© 2020 Tanmay Gupta

REPRESENTATIONS FROM VISION AND LANGUAGE

BY

TANMAY GUPTA

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 at Urbana-Champaign, 2020

Urbana, Illinois

Doctoral Committee:

Professor Derek Hoiem, Chair
Professor Svetlana Lazebnik
Professor Alexander Schwing
Professor Abhinav Gupta

ABSTRACT

Replicating a human-level understanding of the physical world in computers is a monumental task. Achieving this requires building representations of concepts that manifest themselves visually, linguistically or through other senses. Furthermore concepts do not exist in isolation but are related to each other. In this work, we show how to build representations of concepts from visual and textual data, link visual manifestations of concepts to references in text descriptions (a problem known as word or phrase grounding) without strong supervision, and model the interaction between concepts. Specifically, we address the following three challenges faced by existing vision-language models:

The *first* challenge is that of building generalizable and accurate representations of images and words. For generalization across tasks, we build aligned image-word representations that can be shared across multiple tasks like visual recognition and visual question answering and enhance inductive transfer between them. We also augment text-only word embeddings with word embeddings learned from visual co-occurrences to provide more accurate representations of visual concepts.

The *second* challenge is linking references to visual concepts in textual descriptions to the corresponding regions in the image without requiring strong supervision in the form of word-region grounding. We show that maximizing a lower bound on mutual information between image regions and captions leads to state-of-the-art phrase grounding performance.

The *third* challenge is extending vision-language systems to model interactions between visual entities. We build systems that demonstrate this ability in both generation and detection settings. We show how to generate a plausible layout and appearance of entities given a text description of entity actions and interactions. We also develop a state-of-the-art factored model and training techniques for detecting human-object interactions using pretrained object and pose detectors.

To my parents, mentors, teachers, and friends for lessons in living and thinking.

ACKNOWLEDGMENTS

I have admired every member of my thesis committee for their contributions to the field of computer vision and for establishing high standards for the quality of AI research. However, my advisor Associate Professor Derek Hoiem and collaborator Assistant Professor Alex Schwing deserve a special mention. I have deeply valued Derek's timely and astute feedback on technical details, communication of research findings, and the process of scientific enquiry itself. I am thankful to Derek for providing me with immense flexibility over the direction of my thesis research. I am also grateful to Alex whose hands-on approach and technical expertise proved crucial for the completion of 40% of this dissertation within an year of collaboration. So, I owe much to Alex for creating time for two hour long technical discussions on weekdays, weekends, and even holidays.

Parts of this thesis were completed through internship collaborations. I am thankful to Aniruddha Kembhavi, my internship mentor at the Allen Institute for Artificial Intelligence, for thoughtful and inclusive decision making over the course of the challenging Semantic Scene Generation project. I am also grateful to Arash Vahdat, my internship mentor at Nvidia, for an enjoyable and technically enriching collaboration experience. Both Aniruddha and Arash were mentors in the true sense of the word and were deeply invested in the success of the project and their intern. I am indebted to them for their technical and moral support.

I also owe much to my senior University of Illinois at Urbana-Champaign (UIUC) lab mates and collaborators Jason Rock, Kevin Shih, and Saurabh Singh for helping me begin graduate research and publish my first conference papers. My years at UIUC have been incredibly enriching and enjoyable thanks to many graduate students and friends, a few of which are: Zhizhong Li, Rajbir Kataria, Nitish Gupta, Gaurush Hiranandani, Aditya Deshpande, Arun Mallya, Daphne Tsatsoulis, Bryan Plummer, Chuhang Zou, Joe Degol, Tom Paine, Pooya Khorrami, Jyoti Aneja, Daniel McKee, Unnat Jain, Anand Bhattad, Mariya Vasileva, Aiyu Cui, Dominic Roberts, Ayush Jain, Shyam Upadhyay, Sumit Verma, and Ajit Vikram.

This acknowledgement would be incomplete without mentioning Professor Aditya Jagannatham and Professor Sharmishtha Mitra at Indian Institute of Technology Kanpur who laid the foundation for my graduate research journey. I was introduced to computer vision as a sophomore by Professor Jagannatham, and Professor Mitra taught graduate-level matrix theory, an integral part of machine learning and computer vision.

Finally, no words suffice to express my gratitude towards my parents for their unwavering love, encouragement, and support.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

In the last decade, computer vision and machine learning have made huge strides in supervised pattern recognition problems. Conceptually, the solution has been remarkably simple and general - neural networks trained on large amounts of labelled data using stochastic gradient descent. As an evidence of progress, object detection performance on the PASCAL VOC 2007 benchmark has more than doubled from 35 mAP achieved by Deformable Part Models to 79 mAP boasted by Faster-RCNN only half-way through the decade. Currently, carefully engineered object detectors and low-level vision models (e.g edge detectors, monocular depth estimators) are performant and robust enough to be part of safety-critical applications like self-driving cars.

With progress on these fundamental vision applications, new problems have emerged on the horizon that require going beyond detecting objects and attributes in images. Two images, both with a “dog” and a “man” may be drastically different. For instance, consider images described by captions “An old man walking a white dog on a beach” and “A tall man sitting on a couch with a brown dog on his lap”. Generating such descriptions for images or understanding the scenes depicted in the images to answer natural language questions like “What is the man doing on the couch?” requires an understanding of interactions between objects and how natural language may be used to refer to parts of the image in addition to object and attribute detection.

Furthermore, for a human-level understanding of visual and textual concepts, it is important to look beyond categorization for a representation of images and words that is rich enough to express relations between various concepts within and across modalities. For example, humans understand that “dog” relates to other concepts such as “pet”, “needy”, “cute”, “paws”, “tail”, “fluffy” *etc.* and that it would be absurd for a “man” to have a “paw” or be on a “leash”. Image-text embeddings have shown potential for providing such a representation. However, image-text embeddings are often learned in a task-specific manner and generalization across tasks needs further exploration.

In this work, we address questions about learning representations of concepts from images and natural language (text) data. This involves learning generalizable concept representations for objects and attributes, implicit and explicit modeling of interactions between objects, and learning to map textual references to visual manifestation of concepts in images without direct supervision. We will now discuss these 3 challenges in detail.

1.1 CHALLENGE 1: ACCURATE AND GENERALIZABLE REPRESENTATIONS

The right representation could greatly simplify inference and reduce data required to learn such inference for any task. With the popularity of end-to-end training in deep learning frameworks, it is common to learn representations from raw inputs that are tuned for a specific task using supervised learning on task data. This approach has worked well for simple image classification or detection problems where the inference is a simple linear classification layer operating on the image representation. However, for vision-language problems like VQA where a complex inference on the image and word representations needs to be learned in addition to the representations themselves, such an approach poses a challenge. The model could learn a representation that does not generalize but still achieve high training accuracy by overfitting through inference parameters. Compensating for this effect requires training on increasingly large datasets with increasing complexity of inference required.

