82
Ours	**65.16**	**59.87**	**50.75**	**62.21**	**77.45**	**73.76**	**61.04**	**64.32**

Table 4.1: VQA accuracy on novel skill-concept compositions. The highest and lowest numbers of each experiment are emphasized.

Model	lamp	fruit	fridge	surfer	flag	skateb.	oven	sheep	banana	zebra	Overall
XNM	53.69	50.23	57.98	72.68	36.58	70.16	53.49	54.96	**52.35**	61.50	56.36
StackNMN	54.27	46.10	58.97	74.10	**41.31**	74.11	56.30	57.12	50.98	61.25	57.45
X-Att	46.26	33.10	51.52	67.68	31.53	69.73	51.69	49.83	41.71	**64.93**	50.80
X-BERT	44.43	30.72	50.83	61.60	32.46	66.05	48.10	50.32	43.10	57.06	48.47
Base	55.14	52.99	59.06	74.12	39.05	71.67	56.60	63.31	49.83	56.05	57.78
Ours	**57.40**	**54.40**	**60.92**	**74.36**	40.15	**75.27**	**59.91**	**64.04**	50.78	60.77	**59.80**

Table 4.2: VQA Accuracy on individual novel concept split. *skateb.* refers to *skateboarder*.

4.4.3 Novel Skill-Concept Composition VQA

Table 4.1 shows the VQA accuracy on each of the novel compositional subset. Interestingly, although neural module networks are designed to explicitly break down the question answering process into sub-tasks, which in principle should help with adapting these sub-tasks to new questions and thus generalize better, they yield lower performance than transformer models. This may be due to the effective feature learning capacity of self-attention mechanism. Among all transformer models, our base encoder achieves competitive performance to existing networks, demonstrating that it is a strong baseline among multimodal transformers. Finally, our contrastive learning framework outperforms the baseline and all other approaches across each novel composition set. This supports the effectiveness of our framework for generalization to new compositions.

4.4.4 Novel Concept VQA

For this experiment, we are interested in the setting where models are never trained to answer questions about a concept but can make use of the unlabeled image-question pairs, and then are tested on questions that have mentions of this given concept. Similar to the previous experiment, these concepts were sampled to maximize coverage as well as maintaining a reasonable test size. This setting is more challenging than the previous experiment since the model misses the VQA

training supervision on any question that has the given concept, as opposed to any question that has both the given concept and a certain skill.

We report quantitative results in Table 4.2. For this more challenging setting of novel question answering, on average, two of the existing transformer architectures underperform other models by a noticeable margin. This may suggest that the transformer architectures which perform well on large-scale vision and language pre-training may have difficulty specializing to the VQA task. The Base transformer slightly outperforms neural module networks. Lastly, our framework again outperforms all models on average, demonstrating its value in improving VQA generalization ability on novel concepts.

4.4.5 Concept Grounding Results

Since our model learns to weakly ground concepts in questions, we would like to test its grounding efficacy directly beyond using VQA accuracy. To obtain an evaluation set, we manually annotated 320 image-question-concept tuples with the visual region in the image that corresponds to the concept in each tuple. Candidate visual regions are found using Faster-RCNN [58]. We use recall@5 as our grounding metric, considering a grounding correct if the correct visual region falls within the top 5 most similar visual regions to the target concept token. The model trained with our framework achieves a grounding recall of **59.12**, compared to **43.71** of the Base model. Note that our framework obtained this improvement with no additional training data for grounding. As shown in Figure 4.6, our model can often correctly ground a variety of objects, but can be fooled by ambiguous looking concepts like the candle in the incorrect example. Further, it is challenging to learn to differentiate concepts that almost always co-occur (*e.g.,* "*shirt*" and "*person*").

We also notice an interesting phenomena from our approach where the concept mention representation can be most similar to a distinct part of the concept visually (*e.g.,* the ears of the cats in each image in the second row in Figure 4.6). We conjecture that this may be due to the consistent signal that these parts offer in differentiating between positive and negative examples (*e.g.,* almost all cats in the data have ears). Additionally, in the middle example on the bottom row, the model may be distracted by the lengthy reference (*e.g.,* "*... giant, blue thing right behind ...*"), so the model incorrectly grounds "*girl*". This may be a shortcoming the transformer since all text tokens can attend to each other, so the other text tokens can potentially influence the concept mention to a significant degree. Moving forward, it could be fruitful to explore architectures that mitigate erroneous contextual influences. One possible direction would be to learn grounding earlier and/or at different points in the network, as has been done for general VQA [140], which would allow the visual information to be encoded in concept mentions with potentially less influence from other text tokens.

Model	Avg. Novel Count	Avg. Novel Concepts
Base	58.03	60.42
Base+MLM	58.41	60.25
Base+$\mathcal{L}_s$	58.83	61.85
Base+$\mathcal{L}_g$	59.80	62.06
Ours	**60.71**	**63.19**

Table 4.3: Results with different losses on novel skill-concept composition and novel concept VQA.

Model	lamp	fruit	fridge	surfer	flag
Base	55.14	52.99	59.06	74.12	39.05
Random	**+2.80**	+0.49	-0.16	-0.14	+0.50
CCC (Ours)	+2.26	**+1.41**	**+0.98**	**+0.24**	**+1.10**

Table 4.4: Comparing different reference set construction schemes for concept learning across five different concepts.

4.4.6 Loss Ablations

We ablate our losses by sampling three novel compositions and three novel concepts and report their average performance in Table 4.3. Adding our losses leads to consistent gains, with top performance achieved with our full framework. When used alone, our grounding loss seems to contribute a larger benefit compared to the skill loss. Nonetheless, the best performance is achieved by combining the two components, further supporting the value of skill and concept separation. We also experiment with a masked language modeling (MLM) objective [67] that replaces our losses. Our objectives perform better than the MLM objective, implying that the improvements our objectives offer are not simply due to additional data.

4.4.7 CCC Reference Set

To study the effects of our CCC reference set selection strategy, we compare it with the commonly used random sampling method [72, 130] and report novel concept VQA results in Table 4.4. We train both models with our full framework, the only difference being the reference set construction method for the concept loss. Both models improve upon the Base model, with our reference set construction method offering more consistent gains.

Additionally, we experiment with a further refined version of our CCC references. Specifically, we employ ontological constraints when formulating the reference sets. So rather than having

Model	Counting				Color		Subcat.	Overall
	animal	{animals}	{vehicles}	{electronics}	animal	{dishware}	vegetable	
Ours	**65.16**	59.87	50.75	62.21	77.45	73.76	**61.04**	64.32
+OC	64.45	**60.16**	**52.08**	**64.42**	**78.47**	**73.89**	59.29	**64.68**

Table 4.5: VQA accuracy on novel skill-concept compositions comparing our original CCC references (Ours) to CCC references with ontological constraints (+OC).

Model	Test-dev				Test-std	VQA-CP
	Y/N	Num.	Other	All	All	All
Base	85.42	52.78	59.89	69.60	69.99	40.98
Ours	**85.83**	**52.95**	59.88	**69.78**	**70.09**	**41.71**

Table 4.6: Effect of using our components the standard VQA benchmarks and VQA-CP [96]. This is single model performance and both methods see the exact same data for training (*i.e.,* no compositions or concepts are removed). For test-dev and test-std, models are trained on the VQA v2 training and validation data as well as the Visual Genome data, as is typically done [49, 78]. For VQA-CP, models only train on the VQA-CP training split and are evaluated on the testing split.

negative examples only be other examples that don't mention the same concept, we constrain the negative examples such that they must not mention the same concept or any of its hypernyms or hyponyms. This amounts to reducing potential noise in our approach. These results are shown in Table 4.5, where we see that the addition of these ontological constraints generally shows improvements. In our normal setting, we do not assume that we have any knowledge about the concepts. However, these results demonstrate that utilizing knowledge of concepts can improve performance.

4.4.8 General VQA Evaluations

In Table 4.6, we also evaluate on VQA-CP [96] and the test-dev/test-std splits of VQA v2 (Table 4.6). Our approach can be complementary to the Base model and generally improves the results, even when not evaluated in a compositional setting. This can be potentially quite useful for other methods since our approach does not require external data. While we see gains in general, notably, our approach is able to improve on VQA-CP without extra annotations, ensembling/tuning, or a performance drop on VQA v2.

4.4.9 Qualitative Examples

Skill-Concept Compositions We show VQA output examples for novel skill-concept compositions in Figure 4.7 that compare the performance of our approach versus Base. As a reminder, the

models tested here never see labeled image-question pairs during training. Our approach allows the model to adapt to these unseen compositions. We see that, for unseen compositions of counting and different concepts, the base model has difficulty recognizing and counting these concepts. For example, we observe that despite the clear appearance of the animals in the images, the Base model is unable to transfer the skill of counting, whereas the model trained with our framework is able to handle these cases. Similarly, in the bottom two examples of the last column, we see that our approach is able to more precisely locate the specific "*plate*" being referred to.

Novel Concept VQA Example outputs for novel concept VQA are in Figure 4.8. Interesting examples of the improvements that our grounding framework can offer are in the first row, where our model is able to locate the specific object and produce the correct answer. The last two examples show some failure cases, where our model produces plausible yet somewhat generic answers compared to the baseline. Overall, this setting is particularly challenging and could warrant further exploration, such as experimenting with learning concepts in an offline fashion, like pre-training.

4.4.10 Discussion

Our losses can encourage the model to better ground concepts and generalize skills to unseen skill-concept compositions. Some advantages of our approach are that we do not require labeled data or data external to VQA and that our method is general enough to be applied to data from different domains. This can be particularly useful given that automatically generated questions can be easier in many ways compared to generating other aligned image-text pairs, such as captions [141]. A potential extension of our approach would be to generate questions for images in new domains and then employ our approach to generalize to this new domain.

Meanwhile, one issue with our approach is that the performance can vary based on the construction of the candidate references. In future explorations, it could be greatly beneficial to more closely examine the properties of the reference sets that improve or degrade performance, similar to the explorations of different augmentation methods in contrastive learning for object recognition [142]. Additionally, more sophisticated ways to learn skills could potentially encourage better generalization beyond our formulation, since our skill matching loss simply utilizes the most/least similar sentences as positive/negative references and mean pooled token representations.

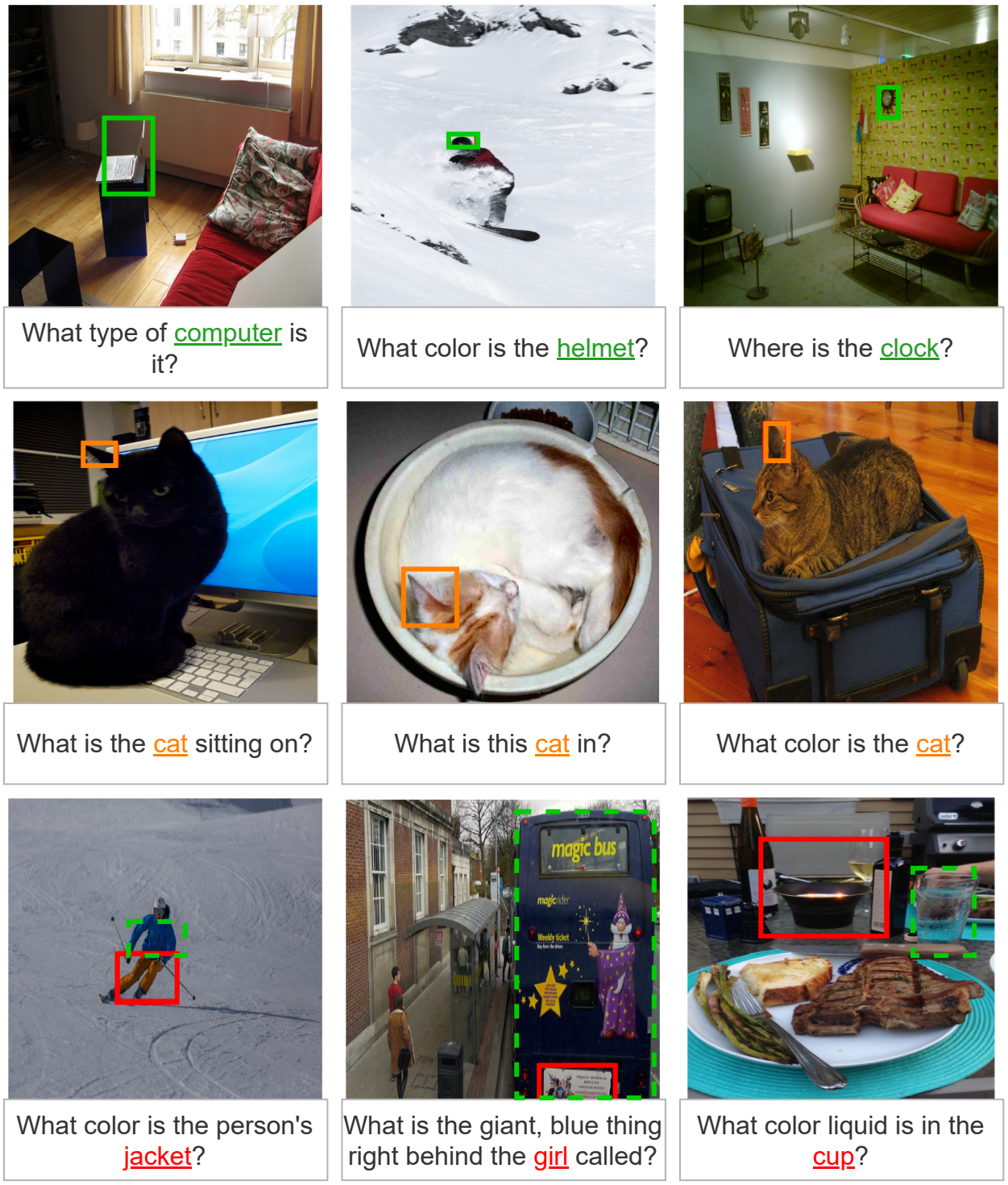


Figure 4.6: Correct, incorrect, and plausible grounding examples. For incorrect examples, the green, dashed bounding box is the box to which the concept should be grounded. We visualize the most similar visual region to the concept mention in the question.

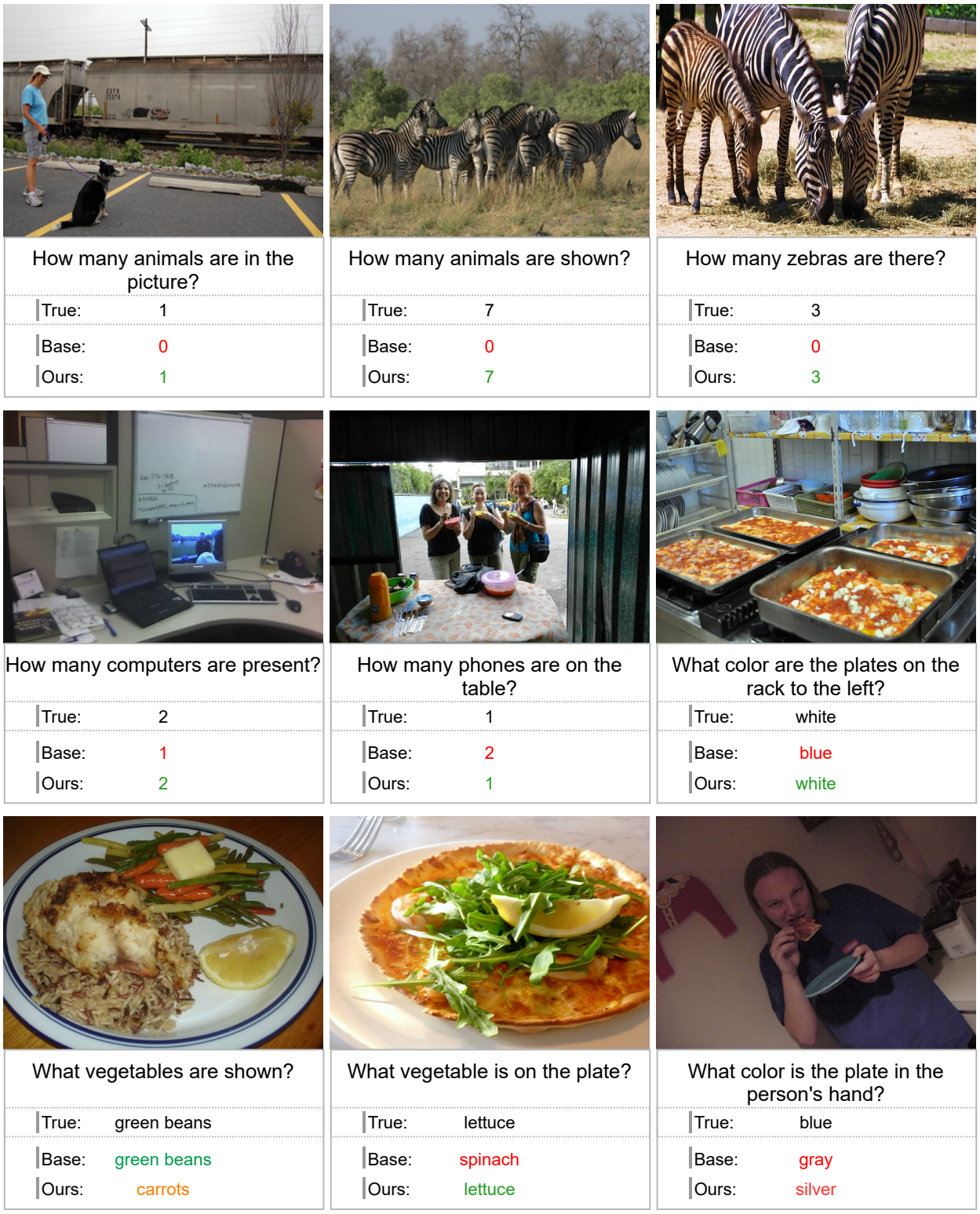


Figure 4.7: Correct, incorrect, and plausible VQA output examples for novel skill-concept compositions comparing the predictions of our approach (Ours) and the Base model.

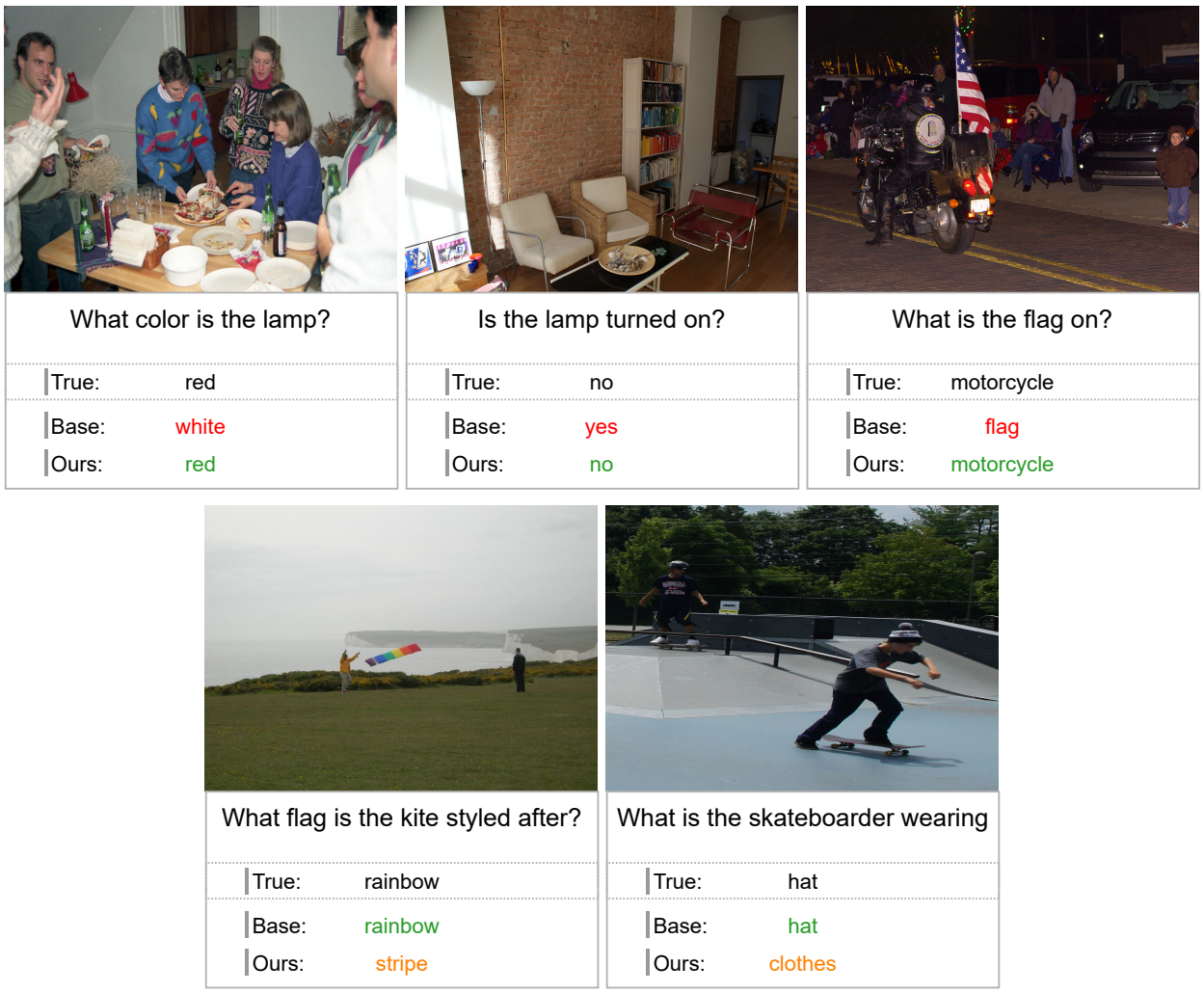


Figure 4.8: Correct, incorrect, and plausible VQA output examples for novel concepts.