Intuition also suggests that concepts such as objects, attributes, relationships or interactions are shared across vision-language tasks. Hence, it is reasonable to expect image and word representations learned from one task to generalize across other tasks, a property currently lacking in vision-language models that are trained end-to-end on a single task.

Another aspect to consider while investigating representations of concepts is the multi-modal nature of such representations. Human understanding of concepts such as "dog" draws from multiple sensory experiences such as seeing various dogs, hearing a dog's bark or growl, and even smelling or feeling the dog's fur. In contrast, popular approaches for learning representations of visual and textual concepts are through learning a visual classifier and modeling word co-occurrences in large text corpora respectively. There exists work on learning image-text embeddings. However, the goal of such approaches is to either learn a mapping across modalities or to jointly represent an input image and text such as a question to predict an answer rather than to construct representations of concepts.

Below we describe our work that addresses the representation challenge. Specifically, we investigate the use of aligned image and word representations that generalize across multiple vision-language tasks, and using images to improve word representations.

1.1.1 Aligned image-word representations that generalize across tasks

There are multiple tasks like visual question answering (VQA), visual recognition (VR), and image captioning that require image and word representations. Our work on learning shared and aligned image and word representations, discussed in Chapter 3, is one of the initial efforts in sharing representations across tasks like VQA and VR. The goal is to learn image and word representations and formulate inference for these tasks using these representations in a way that enhances inductive transfer across tasks. For instance, learning object and attribute recognition should lead to performance gains on VQA and vice-versa. The key insight is that all vision-language tasks share the following:

- **Concept of objects and attributes.** The word "dog" in VQA, VR or any other vision-language task refers to the same concept.
- **Word-region verification sub-task.** Every vision-language tasks needs to solve a common sub-task of verifying whether a word applies to an image-region.

Therefore, we formulate inference for both VR and VQA tasks using shared image-region and word representations with explicit word-region verification as an intermediate step. We use inner product between learned region and word embeddings as the verification mechanism. Thus our work has two key contributions: (i) we share word representations across tasks in addition to sharing visual representations; (ii) interpretations of these representations and the inner product operation remains consistent across tasks which allows consistent training signal for the shared vision-language representations (SVLR) during joint training across multiple vision-language tasks.

Our key result is that SVLR leads to greater inductive transfer from recognition to VQA than sharing image features in multitask learning. This directly leads to highly interpretable attention as demonstrated by high correlation with human attention and the ability to produce object and attribute labels for the selected relevant region.

1.1.2 Using images to improve word representations

Human understanding of concepts draws from a range of senses. For instance, humans integrate visual and other sensory experiences (touch, smell, and sound) of "dogs" along with textual knowledge to fully comprehend the concept of a "dog". In contrast, commonly used word representations like word2vec and GloVe are learned only from co-occurrences of words computed from large text corpora. Learning representations of words using only text has a few key limitations:

- Text often consists of interpretations of concepts or events rather than description of visual appearance. For instance, it is rare to come across a textual description of "dog" as an animal having 4 legs, 2 eyes, 1 tail *etc*. Such information, however, is readily accessible in images.

- Existing word embeddings are learned from a single co-occurrence type *i.e.* does a word occur in the neighborhood of another word. However, words may be related in more than one ways. For example, “apple” and “red” are related through an object-attribute relation whereas “table” and “chair” are related through context.

In Chapter 4, we propose to compute word representations from multiple types of visual co-occurrences extracted from annotated image datasets. We say two words co-occur visually if both words apply to the same image or image-region. We then extend GloVe’s log-bilinear model to learn word embeddings in a multitask fashion from four types of visual co-occurrences. Our approach not only learns a single word embedding for each word but also learns transformation functions that map that embedding to co-occurrence type-specific embedding spaces. We demonstrate qualitatively that modeling multiple co-occurrences provides a richer sense of word relatedness that text only embeddings.

Through unsupervised clustering, supervised partitioning, and zero-shot classification analysis we demonstrate that word embeddings from visual co-occurrences or ViCo complement the information available in text-only embeddings like GloVe. Evaluation on downstream word-only and vision-language tasks demonstrates superior performance to GloVe and random vectors. However, a key finding is that performance of random vectors comes surprisingly close to learned embeddings (GloVe or ViCo) on vision-language tasks. We hypothesize and present evidence that given enough data, vision-language models transform random vectors into useful task-specific embeddings but in data-starved scenarios random vectors perform significantly worse.

1.2 CHALLENGE 2: MAPPING TEXTUAL REFERENCES TO IMAGE-REGIONS WITHOUT STRONG SUPERVISION

Matching words in questions or captions to image regions is fundamental to all vision-language tasks like VQA, image captioning, referring expression comprehension, visual dialog *etc.* In many of these tasks, this matching or grounding is learned as an attention mechanism using only task supervision. In Chapter 5, we explore a mutual information based objectives for learning this word-region mapping from paired image-text data without direct grounding supervision.

For a concrete evaluation, we focus on the task of weakly supervised phrase grounding. Given paired image-caption data such as Flickr30K, the goal is to learn to map noun phrases to image regions. Images are represented using features extracted from a pretrained object detector, and caption-words are represented using contextualized features from a pretrained language model. We maximize the InfoNCE lower bound on mutual information between the set of region features from an image and caption-word representations. Specifically, we maximize compatibility between attention-weighted regions and words in the corresponding caption compared to non-corresponding pairs of images and captions. A key idea is to construct negative captions through word substitutions using a language model instead of randomly sampling negative captions from the training data. By training on either COCO-Captions or the much smaller Flickr30K train set (without grounding annotations), we achieve state-of-the-art performance on Flickr30K entities test set.

In future work, we plan to incorporate this additional objective while training VQA or captioning models to guide relevant visual information extraction from images.

1.3 CHALLENGE 3: MODELING INTERACTIONS BETWEEN OBJECTS

Visual scenes with the same objects could have vastly different interpretations. The reason is that the same object may be interacting with a different object in the two scenes (*e.g.* "man riding a bike" *vs.* "man riding a horse" where both scenes have a man, a bike, and a horse) or interacting with the same object but in a different way (*e.g.* "man walking horse" *vs.* "man riding horse").

To address this challenge, we consider both detection and generation settings. In the generation setting, we aim to generate a video from a text description of entities and their interactions. In the detection setting, our goal is to detect human-object interactions.

1.3.1 Modeling interactions in the generation setting

A natural language description of a scene can succinctly convey information about what are the entities (objects or people) in the scene, and what might be a likely spatial arrangement and appearance of those entities. Note that both spatial location and appearance of an entity depends on other entities, thus requiring joint modeling and understanding of entity interactions.

To study this problem, in Chapter 6, we introduce Semantic Scene Generation (SSG) - the task of generating scene videos with multiple entities given a rich natural language description. A major challenge in this task is jointly modeling the layout and appearance of mentioned entities. This in turn requires an understanding of how actions and interactions mentioned in the description affect entity layout and appearance. In addition, the task also requires world knowledge. For instance, cartoon videos set in the stone age assume a different world knowledge than those set in the future or real world street scenes.

Our approach sequentially adds entities in the scene by predicting the location and scale of the current entity, and retrieving a spatio-temporal entity segment given the layout and appearance of entities added to the scene thus far. Our key technical contributions include sequential training of components of the model while jointly modeling layout and appearances, and auxiliary losses that encourage learning compositional representations for retrieval. We also introduce a new richly annotated video-caption dataset of 25000, 3 second clips from the Flintstones animated series.

1.3.2 Modeling interactions in the detection setting

State-of-the-art vision-language models typically treat images as a bag of regions and have a limited ability to understand the concept of interactions such as "human-driving-car", "human-riding-horse", or "human-walking-horse". To equip the next generation of VQA or captioning models with the ability to understand interactions, in Chapter 7, we study the task of human-object interaction detection. Specifically, we evaluate the efficacy of pretrained object and pose detector outputs in representing and detecting interactions.

Recently, HOI detection literature has seen the use of increasingly sophisticated techniques for encoding appearance (e.g using multi-task learning and attention mechanisms) and layout (e.g. using mixture density networks or interaction patterns). In this work, we show that with an appropriate factorization, and encodings of layout and appearance constructed from outputs of pretrained object detectors, a relatively simple model outperforms more sophisticated approaches on human-object interaction detection. Our model includes factors for detection scores, human and object appearance, and coarse (box-pair configuration) and optionally fine-grained layout (human pose). We also develop training techniques that improve learning efficiency by: (i) eliminating train-inference mismatch; (ii) rejecting easy negatives during mini-batch training; and (iii) using a ratio of negatives to positives that is two orders of magnitude larger than existing approaches while constructing training mini-batches.

CHAPTER 2: BACKGROUND

In this chapter, we introduce literature in both Computer Vision and Natural Language Processing communities that lays the foundation for much of this thesis.

2.1 THE CONCEPT OF A "CONCEPT"

It is important to distinguish between concepts and categories. A category is a collection of instances which are treated as if they are the same. A collection of images which have all been labeled as "dog" form a visual "dog" category. Note that categorization only requires identifying whether an item belongs to a category but does not require any knowledge of how the categories relate to each other.

For the purposes of this work, concepts are similar to categories such that one can assess the degree to which an item is associated with a concept. However, unlike categories, concepts do not exist in isolation but are always defined in relation to other concepts [1]. For example, it is impossible to understand the concept of a "dog" without invoking other concepts like "pet", "hairy", "needy", "cute", "paws", "tail", "fluffy" *etc.* With deep learning, computer vision has come a long way in categorization, but representation of concepts leave much to be desired.

Features *vs.* embeddings as concept representations. Consider a convolutional neural network (CNN) that is trained for the task of classifying images as "dog","cat", or "whale". Any network that maps all images of the same category to a unique point, a "hash code" for the category, in the feature space solves the classification task perfectly. The 3 points in the feature space corresponding to the 3 categories are in a way the perfect features for the task. But are these unique "hash codes" a good representation of visual concepts underlying those categories? Not necessarily. Imagine the feature representation for "dog", "cat" and "whale" in a 3-dimensional space are $[1, 0, 0]$, $[0, 1, 0]$, and $[0, 0, 1]$ respectively. Assuming a euclidean distance metric, such a representation fails to encode that the concept of a "dog" is more similar to "cat" than "whale" because the first two are domestic land-dwelling quadrupeds while the latter is a sea creature.

We will differentiate embeddings of concepts from features as providing a meaningful metric space where distances between embeddings are indicative of relationship between those concepts. Such a metric space might be induced through inductive biases in the network architecture (*e.g.* using convolutional layers instead of fully connected layers) for embedding images or through explicitly enforcing constraints or desirable properties during training.

2.2 VISUAL REPRESENTATIONS

Since the rise of deep learning, the most common visual representations are those learned by CNNs through supervised classification tasks, particularly the ImageNet classification task [2]. ImageNet images assume the image boundaries tightly enclose the object. However, finetuning models initialized with ImageNet trained weights perform well for tasks that require localization such as object detection [3, 4, 5] and segmentation [6, 7]. For vision-language tasks, the following two approaches are common -

Whole Image Representations: The image is fed through a CNN and intermediate convolutional feature maps are used as features. The features may further be spatially aggregated using mean pooling or learned transformations such as fully connected layers [8, 9] or attention [10, 11, 12]. The CNN is typically pretrained on ImageNet classification and finetuned end-to-end on the vision-language task of interest.

Region-level Representations: Object regions are extracted using object detectors or unsupervised methods like Edge Boxes [13] or Selective Search [14]. The regions are then encoded using a CNN or ROI-pooled features from the object detector. Detectors trained on a large number of object and attribute annotations from densely annotated datasets like VisualGenome [15] have been shown to outperform those learned only on a small number of object categories such those found in MSCOCO [16]. For tasks like VQA, region-level representations are aggregated using attention to construct a question relevant visual representation of the whole image [17, 16, 18]. The aggregation is often a linear combination of region features with attention scores as weights. Hence, a well trained attention model is expected to assign high attention scores to regions relevant to answering the question while assigning low scores to irrelevant regions. Note that attention scores are treated as latent variables which are trained only through supervision provided by the downstream VQA task. Region-level representations outperform whole-image representation based approaches in tasks like VQA [18].

More recently, self-supervised or unsupervised representation learning approaches [19, 20, 21] have shown promising results. Features learned completely without ground truth category labels are now able to achieve performance competitive to fully supervised features on image classification, detection and segmentation tasks [22, 23]. An untested hypothesis is that self-supervised features provide better embeddings for images by encoding attribute information. Such visual appearance information may not be encoded in the features learned through fully supervised object classification because they are trained to be invariant to within class appearance variations. Whether self-supervised features benefit vision-language tasks remains to be seen.

2.3 LANGUAGE REPRESENTATIONS

Language representations for vision-language tasks range from word-level representations to sentence level representations.

Word-level Representations. Words in text, especially nouns, adjectives, verbs, adverbs, and prepositions correspond to concepts that may be visual (green), auditory (loud), tactile (soft), olfactory (fragrant), or abstract (gravity) in nature. Recent methods for representing words share the hypothesis that representations of word meaning may be derived by modeling a word's association or co-occurrence with other words in large natural language corpora. In practice, this takes the form of word vector representations obtained through factorization of co-occurrence matrices. The factorization may be explicit and global such as factorization of raw co-occurrence count matrices [24] or transformations of counts such as Positive Pointwise Mutual Information [25], Hellinger distance [26], or log co-occurrence counts [27]. Factorization could also be implicit and local such as performed by Continuous Bag-of-Words and Skip-Gram [28] approaches that scan a document using a local window. It has been established that local window approaches like word2vec [28] perform implicit matrix factorization [29] and are specific instantiations of global approaches like GloVe [27].