4.5 SUMMARY

We propose a new evaluation setting for generalization in VQA: measuring the ability to compose the skills needed to answer questions and the visual concepts that should be grounded to the images. According to our experiments, existing approaches have difficulty generalizing to novel compositions of these two factors. We present an approach that implicitly disentangles skills and concepts, while grounding concepts visually using a novel contrastive learning procedure. Our approach is able to learn from unlabeled VQA data in order to answer questions about previously unseen concepts. Results on VQA v2 show that our approach can achieve state-of-art performance on novel compositions of skills and concepts as well as generalizing from unlabeled data.

CHAPTER 5: LEARNING FROM LEXICAL PERTURBATIONS FOR CONSISTENT VQA

5.1 MOTIVATION

Though great progress has been made towards overall VQA performance, the approaches lack robustness and are sensitive to input variations [12, 97, 144, 145]. In particular, prior work [97] has shown that when presented with a rephrased version of a question, VQA models often produce inconsistent answers. We conjecture that this is likely because most VQA approaches ignore interconnections between related questions and handle each example independently during training, even though learning such relationships is paramount to generalization, robustness, and compositionality. As a result, models must learn the connection between different concept mentions or phrases implicitly.

While a few attempts have been made to improve VQA robustness via augmentation and regularization techniques [97, 123, 146], they encourage consistency among related questions at the answer prediction level, without considering the stronger form of consistency between the intermediate computation steps. Furthermore, proposed augmentations are either costly (human-generated [97]) or suffer from quality control issues [98].

In this work, we propose a novel robust VQA approach that first augments the question and then enforces consistency not only of the answer, but also of the intermediate representations computed by the model. The intuition is that (like humans) VQA models should follow the same reasoning steps to solve two differently-phrased questions that have the same meaning. For example, to answer both questions *"Are the buildings tall?"* and *"Are the buildings short?"* a model should follow the same process (*i.e.,* detect buildings in the image, then classify their height).

How to enforce such consistency of the intermediate reasoning steps appears to be a hard problem in general. We propose that an effective way to improve this stronger notion of *reasoning consistency* is by maintaining the associated computation steps between questions that differ by controlled variations. We leverage the family of interpretable, compositional VQA models called *Neural Module Networks (NMNs)* [25, 26, 27, 28] which explicitly represent sub-tasks like object detection and spatial reasoning as *modules* within the network and predict sequences of weights over these modules (akin to *programs*) to solve each question. However, unlike existing NMN work, we regularize the model to learn not only how to compose sub-tasks, but also follow the same sequences of sub-tasks to answer two variations of the same question, illustrated in Figure 5.4.

In addition, we also propose a novel data augmentation algorithm based on auxiliary linguistic knowledge freely available in text-only corpora. Examples like in Figure 5.1, seem to suggest

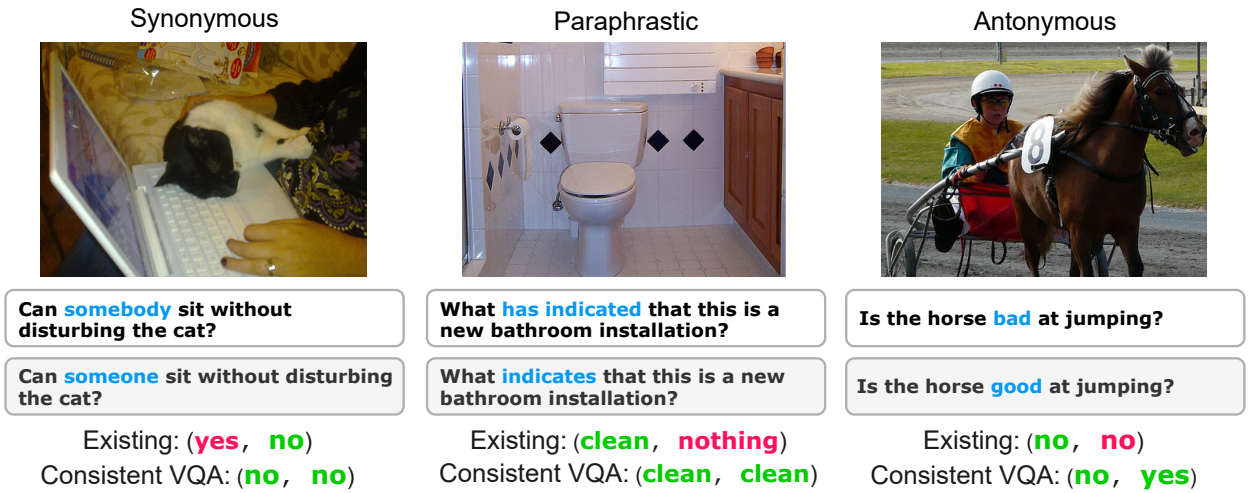


Figure 5.1: Existing VQA models are often sensitive to question variations, including simple lexical changes. Examples show three types of lexical changes, along with answers predicted by a recent model [143]. Our goal is to produce consistent answers to related questions (bottom row) while maintaining the overall accuracy.

that existing VQA models ignore the relatedness between questions with mild linguistic modifications. Questions like the antonymous pair in Figure 5.1 are related in that they query the same property of the same object, but differ in that they ask the opposite of each other. Existing work ignores such modifications and focuses either on human-written paraphrased VQA questions [97] or auto-generated paraphrases using back-translation [98]. Human paraphrases tend to add filler phrases or even change the question meaning (see Section 5.2) and are very costly to annotate, while back-translations are also hard to control and have quality problems. Motivated by this, we propose to augment data with changes at a low level, such as simple lexical substitutions. We create variations by substituting parts of the questions using rules extracted from large-scale language resources [147, 148] which keeps the meaning and the realistic distribution of the original questions while avoiding the cost and/or semantic incoherence commonly found with prior work.

Finally, we present VQA Perturbed Pairings (VQA P2), a dataset of perturbed questions derived from VQA v2 that can be used to measure robustness to the specific linguistic phenomena not previously evaluated in VQA literature, covering the usage of different synonyms (*Synonymous*), different phrasings (*Paraphrastic*), and opposite attributes (*Antonymous*). This benchmark evaluates the robustness of models different concept mentions and phrasings of questions. VQA P2 is comprised of 26.5k VQA questions, where each question has a corresponding question in VQA v2 [12] that it differs from by controlled perturbations. VQA P2 can be easily expanded without the need for expensive human annotations [97, 100, 146], while maintaining control over the perturbations used.

5.2 VQA PERTURBED PAIRINGS (VQA P2) BENCHMARK

As noted, VQA approaches tend to ignore relationships among questions and can be inconsistent, even when the questions only differ by slight linguistic changes. We aim to create a benchmark to measure the progress of robustness in VQA models. Specifically, we measure the consistency of predictions under different linguistic perturbations of the input questions. Next, we provide the design considerations for choosing the best source to create perturbations. We then introduce a knowledge-driven pipeline which we used to create high-quality, semantically coherent questions in an efficient fashion.

5.2.1 Sources of Linguistic Perturbations

A natural choice to create question variations is to use human annotations (as done by [97]). Beyond the obvious drawback of being cost-intensive, human annotations have a number of issues for our specific setting. First, human-rephrased questions tend to bias towards changes that immediately come to mind when prompted with the original question instead of coming up with substitute words, which is desired to study linguistic perturbations. For example, adding filler phrases or changing word arrangement tend to be common for annotators (Figure 5.2).[1] Second, human paraphrasing can introduce multiple sources of variations, such as introducing commonsense related items, sentence structural changes as well as lexical alterations. Therefore, using human-written variations as a diagnostic benchmark lacks a level of precision needed to diagnose and understand model performance on linguistic perturbations.

On the other hand, fully automated methods, like question generation [149, 150] or using back-translation [98], suffer from many quality control issues, which may make them suitable for training, but not benchmarking. Figure 5.2 shows how this type of methods could generate mismatched phrasal replacement or semantic drift, where the generated questions no longer hold the same meaning. In general, controlled generation of text remains a difficult, open challenge [151].

Considering the limitations of human-written and fully-automated options, we propose to use a knowledge-driven approach to creating perturbed questions. We notice that large-scale linguistic resources have a rich, expressive repertoire of candidates for lexical changes, which yields richer and less biased lexical variations than a human annotator could typically provide. Instead of generating full questions, we curate substitution rules first and then apply the rules to existing, human-written questions to create perturbed questions. This procedure is more controlled than fully-automated methods, which greatly reduces the margin for error and ensures the data qual-

[1] Randomly sampling 100 human-written question pairs from VQA-Rephrasings [97] shows that approximately 36% of these human paraphrases simply add fillers (*e.g.,* "*Can you tell me...*") or are simple rearrangements (*e.g.,* "*What color is the van?*" to "*The van is what color?*").

ity needed for a benchmark dataset. Next, we describe the pipeline that we designed to generate high-quality, semantically coherent questions in an efficient fashion.

5.2.2 Perturbation Pipeline

Perturbation	**Example rules** (source → target)
Synonymous	*car* → *automobile* *refrigerator* → *fridge* *phone* → *telephone*
Paraphrastic	*performed in* `[NP]` → *carried out within* `[NP]` *be considered* `[NP]` → *be viewed as* `[NP]` *participating in* `[VP]` → *taking part in* `[VP]`
Antonymous	*open* → *closed* *wet* → *dry* *full* → *empty*

Table 5.1: Example rules used to generate the data. Synonymous and antonymous rules are single word, while paraphrastic rules can contain multi-word expressions and grammatical constraints.

Substitution Extraction We extract lexical substitution rules from the Paraphrase Database 2.0 (PPDB) [148], a lexical database containing over 100 million paraphrases automatically mined from human-written text, as well as WordNet [147], two large-scale linguistic resources, and apply these rules to existing VQA v2 questions. We create three types of perturbations: 1) *synonymous* perturbations that substitute a single word with its synonym; 2) *paraphrastic* perturbations that substitute multi-word phrases for single/multi-word phrases with the same meaning; and 3) *antonymous* perturbations where adjectives or adverbs are substituted with their antonyms, which explores the ability to understand opposite states of an attribute or action. We extract synonymous and paraphrastic rules from the lexical and syntactic subsets of PPDB respectively, and accept the rule if the target is equivalent or entailed by the source, according to PPDB's constraints. We gather antonymous rules from WordNet.

Rule Refinement We automatically determine which rules can be applied by matching the source phrases and grammar requirements to the questions, discarding rules that are not applicable to any questions. We ensure that our rules are not simply adding unknown words by removing rules whose source or target contain words that don't appear in the VQA vocabulary. To prevent low quality/frequency substitutions, we filter the synonymous and paraphrastic rules with a minimum

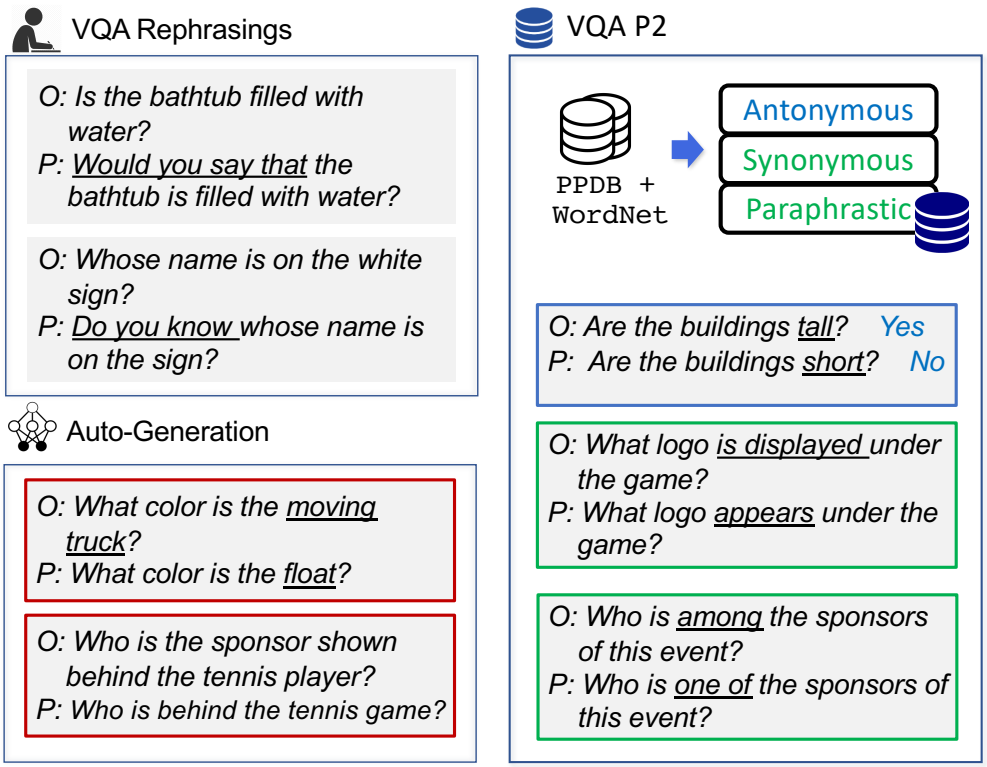


Figure 5.2: Examples of VQA P2, human generated [97], and back-translated (Auto-Generation) variants (P) of input questions (O). VQA P2 uniquely offers three types of controlled linguistic variations to benchmark robustness.

confidence threshold using PPDB confidence scores, which are well-aligned with human judgements [148]. Example rules are shown in Table 5.1, where each rule is a mapping from a source to target phrase.

Applying Substitution Rules We apply the filtered rules to VQA v2 questions and obtain our final benchmark. Since consistency metrics require the ground truth annotation for the each question, and VQA v2 test sets do not have publicly available answer annotations, we use the VQA v2 val set as the basis for our benchmark. This practice is common among methods that require per-question answer annotations [96, 97]. During the rule application process, for each question, multiple substitutions for a specific type of perturbation can be made, which increases variations. We use the grammatical constraints and entailment relationships from PPDB/WordNet as well as word sense disambiguation [152] to ensure that the senses of the words/phrases in the question match the senses of the rules. For antonymous rules, we limit their application to yes/no questions of the form "*is/are the*" and "*is/are this/these*" where the WordNet synsets of the source word in both the question and rule match. Antonymous rules are limited to these kinds of questions so we

only apply the antonym substitutions to attributes/states that are directly queried (*e.g.,* "*Is the window open?*"), and not simply mentioned in the question (*e.g.,* "*What's near the open window?*"). For answers, synonymous and paraphrastic questions use the same answers as their VQA v2 counterparts as they share the same meaning and antonymous questions take the opposite answers.

5.2.3 Benchmark Summary and Evaluation Settings

We are able to build a high-quality benchmark in an efficient way, thanks to the quality control checks at each step of the data creation, including using PPDB confidence scores which correlate well with human judgements [148], word sense and grammar matching, and manual filtering of rules. The final outcome is a set of $26,512$ question-answer pairs that are perturbed counterparts of VQA v2 questions (Figure 5.2). Due to the fast, automated nature of this process, VQA P2 is easily extensible to more linguistic variations and larger sets of perturbations following our pipeline. VQA P2 is distinct from, and complementary to, the existing VQA robustness benchmark, VQA-Rephrasings [97]. We target linguistic perturbations to factor out other sources of variations, while [97] targets general rephrasings. The perturbations are much richer in our dataset: the synonymous and paraphrastic changes in VQA-Rephrasings only cover $\sim$20% of the amount that VQA P2 covers, and VQA-Rephrasings contains *none* of the antonymous changes. We provide examples of VQA P2 in Figure 5.3.

Since our dataset provides information about the types of perturbations (synonymous, paraphrastic, and antonymous), we can evaluate the overall consistency as well as a model's capacity for addressing specific types of perturbations, which offers more diagnostic insight. We can also evaluate the effectiveness of robustness approaches, augmentation methods, and their combinations at capturing these perturbations, including knowledge-aware (*i.e.,* standard setting, having access to substitution rules) and knowledge-agnostic (*i.e.,* extended setting, agnostic of substitution rules) settings.

5.2.4 Discussion: Perturbing Images

In this work, we explore perturbing questions to measure robustness. However, one may also consider perturbing the images as well. One method for this is to perturb images using imperceptible, adversarial perturbations [153, 154]. The use of adversarial examples for data augmentation has been shown to be effective for improving VQA performance [98]. However, for benchmarking, these perturbations may not provide interpretable results, since they are imperceptible by design. Alternatively, one may perform semantic editing of the images, akin to our lexical perturbations, which has also been used for data augmentation in VQA [155]. For example, one can remove

objects (and do inpainting [156]) or change their colors, and then edit the question-answer pairs accordingly. Determining the correct answers for the perturbed images requires careful consideration to ensure that the image-question-answer tuples are coherent. This process is more difficult for questions involving commonsense, so approaches for perturbing images may be limited to certain image-question pairs. Overall, perturbing images in semantically meaningful ways for benhcmarking is an intriguing and challenging direction for future work.

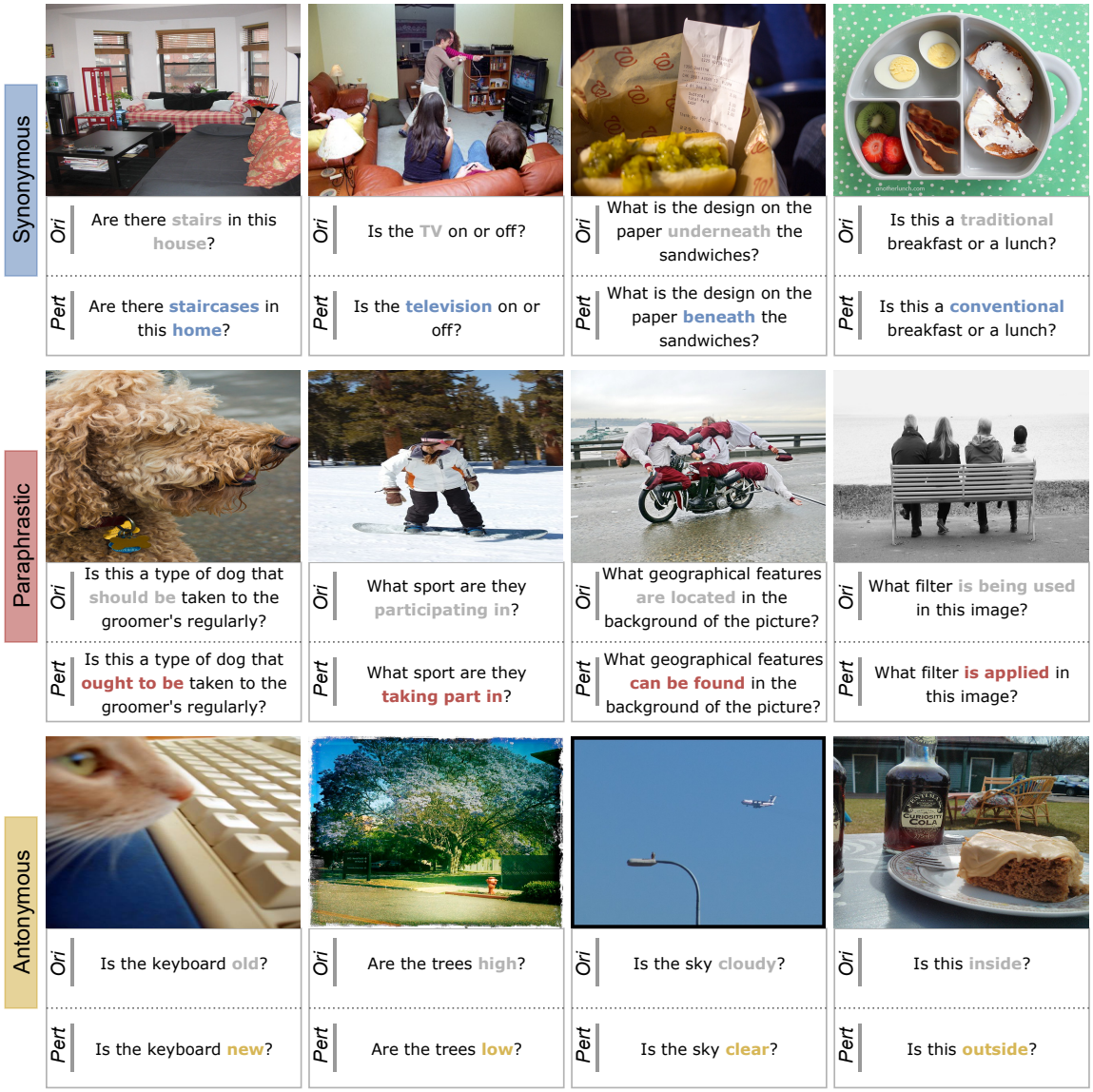


Figure 5.3: Example questions for each perturbation type (synonymous, paraphrastic, and antonymous) from VQA P2 (*Pert*) and their VQA v2 counterparts (*Ori*). The gray text in the original question delineates the source words/phrases to be replaced and the words/phrases in color indicate the replacement words/phrases.

5.3 APPROACH

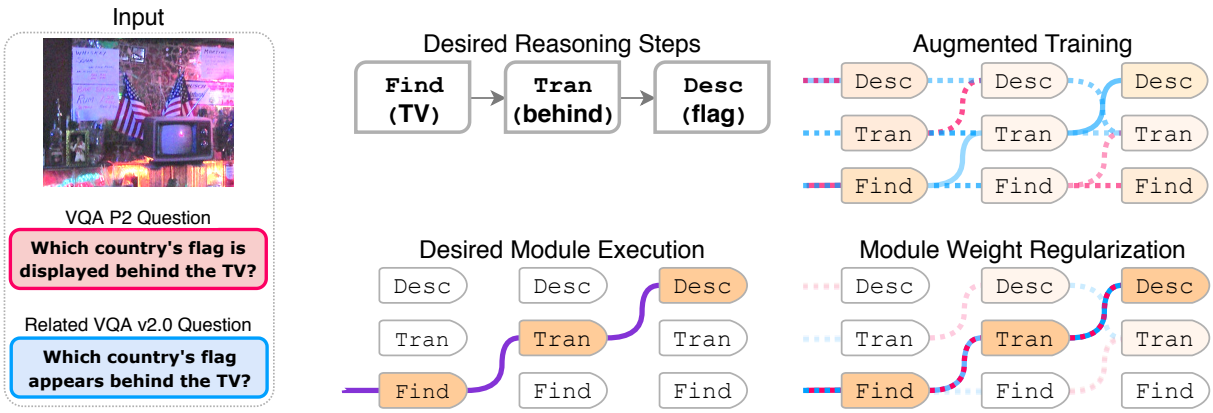


Figure 5.4: We propose a novel Q3R framework that improves VQA robustness against linguistic variations by augmenting questions and encouraging similar module weights between related questions.

Given an image-question pair, (I, Q), a VQA model maps the pair to a distribution over an answer set, $f(I, Q) \rightarrow \hat{a}$. Existing approaches most often treat this as a classification task, minimizing the prediction loss between the predicted answer and the ground truth answer. This standard approach, however, does not take into account possible relations among questions. The goal of our approach is to train a model to be aware of question relationships, and thereby learn to be more consistent when answering.

Modeling Question Relatedness. Typically, given an input question, a predictable set of reasoning steps are expected to answer the question. For example, in Figure 5.4, the sub-networks which can answer "*Which country's flag is displayed behind the TV?*" should be able to decompose this task into components such as "`Find`(*TV*)", "`Transform`(*Behind*)", and "`Describe`(*which country's flag*)" and learn to transfer all sub-networks to the question: "*Which country's flag appears behind the TV?*" Essentially, there should be a set of elementary operations where each one solves a less complex task than the original question and related questions can share these operations, following a similar order of execution. Based on this intuition, we propose our **Question-Relatedness Regularized Reasoning** (**Q3R**) framework. Our overall approach is illustrated in Figure 5.5

5.3.1 Question-Relatedness Regularized Reasoning (Q3R) Framework

Our framework is comprised of three components: 1) a method to create linguistic variations of input questions; 2) a compositional backbone model, guided by question-based module selection; and 3) a mechanism to enforce similarities of related questions at the module level.

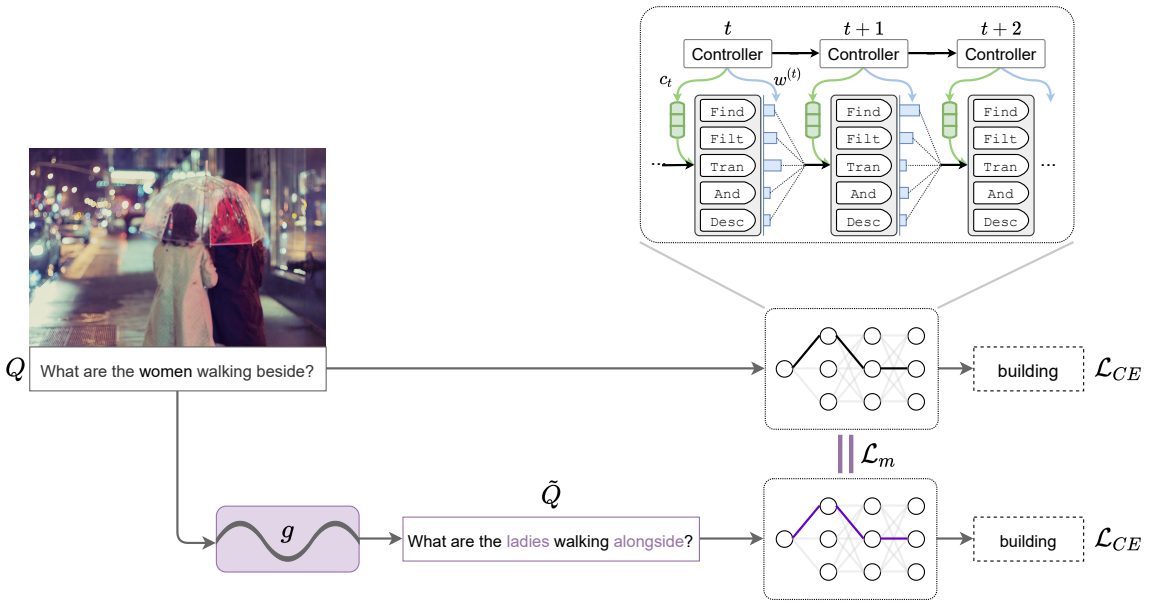


Figure 5.5: Illustration of our Question-Relatedness Regularized Reasoning (Q3R) framework with an example backbone model (top right).

Creating Related Questions The input to our pipeline is an image-question pair, (I, Q), and, during training, a corresponding answer A. We create a module, $g(Q, A) \rightarrow (\tilde{Q}, \tilde{A})$, that takes Q and A as input and outputs a related question-answer pair $(\tilde{Q}, \tilde{A})$. In practice, any approach that can create linguistic variants of a given question-answer pair can be applied here. Our main setting uses perturbation rules from PPDB and WordNet rules to create linguistic substitutions, making g a function of the perturbation type, τ, as well. We also experiment with a different source of changes by utilizing back-translation to create perturbations.

Backbone Model Our backbone model is comprised of input encoders, a controller network, and a set of re-usable modules, $\mathcal{M}$. The input encoders compute a set of visual and textual features for I and Q, respectively. As shown in Figure 5.5, the controller is responsible for decomposing the reasoning process into a sequence of steps that are executed by the network. At each step, t, the controller reads the question and, based on this question, produces module weights, $w^{(t)} \in \mathbb{R}^{|\mathcal{M}|}$, which are used for module selection, as well as a textual parameter, c_t, which is an input to the modules. In this formulation, module selection is essentially an attention over the module outputs, which allows for end-to-end training. The sequence of weights over all reasoning steps represents a soft layout that specifies what modules are utilized at each step. The modules implement different sub-tasks that the model has at its disposal during reasoning. Our framework is agnostic to the specific designs of the components of the backbone network and can work within the general controller-module framework. The backbone architectures can be instantiated differently to realize

Module	Input	Output	StackNMN [27]	XNM [28]
`Find`	x, c_t	$\hat{a}$	$\text{conv}_2(\text{conv}_1(x) \odot Wc_t)$	$\text{conv}(\phi(x, Wc_t))$
`Transform`	x, c_t, a	$\hat{a}$	$\text{conv}_2(\text{conv}_1(x) \odot W_1 \sum(a \odot x) \odot W_2 c_t)$	$\text{norm}((\sigma(RWc_t) \odot M)a)$
`Filter`	x, c_t, a	$\hat{a}$	$\texttt{And}(\texttt{Find}(x, c_t), a)$	same as StackNMN
`And`	a_1, a_2	$\hat{a}$	$\min(a_1, a_2)$	same as StackNMN
`Describe`	x, c_t, a	$z^{(t)}$	$W_1(W_2 \sum(a \odot x) \odot W_3 c_t)$	$\sum a \odot x$
`NoOp`	none	-	-	-

Table 5.2: Neural modules of StackNMN and XNM. Here x are the visual features, c_t is the textual parameter, each a (and $\hat{a}$) is an attention map over the image regions, $z^{(t)}$ is used to compute the final answer prediction, and all W are learned parameters. For XNM, R is the set of edge features and M is the adjacency matrix of the scene graph.

a variety of models with distinct module functionalities, reasoning steps, feature backbones, etc. In this work, we adopt three designs of the controller-module models to test the effects of our Q3R training framework.

For **StackNMN** [27], the original method uses grid features from a CNN [71] as visual features. For fair comparison with other models, we use object features [49] for StackNMN, which is a stronger feature backbone for VQA. The modules in this architecture largely use elementwise multiplications to fuse visual and linguistic features, compute different attention maps, or obtain answer vectors. Additionally, in `Find` and `Transform`, 1D convolutions are also used to compute weighted visual and multimodal features. The specific module designs for StackNMN are shown in Table 5.2.

For **XNM** [28], the visual features, $x \in \mathbb{R}^{K \times d_v}$, are object features that represent nodes in the input scene graph. The edge features, $R \in \mathbb{R}^{K \times K \times 2d_v}$, for the scene graph are the concatenations of neighboring edges and $M \in \mathbb{R}^{K \times K}$ is the adjacency matrix of the scene graph. The `Transform` module is the only module that considers the scene graph and uses the graph information to shift the visual attention according to the graph connectivity. These modules adopt the naming conventions of StackNMN, but XNM, in particular, has a different `Transform` implementation, which learns attention transforms on image scene graph representations, and an alternative multimodal fusion method [157], call it $\phi(\cdot, \cdot)$. The detailed module information is shown in Table 5.2.

We also present a hybrid of StackNMN and XNM, called **HybridNet**, for diagnostic and visualization purposes. Specifically, HybridNet utilizes the `Transform` module of StackNMN, while maintaining the rest of the design from XNM.

Regularization Method We propose to regularize the training of the backbone compositional model to improve its consistency and robustness against linguistic variations at the module level. Controlling and regularizing question relatedness at the module level offers several benefits. First, it provides finer control of the model's active sub-networks than only using supervision at the

Algorithm 5.1: Q3R Training Procedure

```
input: steps N; input data D; module g; model f
for i ∈ {1, ..., N} do
    sample (I_i, Q_i, A_i) from D
    compute L_CE w.r.t. (I_i, Q_i, A_i)
    u ~ Bernoulli(r)
    if u = 1 then
        sample (I_j, Q_j, A_j) from D
        τ ~ cat(T|ρ)
        (Q̃_j, Ã_j) ← g(Q_j, A_j, τ)
        compute L_m w.r.t. (I_j, Q_j, Q̃_j)
        L = L_CE + L_m
    else
        L = L_CE
    end
    update f to minimize L
end
```

output layer. Second, the related question pairs share intermediate activation similarity but do not necessarily need to match one another at a lower level (*e.g.*, attention maps within modules), making the model less sensitive to surface-level sentence variation. We center our regularization method around these two observations. Given an image, I_j, as well as a pairing of an original question, Q_j, and its perturbed version, $\tilde{Q}_j$, the controller maps each question to a set of module weights at each reasoning step, $w_j^{(t)}$ and $\tilde{w}_j^{(t)}$, respectively. Across all reasoning steps, these module weights can be interpreted as selecting "paths" over the grid of all modules across all time steps, especially when the weights are computed with Gumbel softmax [158] as done with XNM [28]. Ideally, if two questions agree on the basic sub-tasks, they should also agree on the activated module paths (as in Figure 5.4). Thus, we define the regularization loss term as:

$$\mathcal{L}_m(I_j, Q_j, \tilde{Q}_j) = \lambda \sum_{t=1}^{T} d(w_j^{(t)}, \tilde{w}_j^{(t)}), \tag{5.1}$$

where λ is a hyperparameter to scale the loss term and d is a distance metric between two distributions. We find that KL divergence or common vector norm losses, such as L1-norm, work well as d in the proposed loss term.

Training Procedure Given the regularization term, $\mathcal{L}_m$, and the VQA loss [30] between the predicted answer distribution and the ground truth answer prediction, $\mathcal{L}_{CE}(\hat{a}_i, a_i | I_i, Q_i)$ (see Chapter 2), we employ a multi-task training procedure [136, 137], where we treat each perturbation

type as a different task, optimizing for each perturbation type individually. As in Algorithm 5.1, for each input image-question pair sampled from VQA v2, we compute $\mathcal{L}_{CE}$. Then, with probability r, we sample a perturbed pairing to compute our module weight loss. Otherwise, we simply update the model using $\mathcal{L}_{CE}$. When utilizing our loss, we sample a perturbation type τ from a categorical distribution, denoted cat$(\mathcal{T}|\rho)$, over the set of perturbation types $\mathcal{T}$ with probabilities ρ.[2] We input the question Q_j and the perturbation type into g to obtain $\tilde{Q}_j$ that differs from the input question according to the sampled type. We use the perturbed question and original image-question pair, $(I_j, Q_j, \tilde{Q}_j)$, to compute $\mathcal{L}_m$. The network is then updated to minimize the sum of these loss terms. In practice, when operating on batches, we sample a batch of a particular perturbation type. We notice that this procedure offers a more stable learning process and better performance than mixing different perturbation types in a single batch.

5.3.2 Comparison to Expert Layout Supervision

Many NMNs adopt expert layouts, or programs, to guide the search for optimal module selections [25, 26]. These layouts are useful for guiding the reasoning process during training and parsers can be trained to parse questions into these structured, programtic representations [159]. However, while this approach is suitable for synthetic datasets with simple scenes and spatial reasoning [23], it has more limited success on realistic images [27]. Our loss can be viewed as providing weak supervision to module layouts, avoiding the need for ground-truth module layout annotation, which is costly and not as clearly defined for natural questions about real images.

5.4 EXPERIMENTS

Data and Metrics We use VQA v2 [12] for training and VQA P2 for our main evaluations. Since our goal is to benchmark and advance existing models' consistency, it requires ground truth answer labels for each test question, so we utilize the validation set for evaluation as is common practice [96, 97]. Our metrics are standard VQA accuracy as well as the consensus score (CS) [97] between pairs of related questions. A non-zero CS for a pair of questions requires a model to answer both questions correctly [97].

Models We pick representative models for our evaluations:

- **BAN** [104] is a bilinear model a top performing model on VQA v2 without external data.

[2] In all our experiments, we use a uniform distribution over the perturbation types.

- **Pythia** [143] is effectively a variant of the bottom-up top-down model [49] with stronger hyperparameter tuning.
- **Transformer** [67, 68, 73] is an architecture that acts as a multimodal encoder.
- **MCAN** [78], which is the state-of-the-art model and is composed of transformer layers, which are used to perform self-attention within each modality and co-attention between the modalities.

As described in Section 5.3.1, **StackNMN** [27] and **XNM** [28] are examples of expert-free, end-to-end trainable NMNs. For diagnostic and visualization purposes, we also experiment with a hybrid of these two networks, called HybridNet. Experiments with BAN, Pythia, Transformer, and MCAN examine the consistency of non-NMN state-of-the-art architectures. The NMN experiments explore the performance of different module implementations when employing our framework.

Settings All models use bottom-up top-down visual features [49] and pre-trained GloVe embeddings [66]. XNM, StackNMN, and HybridNet use the implementation provided by [28] to ensure consistency amongst different NMN models. We implement the modules and controller of StackNMN to match the paper description and official implementation. All NMN models are trained with the Adam optimizer [139] and have the same learning rate of 0.0008 and batch size of 256. Following their implementations, we use hidden dimension sizes of 512 for StackNMN and 1024 for XNM, while we use 1024 for HybridNet to match XNM. We use the recommended number of reasoning steps, $T = 3$, for XNM and use the same for StackNMN. HybridNet uses $T = 4$ to test longer reasoning sequences as well as for visualization purposes. With our Q3R framework, $\lambda = 1.0$ and $r = 0.2$ for XNM as well as HybridNet, and $\lambda = 0.5$ and $r = 0.1$ for StackNMN.

For BAN, we use the 8-glimpse model provided by the authors and adopt their training settings. We do not use the counting module [157] nor additional training data from Visual Genome [24]. For the transformer model, we use LXMERT [73] as the architecture and utilize the publicly available code with the settings from the authors. For fair comparison, we do not use large-scale pre-training. Pythia [143] utilizes the implementation provided by the authors as well as the recommended training settings. We also experiment with the cycle consistency [97] VQA robustness technique in combination with Pythia, which uses the settings provided by the author. For MCAN [78], we again use the provided implementation and respective recommended settings. We use the six layer version of MCAN for computational efficiency.

Model		VQA v2	Synonymous			Paraphrastic			Antonymous		
			Pert	*Ori*	Δ	*Pert*	*Ori*	Δ	*Pert*	*Ori*	Δ
BUTD	Pythia [143]	65.8	64.6	66.4	-1.8	54.8	56.8	-2.0	74.3	86.7	-12.4
Bilin.	BAN [104]	66.1	64.5	66.3	-1.8	56.3	56.7	-0.4	73.9	86.0	-12.1
Self-att.	Transformer [73]	63.5	61.0	64.2	-3.2	53.0	54.7	-1.7	73.0	84.2	-11.2
	MCAN [78]	67.3	65.9	67.8	-1.9	56.6	58.4	-1.8	77.4	88.4	-11.0
NMN	StackNMN [27]	62.6	61.2	63.5	-2.3	53.2	53.6	-0.4	74.8	84.9	-10.1
	XNM [28]	64.5	62.8	65.2	-2.4	55.6	56.8	-1.2	74.3	85.1	-10.8

Table 5.3: Benchmarking the robustness of existing VQA models on VQA P2. Accuracy on the VQA v2 validation set as well as VQA P2 subsets. *Ori* and *Pert* refer to the set of questions before and after applying a particular perturbation, respectively, and Δ is the difference in performance between these sets. For fair comparison, all models are trained from scratch with the training split of the VQA v2 dataset.

5.4.1 Benchmarking Robustness with VQA P2

We benchmark existing models on their robustness against controlled lexical perturbations, as shown in Table 5.3. Interestingly, we see that the different classes of models have less trouble with paraphrastic changes than they do with synonymous, with the average drop in performance being -1.3 compared to -2.2. This is likely due to the fact that paraphrastic changes tend to effect transitional phrases (*e.g.*, "*be considered*"), which models may ignore [160], whereas synonymous changes effect these as well as concept mentions (*e.g.*, "*car*") that are needed to answer the question. We see that models struggle the most with antonymous changes, dropping at least -10.1. Despite having lower VQA v2 accuracy, the NMN architectures perform better on the perturbed antonymous questions compared to BAN and Transformer. The results suggest that the Transformer model trained from scratch is one of the less robust architectures across the different types of perturbations. Alternatively, MCAN offers some of the highest scores, but still exhibits drops in performance on perturbed questions similar to Pythia or BAN. We also see that Pythia performs similarly to BAN on each set except for the paraphrastic perturbed questions, where it drops performance noticeably more than BAN. Overall, we see that existing models generally struggle with these controlled variations, where the largest difficulties appear on the logical consistency measured with antonymous perturbations and concept mention consistency measured with synonymous perturbations. To our knowledge, this is the first study of VQA robustness analysis on different types of linguistic variations.

5.4.2 Benchmarking Cycle Consistency on VQA P2

An existing approach to consistent visual question answering is cycle consistency (CC) [97]. This approach jointly trains a question generator along with the question answering model. The question answering model is trained to yield consistent answers between the original and generated questions. During the training process, the generated questions used to train the question answering model are filtered such that the similarity (*i.e.*, cosine similarity) must be at least a threshold (*e.g.*, 0.9). Shown in Table 5.4, we evaluate their model trained with and without this framework on VQA P2. Although CC is technically model agnostic, we use Pythia for a consistent comparison with the results seen on VQA-Rephrasings. We can see that the CC model drops in performance on the perturbed data and CS compared to the base model, while maintaining or slightly improving the performance on the original data and VQA v2 as a whole. This result is interesting since, in addition to enforcing consistency, this approach essentially augments the questions with similar ones that are within a similarity threshold of the original, which presumably many of the lexically altered questions in VQA P2 would fall within. However, this approach hinders the consistency towards these lexical perturbations. This also highlights the usefulness of VQA P2 as a diagnostic benchmark with realistic, non-trivial variations not found in VQA-Rephrasings.