Sentence-level Representations. Some vision-language tasks such as VQA [8], or Caption-Image retrieval [30, 31, 32] might require a vector representation of the entire sentence. This is often done by feeding the sequence of words through a recurrent model such as LSTM [33] or GRU [34], and the hidden representation output at the last time step is used as the sentence embedding [35]. These sentence level representations are usually trained directly on the downstream task with parameters of the recurrent model learned using Backpropogation Through Time [36].

Contextualized Word Representations. Recently, language models pretrained on large text corpora have shown strong performance as feature extractors that simultaneously encode word representations and sentence context. The first work to demonstrate strong performance of language model features on a wide range of NLP tasks is ELMo [37]. The language model is an LSTM pretrained to maximize the log likelihood of sentences in a large text corpora. The hidden layer outputs at different time steps were used as contextualized word representations for the corresponding words. Note that language models are pretrained in a completely self-supervised fashion. Therefore, generalization of language model features for tasks such as question answering, semantic role labelling, coreference resolution, named entity recognition *etc.* is a significant milestone in computational methods for natural language understanding.

Another breakthrough in contextualized word representation occurred through a novel attention based sequence encoder known as a Transformer [38]. Transformers consist of a stack of key-value attention layers interleaved with fully connected layers. The attended representation from the context of a word are added as a residual to the transformed representation of the word created by the previous layer. BERT [39] is a successful transformer based model. Unlike previous language models which are trained to maximize the log likelihood of sentences in a text corpora, BERT is trained using a masked language modeling (MLM) objective. MLM training randomly masks a fraction of the words in the input to the Transformer and maximizes the log likelihood of the masked words. Note that this is similar to the Continuous Bag-of-Words (CBOW) model used for word representations where words in a context window are used to predict the current word. However, CBOW learns a global uncontextualized vector representation for each word and the task of contextualization is expected to be learned by the downstream task models. BERT shows that not only word representations but also contextualization can be learned through a generative model of natural language.

2.4 MULTITASK AND TRANSFER LEARNING

Vision-language tasks such as VQA, Image Captioning, Text-Image Retrieval, Phrase Grounding *etc.* have a lot in common. These tasks not only share concepts (*e.g.* a "dog" refers to the same concept in each task) but likely involve similar inference over images and text (*e.g.* matching words to image regions). Hence it is important to investigate ways of sharing knowledge across different vision-language tasks. Below we discuss two perspectives on sharing knowledge -

Multitask Learning. Multitask learning [40] refers to simultaneously learning to solve multiple tasks. The most common way of multitask learning for vision applications is sharing a set of base visual features across tasks with task-specific layers operating on the shared features to address individual tasks. However, such an approach requires each task-specific head to learn to reinterpret the base visual features and conflicting training signals could lead to worse performance than training on individual tasks. Therefore, multitask learning is most successful when the tasks are related. Multitask learning can be viewed as a regularization alternative to uniformly penalizing all complexity, such as through weight decay, by requiring that the representations or the inference algorithm work well on a related task.

Transfer Learning. Transfer learning [41] is a more general term that refers to learning a skill or a concept from one task that is useful in solving another task. This is also referred to as inductive transfer since learning from one task induces a more general principle or representation that is applicable to another task. Multitask learning is one approach to transfer learning which requires training on multiple tasks simultaneously while sharing representations across tasks. Another popular approach is pretraining on one task and finetuning on the other task. This approach suffers from catastrophic forgetting of the old task and various methods like LWF [42], iCARL [43], and DeepInversion [44] seek to address this problem. In addition to generalization across tasks, transfer learning also applies to generalization of learning across domains for the same task (*e.g.* synthetic to real images), learning complex skills from previously learned simpler skills (often studied under curriculum learning [45, 46]), generalization of inference to novel concepts (*e.g.* generating captions about novel objects unseen in captioning training data [47]), and generalization to parts of the target data distribution that were undersampled or unseen during training.

CHAPTER 3: GENERALIZABLE SHARED VISION-LANGUAGE REPRESENTATIONS

3.1 INTRODUCTION

The application of knowledge learned while solving one task to solve another task is known as transfer learning or inductive transfer. In literature, deep features or weights learned through pre-training or multitask learning are used as foundation for learning new tasks. However, the relation of features to each new task needs to be re-learned using the new tasks data. In this chapter, we present a shared vision-language representation (SVLR) space as a means to achieve inductive transfer between related vision-language tasks without the need to re-learn this mapping.

We focus on transfer between visual recognition (VR) and attention-based visual question answering (VQA). Towards this goal we create an SVLR module (Fig. 3.1) that represents an image region as a vector using a CNN and a word as another vector of the same dimension obtained by transforming the corresponding word2vec embedding through fully connected layers. We then formulate a region's score for a given class in VR in terms of inner product of the region and word representation produced by SVLR. When the model is trained on VR, SVLR representations of a region and words that apply to that region are mapped close together while inapplicable words are mapped further away. We refer to this process as the *alignment* of image and word representations.

Inference in VQA is now formulated to use these aligned representations in three ways. *First*, each EdgeBox region proposal is represented as vector of pre-selected object and attribute class scores computed using SVLR. *Second*, these region representations are pooled to get image representation using attention scores as weights. These attention score are also computed using SVLR. Here, we make an assumption that a region is relevant to a question and candidate answer (QA) pair iff it contains a noun or an adjective present in the QA. For example, to evaluate if the answer "red" is correct for the question "What color is the skier's jacket?", a region is relevant iff it contains the adjective "red" or one of the nouns "color", "skier", or "jacket". *Third*, the pooled image representation is concatenated with QA representations which is scored by a set of bimodal pooling and fully connected layers. The QA representation used here are also constructed from SVLR word representations. In our framework, not only do we expect training on VR to help VQA, but also training on VQA to form newer region-word alignments and reinforce existing ones.