	VQA P2			VQA v2
Model	*Pert*	*Ori*	CS	
Pythia	**65.9**	69.0	**70.2**	65.8
+CC [97]	65.6	69.0	69.8	**65.9**

Table 5.4: Performance of the cycle consistency method [97] on VQA P2.

5.4.3 Improving Model Robustness Using Q3R

We evaluate the effect of adding Q3R to different NMN architectures (+Q3R) and compare to knowledge-aware augmentation alone (+*Aw*). Table 5.5 shows that, on VQA P2, both data augmentation and Q3R result in significant improvements over the base models for both accuracy and CS. Comparing Q3R and *Aw* augmentation on VQA P2, we observe performance gains on all metrics, including a CS improvement of $+0.3$ and $+0.5$ for StackNMN and XNM, respectively. Although our focus is on consistency, not overall VQA v2 accuracy, results show that models regularized by Q3R generally see better performance on VQA v2 accuracy.

Generalizing to Human Paraphrased Questions Human-written rephrasings typically contain various sources of change: such as those that involve common sense knowledge, or structural

Model	VQA P2			VQA v2	VQA-R	
	Pert	*Ori*	CS		*Rep*	CS
StackNMN	63.3	66.9	66.2	62.6	52.9	51.7
+*Aw*	66.7	67.3	72.1	62.6	52.8	51.6
+Q3R	**67.1**	**67.5**	**72.4**	**62.8**	**53.3**	**52.0**
XNM	64.7	68.3	68.8	64.5	54.8	54.6
+*Aw*	68.0	68.6	73.9	64.3	55.0	54.8
+Q3R	**68.1**	**68.9**	**74.4**	**64.7**	**55.5**	**55.0**

Table 5.5: Effect of adding Q3R to different NMNs, measured by accuracy and CS. VQA-R is VQA-Rephrasings and *Rep* means rephrased questions.

level change of sentences. Nonetheless, we are interested in observing the outcome of the additional study on VQA-Rephrasings. Shown in Table 5.5, when trained with our Q3R framework, XNM and StackNMN receive a $+0.7$ and $+0.4$ accuracy improvement on paraphrased questions as well as a $+0.4$ and $+0.3$ improvement on CS score, respectively. We find that augmentation alone is less effective than Q3R on this data, and actually decreases performance for StackNMN. Q3R's moderate gain may be explained by the fact that linguistic perturbations are present in some human-written questions, so the model's robustness on this dataset benefits from our training framework.

Analysis by Perturbation Types A benefit of having information on the specific types of linguistic variations is that we can profile a model's performance by each type to assist understanding and diagnosis. Table 5.6 shows that models are generally more confident with the antonymous perturbations, likely because "yes/no" questions have higher answer prediction scores in general. We note that performance gains are significant on single-word changes and less obvious on multi-word changes.

Model	Syn.		Par.		Ant.	
	μ	CS	μ	CS	μ	CS
StackNMN	62.6	64.7	51.6	53.8	79.8	76.1
+Q3R	**64.3**	**70.6**	**53.4**	**58.6**	**84.3**	**83.9**
XNM	64.1	67.6	**56.7**	60.7	79.7	76.0
+Q3R	**65.7**	**72.9**	56.4	**61.8**	**84.4**	**84.7**

Table 5.6: Average accuracy (μ) between *Pert*+*Ori* and CS on each type of perturbation: Synonymous (Syn.), Paraphrastic (Par.) and Antonymous (Ant.).

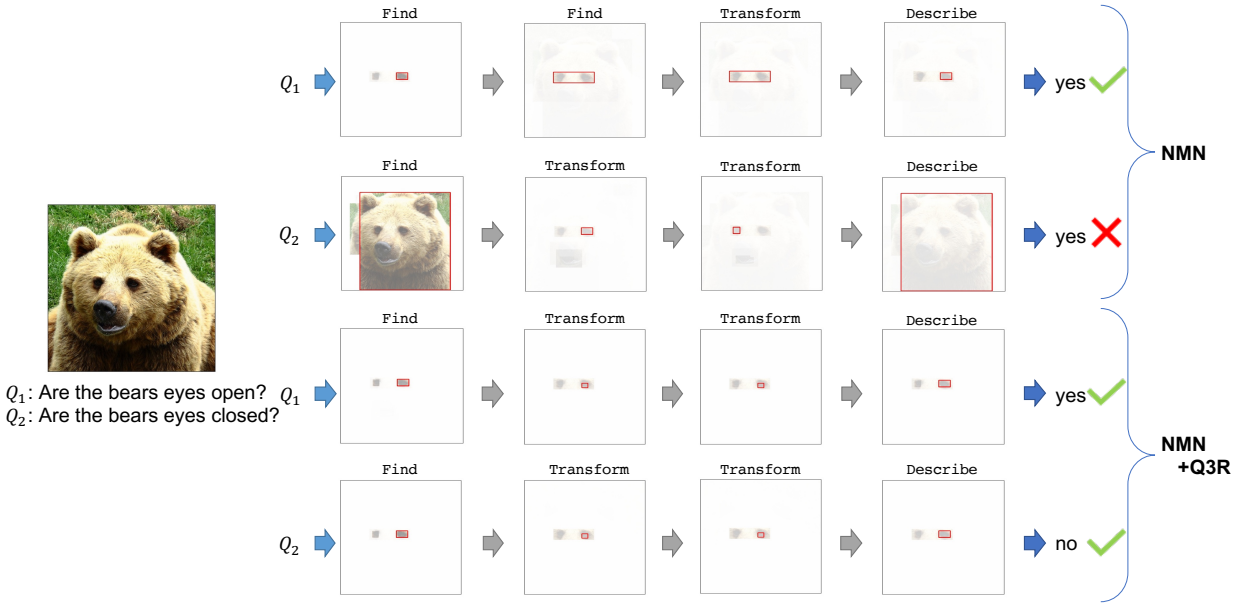


Figure 5.6: Visual attentions and selected modules across reasoning steps of HybridNet. Attention weights are indicated by the transparency of the bounding box; red boxes are the highest weighted regions.

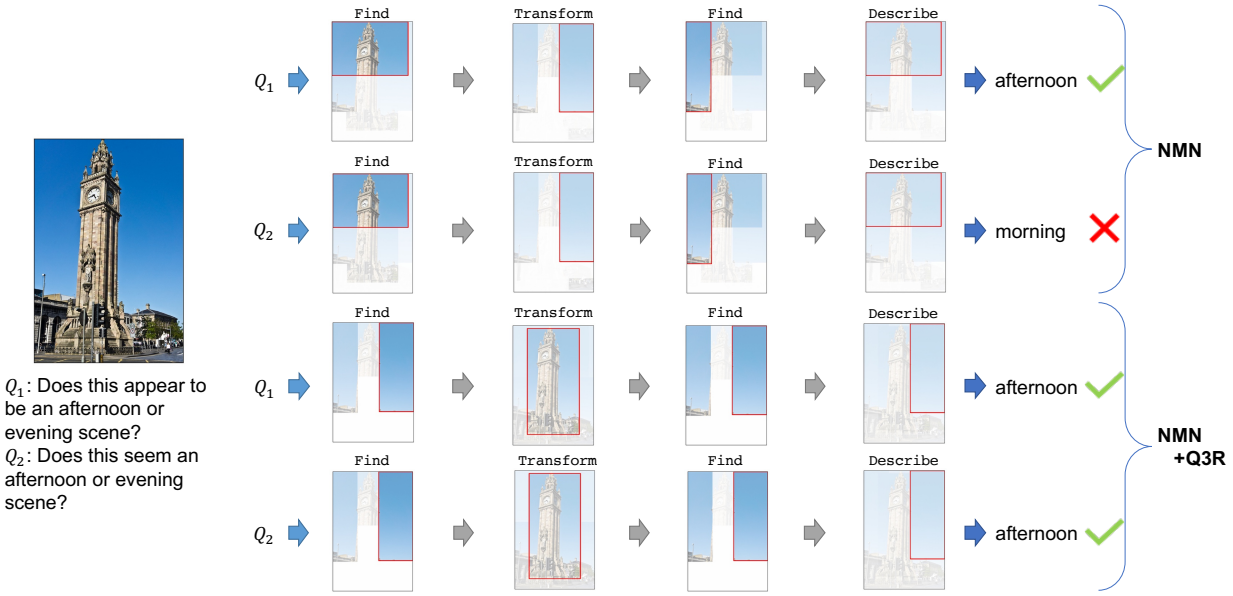


Figure 5.7: Example reasoning outputs for a paraphrastic pair of questions. Again, the top two rows are from HybridNet trained on VQA v2 and the bottom two are from HybridNet+Q3R.

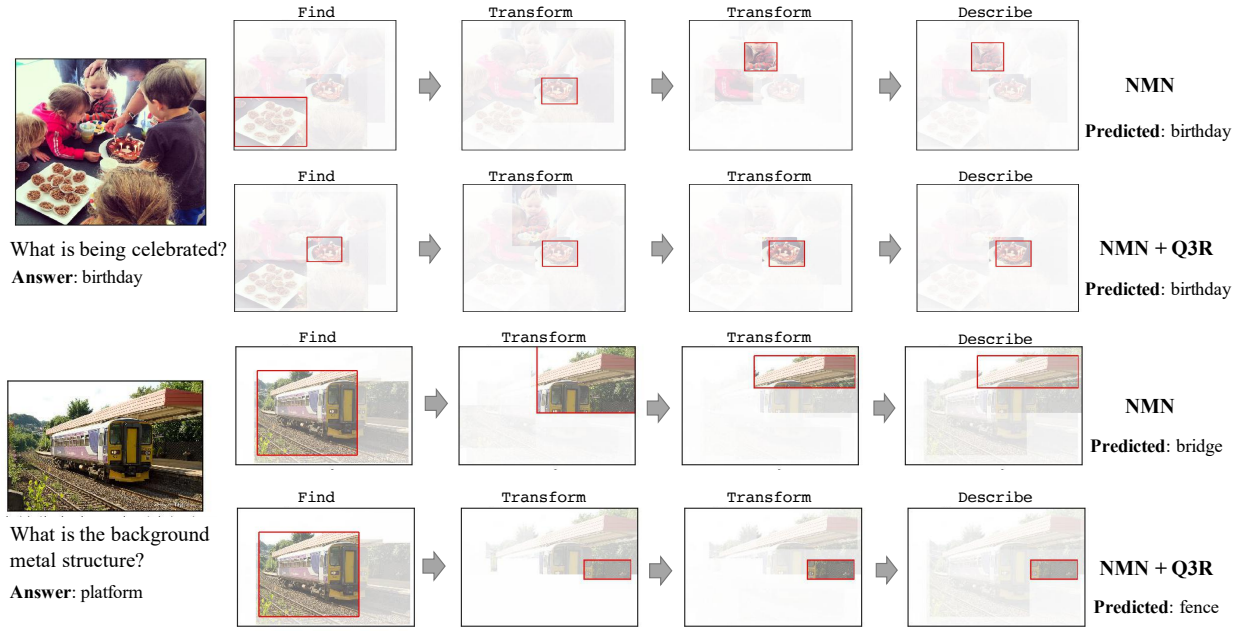


Figure 5.8: Another example of reasoning outputs at different steps with and without our framework.

Regularization Improves Interpretability To further understand the effect of applying our loss on the reasoning steps involved in executing NMNs, we compute different statistics using HybridNet. For each question in VQA P2, we compute the average normalized difference in output module weights between the perturbed question, $\tilde{Q}_j$, and the original question, Q_j for all steps. Applying Q3R leads to a 93% reduction (7.4×10^{-3} to 0.5×10^{-3}) on the average difference of the module weight distribution at each step. In other words, our framework encourages more consistent reasoning steps between related questions.

It also appears that our framework can lead to more intuitive visual attentions, as shown in Figure 5.6, Figure 5.7, and Figure 5.8. In both examples, we see that both the module selections and visual attentions become more consistent when utilizing Q3R. In Figure 5.6, the model trained without Q3R maintains its answers despite the perturbation, whereas the model trained with our framework predicts the appropriate answer and also matches visual attentions between them. Then, in Figure 5.7, we see an example where the questions share the same meaning and the same modules are selected for both questions and both models, but the model without our framework yields inconsistent answers. Finally, we observe in Figure 5.8 that the model trained with Q3R attends to more intuitive visual regions throughout the reasoning process. These results suggest that our framework may give rise to more interpretable and consistent visual reasoning.

Model	VQA P2			VQA v2
	Pert	*Ori*	CS	
StackNMN	63.3	66.9	66.2	62.6
+*Ag*	63.4	66.9	67.0	62.7
+Q3R	**64.0**	**67.1**	**67.8**	**62.9**
StackNMN	63.3	66.9	66.2	62.6
+*Aw*	66.7	67.3	72.1	62.6
+Q3R	**67.1**	**67.5**	**72.4**	**62.8**

Table 5.7: Comparison with different sources of augmentation: knowledge-aware (*Aw*) and knowledge-agnostic (*Ag*).

5.4.4 Other Perturbation Sources

As noted in Section 5.2.3, we can explore the impact of knowledge-aware (*Aw*) and knowledge-agnostic augmentation (*Ag*) methods as well as how they affect the performance of Q3R (*i.e.*, different g). For knowledge-agnostic augmentation, we employ a top-performing, pre-trained machine translation model [161] to generate back-translated questions. To quantify the impact of the different methods, we compare against the base model and augmentation alone. Table 5.7 shows the performance with StackNMN, where Q3R improves performance beyond augmentation for both sources. Even in the challenging knowledge-agnostic setting, Q3R improves consistency by $+0.8$ over augmentation and $+1.6$ over the base model. This suggests that our framework can be effective for improving robustness regardless of the source of augmentation, whereas augmentation alone is more dependent upon the source.

5.4.5 Regularizing Transformer-based VQA Models

Our results seem to suggest an advantage of representing and incorporating inductive bias at the modular level, rather than just using answer-level supervision. We have demonstrated our method using NMNs, which can also generalize to other interpretable VQA architecture that involves the computation of sub-tasks, such as those based on executable symbolic programs [159, 162]. Given the improved robustness from Q3R, it would be ideal to be able to employ similar regularization techniques to other state-of-the-art architectures, such as transformers [73, 78]. We believe the dataset presented by our work is a new and useful resource, which helps elucidate the shortcomings of such models and steer more interest into improving the compositionality of transformer models so that they can be more congruent with modular/symbolic regularization.

We perform an initial study to explore whether or not state-of-the-art transformer-based archi-

Model	VQA P2			VQA v2
	Pert	*Ori*	CS	
MCAN	67.5	**70.5**	72.2	**67.3**
+*Ag*	67.5	70.1	72.4	66.9
+Reg.	**67.8**	70.2	**72.6**	66.9
MCAN	67.5	70.5	72.2	**67.3**
+*Aw*	70.4	70.5	76.3	67.1
+Reg.	**70.5**	**70.7**	**76.6**	67.2

Table 5.8: Performance of augmentation and applying regularization similar to Q3R on MCAN.

tectures can exploit augmented data to improve robustness as well as the potential for applying robustness regularization to these models. To do so, we again experiment with knowledge-aware and knowledge-agnostic augmentation as well as regularization. For adding regularization, we use our Q3R training algorithm, but we adapt the loss function to regularize the self-attention distributions of the multimodal layers of MCAN. This is similar to knowledge distillation approaches that have seen successful uses for shrinking large language models [163]. Specifically, let $A^{(l)} \in \mathbb{R}^{H \times K \times K}$ be the self-attention values of the queries in MCAN's l^{th} guided attention layer for the original question, where H is the number of attention heads and K is the number of query vectors. Likewise, let $\tilde{A}^{(l)} \in \mathbb{R}^{H \times K \times K}$ be the same tensor for the perturbed question. The loss is given by

$$\mathcal{L}_m(I_j, Q_j, \tilde{Q}_j) = \lambda \sum_{l=1}^{L} d(A^{(l)}, \tilde{A}^{(l)}) = \lambda \sum_{l=1}^{L} \frac{\xi^{(l)}}{HK} \sum_{h=1}^{H} \sum_{k=1}^{K} D_{KL}(\tilde{A}^{(l)}_{h,k} \mid\mid A^{(l)}_{h,k}), \tag{5.2}$$

where, for both A and $\tilde{A}$, $A^{(l)}_{h,k} \in \mathbb{R}^K$ and $\sum_i A^{(l)}_{h,k,i} = 1$ for $k = 1, ..., K$. $\xi^{(l)}$ is a layer specific scalar used to control the influence of each layer. In our experiments, we apply this loss to the first three out of six layers and use $\xi^{(l)} = l/3$. For our training algorithm, we use $\lambda = 1.0$ and $r = 0.2$. Note, when adapting our framework to this transformer-based model, the rest of Algorithm 5.1 remains exactly the same, so we simply use a different backbone model (*i.e.,* MCAN) and loss (*i.e.,* Equation 5.2).

We see that the model is able to exploit the augmented data, but these improvements are more tied to the augmentation source than the interpretable models. For example, when using knowledge-agnostic augmentation, we see decreases in performance on the original questions and VQA v2 overall. Meanwhile, with knowledge-aware augmentation, the model is able to maintain performance on the original questions. This sort of asymmetry in performance between sources is not observed for the interpretable models. We further observe that regularization can help improve per-

formance, particularly consistency, with either augmentation method. However, this transformer-based model is not as amenable to regularization and the added regularization does not fully recover the drop in performance on the original questions and overall VQA v2. More investigation is needed to further improve robustness of these transformer-based models using different regularization techniques or other methods. These results underscore one of the potential strengths in interpretable architectures, which is that the design of these architectures may be more amenable to regularization and diagnosis compared to other architectures that are more opaque, such as MCAN.

5.4.6 HybridNet Performance

Although we primarily use HybridNet for visualization and diagnosis, we also benchmark HybridNet to verify the performance. These results are presented in Table 5.9 and Table 5.10. Although we use HybridNet for visualizations purposes, we see that this model also benefits from our framework.

Model	VQA P2			VQA v2
	Pert	*Ori*	CS	
HybridNet	63.3	67.0	66.6	63.0
+Q3R	**67.0**	**67.4**	**72.5**	**63.1**

Table 5.9: Performance of HybridNet with and without Q3R.

Model	Syn.		Par.		Ant.	
	μ	CS	μ	CS	μ	CS
HybridNet	62.5	65.0	53.3	55.7	79.9	76.4
+Q3R	**64.1**	**70.8**	**55.0**	**59.4**	**84.0**	**83.7**

Table 5.10: Performance of HybridNet by perturbation type.

5.5 SUMMARY

We show that a promising direction to improve the robustness and consistency of VQA models is by learning from lexical perturbations. We demonstrate that using such lexical perturbations allow models to learn the relationships amongst different concept mentions and phrasings. We propose a novel approach based on modular networks, which creates two questions related by linguistic

perturbation and regularizes the visual reasoning process between them to be consistent during training. We introduce a new benchmark, VQA P2, that features controlled, non-trivial linguistic variations that allows us to investigate and diagnose sources of inconsistencies in model predictions. We show that existing models have difficulties with different types of linguistic variations and that our approach is effective towards improving robustness and generalization ability.

CHAPTER 6: KNOWLEDGE-AWARE VIDEO CAPTIONING

6.1 MOTIVATION

When operating in complex domains, such as the news domain, it becomes important for models to not only link concept mentions across modalities, but also learn associations between concepts in background knowledge and the visual content. For example, if a model is generating a description of a news image that shows rubble and the aftermath of a catastrophic event, then the model should be able to associate concepts, such as events and entities, found in background knowledge in order to include some amount of insight into what caused the destruction seen in the video (*e.g.,* an attack or natural disaster). In this work, we investigate captioning in the news domain and present a technique for learning such associations in an end-to-end fashion.

Video captioning is a challenging task that seeks to automatically generate a natural language description of the content of a video. Many video captioning efforts focus on learning video representations that model the spatial and temporal dynamics of the videos [55, 87, 91]. Although the language generation component within this task is of great importance, less work has been done to enhance the contextual knowledge conveyed by the descriptions.

The descriptions generated by existing methods tend to be generic, describing only what is evidently visible and lacking specific knowledge, like named entities and event participants, as in Figure 6.1a. In many situations, however, generic descriptions are uninformative as they do not provide contextual knowledge. For example, in Figure 6.1b, details such as *who is speaking* and *why* are imperative to truly understanding the video, since contextual knowledge gives the surrounding circumstances or cause of the depicted events.

To address this problem, we collect a news video dataset, where each video is accompanied by meta-data (*e.g.,* tags and date) and a natural language description of the content in, and/or context around, the video. We create an approach to this task that is motivated by two observations.

First, the video content alone is insufficient to generate the description. Named entities or events are necessary to identify the participants, location, and/or cause of the video content. These could potentially be mined from visual evidence, but training such a system is exceedingly difficult [93]. Further, not all the knowledge necessary for the description may appear in the video. In Figure 6.2a, the video depicts much of the description content, but knowledge of the speaker ("*Carles Puigdemont*") is missing from the visual evidence since the speaker never appears in the video.

Second, one may use a video's meta-data to retrieve topically related news documents that contain the named entities or events that appear in the video's description, but these may not be specific to the video content. For example, in Figure 6.2b, the video discusses the "*heightened security*"

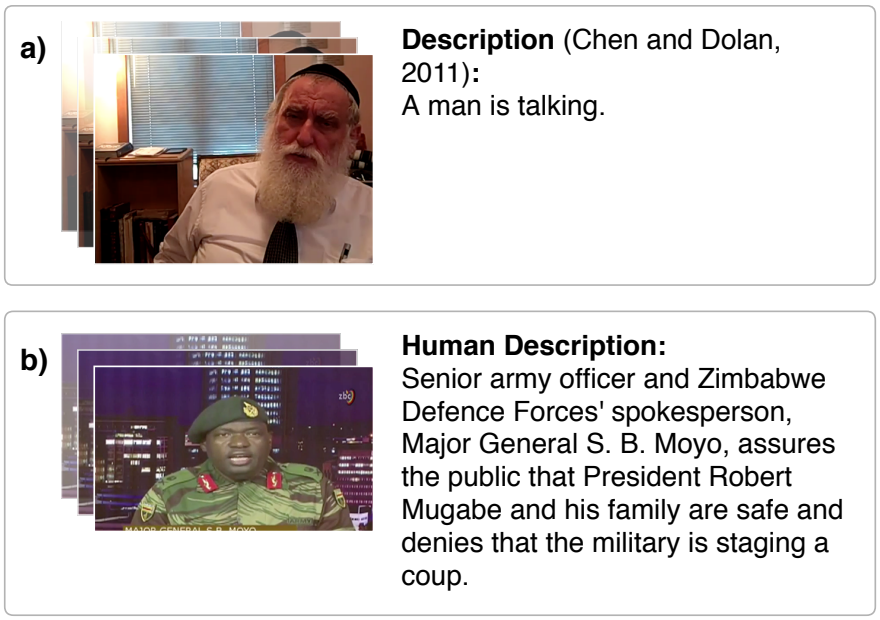


Figure 6.1: Comparison of machine (a) and human (b) generated descriptions.

and does not depict the arrest directly. Related documents capture background knowledge about the attack that led to the "*heightened security*" as well as the arrest, but they may not describe the actual video content, which displays some of the increased security measures.

Thus, we propose to retrieve weakly aligned, topically related news documents from which we seek to extract named entities [164] and events [165] likely relevant to the video. We then propose to use this knowledge in the generation process through an *entity pointer network*, which learns to dynamically incorporate extracted entities into the description, and through a new *knowledge gate*, which conditions the generator on the extracted event and entity types. We include the video content in the generation by learning video representations using a spatio-temporal hierarchical attention that spatially attends to regions of each frame and temporally attends to different frames. We call the combination of these generation components the *Knowledge-aware Video Description* (KaVD) network.

6.2 APPROACH

6.2.1 Document Retrieval and Knowledge Extraction

We gather weakly aligned, topically related news documents as a source of background knowledge using the video meta-data. For each video, we use the corresponding tags to perform a

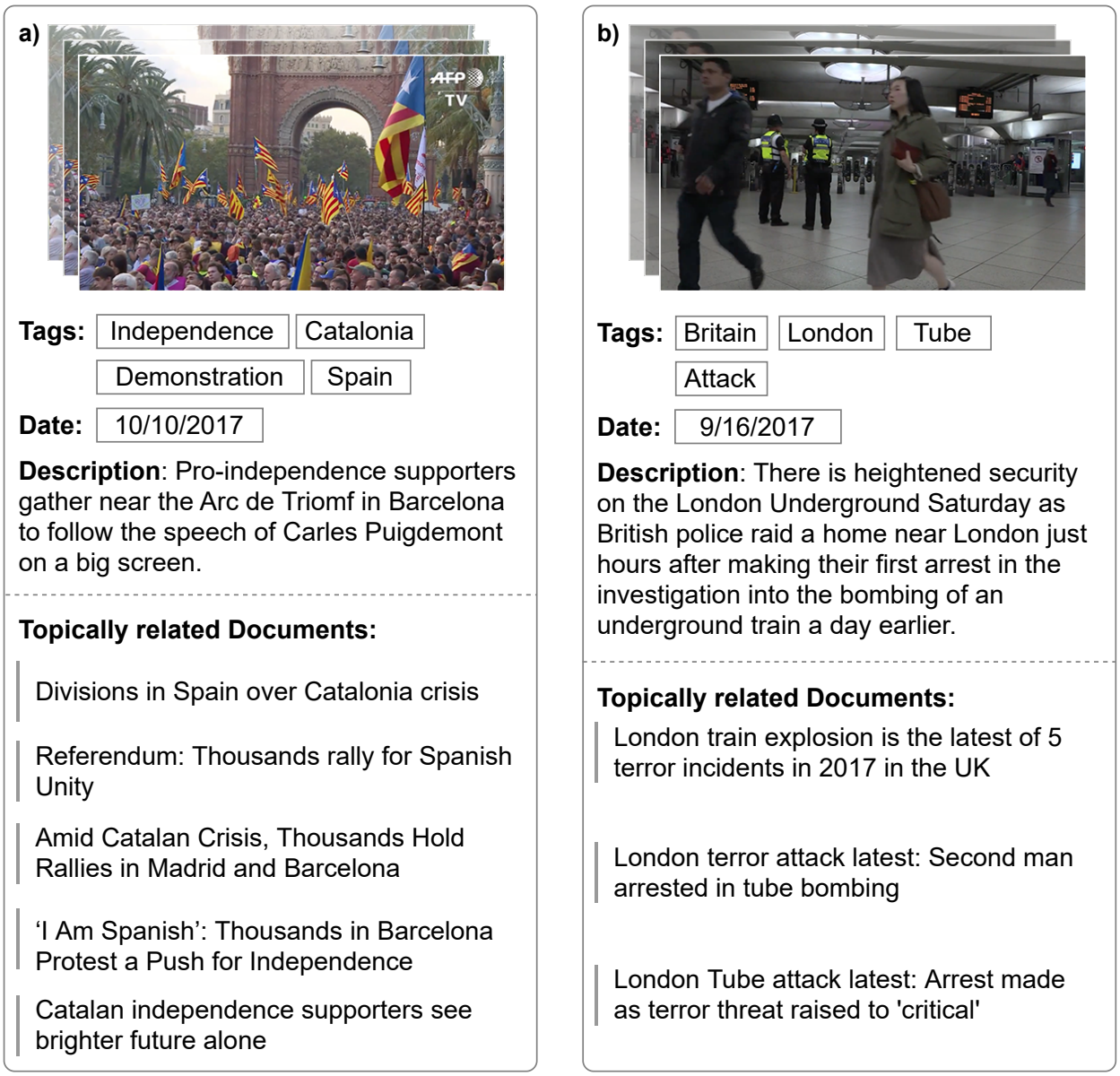


Figure 6.2: Examples from our dataset with some retrieved topically related documents.

keyword search on documents from a number of popular news outlet websites.[1] We filter these documents by the date associated with video, only keeping documents that are written within d days before and after the video upload date.[2] The keyword search gathers documents that are at least somewhat topically relevant and filtering by date increases the likelihood that the documents reference the specific events and entities of the video, since the occurrences of entity and event mentions across news documents tend to be temporally correlated. We retrieve an average of 3.1

[1] BBC, CNN, and New York Times.

[2] $d = 3$ in our experiments.

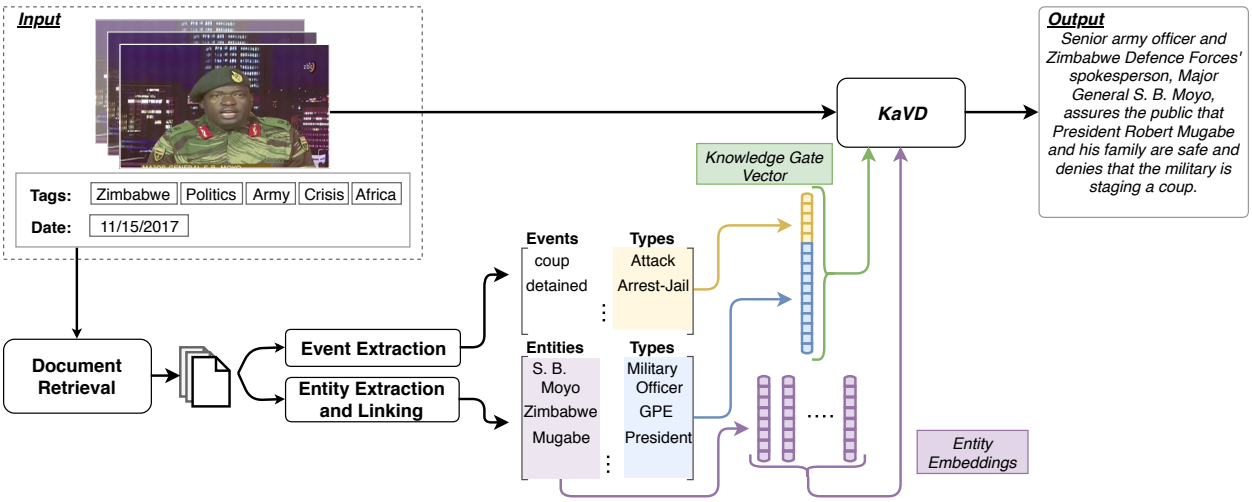


Figure 6.3: Overview of our knowledge-aware video captioning approach.

articles per video and find that on average 68.8% of the event types and 70.6% of the entities in the ground truth description also appear in corresponding news articles. In Figure 6.3, the retrieved background documents include the entity "*Mugabe*" and the event "*detained*", which are relevant to the video description.

We apply a publicly available entity discovery and linking system [164] to extract named entities and their fine-grained types (*e.g.*, "`President`" versus "`Military Officer`"). Additionally, we use an event extraction system [165] to extract events and their arguments. For example, in Figure 6.3, we get entities "*S. B. Moyo*", "*Zimbabwe*", and "*Mugabe*" with their respective types, "`Military Officer`", "`GPE`" (Geo-political Entity), and "`President`". Likewise, we obtain events "*coup*" and "*detained*" with their respective types, "`Attack`" and "`Arrest-Jail`".

We encode the entities and events into representations that can be fed to the model. First, we obtain an entity embedding, $\mathbf{e}_m$, for each entity by averaging the embeddings of the words in the entity mention. Second, we encode the entity and event types into a one-hot knowledge gate vector, $\mathbf{k}_0$. Each element of $\mathbf{k}_0$ corresponds to an event or entity type (*e.g.*, "`Arrest-Jail`" event type or "`President`" entity type), so the jth element, $k^{(j)}$, is 1 if the entity or event type is found in the related documents and 0 otherwise. $\mathbf{k}_0$ serves as the initial knowledge gate vector of the decoder. The entity embeddings provide semantic representations of the entities that can appear in the output. Meanwhile, the knowledge gate vector aids the generation process by providing the model with the event and entity types. By doing so, the model can learn to associate patterns of knowledge elements (*e.g.*, "`Arrest-Jail`" events alongside "`Military Officer`" and "`President`") with the visual content (*e.g.*, military police patrolling the streets) in order to generate descriptions with complex events (*e.g.*, "*coup*").

6.2.2 KaVD Network

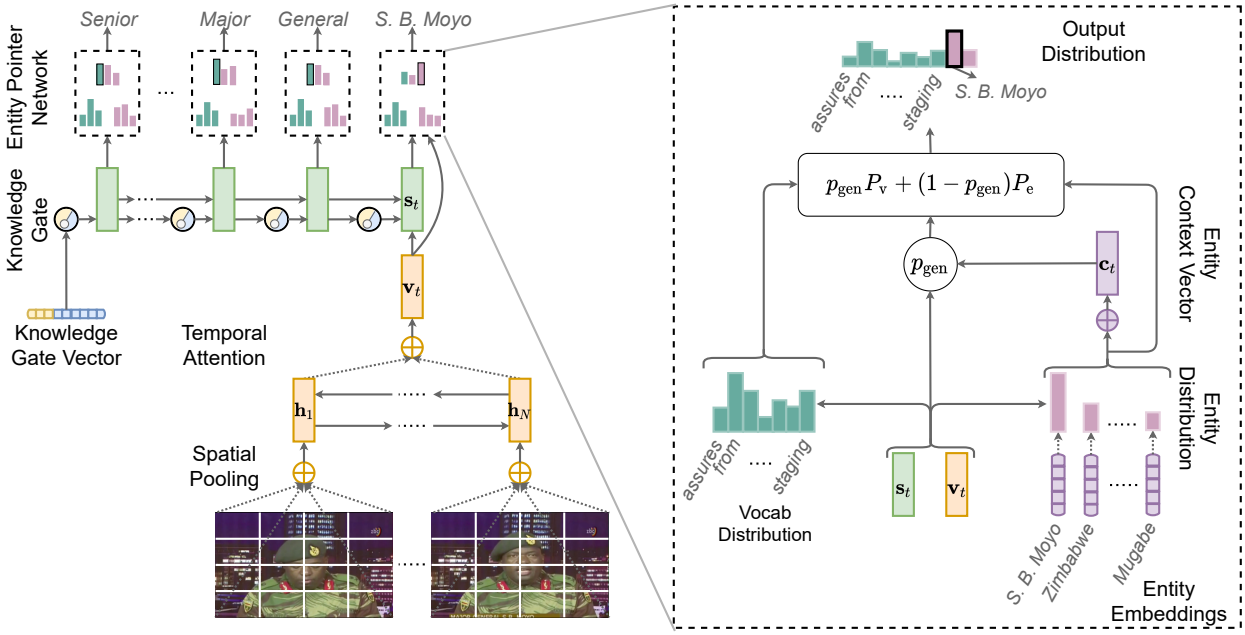


Figure 6.4: KaVD Network. At each decoder time step, the model computes p_{gen} to determine whether to emit a vocabulary word or a named entity from the topically related documents.

Our model learns video representations using hierarchical, or multi-level, attention [166, 167]. The encoder is comprised of a spatial pooling layer [54] and bi-directional Long Short-Term Memory network (LSTM) [46] temporal encoder. The spatial pooling allows the model to consider different locations of each frame (Figure 6.4). The temporal encoder incorporates motion into the frame representations by encoding information from the preceding and subsequent frames [55]. We use a LSTM decoder, which applies a temporal attention [48] to the frame representations at each step. To generate each word, the decoder computes its hidden state, adjusts this hidden state with the *knowledge gate* output at the current time step, and determines the most probable word by utilizing the *entity pointer network* to decide whether to generate a named entity or vocabulary word. Pointer networks are effective at incorporating out-of-vocabulary (OOV) words in output sequences [107, 168]. In previous research, OOV words may appear in the input sequence, in which case they are copied into the output. Analogously, in our approach, named entities can be considered as OOV words that are from a separate set instead of the input sequence. In the following equations, where appropriate, we omit bias terms for brevity.

Encoder The input to the encoder is a sequence of video frames, $\{F_1, ..., F_N\}$. First, we extract frame-level features by applying a Convolutional Neural Network (CNN) [169, 170, 171, 172, 173] to each frame, F_i, and obtaining the response of a convolutional layer, $\{\mathbf{a}_{i,1}, ..., \mathbf{a}_{i,L}\}$, where $\mathbf{a}_{i,l}$ is

a D-dimensional representation of the lth location of the ith frame (*e.g.*, the top left box of the first frame in Figure 6.4). We apply the spatial pooling to these location representations:

$$\mathbf{z}_i = \sum_{l=1}^{L} \xi_{i,l} \mathbf{a}_{i,l}. \tag{6.1}$$

By default, we use attentive pooling, where

$$\alpha_{i,l} = \tanh\left(\mathbf{W}_\alpha \mathbf{a}_{i,l} + \mathbf{b}_\alpha\right), \tag{6.2}$$

$$\xi_{i,l} = \text{softmax}\left(\alpha_{i,l}\right), \tag{6.3}$$

but we also experiment with mean pooling. These frame representations, $\{\mathbf{z}_1, ..., \mathbf{z}_N\}$, are input to a bi-directional LSTM, producing temporally encoded frame representations $\{\mathbf{h}_1, ..., \mathbf{h}_N\}$.

Decoder The decoder is a LSTM cell with the addition of a temporal attention mechanism, a *knowledge gate* and an *entity pointer network*. At each decoder step t, we apply a temporal attention to the frame representations:

$$\beta_{t,i} = \tanh\left(\mathbf{W}_{\beta,h} \mathbf{h}_i + \mathbf{W}_{\beta,s} \mathbf{s}_{t-1} + \mathbf{b}_\beta\right), \tag{6.4}$$

$$\eta_{t,i} = \text{softmax}\left(\beta_{t,i}\right), \tag{6.5}$$

$$\mathbf{v}_t = \sum_{i=1}^{N} \eta_{t,i} \mathbf{h}_i, \tag{6.6}$$

where $\mathbf{s}_{t-1}$ is the previous decoder hidden state. This yields a single, spatio-temporally attentive video representation, $\mathbf{v}_t$. We then compute an intermediate hidden state, $\hat{\mathbf{s}}_t$, by applying the decoder LSTM to $\mathbf{s}_{t-1}$, $\mathbf{v}_t$, and previous word embedding, $\mathbf{x}_{t-1}$. The final decoder hidden state is determined after the knowledge gate computation.

The motivation for the knowledge gate is that it biases the model to generate sentences that contain specific knowledge relevant to the video and topically related documents, acting as a kind of coverage mechanism [174]. It essentially helps the model learn to associate patterns of entity and event types with the visual content and the concepts therein. For example, given the retrieved event types in Figure 6.3, the knowledge gate encourages the decoder to generate the event trigger "*coup*" due to the presence of the "`Attack`" event type. The knowledge gate, $\mathbf{g}_t$, is given by

$$\mathbf{g}_t = \sigma\left(\mathbf{W}_{g,v}[\mathbf{x}_{t-1}, \mathbf{v}_t] + \mathbf{W}_{g,s} \hat{\mathbf{s}}_t + \mathbf{b}_g\right), \tag{6.7}$$

$$\mathbf{k}_t = \mathbf{g}_t \odot \mathbf{k}_{t-1}, \tag{6.8}$$

where $[\mathbf{x}_{t-1}, \mathbf{v}_t]$ is the concatenation of these two vectors. This gating step determines the amount of the entity and event type features contained in $\mathbf{k}_{t-1}$ to carry to the next step. With the updated $\mathbf{k}_t$, we compute the decoder hidden state, $\mathbf{s}_t$, as

$$\mathbf{s}_t = \hat{\mathbf{s}}_t + (\mathbf{o}_t \odot \tanh(\mathbf{W}_{s,k}\mathbf{k}_t + \mathbf{b}_s)), \tag{6.9}$$

where $\mathbf{o}_t$ is the output gate of the LSTM.

Our next step is to generate the next word. The model needs to produce named entities (*e.g.*, "*S. B. Moyo*" and "*Robert Mugabe*") throughout the generation process. These named entities tend to occur rarely if at all in many datasets, including ours. We overcome this issue by using the entity embeddings from the topically related documents as potential entities to incorporate in the description. We adopt a soft switch pointer network [107], as our entity pointer network, to perform the selection of generating words or entities.

For our entity pointer network to predict the next word, we first predict a vocabulary distribution, $P_{\text{v}} = \psi(\mathbf{s}_t, \mathbf{v}_t)$, where $\psi(\cdot)$ is a softmax output layer. $P_{\text{v}}(w)$ is the probability of generating word w from the decoder vocabulary. Next, we compute an entity context vector, $\mathbf{c}_t$, using a soft attention:

$$\gamma_{t,m} = \tanh(\mathbf{W}_{\gamma,\text{e}}\mathbf{e}_m + \mathbf{W}_{\gamma,s}\mathbf{s}_t + \mathbf{W}_{\gamma,v}\mathbf{v}_t + \mathbf{b}_\gamma), \tag{6.10}$$

$$\epsilon_{t,m} = \text{softmax}(\gamma_{t,m}), \tag{6.11}$$

$$\mathbf{c}_t = \sum_{m=1}^{M} \epsilon_{t,m}\mathbf{e}_m. \tag{6.12}$$

We use the scalars $\epsilon_{t,m}$ as our entity probability distribution, P_{e}, where $P_{\text{e}}(E_m) = \epsilon_{t,m}$ is the probability of generating entity mention E_m. We compute the probability of generating a word from the vocabulary, p_{gen}, as

$$p_{\text{gen}} = \sigma(\mathbf{w}_c^\intercal\mathbf{c}_t + \mathbf{w}_s^\intercal\mathbf{s}_t + \mathbf{w}_x^\intercal\mathbf{x}_{t-1} + \mathbf{w}_v^\intercal\mathbf{v}_t). \tag{6.13}$$

Finally, we predict the probability of word w as

$$P(w) = p_{\text{gen}}P_{\text{v}}(w) + (1 - p_{\text{gen}})P_{\text{e}}(w), \tag{6.14}$$

and select the word of maximum probability. In Equation 6.14, $P_{\text{e}}(w)$ is 0 when w is not a named entity. Likewise, P_{v} is 0 when w is an OOV word. For the example in Figure 6.4, the vocabulary distribution, P_{v}, has the word "*from*" as the most probable word and the entity distribution, P_{e}, has the entity "*S. B. Moyo*" as the most probable entity. However, by combining these two distribution using p_{gen}, the model switches to the entity distribution and correctly generates "*S. B. Moyo*".