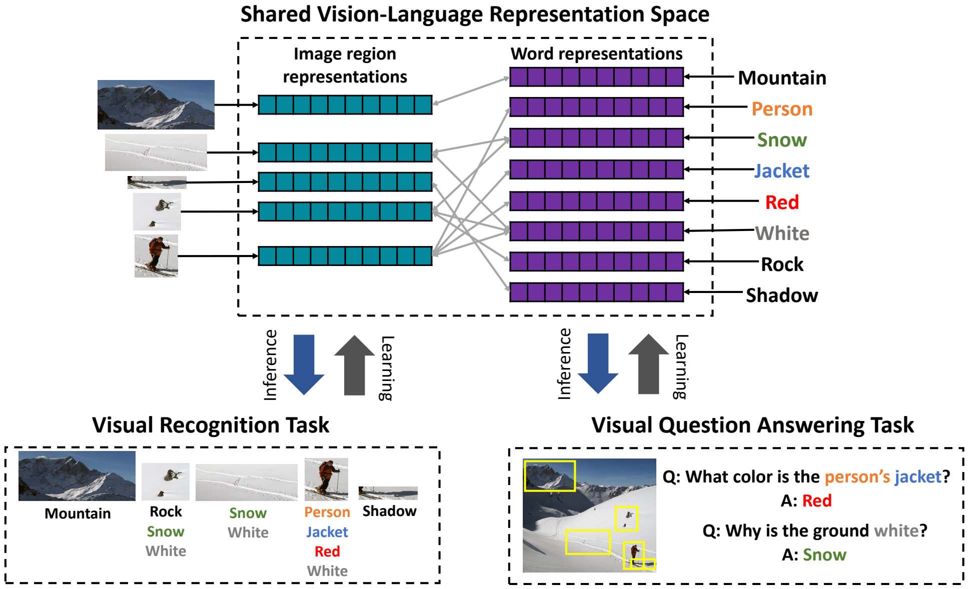


Figure 3.1: **Sharing region and word representations across multiple vision-language tasks:** The SVLR module projects image-regions and words into visual and textual embeddings which are shared across tasks like Visual Recognition and VQA. The models for individual tasks are formulated in terms of inner products of region and word representations enforcing an alignment between them during training.

3.2 RELATED WORK

Our framework is motivated by the never-ending learning (NEL) paradigm [48, 49, 50, 51, 52]. NEL aims to continuously learn from multiple tasks such that learning to solve newer problems becomes easier. Representation learning [53], multitask learning [40], and curriculum learning [45] are different aspects of this larger paradigm. Inductive transfer through shared representations is a necessary first step for NEL. Most works focus on building transferable representations within a single modality such as language or vision only. We extend this framework to learn a shared vision-language representation space which enables a much larger class of vision-language tasks to easily build on and contribute to the shared representation. We now describe how our formulation of VR and VQA in the joint learning setup differs from models that focus on these tasks independently.

Recognition using vision-language embeddings. Traditionally, visual recognition has been posed as multiclass classification over discrete labels [54, 55, 56]. Using these recognizers for tasks like VQA and image captioning is challenging because of the open-vocabulary nature of these problems. Availability of continuous word embeddings (e.g. word2vec [57]) has allowed reformulation of visual recognition as a nearest neighbor search in a learned image-language embedding space [58]. Such embeddings have been successfully applied to a variety of tasks that require recognition such as image captioning [59, 60], phrase localization [61, 15], referring expressions [62, 63], and VQA [8, 64, 65].

Our recognition model is related to previous open-vocabulary recognition and localization models [58, 66, 67]. However, we specifically focus on the multitask setting where VR forms a part of a higher-level vision-language task such as VQA. Since the SVLR module is reused in both tasks with inner products in the embedding space forming the basis for both models, during joint training VQA provides a weak supervision for recognition as well. Fang et al. [68] also learn object and attribute classifiers from weak supervision in the form of image-caption pairs using a multiple instance learning (MIL) framework, but do not use vision-language embeddings. Liu et al. [69] use VR annotation from Flickr30K entities [61] to co-supervise attention in a caption generation model on the same dataset. Our work goes further by allowing the supervision to come from separate datasets, thereby increasing the amount of training data available for the shared parameters.

Visual Question Answering. VQA involves responding to a natural language query about an image. Our VQA model is closely related to attention-based VQA models [70, 71, 10, 72, 73, 11, 74, 75, 76, 77] which attempt to compute a distribution (region relevance or attention) over the regions/pixels in an image using inner product of image-region and the full query embedding [72, 73, 71, 10]. Attention scores are used as weights to pool relevant visual information which is usually combined with the language representation to create a multimodal representation. Various methods of pooling such as elementwise-addition, multiplication, and outer-products have been explored [11, 70]. Attention models are themselves an active area of research with applications in visual recognition [78, 79], caption generation [80], question answering [81, 82, 76], machine comprehension [83], translation [84, 85], and neural turing machines [86].

Our model explicitly formulates attention in VQA as image localization of nouns and adjectives mentioned in a candidate QA pair. Ilievski et al. [71] use a related approach for attention. They use word2vec to map individual words in the question to the class labels of a pre-trained object detector which then generates the attention map by identifying regions for those labels. Tommasi et al. [77] similarly use a pre-trainined CCA [67] vision-language embedding model to localize noun phrases, then extracts scene, attribute, and object features to answer VQA questions. Our model differs from these methods in two ways: (i) vision-language embeddings for VR allow for end-to-end trainability, and (ii) jointly training on VR provides additional supervision of attention through a different (non-VQA) dataset.

Similar to our work, Andreas et al. [74, 75] build a compositional and interpretable model