6.3 NEWS VIDEO DATASET

Dataset	Domain	#Videos	#Sentences	Vocab Size	Named Entities/Sentence
TACos M-L [37]	Cooking	14,105	52,593	2,864	0.1×10^{-4}
MSVD [31]	Multi-category	1,970	70,028†	13,010	0.4×10^{-2}
MSR-VTT [11]	20 categories	10,000	200,000†	29,316	1.4×10^{-1}
News Video (Ours)	News	2,883	3,302	9,179	2.1

Table 6.1: Comparison of our news video dataset to other datasets. † indicates that the dataset has multiple, single-sentence reference descriptions for each video.

Current datasets for video description generation focus on specific [37] and general [11, 31] domains, but do not contain a large proportion of descriptions with specific knowledge like named entities as shown in Table 6.1. In our news video dataset, the descriptions are replete with important knowledge that is both necessary and challenging to incorporate into the generated descriptions.

Our news video dataset contains AFP international news videos from YouTube.[3] These videos are from October, 2015 to November, 2017 and cover a variety of topics, such as protests, attacks, natural disasters, trials, and political movements. The videos are "on-the-scene" and contain some depiction of the content in the description. For each video, we take the YouTube descriptions given by AFP News as the ground-truth descriptions we wish to generate. We collect the tags and metadata (*e.g.*, upload date). We filter videos by length, with a cutoff of 2 minutes, and remove videos which are videographics or animations.

For preprocessing, we tokenize each sentence, remove punctuation characters other than periods, commas, and apostrophes, and replace numerical quantities and dates/times with special tokens. For efficiency, we sample frames at a rate of 1 frame per second. We randomly select 400 videos for testing, 80 for validation, and 2,403 for training.

6.4 EXPERIMENTS

6.4.1 Model Comparisons

We test our method against different baselines that control for the effects of each of our major components:

- **Article-only.** We use a summarization model [107] to generate the description by summarizing the topically related documents.

[3] `https://www.youtube.com/user/AFP`

- **Video-only (VD).** We train a model that does not receive any background knowledge and generates the description directly from the video.
- **Video with the knowledge gate (VD+KG).** This model only sees the video and the knowledge gate encoding of patterns of knowledge elements found in the related articles.
- **Video with the entity pointer network (VD+EP).** We train a model that can describe the video and copy relevant entities into the description, but is not conditioned on the events and entity types in the related documents.
- **No video (EP+KG).** Similar to the article-only model, we experiment with a model that just generates descriptions with the knowledge gate and entity pointer network to examine the impact of the video in isolation.
- **KaVD.** Our full model is comprised of all of the components (VD+EP+KG). Under this full setting, we can observe the performance of the entire ensemble.

Each model uses a cross entropy loss for training [45, 47] (see Chapter 2). Video-based models are trained using the Adam optimizer [139] with a learning rate of 0.0002 and have a hidden state size of 512 as well as an embedding size of 300. We use Google News pre-trained word embeddings [65] to initialize our word embeddings and compute entity embeddings. For visual features, we use the Conv3-512 layer response of VGGNet [170] pre-trained on ImageNet [52].

We also experiment with a slightly modified version of our KaVD model, called KaVD*, which uses a stronger visual feature backbone to obtain frame representations and simplifies the video encoder. Specifically, we utilize the very deep ResNet-101 [71] model, which is also pre-trained on ImageNet [52], and we replace the attentive pooling with mean pooling. We extract features from the Conv5_3 layer. We find that these modifications improve the input representations of the video and thereby improve performance overall.

6.4.2 Evaluation Metrics

We use METEOR [39] and ROUGE-L [40] as metrics for evaluating the generated descriptions. METEOR accounts for stemming and synonym matching, which is well-suited for our scenario since we only have one reference per video. We also use ROUGE-L for comparison to summarization work and for its longest common sequence comparison, which can capture named entities with multiple words as well as event mentions. These capture the coherence and relevance of the generated descriptions. Other metrics that can be used to evaluate generated descriptions are BLEU [38], CIDEr [41], and SPICE [42]. However, we do not employ these metrics for several reasons: BLEU

does not account for recall, meaning that it does not reflect the coverage of the content in the generated description relative to the ground truth, which is important for knowledge-aware generation as coverage here indirectly indicates the factual accuracy of the generated description. CIDEr may not be very robust in this scenario since the ground truth descriptions are quite linguistically and topically diverse and the dataset is of moderate size, so the global word frequency statistics that CIDEr uses may be less reflective of the important information in the descriptions. Lastly, the semantic parsing required by the SPICE metric is ill-suited for the knowledge-centric descriptions in our task since it is designed for scene graphs with common objects/attributes, like those in Visual Genome [24], and not the complex entities, relations, and events found in our setting.

Generating these descriptions is concerned with not only generating fluent text, but also the amount of knowledge conveyed and the accuracy of the knowledge elements (*e.g.*, named entities or event structures). Previous work in natural language generation and summarization [175, 176, 177, 178] scores and/or assigns weights to overlapping text, salient phrases, or information units (*e.g.*, entity relations [177]). However, knowledge elements cannot be simply represented as a set of isolated information units since they are inherently interconnected through some structure.

Therefore, for this knowledge-centric generation task, we compute F1 scores on event and entity extraction results from the generated descriptions against the extraction results on the ground truth. For entities, we measure the F1 score of the named entities in the generated description compared to the ground truth. For events, given a generated description, w^s, and the ground truth description, w^c, we extract a set of event structures, $\mathcal{Y}^s$ and $\mathcal{Y}^c$, for both descriptions such that $\mathcal{Y} = \{(t_k, r_{k,1}, a_{k,1}, ..., r_{k,m}, a_{k,m})\}_{k=1}^{K}$ where there are K events extracted from the description, t_k is the kth event type, $r_{k,m}$ is the mth argument role of t_k, and $a_{k,m}$ is the mth argument of t_k. For the description in Figure 6.2a, one may obtain:

$$\begin{aligned}\mathcal{Y} = \{(&\texttt{Demonstrate},\\ &\texttt{Entity}, \text{“}\textit{Pro-independence supporters}\text{”},\\ &\texttt{Place}, \text{“}\textit{Barcelona}\text{”})\}.\end{aligned} \tag{6.15}$$

Next, we form event type, argument role, and argument triples $(t_k^s, r_{k,m}^s, a_{k,m}^s)$ and $(t_j^c, r_{j,m}^c, a_{j,m}^c)$ for each event structure in $\mathcal{Y}^s$ and $\mathcal{Y}^c$, respectively. We compute the F1 score of the triples, considering a triple correct if and only if it appears in the ground truth triples.[4] This metric enables us to evaluate how well a generated description captures the overall events, while still giving credit to partially correct event structures. We compute these F1 scores on 50 descriptions based on manually annotated event structures. We also perform automatic F1 score evaluation on the entire

[4]This criterion is used for computing precision and recall.

Model	METEOR	ROUGE-L	Entity F1	A-Entity F1	Event F1	A-Event F1
Article-only	8.6	13.2	8.7	8.5	1.9	3.6
VD	9.1	17.9	2.5	1.5	1.0	7.3
VD+EP	9.7	18.1	15.3	13.6	5.7	7.0
VD+KG	9.8	18.5	10.2	10.7	6.7	8.3
EP+KG	10.1	18.7	23.7	20.9	2.2	9.9
KaVD	**10.2**	18.9	22.1	19.7	9.6	8.9
KaVD*	**10.2**	**19.6**	**24.4**	**21.1**	**14.2**	**10.3**

Table 6.2: METEOR, ROUGE-L, and manual/automated entity (Entity F1/A-Entity F1) and event (Event F1/A-Event F1) F1 scores of the baselines and KaVD network on our dataset.

test set using the entity [164] and event [165] extraction systems of and, respectively. The manual evaluations offer accurate comparisons and control for correctness, while the automated evaluations explore the viability of using automated IE tools to measure performance, which is desirable for scaling to larger datasets for which manual evaluations are too expensive.

Our knowledge element metrics can be thought of as being somewhat similar to SPICE in that they measure the agreement of semantic meaning between descriptions, where this similarity is measured by the overlap of elements in scene graphs for SPICE or knowledge graphs for our metrics. However, in addition to handling different kinds of semantics (*i.e.,* scene graphs vs. knowledge graphs), our metrics differ from SPICE in that we do not require a specific set of entity types and entities or events can be non-visual (*e.g.,* an organization or "`Nominate`"). This makes our metrics better suited to this knowledge-centric evaluation. However, there are relevant limitations to these metrics that should be explored in future work. In particular, state-of-the-art IE tools are reliant on a target ontology that defines what entities, relations, and events are to be extracted. Consequently, these metrics may cover salient yet not fully comprehensive set of knowledge elements. While the coverage for the purposes of our dataset is sufficient as the domains of the videos and IE tools match, moving forward, utilizing OpenIE [179] or other general purpose knowledge extraction tools would be very useful for improving the generality of our metrics.

6.4.3 Results and Analysis

The KaVD network outperforms almost all of the baselines, as shown in Table 5.3, achieving statistically significant improvements in METEOR and ROUGE-L w.r.t. all other models besides the no-video model ($p < 0.05$).[5] The additions of the entity pointer network and knowledge gate are complementary and greatly improve the entity incorporation performance, increasing the en-

[5]Computed via paired bootstrap resampling [180].

tity F1 scores by at least 6% in both the manual and automatic evaluations. In Figure 6.5a, the entity pointer network is able to incorporate the entity "*Abdiaziz Abu Musab*", who is a leader of the group responsible for the attack. We find that the entity and event type features from the knowledge gate help generate more precise entities. However, noise in the article retrieval process and entity extraction system limits our entity incorporation capabilities, since on average 70.6% of the entities in the ground truth description are retrieved from the articles. This is significant coverage, but still leaves room for missed entities. The video encoder helps generate the correct events and offers qualitative benefits, such as allowing the model to generate more concise and diverse descriptions, though it negatively affects the entity incorporation performance. Enhancing the visual representations and simplifying the spatially attentive pooling (*i.e.,* KaVD*) offers further performance benefits as the model is able to optimize a slightly less complex function, while also using richer input features.

The video alone is insufficient to generate the correct entities (Table 6.2). In Figure 6.5a, the VD baseline generates the correct event, but generates the incorrect location "*Kabul*". We observe that when the visual evidence is ambiguous, this model may fail to generate the correct events and entities. For example, if a video depicts the destruction of buildings after a hurricane, then the VD baseline may mistakenly describe the video as an explosion since the visual evidence is similar. These are issues that the knowledge elements from related documents can help disambiguate since information about the events surrounding the video offers insight into what caused the evidence in the video (*e.g.,* if the video shows destruction, but no conflict event occurred, then the destruction may be more likely due to a natural disaster).

The article-only baseline tends to mention the correct entities as shown in Figure 6.5a, where the description is generally on topic but provides some irrelevant information. Indeed, this model can generate descriptions unrelated to the video itself. In Figure 6.5b, the article-only baseline's description contains some correct entities (*e.g.,* "*Colombia*"), but is not focused on the announcement depicted in the video. As has been discussed in prior work [107], this model can be more extractive than abstractive, copying many sequences from the documents. This can lead to irrelevant descriptions as the articles may not be specific to the video.

Our entity and event F1 score based metrics correlate well with the correctness of the knowledge conveyed in the generated description. The consistency in model rankings between the manual and automatic entity metrics shows the potential of using automated entity extraction approaches to evaluate with this metric. We observe discrepancies between the manual and automatic event metrics, in part, due to errors in the automated extraction and the addition of more test points. For example, in the generated sentence, "*Hundreds of people are to take to the streets of...*", the event extraction system mistakenly assigns a "`Transport`" event type instead of the correct "`Demonstrate`" event type. In contrast, such mistakes do not appear in the manual evaluations.

a) 

Title: Deadly Shabaab attack rocks Somali capital

b) 

Title: Santos: 'Green light' for referendum on Colombia peace deal

Model	Description
Article-only	somali capital mogadishu on saturday. at least 276 people have died and the government news agency sonna says only 111 of them have been identified. a turkish military but instead witnessed her burial. no group has yet said it was behind on instead he attended her burial. "anfa'a said she had spoken to her sister 20 minutes before on
VD	a suicide bomber killed # people in a bus carrying # people killed in a bus in central kabul.
VD+EP	A suicide bomber killed # people were killed in a bus near the northern city of Mogadishu, police said.
VD+KG	At least # people were killed and # wounded when a busy bus station in Kabul, killing at least # people dead and others who died in the rubble of the deadliest attack in the country.
EP+KG	At least # people were killed in a suicide car bomb attack on a suicide car bomb attack on a police vehicle in Mogadishu, police said.
KaVD	A suicide bombing claimed by the Abdiaziz Abu Musab group `time` killed # people in Somalia's capital Mogadishu, killing # people, officials said.

Model	Description
Article-only	colombia's marxist rebels against her family. and last year, when given the leg of helena gonzález's nephew years ago is still fresh the as pope francis arrived in colombia on wednesday for a six-day the
VD	president donald trump says that he will be talks to be to be talks to be talks in the country's country to be talks, saying he says he would be no evidence's state and kerry says.
VD+EP	President Maduro says the FARC president warns that the ceasefire to Prime Minister says that he will be ready to help President Maduro says that he is no evidence of President Bashar talks in Bogota.
VD+KG	US Secretary of State John Kerry, who will not any maintain in Syria, after a ceasefire in Syria, saying that the United Nations says, it will not to be into a speech in its interview.
EP+KG	Venezuela's President FARC envoy to Colombia is a definitive ceasefire in the FARC conflict, with FARC rebels, the FARC rebels.
KaVD	Colombia's government, signed the peace agreement with the FARC peace accord in the FARC rebels.

Figure 6.5: Comparison of generated descriptions. The KaVD network generates the correct entities and correct events, while other models may contain some wrong entities or wrong events.

Figure 6.6 shows another example output along with the ground truth description and entries from the related articles. Here, the full model generates a rather coherent description that generally aligns with the ground truth description: the entities mentioned are correct and the events discussed are approximately accurate. In comparison to the no video model, EP+KG we see that the description does not mention as many erroneous entities. This may be due to the influence of the visual inputs which show "*Iraqi forces*" as well as "*Mosul*" and provide an association between these two entities since they are involved in the same events. Meanwhile, incorrect entities like the "*Tigris River*" are not shown in the video, so the model has a signal to filter such entities out.

The example in Figure 6.6 underscores the importance of evaluating knowledge elements. When

Ground Truth	Iraqi forces enter western Mosul neighborhoods, a key stronghold in the shrinking "caliphate" of the Islamic State group, which replied with deadly suicide attacks in Iraq and Syria.
EP+KG	Iraqi forces have retaken the Rapid Response Division who have taken into the Tigris area in Baghdad, the latest of the Islamic State group.
KaVD*	Iraqi forces continue to retake the city of Mosul, the latest of remaining offensive in the city of Mosul.
Related Documents	The battle for western Mosul is likely to be tougher than that for the eastern half of the city... ... with escape routes across the desert and toward Syria blocked off... The thunderous booms from howitzers near Hamam al-Alil, a town along the Tigris River... ... the Governor of Salahudin province appealed for help from Baghdad after a spike in attacks... Hamam al-Alil was taken by Iraqi forces in November... The Islamic State's military tactics have also added to the challenge. Nowhere is that more true than in Mosul, Iraq's second largest city. ... commander of the rapid response unit of the Federal Police.

Figure 6.6: Output examples, the ground truth description, and sentences from related documents.

generating these descriptions, it is important to generate factually accurate outputs and these evaluations provide a means to probe for this information. In future work, it is important to improve automated event and entity extraction methods so that these can be reliably used for evaluation purposes. Moreover, expanding these systems to handle a wider and more fine-grained set of event and entity types would help evaluate the veracity of the generated descriptions even further.

6.5 SUMMARY

We collect a news video captioning dataset with knowledge-rich descriptions and present an approach to this task that uses a novel Knowledge-aware Video Description network, which utilizes the video and background knowledge mined from topically related documents. Our approach is able to learn associations between concepts found in the background knowledge and the visual content. We present new metrics that measure a model's ability to incorporate named entities and specific events into the descriptions. We show the effectiveness of our approach and set a new benchmark for this dataset.

CHAPTER 7: CONCLUSIONS AND FUTURE DIRECTIONS

7.1 CONCLUSIONS

Vision and Language is an important area that studies fundamental questions about how machines may gain, refine, and utilize an understanding of the physical world as well as communicate this understanding. At the heart of many of these questions is the acquisition and organization of concepts that are manifested in the environment. In this dissertation, we have explored this important area through improving vision and language models, particularly VQA and video captioning models, in three aspects:

- **Linking instances of concepts across modalities.** The ability to ground concept mentions to image regions is important for many vision and language tasks, and is often an expected byproduct of training on vision and language tasks. However, in VQA, models tend to rely on contextual cues or learned priors instead of actually recognizing and linking concepts across modalities. Consequently, when a concept appears in a new context, models often fail to adapt.

- **Consistency towards different mentions of the same concept.** Instances of a concept can take many different forms, such as the appearance of a concept in different images or the use of synonyms in text, and it can be difficult for models to infer these relationships from the training data alone. In VQA models, this, along with different phrasings of questions, can lead to inconsistencies in predictions and an overall degradation in performance.

- **Modeling associations between related concepts in complex domains.** In scenarios where multiple related sources of information (*e.g.,* visual inputs and external knowledge) need to be considered, models must be able to connect concepts found within and across these different sources. Existing models require further extension in order to handle these different input sources and use them effectively. We explore this in the context of incorporating structured knowledge in video captioning.

In order to address these issues, at the core of this dissertation, we propose three principles that guide our development of solutions for vision and language models:

- **Leveraging structured knowledge.** In many instances, it may be difficult for vision and language models to learn concepts, their different appearances/mentions, and their relationships through the training data alone. We take a step further by bolstering models ability to learn associations between concepts as well as their mentions using structured knowledge

to augment training as well as perform knowledge-aware tasks. We discuss this principle in Chapter 5 and Chapter 6.

- **Utilizing relationships amongst examples.** Vision and language models largely ignore the interconnections between related examples (*e.g.,* sharing the same concepts, having the same appearance or meaning) and handle each example independently during training. This means that relationships amongst examples must be learned implicitly. Encoding the relationships through regularization and joint objectives can improve generalization, robustness, and compositionality of vision and language models. This is elaborated in Chapter 4 and Chapter 5.

- **Compositionality.** For a wide range of problems, like VQA or knowledge-aware tasks, there are two important processes involved: 1) recognizing and representing concepts; 2) performing task-specific operations using the concepts or other information, such as answering a question or generating a caption based on the concepts in an image or video. The ability to seamlessly compose concept representations with the task-specific capabilities can allow models to better generalize to new domains or instances. We employ and expound this in Chapter 4 and Chapter 6.

Using these principles, we explore two important vision and language tasks: VQA and video captioning. This work proposes a number of models, approaches, and problems:

- We present a self-supervised, contrastive learning approach to learn to ground concept mentions jointly alongside VQA, without the need for annotations or data external to the task. We further propose a new view and evaluation setting of compositionality in VQA that examines models ability to learn concepts and compose them with the skills needed to answer questions. (Chapter 4, [181])

- The VQA P2 benchmark we put forth provides a means for evaluating VQA model's consistency towards different concept mentions and phrasings. Further, we introduce a novel regularization framework (Q3R) for modular architectures that improves model consistency and overall performance. (Chapter 5, [99])

- Knowledge-aware video captioning is the first of its kind and represents an important step towards the use of captioning models for real world data and applications. We present a Knowledge-aware Video Description network, which utilizes weakly aligned knowledge elements to generate descriptions of news videos. Our model is able to learn associations between the visual content and patterns of entities as well as events. (Chapter 6, [7])

7.2 BROADER APPLICATIONS

The principles, ideas, and approaches laid out in this dissertation are also generalizable to a range of problems and domains. In this section, we present some discussion of other problems, tasks, or domains to which the work in this dissertation can be applied.

7.2.1 Separating Skills and Concepts

The idea of disentangling skills and concepts from Chapter 4 can be employed in a number of different settings. While the specific implementations of skills may differ based on the input modalities or the domain, models doing any type of question answering can potentially benefit from this idea. For example, in reading comprehension [182], models could be trained to first recognize the content (concepts) in the question that should be used to filter for the correct information in the accompanying passage, and then apply the skills in order determine the exact answer span. Our knowledge-aware captioning work in Chapter 6 also shares a similar separation, where entities are regarded as being separate from the text generation skills of the decoder. Likewise, in vision-and-language navigation, models should be able to compose the step-by-step decision making processes with the concepts present in the surrounding environment. Therefore, models may be able to employ similar losses to our skill matching and concept grounding in order to encourage better generalization to new scenes/environments or unseen compositions of actions and concepts. Lastly, having compositional models that can separate skills and concepts can help VQA models expand to knowledge-aware visual question answering [183] or the news domain more generally. When operating in these complex domains, it becomes important to not only consider the visual content, but also background knowledge, as shown in Chapter 6. In this setting, where the background knowledge may be well-aligned and represented as a graph, the processes of reasoning over the graph and visual content can be seen as the set of skills. Inducing a separation between the skills needed to answer the questions, such as identifying named entities or determining occupations, and the representations of concepts present in the image and/or background knowledge, such as people or relations, could make models more effective in this setting because the entities and graph structures found in the inputs may be quite diverse, making compositionality even more important. This touches on building more compositional models, in general, which is also elaborated for future work in Section 7.4.1.

Beyond learning a similar disentanglement for other problems, the use of novel compositions for evaluations is important and can also be generalized. For example, in event extraction [184], there are useful constraints or inductive biases that are desirable for the model to learn [165, 185], such as the victim of an "`Injure`" event must be a "`Person`" entity. However, it can be important for

these models to be compositional as well, such as learning that the attacking agent of an "`Injure`" event can be a "`Person`", "`Organization`", or "`Geo-political Entity`" and being able to compose "`Injure`" events with any of these agent types. Therefore, it may be advantageous to create evaluations for event extraction systems that probe their ability to compose events with arguments of different types.

7.2.2 Controllable Perturbed Data for Diagnostics

Utilizing perturbed data as a diagnostic tool, like in Chapter 5, is an important idea that can be widely applied. For example, generating relevance preserving perturbed data has been utilized in Information Retrieval (IR) as a way to probe how well retrieval models implement different heuristics [186]. In the context of IR, perturbations, such as document scaling [186] where a given document is concatenated with itself multiple times, can provide a means for exploring how sensitive a given retrieval model is to document length. Similarly, for object recognition [52], ObjectNet [187] varies the backgrounds and viewpoints of objects in the images to demonstrate that existing models can have difficulties recognizing objects under these different controls. Overall, controlled perturbations or variations in data can provide a useful diagnostic tool with which model performance can be probed at a fine-grained level and offer insights that may not easily gleaned from existing large-scale benchmarks.

7.2.3 Representing and Incorporating External Knowledge

External knowledge can be an useful signal for models to leverage during training and inference by providing rich, structured context for models, as in knowledge-aware video captioning in Chapter 6. For instance, in healthcare, one may want to utilize a captioning system to create medical image reports [3, 4], which requires recognition of concepts in scans or images as well as associating these concepts with knowledge elements from external knowledge or health records. Another example is in textual entailment where external knowledge can be utilized to provide extra context beyond the local textual context [188].

Given the utility of external knowledge, the question naturally arises of how to best represent this knowledge so that it can be effectively incorporated into state-of-the-art, neural models. There are two essential pieces to this: 1) How to represent/encode the knowledge? 2) How to use this knowledge in the model? One way of representing knowledge this could be through simple encodings, like the knowledge gate vector in Chapter 6. A promising and more general purpose method is to utilize graph neural networks [189, 190], which can encode the structure of the knowledge into the resulting representations. These representations can then be given as inputs to a model.

Existing neural models are largely not adapted to handling external knowledge. However, some architectures, like Memory Networks [191, 192], contain mechanisms for handling these different sources of inputs. When designing neural approaches that take external knowledge as input, there are roughly three ways of utilizing the knowledge. First, as additional features to be used for the final prediction [188]. Second, as a signal to influence the intermediate computations or representations [7]. Third, as a means to enhance the input representations of concepts [193]. Each of the aforementioned methods can be used depending upon several factors, including the alignment of the knowledge with the inputs as well as the specific downstream task.

Using knowledge as a means to enhance input representations can be an easier way to inject external knowledge into downstream models. In this direction, a possible area of exploration may be joint knowledge and multimodal data representation learning. This would entail learning multimodal representations that encode the distributional information of images and text as well as the structural information from external knowledge. Such representations could be useful for a variety of downstream tasks, such as information extraction [74, 165, 194] or knowledge-aware VQA [183]. One potential means of doing this could be to use transformer layers along with clever masking [195] and contrastive learning [193] as an inductive bias for the model to learn to encode both the context from each data modality as well as the structured context from the external knowledge. Overall, this is a potentially interesting direction that warrants further exploration.

7.3 LIMITATIONS

While the aforementioned advances are important steps forward, there are still limitations of our approaches to be improved upon. We outline some of these limitations in this section.

7.3.1 Model and Supervision for Learning Grounding

When learning concept grounding using contrastive learning in Chapter 4, we demonstrate that the construction of the reference sets plays a role in the performance of the approach. Although our CCC references do improve the quality of the reference sets, certain ambiguities are difficult to overcome. In particular, concepts that very frequently co-occur in the training data may require extra data or further filtering to construct reference sets that can separate these concepts. For example, in the VQA data, "*shirt*" nearly always co-occurs with "*person*" (or other sub-concepts of "*person*"), so identifying contrasting examples where these two concepts are separate can be challenging without involving more data. This source of noise can be difficult to filter out, regardless of the specific approach or task. Improvements like our CCC reference set formulation or other negative example mining techniques are moving in the right direction, but there is more to be done

to refine these methods as well as show that they can be effective regardless of the domain. Furthermore, questions around whether or not the transformer architecture is the optimal model for learning ground remain, as discussed in Chapter 4.

7.3.2 Expanding to the Open Domain

A significant portion of vision and language efforts center around datasets that each cover specific sets of concepts. While these are important for advancing the field, as approaches continue to perform better on many of these datasets, a need for larger and more open domain datasets grows. Moving towards using datasets with more expansive ontologies and larger quantities of data offering weak supervision is important for improving the utility as well as portability of vision and language models. In our work, we are not necessarily limited to a specific ontology, meaning our methods could potentially be easily ported to a wider range of data. However, as we demonstrate in Chapter 4 and Chapter 5, there are variations in performance depending upon what concepts appear or how they appear. Expanding the coverage of the datasets used to train models and measure their generalization and robustness would help better improve and evaluate the progress made on vision and language models.

7.3.3 Veracity in Knowledge-aware Tasks

When performing knowledge-aware tasks, such as knowledge-aware captioning in Chapter 6, the output being factually accurate is paramount as the spread of misinformation is an ever-growing issue around the world [196, 197]. So while there is clear utility in having technology that can describe and summarize images or video in natural language, ensuring that the generated text is factually correct is a challenging problem that we must solve before something like this is to be deployed. For example, in Figure 6.6, the full model misses the attacks mentioned in the ground truth, which are important for the description. The knowledge-centric metrics that we propose in Chapter 6 are a useful step and can capture incorrect or missing elements, but they are dependent upon the ontologies used for the extraction components as well as the performance of these components. Indeed, there are some event structures that models may simply miss due to them not being represented in the target ontology. Moreover, there are other fine-grained details that knowledge-aware models may generate that are inaccurate but are not specifically entities or events (*e.g.*, exaggerating narratives or introducing false relations between entities). Overall, improving and guaranteeing the veracity of the output is imperative for knowledge-aware tasks.

7.4 FUTURE DIRECTIONS

7.4.1 Compositionality

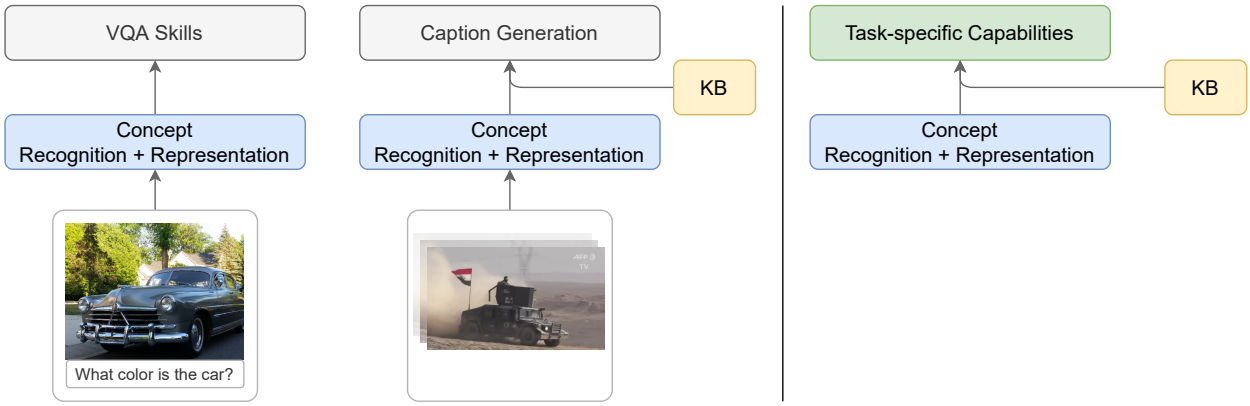


Figure 7.1: Overview of compositionality as part of our approaches (left) and more generally (right). KB means knowledge base, which serves as a source of external knowledge relevant to the concepts.

When training a model on a downstream task, the process of learning and representing concepts is inherently intertwined with the objective and data of the task. For example, as we show in Chapter 4, many model's representations of concepts are dependent upon the skills required to answer the question as well as the question context. This dependence is a significant obstacle as it makes models less able to transfer to new domains or tasks. Furthermore, the need for compositionality goes beyond vision and language models. For instance, in relation extraction, where the goal is to predict a relationships between entities found in text, models must be able to predict the correct relation between any two entities and certain relation types should not be tied to specific entities [193].[1] This is similar for event extraction [184] in which models are expected to compose event argument roles with entities found in the sentence.

In the future, it would be greatly beneficial to further disentangle learning concepts and the capabilities of the downstream task (Figure 7.1). Essentially, models and their learned representations should be designed and trained to be more compositional. We, in part, demonstrate the potential of this in Chapter 4 and Chapter 6. In Chapter 4, we show how separating concept and skill representations can improve generalization. Likewise, in Chapter 6, we illustrate how a model can learn to compose captions that contain different entities. However, these are scratching the surface of a much deeper area of study.

There are likely many ways to design models to exhibit compositionality. For example, the

[1] https://catalog.ldc.upenn.edu/LDC2006T06

explicit compositional design of architectures is one such method. This is in the direction of Neural Module Networks, as introduced in Chapter 5. Indeed, one could create models that have input encoders and then break the remainder of the model down into composable sub-tasks, where the sub-tasks can be defined at various levels of complexity, such as elementary operations or skills. Alternatively, regularization or other strategies to control the internal representations of the model could be another viable option, such as the approach elaborated in Chapter 4.

Central to compositionality is the separation of concept recognition as well as representation and the task-specific inference algorithm. This property can allow for models to adapt to unseen compositions within a specific task. Additionally, this can facilitate the transfer between domains or scenarios. For example, if one were to train a model for language generation, such as GPT-3 [198], as a general decoder that is concept-agnostic, then one may be able to compose this decoder with many different concept representations for different tasks, such as captioning. Another example, could be learning joint knowledge and multimodal data representations via pre-training as well as learning VQA skills on large-scale VQA data, and then composing these components to transfer the VQA model to a new domain, such as news. Overall, compositionality is a desirable property to build into our models that can enable better generalization and transfer to new examples and domains.

7.4.2 Improving Data Efficiency

Large-scale pre-training has been shown to improve downstream vision and langauge tasks [8, 9, 73, 76, 77, 199]. One of the motivations for this is to learn representations of concepts and their alignments in different modalities. We observe in Chapter 4 that utilizing contrastive learning and forcing the model to reconstruct concept mentions by pointing can help the model learn grounding. Furthermore, our approach as well as other grounding approaches [72, 75] are able to learn effectively on far less data.

A potentially fruitful direction to explore would be to build such grounding objectives into the pre-training tasks in order to improve the data efficiency of pre-training. This can be done, for example, through contrastive objectives that force the model to better distill the correspondences between concept across the modalities. Some efforts show the effectiveness of refining the negative and positive examples used in contrastive learning setups [75, 181]. Applying similar principles to pre-training may allow these models to be just as, if not more, effective with less data.

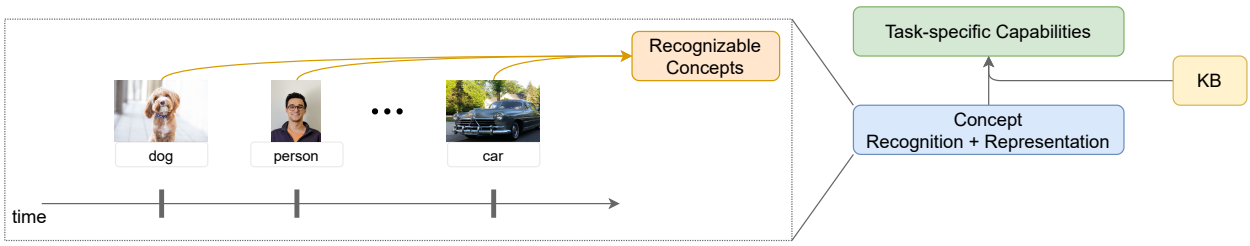


Figure 7.2: Illustration of compositionality with dynamic concept acquisition.

7.4.3 Dynamically Acquiring Concepts

Often times when models are trained for specific tasks, they are constrained by the concepts with which they are trained. Utilizing weak supervision from pairs of images and text can provide a means for expanding the concepts that models can use and/or reducing the need for image annotations. For example, one may use natural language supervision to adapt to different tasks [199, 200], such as object detection. Taking this further, one research direction would be to learn to acquire concepts progressively using natural language supervision, similar to language acquisition [201]. As the model acquires more concepts, one may be able to use the model to select training instances for downstream models or as a component of the approach to a downstream task. The ability to adapt to previously unseen concepts using natural language supervision could be used to make other technologies, such as object detectors, more portable to new scenarios by reducing task-specific label requirements. As shown in Figure 7.2, combining dynamic concept acquisition with compositionality could potentially lead to models that can handle an ever-expanding, diverse sets of concepts and compose these concepts with the learned task-specific capabilities, making generalization to new domains even more seamless.

7.5 CLOSING REMARKS

A prodigious amount of progress has been made in AI. It is truly remarkable that we have been able to make such great strides towards complex and challenging vision and language problems, like visual question answering and captioning. The work in this dissertation moves towards improving the ability of vision and language models to represent concepts in a unified fashion, meaning we enhance their ability to recognize and link concepts within and across modalities as well as learn associations between concepts. The state-of-the-art still has much to improve upon in order to demonstrate the proficient understanding of the physical world necessary to operate in unconstrained settings. As we progress, moving towards more compositional, transferable approaches will be key to increasing the utility and portability of vision and language models.

REFERENCES

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, "Vqa: Visual question answering," in *ICCV*, 2015.

[2] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. Mooney, T. Darrell, and K. Saenko, "Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition," in *ICCV*, 2013.

[3] B. Jing, P. Xie, and E. Xing, "On the automatic generation of medical imaging reports," in *ACL*, 2018.

[4] C. Y. Li, X. Liang, Z. Hu, and E. P. Xing, "Knowledge-driven encode, retrieve, paraphrase for medical image report generation," in *AAAI*, 2019.

[5] G. Murphy, *The big book of concepts*. MIT press, 2004.

[6] D. Lu, S. Whitehead, L. Huang, H. Ji, and S.-F. Chang, "Entity-aware image caption generation," in *EMNLP*, 2018.

[7] S. Whitehead, H. Ji, M. Bansal, S.-F. Chang, and C. Voss, "Incorporating background knowledge into video description generation," in *EMNLP*, 2018.

[8] X. Li, X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei et al., "Oscar: Object-semantics aligned pre-training for vision-language tasks," in *ECCV*, 2020.

[9] Y.-C. Chen, L. Li, L. Yu, A. E. Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu, "Uniter: Universal image-text representation learning," in *ECCV*, 2020.

[10] 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 *ECCV*, 2014.

[11] J. Xu, T. Mei, T. Yao, and Y. Rui, "Msr-vtt: A large video description dataset for bridging video and language," in *CVPR*, 2016.

[12] 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.

[13] A. Singh, V. Natarajan, M. Shah, Y. Jiang, X. Chen, D. Batra, D. Parikh, and M. Rohrbach, "Towards vqa models that can read," in *CVPR*, 2019.

[14] 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.

[15] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh, and D. Batra, "Visual Dialog," in *CVPR*, 2017.

[16] 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.

[17] J. Devlin, S. Gupta, R. Girshick, M. Mitchell, and C. L. Zitnick, "Exploring nearest neighbor approaches for image captioning," *arXiv preprint arXiv:1505.04467*, 2015.

[18] B. Zhou, Y. Tian, S. Sukhbaatar, A. Szlam, and R. Fergus, "Simple baseline for visual question answering," *arXiv preprint arXiv:1512.02167*, 2015.

[19] A. Agrawal, D. Batra, and D. Parikh, "Analyzing the behavior of visual question answering models," in *EMNLP*, 2016.

[20] A. Jabri, A. Joulin, and L. Van Der Maaten, "Revisiting visual question answering baselines," in *ECCV*, 2016.

[21] P. Zhang, Y. Goyal, D. Summers-Stay, D. Batra, and D. Parikh, "Yin and yang: Balancing and answering binary visual questions," in *CVPR*, 2016.

[22] K. Kafle and C. Kanan, "Visual question answering: Datasets, algorithms, and future challenges," *Computer Vision and Image Understanding*, vol. 163, pp. 3–20, 2017.

[23] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick, "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning," in *CVPR*, 2017.

[24] 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," *International Journal of Computer Vision*, vol. 123, no. 1, pp. 32–73, 2017.

[25] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, "Neural module networks," in *CVPR*, 2016.

[26] R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko, "Learning to reason: End-to-end module networks for visual question answering," in *ICCV*, 2017.

[27] R. Hu, J. Andreas, T. Darrell, and K. Saenko, "Explainable neural computation via stack neural module networks," in *ECCV*, 2018.

[28] J. Shi, H. Zhang, and J. Li, "Explainable and explicit visual reasoning over scene graphs," in *CVPR*, 2019.

[29] Q. Wu, D. Teney, P. Wang, C. Shen, A. Dick, and A. van den Hengel, "Visual question answering: A survey of methods and datasets," *Computer Vision and Image Understanding*, vol. 163, pp. 21–40, 2017.

[30] D. Teney, P. Anderson, X. He, and A. Van Den Hengel, "Tips and tricks for visual question answering: Learnings from the 2017 challenge," in *CVPR*, 2018.

[31] D. L. Chen and W. B. Dolan, "Collecting highly parallel data for paraphrase evaluation," in *ACL*, 2011.

[32] B. Pan, H. Cai, D.-A. Huang, K.-H. Lee, A. Gaidon, E. Adeli, and J. C. Niebles, "Spatio-temporal graph for video captioning with knowledge distillation," in *CVPR*, 2020.

[33] Z. Zhang, Y. Shi, C. Yuan, B. Li, P. Wang, W. Hu, and Z.-J. Zha, "Object relational graph with teacher-recommended learning for video captioning," in *CVPR*, 2020.

[34] Q. Zheng, C. Wang, and D. Tao, "Syntax-aware action targeting for video captioning," in *CVPR*, 2020.

[35] M. Regneri, M. Rohrbach, D. Wetzel, S. Thater, B. Schiele, and M. Pinkal, "Grounding action descriptions in videos," *Transactions of the Association for Computational Linguistics*, vol. 1, pp. 25–36, 2013.

[36] M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele, "Translating video content to natural language descriptions," in *ICCV*, 2013.

[37] A. Rohrbach, M. Rohrbach, W. Qiu, A. Friedrich, M. Pinkal, and B. Schiele, "Coherent multi-sentence video description with variable level of detail," in *GCPR*, 2014.

[38] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *ACL*, 2002.

[39] M. Denkowski and A. Lavie, "Meteor universal: Language specific translation evaluation for any target language," in *WMT*, 2014.

[40] C.-Y. Lin, "Rouge: a package for automatic evaluation of summaries," in *Text summarization branches out: ACL workshop*, 2004.

[41] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *CVPR*, 2015.

[42] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "Spice: Semantic propositional image caption evaluation," in *ECCV*, 2016.