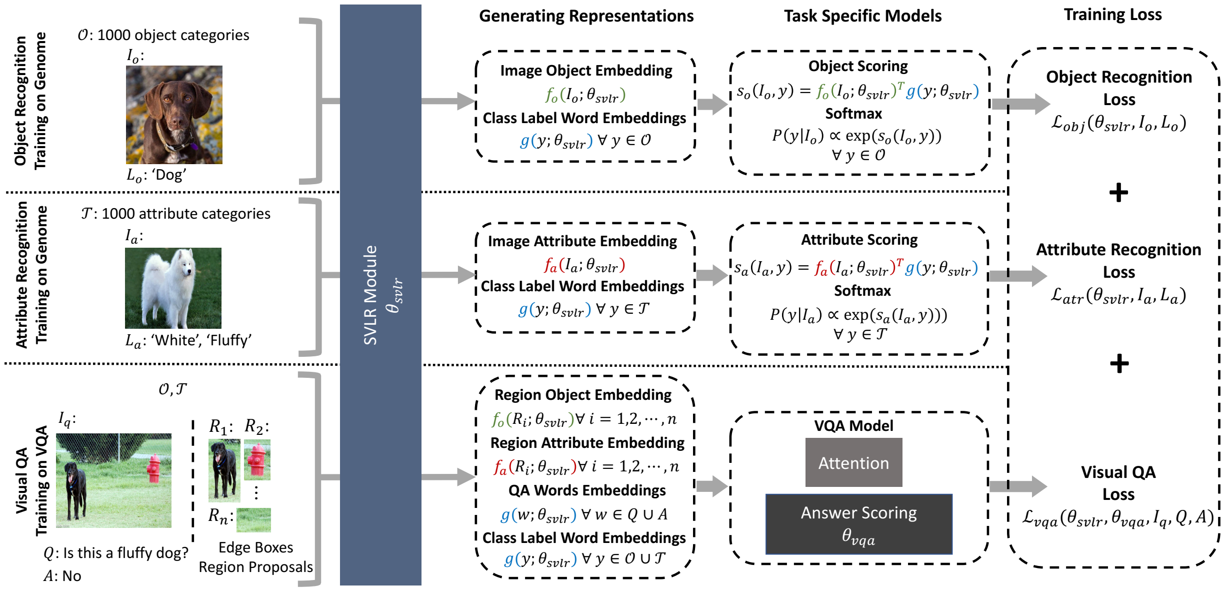

Object Recognition Training on Genome
$\mathcal{O}$: 1000 object categories
I_o:
L_o: 'Dog'
Attribute Recognition Training on Genome
$\mathcal{T}$: 1000 attribute categories
I_a:
L_a: 'White', 'Fluffy'
Visual QA Training on VQA
$\mathcal{O}, \mathcal{T}$
I_q:
Q: Is this a fluffy dog?
A: No
R_1: R_2:
R_n:
Edge Boxes Region Proposals
SVLR Module θ_{svlr}
Generating Representations
Image Object Embedding
$f_o(I_o; \theta_{svlr})$
Class Label Word Embeddings
$g(y; \theta_{svlr})\ \forall\ y \in \mathcal{O}$
Image Attribute Embedding
$f_a(I_a; \theta_{svlr})$
Class Label Word Embeddings
$g(y; \theta_{svlr})\ \forall\ y \in \mathcal{T}$
Region Object Embedding
$f_o(R_i; \theta_{svlr}) \forall\ i = 1,2,\cdots,n$
Region Attribute Embedding
$f_a(R_i; \theta_{svlr}) \forall\ i = 1,2,\cdots,n$
QA Words Embeddings
$g(w; \theta_{svlr})\ \forall\ w \in Q \cup A$
Class Label Word Embeddings
$g(y; \theta_{svlr})\ \forall\ y \in \mathcal{O} \cup \mathcal{T}$
Task Specific Models
Object Scoring
$s_o(I_o, y) = f_o(I_o; \theta_{svlr})^T g(y; \theta_{svlr})$
Softmax
$P(y|I_o) \propto \exp(s_o(I_o, y))$
$\forall\ y \in \mathcal{O}$
Attribute Scoring
$s_a(I_a, y) = f_a(I_a; \theta_{svlr})^T g(y; \theta_{svlr})$
Softmax
$P(y|I_a) \propto \exp(s_a(I_a, y)))$
$\forall\ y \in \mathcal{T}$
VQA Model
Attention
Answer Scoring
θ_{vqa}
Training Loss
Object Recognition Loss
$\mathcal{L}_{obj}(\theta_{svlr}, I_o, L_o)$
+
Attribute Recognition Loss
$\mathcal{L}_{atr}(\theta_{svlr}, I_a, L_a)$
+
Visual QA Loss
$\mathcal{L}_{vqa}(\theta_{svlr}, \theta_{vqa}, I_q, Q, A)$

Figure 3.2: **Joint Training on Visual Recognition(VR) and Visual Question Answering(VQA) with SVLR Module:** The figure depicts sharing of image and word representations through the SVLR module during joint training on object recognition, attribute recognition, and VQA. The recognition tasks use object and attribute labelled regions from Visual Genome while VQA uses images annotated with questions and answers from the VQA dataset. The benefit of joint training is that while the VQA dataset does not provide region groundings of nouns and adjectives in the QA (e.g. "fluffy","dog"), this complementary supervision is provided by the Genome recognition dataset. Models for each task involve image and word embeddings produced by SVLR module or their inner products (See Fig 3.3 for VQA model architecture).

for VQA that relies on the syntactic parse to dynamically arrange a set of parameterized neural modules that are then applied to the image. Each module performs a specific function such as localizing a specific word or verifying relative locations. In contrast, our approach uses a static model but relies on our shared representations and attention based on the QA parse for modularity and interpretability.

3.3 METHOD

We propose an SVLR module to facilitate greater inductive transfer across vision-language tasks. Fig. 3.2 depicts joint training of SVLR along with VR and VQA models. We now describe the architecture of our proposed SVLR module, and inference and training procedures for VR and VQA in terms of region and word representations produced by SVLR.

3.3.1 SVLR

The SVLR module converts words and image-regions into feature representations that are aligned to each other and shared across tasks.

Word Representations. The representation $g(w)$ for a word w is constructed by applying two fully connected layers (with 300 output units each) to pretrained word2vec representation [28] of w with ReLU after the first layer.

Region Representations. A region R is represented using two 300 dimensional feature vectors $f_o(R)$ and $f_a(R)$ that separately encode the objects and attributes contained. We used two representations instead of one to encourage disentangling of these two factors of variation. For example, we do not expect "red" to be similar to "apple", but we expect $f_o(R)$ and $f_a(R)$ to be similar to $g(\text{"}\textit{red}\text{"})$ and $g(\text{"}\textit{apple}\text{"})$ if R depicts a red apple. The features are constructed by extracting the average pooled features from Resnet [54] pretrained on ImageNet and then passing through separate object and attribute networks. Both networks consist of two fully connected layers (with 2048 and 300 output units) with batch normalization [87] and ReLU activations.

3.3.2 Recognition with SVLR

3.3.2.1 Inference

The visual recognition task is to classify image regions into one or more object and attribute categories. The classification score for region R and object category w is $f_o^T(R)g(w)$. The classification score for an attribute category v is $f_a^T(R)g(v)$. Attributes may include adjectives and adverbs (e.g., "standing"). Though our recognition dataset has a limited set of object categories $\mathcal{O}$ and attribute categories $\mathcal{T}$, our model can produce classification scores for any object or attribute label given its word2vec representation. In experiments, the $\mathcal{O}$ and $\mathcal{T}$ consist of 1000 most frequent object and attribute categories in the Visual Genome dataset [15].

3.3.2.2 Training

Our VR model is trained using the Visual Genome dataset which provides image regions annotated with object and attribute labels. VR uses only the parameters for the embedding functions f_o, f_a and g that are part of the SVLR module. The parameters of f_o receive gradients from the object loss while those of f_a receive gradients from the attribute loss. The parameters of word embedding model g receive gradients from both losses.