[43] C.-W. Liu, R. Lowe, I. 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," in *EMNLP*, 2016.

[44] 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," in *EMNLP*, 2014.

[45] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence – video to text," in *ICCV*, 2015.

[46] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, 1997.

[47] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *CVPR*, 2015.

[48] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR*, 2015.

[49] 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.

[50] A. Burns, R. Tan, K. Saenko, S. Sclaroff, and B. A. Plummer, "Language features matter: Effective language representations for vision-language tasks," in *ICCV*, 2019.

[51] H. Jiang, I. Misra, M. Rohrbach, E. Learned-Miller, and X. Chen, "In defense of grid features for visual question answering," in *CVPR*, 2020.

[52] 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.

[53] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev et al., "The kinetics human action video dataset," in *CVPR*, 2017.

[54] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *ICML*, 2015.

[55] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville, "Describing videos by exploiting temporal structure," in *ICCV*, 2015.

[56] X. Guo, H. Wu, Y. Cheng, S. Rennie, G. Tesauro, and R. Feris, "Dialog-based interactive image retrieval," in *NeurIPS*, 2018.

[57] A. Miech, J.-B. Alayrac, L. Smaira, I. Laptev, J. Sivic, and A. Zisserman, "End-to-end learning of visual representations from uncurated instructional videos," in *CVPR*, 2020.

[58] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NeurIPS*, 2015.

[59] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[60] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *ICCV*, 2015.

[61] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?" in *CVPR*, 2018.

[62] D. Ghadiyaram, D. Tran, and D. Mahajan, "Large-scale weakly-supervised pre-training for video action recognition," in *CVPR*, 2019.

[63] Z. S. Harris, "Distributional structure," *Word*, vol. 10, no. 2-3, pp. 146–162, 1954.

[64] K. Lund and C. Burgess, "Producing high-dimensional semantic spaces from lexical co-occurrence," *Behavior research methods, instruments, & computers*, vol. 28, no. 2, pp. 203–208, 1996.

[65] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Workshops at ICLR*, 2013.

[66] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *EMNLP*, 2014.

[67] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019.

[68] 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.

[69] 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.

[70] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *EMNLP*, 2015.

[71] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[72] H. Akbari, S. Karaman, S. Bhargava, B. Chen, C. Vondrick, and S.-F. Chang, "Multi-level multimodal common semantic space for image-phrase grounding," in *CVPR*, 2019.

[73] H. Tan and M. Bansal, "LXMERT: Learning cross-modality encoder representations from transformers," in *EMNLP*, 2019.

[74] M. Li, A. Zareian, Q. Zeng, S. Whitehead, D. Lu, H. Ji, and S.-F. Chang, "Cross-media structured common space for multimedia event extraction," in *ACL*, 2020.

[75] T. Gupta, A. Vahdat, G. Chechik, X. Yang, J. Kautz, and D. Hoiem, "Contrastive learning for weakly supervised phrase grounding," in *ECCV*, 2020.

[76] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," in *NeurIPS*, 2019.

[77] G. Li, N. Duan, Y. Fang, M. Gong, D. Jiang, and M. Zhou, "Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training." in *AAAI*, 2020.

[78] Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian, "Deep modular co-attention networks for visual question answering," in *CVPR*, 2019.

[79] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, "Translating videos to natural language using deep recurrent neural networks," in *NAACL*, 2015.

[80] R. Pasunuru and M. Bansal, "Multi-task video captioning with video and entailment generation," in *ACL*, 2017.

[81] J. Lu, J. Yang, D. Batra, and D. Parikh, "Neural baby talk," in *CVPR*, 2018.

[82] L. Zhou, Y. Zhou, J. J. Corso, R. Socher, and C. Xiong, "End-to-end dense video captioning with masked transformer," in *CVPR*, 2018.

[83] H. Agrawal, K. Desai, Y. Wang, X. Chen, R. Jain, M. Johnson, D. Batra, D. Parikh, S. Lee, and P. Anderson, "nocaps: novel object captioning at scale," in *ICCV*, 2019.

[84] P. Pan, Z. Xu, Y. Yang, F. Wu, and Y. Zhuang, "Hierarchical recurrent neural encoder for video representation with application to captioning," in *CVPR*, 2016.

[85] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu, "Video paragraph captioning using hierarchical recurrent neural networks," in *CVPR*, 2016.

[86] M. Zanfir, E. Marinoiu, and C. Sminchisescu, "Spatio-temporal attention models for grounded video captioning," in *ACCV*, 2016.

[87] Y. Yu, H. Ko, J. Choi, and G. Kim, "End-to-end concept word detection for video captioning, retrieval, and question answering," in *CVPR*, 2017.

[88] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 67–78, 2014.

[89] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. C. Niebles, "Dense-captioning events in videos," in *ICCV*, 2017.

[90] C. C. Park, B. Kim, and G. Kim, "Attend to you: Personalized image captioning with context sequence memory networks," in *CVPR*, 2017.

[91] S. Venugopalan, L. A. Hendricks, R. Mooney, and K. Saenko, "Improving lstm-based video description with linguistic knowledge mined from text," in *EMNLP*, 2016.

[92] X. Hu, X. Yin, K. Lin, L. Wang, L. Zhang, J. Gao, and Z. Liu, "Vivo: Surpassing human performance in novel object captioning with visual vocabulary pre-training," in *AAAI*, 2021.

[93] K. Tran, X. He, L. Zhang, J. Sun, C. Carapcea, C. Thrasher, C. Buehler, and C. Sienkiewicz, "Rich image captioning in the wild," in *CVPR Workshops*, 2016.

[94] A. F. Biten, L. Gomez, M. Rusinol, and D. Karatzas, "Good news, everyone! context driven entity-aware captioning for news images," in *CVPR*, 2019.

[95] A. Tran, A. Mathews, and L. Xie, "Transform and tell: Entity-aware news image captioning," in *CVPR*, 2020.

[96] A. Agrawal, D. Batra, D. Parikh, and A. Kembhavi, "Don't just assume; look and answer: Overcoming priors for visual question answering," in *CVPR*, 2018.

[97] M. Shah, X. Chen, M. Rohrbach, and D. Parikh, "Cycle-consistency for robust visual question answering," in *CVPR*, 2019.

[98] R. Tang, C. Ma, W. E. Zhang, Q. Wu, and X. Yang, "Semantic equivalent adversarial data augmentation for visual question answering," in *ECCV*, 2020.

[99] S. Whitehead, H. Wu, Y. R. Fung, H. Ji, R. Feris, and K. Saenko, "Learning from lexical perturbations for consistent visual question answering," *arXiv preprint arXiv:2011.13406*, 2020.

[100] D. A. Hudson and C. D. Manning, "Gqa: A new dataset for real-world visual reasoning and compositional question answering," in *CVPR*, 2019.

[101] D. A. Hudson and C. D. Manning, "Compositional attention networks for machine reasoning," in *ICLR*, 2018.

[102] D. A. Hudson and C. D. Manning, "Learning by abstraction: The neural state machine," in *NeurIPS*, 2019.

[103] Z. Yu, J. Yu, J. Fan, and D. Tao, "Multi-modal factorized bilinear pooling with co-attention learning for visual question answering," in *ICCV*, 2017.

[104] J.-H. Kim, J. Jun, and B.-T. Zhang, "Bilinear attention networks," in *NeurIPS*, 2018.

[105] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," in *CVPR*, 2016.

[106] T.-H. Wen, M. Gašić, N. Mrkšić, P.-H. Su, D. Vandyke, and S. Young, "Semantically conditioned lstm-based natural language generation for spoken dialogue systems," in *EMNLP*, 2015.

[107] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *ACL*, 2017.

[108] D. Mirkovic, L. Cavedon, M. Purver, F. Ratiu, T. Scheideck, F. Weng, Q. Zhang, and K. Xu, "Dialogue management using scripts and combined confidence scores," 2011, uS Patent 7,904,297.

[109] A. Oh and A. Rudnicky, "Stochastic language generation for spoken dialogue systems," in *ANLP-NAACL 2000 Workshop: Conversational Systems*, 2000.

[110] T.-H. Wen, M. Gašić, N. Mrkšić, L. M. Rojas-Barahona, P.-H. Su, D. Vandyke, and S. Young, "Multi-domain neural network language generation for spoken dialogue systems," in *NAACL*, 2016.

[111] V.-K. Tran and L.-M. Nguyen, "Natural language generation for spoken dialogue system using rnn encoder-decoder networks," in *CoNLL*, 2017.

[112] L. Chen, B. Lv, C. Wang, S. Zhu, B. Tan, and K. Yu, "Schema-guided multi-domain dialogue state tracking with graph attention neural networks," in *AAAI*, 2020.

[113] S. Gehrmann, Y. Deng, and A. M. Rush, "Bottom-up abstractive summarization," in *EMNLP*, 2018.

[114] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *NeurIPS*, 2015.

[115] J. Gu, Z. Lu, H. Li, and V. O. Li, "Incorporating copying mechanism in sequence-to-sequence learning," in *ACL*, 2016.

[116] Q. Wang, X. Pan, L. Huang, B. Zhang, Z. Jiang, H. Ji, and K. Knight, "Describing a knowledge base," in *INLG*, 2018.

[117] W. Lehnert, "Human and computational question answering," *Cognitive Science*, vol. 1, no. 1, pp. 47–73, 1977.

[118] J. Weston, A. Bordes, S. Chopra, A. M. Rush, B. van Merriënboer, A. Joulin, and T. Mikolov, "Towards ai-complete question answering: A set of prerequisite toy tasks," in *ICLR*, 2016.

[119] X. Zeng, Y. Wang, T.-Y. Chiu, N. Bhattacharya, and D. Gurari, "Vision skills needed to answer visual questions," *Proc. ACM Hum.-Comput. Interact.*, 2020.

[120] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, "Building machines that learn and think like people," *Behavioral and brain sciences*, 2017.

[121] S. Ramakrishnan, A. Agrawal, and S. Lee, "Overcoming language priors in visual question answering with adversarial regularization," in *NeurIPS*, 2018.

[122] R. R. Selvaraju, S. Lee, Y. Shen, H. Jin, S. Ghosh, L. Heck, D. Batra, and D. Parikh, "Taking a hint: Leveraging explanations to make vision and language models more grounded," in *ICCV*, 2019.

[123] J. Wu and R. Mooney, "Self-critical reasoning for robust visual question answering," in *NeurIPS*, 2019.

[124] L. A. Hendricks, J. Mellor, R. Schneider, J.-B. Alayrac, and A. Nematzadeh, "Decoupling the role of data, attention, and losses in multimodal transformers," *arXiv preprint arXiv:2102.00529*, 2021.

[125] A. Singh, V. Goswami, and D. Parikh, "Are we pretraining it right? digging deeper into visio-linguistic pretraining," *arXiv preprint arXiv:2004.08744*, 2020.

[126] M. Acharya, K. Kafle, and C. Kanan, "Tallyqa: Answering complex counting questions," in *AAAI*, 2019.

[127] K. Kafle and C. Kanan, "An analysis of visual question answering algorithms," in *ICCV*, 2017.

[128] D. Teney and A. v. d. Hengel, "Zero-shot visual question answering," *arXiv preprint arXiv:1611.05546*, 2016.

[129] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*, 2020.

[130] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[131] J. Dodge, A. Goyal, X. Han, A. Mensch, M. Mitchell, K. Stratos, K. Yamaguchi, Y. Choi, H. Daumé III, A. Berg et al., "Detecting visual text," in *NAACL*, 2012.

[132] J. Hessel, D. Mimno, and L. Lee, "Quantifying the visual concreteness of words and topics in multimodal datasets," in *NAACL*, 2018.

[133] G. Kehat and J. Pustejovsky, "Integrating vision and language datasets to measure word concreteness," in *IJCNLP*, 2017.

[134] M. Honnibal and I. Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," 2017, to appear.

[135] S. Ganju, O. Russakovsky, and A. Gupta, "What's in a question: Using visual questions as a form of supervision," in *CVPR*, 2017.

[136] D. Dong, H. Wu, W. He, D. Yu, and H. Wang, "Multi-task learning for multiple language translation," in *ACL*, 2015.

[137] M.-T. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser, "Multi-task sequence to sequence learning," in *ICLR*, 2016.

[138] K. Terao, T. Tamaki, B. Raytchev, K. Kaneda, and S. Satoh, "Rephrasing visual questions by specifying the entropy of the answer distribution," *arXiv preprint arXiv:2004.04963*, 2020.

[139] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[140] K. Gouthaman and A. Mittal, "Reducing language biases in visual question answering with visually-grounded question encoder," in *ECCV*, 2020.

[141] S. Uppal, A. Madan, S. Bhagat, Y. Yu, and R. R. Shah, "C3vqg: Category consistent cyclic visual question generation," *arXiv preprint arXiv:2005.07771*, 2020.

[142] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*, 2020.

[143] Y. Jiang, V. Natarajan, X. Chen, M. Rohrbach, D. Batra, and D. Parikh, "Pythia v0.1: the winning entry to the vqa challenge 2018," *arXiv preprint arXiv:1807.09956*, 2018.

[144] A. Agrawal, D. Batra, D. Parikh, and A. Kembhavi, "Don't just assume; look and answer: Overcoming priors for visual question answering," in *CVPR*, 2018.

[145] X. Xu, X. Chen, C. Liu, A. Rohrbach, T. Darrell, and D. Song, "Fooling vision and language models despite localization and attention mechanism," in *CVPR*, 2018.

[146] A. Ray, K. Sikka, A. Divakaran, S. Lee, and G. Burachas, "Sunny and dark outside?! improving answer consistency in VQA through entailed question generation," in *EMNLP*, 2019.

[147] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[148] E. Pavlick, P. Rastogi, J. Ganitkevitch, B. Van Durme, and C. Callison-Burch, "Ppdb 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification," in *ACL*, 2015.

[149] Y. Li, N. Duan, B. Zhou, X. Chu, W. Ouyang, X. Wang, and M. Zhou, "Visual question generation as dual task of visual question answering," in *CVPR*, 2018.

[150] F. Liu, T. Xiang, T. M. Hospedales, W. Yang, and C. Sun, "ivqa: Inverse visual question answering," in *CVPR*, 2018.

[151] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, and E. P. Xing, "Toward controlled generation of text," in *ICML*, 2017.

[152] L. Tan, "Pywsd: Python implementations of word sense disambiguation (wsd) technologies [software]," https://github.com/alvations/pywsd, 2014.

[153] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *ICLR*, 2014.

[154] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *ICLR*, 2015.

[155] T. Gokhale, P. Banerjee, C. Baral, and Y. Yang, "MUTANT: A training paradigm for out-of-distribution generalization in visual question answering," in *EMNLP*, 2020.

[156] S. Zhao, J. Cui, Y. Sheng, Y. Dong, X. Liang, E. I. Chang, and Y. Xu, "Large scale image completion via co-modulated generative adversarial networks," in *ICLR*, 2021.

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

[158] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in *ICLR*, 2017.

[159] J. Mao, C. Gan, P. Kohli, J. B. Tenenbaum, and J. Wu, "The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision," in *ICLR*, 2019.

[160] S. Subramanian, S. Singh, and M. Gardner, "Analyzing compositionality in visual question answering." in *ViGIL@ NeurIPS*, 2019.

[161] N. Ng, K. Yee, A. Baevski, M. Ott, M. Auli, and S. Edunov, "Facebook fair's wmt19 news translation task submission," *arXiv preprint arXiv:1907.06616*, 2019.

[162] K. Yi, J. Wu, C. Gan, A. Torralba, P. Kohli, and J. Tenenbaum, "Neural-symbolic vqa: Disentangling reasoning from vision and language understanding," in *NeurIPS*, 2018.

[163] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers," in *NeurIPS*, 2020.

[164] X. Pan, B. Zhang, J. May, J. Nothman, K. Knight, and H. Ji, "Cross-lingual name tagging and linking for 282 languages," in *ACL*, 2017.

[165] Q. Li, H. Ji, and L. Huang, "Joint event extraction via structured prediction with global features," in *ACL*, 2013.

[166] Z. Yang, D. Yang, C. Dyer, X. He, A. J. Smola, and E. H. Hovy, "Hierarchical attention networks for document classification," in *NAACL*, 2016.

[167] Y. Qin, D. Song, H. Cheng, W. Cheng, G. Jiang, and G. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," in *IJCAI*, 2017.

[168] Y. Miao and P. Blunsom, "Language as a latent variable: Discrete generative models for sentence compression," in *EMNLP*, 2016.

[169] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NeurIPS*, 2012.

[170] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2014.

[171] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015.

[172] 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.

[173] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning." in *AAAI*, 2017.

[174] Z. Tu, Z. Lu, Y. Liu, X. Liu, and H. Li, "Modeling coverage for neural machine translation," in *ACL*, 2016.

[175] A. Nenkova and R. Passonneau, "Evaluating content selection in summarization: The pyramid method," in *NAACL*, 2004.

[176] J. Novikova, O. Dušek, A. C. Curry, and V. Rieser, "Why we need new evaluation metrics for nlg," in *EMNLP*, 2017.

[177] S. Wiseman, S. M. Shieber, and A. M. Rush, "Challenges in data-to-document generation," in *EMNLP*, 2017.

[178] R. Pasunuru and M. Bansal, "Multi-reward reinforced summarization with saliency and entailment," in *NAACL*, 2018.

[179] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, "Open information extraction from the web," in *IJCAI*, 2007.

[180] P. Koehn, "Statistical significance tests for machine translation evaluation," in *EMNLP*, 2004.

[181] S. Whitehead, H. Wu, H. Ji, R. Feris, and K. Saenko, "Separating skills and concepts for novel visual question answering," in *CVPR*, 2021.

[182] P. Rajpurkar, R. Jia, and P. Liang, "Know what you don't know: Unanswerable questions for SQuAD," in *ACL*, 2018.

[183] N. Y. Sanket Shah, Anand Mishra and P. P. Talukdar, "Kvqa: Knowledge-aware visual question answering," in *AAAI*, 2019.

[184] Z. Song, A. Bies, S. Strassel, T. Riese, J. Mott, J. Ellis, J. Wright, S. Kulick, N. Ryant, and X. Ma, "From light to rich ERE: Annotation of entities, relations, and events," in *Proceedings of the The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, 2015.

[185] L. Huang, H. Ji, K. Cho, I. Dagan, S. Riedel, and C. Voss, "Zero-shot transfer learning for event extraction," in *ACL*, 2018.

[186] H. Fang, T. Tao, and C. Zhai, "Diagnostic evaluation of information retrieval models," *ACM Transactions on Information Systems (TOIS)*, 2011.

[187] A. Barbu, D. Mayo, J. Alverio, W. Luo, C. Wang, D. Gutfreund, J. Tenenbaum, and B. Katz, "Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models," in *NeurIPS*, 2019.

[188] P. Kapanipathi, V. Thost, S. S. Patel, S. Whitehead, I. Abdelaziz, A. Balakrishnan, M. Chang, K. Fadnis, C. Gunasekara, B. Makni et al., "Infusing knowledge into the textual entailment task using graph convolutional networks," in *AAAI*, 2020.

[189] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, 2008.

[190] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[191] J. Weston, S. Chopra, and A. Bordes, "Memory networks," in *ICLR*, 2014.

[192] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "End-to-end memory networks," in *NeurIPS*, 2015.

[193] Y. Qin, Y. Lin, R. Takanobu, Z. Liu, P. Li, H. Ji, M. Huang, M. Sun, and J. Zhou, "Erica: Improving entity and relation understanding for pre-trained language models via contrastive learning," *arXiv preprint arXiv:2012.15022*, 2020.

[194] Y. Lin, H. Ji, F. Huang, and L. Wu, "A joint neural model for information extraction with global features," in *ACL*, 2020.

[195] Y. Kant, D. Batra, P. Anderson, A. Schwing, D. Parikh, J. Lu, and H. Agrawal, "Spatially aware multimodal transformers for textvqa," in *ECCV*, 2020.

[196] R. Faris, H. Roberts, B. Etling, N. Bourassa, E. Zuckerman, and Y. Benkler, "Partisanship, propaganda, and disinformation: Online media and the 2016 us presidential election," *Berkman Klein Center Research Publication*, vol. 6, 2017.

[197] C. Wardle and H. Derakhshan, "Information disorder: Toward an interdisciplinary framework for research and policy making," *Council of Europe report*, vol. 27, pp. 1–107, 2017.

[198] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell et al., "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[199] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark et al., "Learning transferable visual models from natural language supervision," *arXiv preprint arXiv:2103.00020*, 2021.

[200] A. Zareian, K. D. Rosa, D. H. Hu, and S.-F. Chang, "Open-vocabulary object detection using captions," *arXiv preprint arXiv:2011.10678*, 2020.

[201] D. Surís, D. Epstein, H. Ji, S.-F. Chang, and C. Vondrick, "Learning to learn words from visual scenes," in *ECCV*, 2020.