Object loss. We use a multi-label loss as object classes may not be mutually exclusive (e.g., "man" *is a* "person"). For a region R_j, we denote the set of annotated object categories and their hypernyms extracted from WordNet [88] by $\mathcal{H}_j$. The object loss forces the true labels and their hypernyms to score higher than all other object labels by a margin η_{obj}. For a batch of M samples $\{(R_j, \mathcal{H}_j)\}_{j=1}^{M}$ the object loss is:

$$\mathcal{L}_{obj} \quad = \quad \frac{1}{M}\sum_{j=1}^{M}\frac{1}{|\mathcal{H}_j|}\sum_{l\in\mathcal{H}_j}\frac{1}{|\mathcal{O}|}\sum_{k\in\mathcal{O}\setminus\mathcal{H}_j}\max\{0, \eta_{obj} \ + \ f_o^T(R_j)g(k) \ - \ f_o^T(R_j)g(l)\} \tag{3.1}$$

Attribute Loss. The attribute loss is a multi-label classification loss with two differences from object classification. Attribute labels are even less likely to be mutually exclusive than object labels. As such, we predict each attribute with independent cross entropy losses. We also weigh the samples based on fraction of positive labels in the batch to balance the positive and negative labels in the dataset. For a batch with M samples $\{(R_j, \mathcal{T}_j)\}_{j=1}^{M}$ where $\mathcal{T}_j$ is the set of attributes annotated for region R_j, the attribute loss is:

$$\begin{aligned}\mathcal{L}_{atr} = \frac{1}{M}\sum_{j=1}^{M}\sum_{t\in\mathcal{T}} \mathbb{1}\left[t\in\mathcal{T}_j\right](1-\Gamma(t))\log\left[\sigma(f_a^T(R_j)g(t))\right] + \\ \mathbb{1}\left[t\notin\mathcal{T}_j\right]\Gamma(t)\log\left[1-\sigma(f_a^T(R_j)g(t))\right]\end{aligned} \tag{3.2}$$

where σ is a sigmoid activation function and $\Gamma(t)$ is the fraction of positive samples for attribute t in the batch.

3.3.3 VQA with SVLR

3.3.3.1 Inference

Our VQA model is illustrated in Fig. 3.3. The input to our VQA model is an image, a question, and a candidate answer. Regions are extracted from the image using Edge Boxes [13]. The same SVLR module used by VR is explicitly applied to VQA for attention and answer scoring. Our system assigns attention scores to each region according to how well it matches words in the question/answer, then scores each answer based on the question, answer, and attention-weighted scores for all objects ($\mathcal{O}$) and attributes ($\mathcal{T}$).

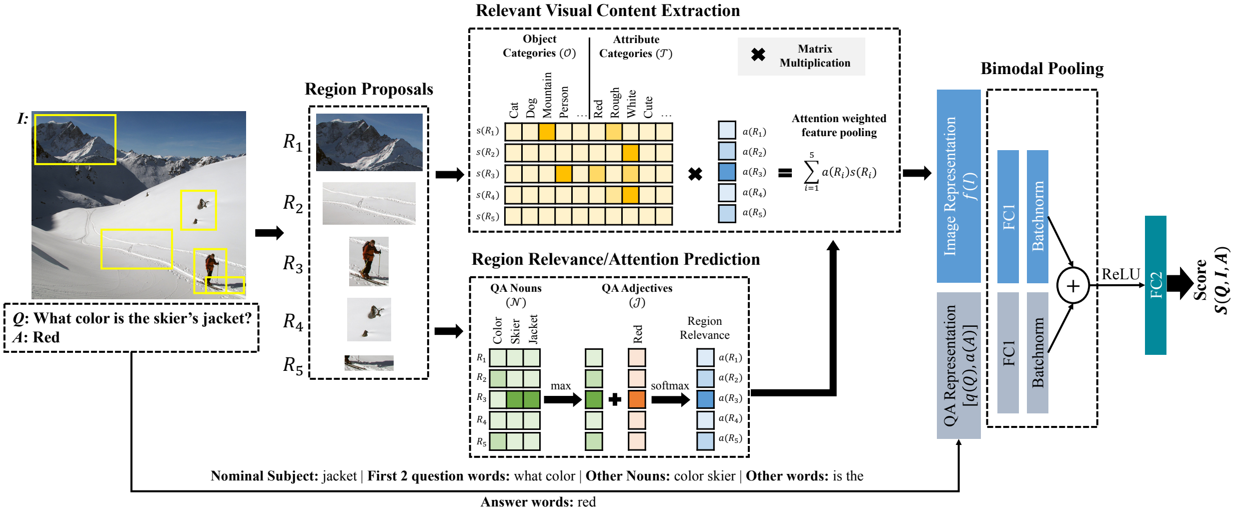

Relevant Visual Content Extraction
Object Categories ($\mathcal{O}$)
Attribute Categories ($\mathcal{T}$)
Matrix Multiplication
Cat
Dog
Mountain
Person
...
Red
Rough
White
Cute
...
$s(R_1)$
$s(R_2)$
$s(R_3)$
$s(R_4)$
$s(R_5)$
$a(R_1)$
$a(R_2)$
$a(R_3)$
$a(R_4)$
$a(R_5)$
Attention weighted feature pooling
$\sum_{i=1}^{5} a(R_i)s(R_i)$
Region Proposals
I:
R_1
R_2
R_3
R_4
R_5
Q: What color is the skier's jacket?
A: Red
Region Relevance/Attention Prediction
QA Nouns ($\mathcal{N}$)
QA Adjectives ($\mathcal{J}$)
Color
Skier
Jacket
Red
Region Relevance
max
softmax
Bimodal Pooling
Image Representation $f(I)$
QA Representation $[q(Q), a(A)]$
FC1
Batchnorm
FC1
Batchnorm
ReLU
FC2
Score $S(Q, I, A)$
Nominal Subject: jacket | First 2 question words: what color | Other Nouns: color skier | Other words: is the
Answer words: red

Figure 3.3: **Inference in our VQA model:** The image is first broken down into Edge Box region proposals[13]. Each region R is represented by visual category scores $s(R) = [s_o(R), s_a(R)]$ obtained using the visual recognition model. Using the SVLR module, the regions are also assigned an attention score using the inner products of region features with representations of nouns and adjectives in the question and answer. The region features are then pooled using the relevance scores as weights to construct the *attended* image representation. Finally, the image and question/answer representations are combined and passed through a neural network to produce a score for the input question-image-answer triplet.

Region Relevance. Unlike other attention models [11, 10] that are free to learn any correlation between regions and question/answers, our attention model encodes an explicit notion of vision-language grounding. Let $\mathcal{R}$ be the set of region proposals extracted from the image, and $\mathcal{N}$ and $\mathcal{J}$ denote the set of nouns and adjectives in the (Q, A) pair. Each region $R \in \mathcal{R}(I)$ is assigned an attention score $a(R)$ as follows:

$$a'(R) = \max_{n \in \mathcal{N}} f_o^T(R) g(n) + \max_{j \in \mathcal{J}} f_a^T(R) g(j) \tag{3.3}$$

$$a(R) = \frac{\exp(a'(R))}{\sum_{R' \in \mathcal{R}(I)} \exp(a'(R'))} \tag{3.4}$$

Thus, a region's attention score is the sum of maximum adjective and noun scores for words mentioned in the question or answer (which need not be in sets $\mathcal{O}$ and $\mathcal{T}$).

Image Representation. To score an answer, the content of region R is encoded using the

VR scores for all objects and attributes in $\mathcal{O}$ and $\mathcal{T}$, as presence of unmentioned objects or attributes may help answer the question. The image representation is an attention-weighted average of these scores across all regions:

$$f(I) = \sum_{R \in \mathcal{R}(I)} a(R) \begin{bmatrix} s_o(R) \\ s_a(R) \end{bmatrix} \tag{3.5}$$

where I is the image, $s_o(R)$ are the scores for 1000 objects in $\mathcal{O}$ for each image region R, $s_a(R)$ are the scores for 1000 attributes in $\mathcal{T}$, and $a(R)$ is the attention score.

Question/Answer Representation. To construct representations $q(Q)$ and $a(A)$ for the question and answer, we follow Shih et al. [73], dividing question words into 4 bins, averaging word representations in each bin, and concatenating the bin representations resulting in a 1200 ($= 300 \times 4$) dimensional vector $q(Q)$. The answer representation $a(A) \in \mathbb{R}^{300}$ is obtained by averaging the word representations of all answer words. The word representations used here are produced by the SVLR module.

Answer Scoring. We combine the image and Q/A representations to jointly score the (Q, I, A) triplet. To ensure equal contribution of language and visual features, we apply batch normalization [87] on linear transformations of these features before adding them together to get a bimodal representation $\beta(Q, I, A) \in \mathbb{R}^{2500}$:

$$\beta(Q, I, A) = \mathcal{B}_1(W_1 f(I)) + \mathcal{B}_2 \left(W_2 \begin{bmatrix} q(Q) \\ a(A) \end{bmatrix} \right) \tag{3.6}$$

Here, $\mathcal{B}_1, \mathcal{B}_2$ denote batch normalization and $W_1 \in \mathbb{R}^{2500 \times 2000}$ and $W_2 \in \mathbb{R}^{2500 \times 1500}$ define the linear transformations. The bimodal representation is:

$$\mathcal{S}(Q, I, A) = W_3 \text{ ReLU}(\beta(Q, I, A)) \tag{3.7}$$

with $W_3 \in \mathbb{R}^{1 \times 2500}$.

3.3.3.2 Training

We use the VQA dataset [8] for training parameters of our VQA model: W_1, W_2, W_3, and scales and offsets of batch normalization layers. In addition, the VQA loss backpropagates into f_o, f_a, and g which are part of the SVLR module. Each sample in the dataset consists of a question Q about an image I with list of answer options including a positive answer A^+ and N negative answers $\{A^-(i)|i = 1, \cdots, N\}$.

The VQA loss encourages the correct answer A^+ to be scored higher than all incorrect answer options $\{A^-(i)|i = 1, \cdots, N\}$ by a margin η_{ans}. Given batch samples $\{(Q_j, I_j, A_j)\}_{j=1}^P$, the loss is written as

$$\mathcal{L}_{ans} = \frac{1}{NP} \sum_{j=1}^{P} \sum_{i=1}^{N} \max\{0, \eta_{ans} + \mathcal{S}(Q_j, I_j, A_j^-(i)) - \mathcal{S}(Q_j, I_j, A_j^+)\} \tag{3.8}$$

3.3.3.3 Zero-Shot VQA

The representations produced by SVLR module should be directly usable in related vision-language tasks without any additional learning. To demonstrate this *zero-shot cross-task transfer*, we train the SVLR module using Genome VR data only and apply to VQA. Since bimodal pooling and scoring layers cannot be learned without VQA data, we use a proxy scoring function constructed using region-word scores only. For each region, we compute $p_q(R)$ as the sum of its scores for the maximally aligned question nouns and question adjectives (Eq. 3.3 with only question words). A score $p_a(R)$ is similarly computed using answer nouns and adjectives. The final score for the answer is defined by

$$S(Q, I, A) = \sum_{R \in \mathcal{R}} a(R) \min(p_q(R), p_a(R)) \tag{3.9}$$

where a is the attention score computed using Eq. 3.4. Therefore, the highest score is given to QA pairs where question as well as answer nouns and adjectives can be localized in the image. Note that since the model is not trained on even a *single question* from VQA, the zero-shot VQA task also shows that our model does use the image to answer questions instead of solely relying on the language prior which is a common concern with most VQA models [89, 90].

3.4 EXPERIMENTS

Our experiments investigate the extent to which using SVLR as a core representation improves transfer in multitask learning. We first analyze how including the VR task improves VQA (Sec. 3.4.2, Tab. 3.1). We find that using SVLR doubles the improvement compared to standard multitask learning, and demonstrate performance well above chance in a zero-shot setup (trained only on VR, applied to VQA). We then analyze improvement to VR due

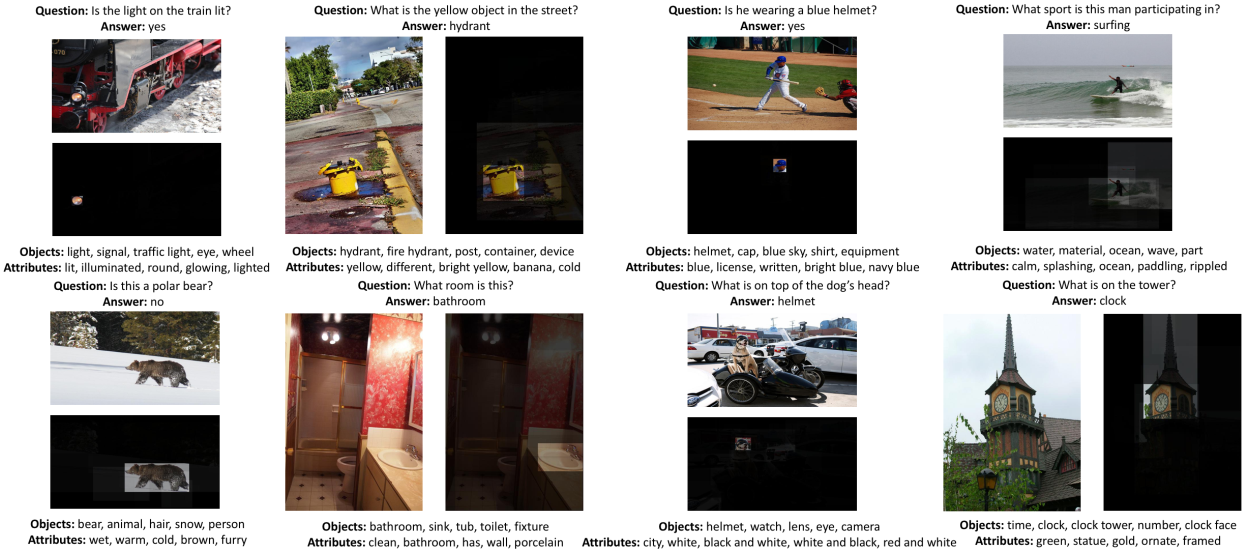

Question: Is the light on the train lit?
Answer: yes
Objects: light, signal, traffic light, eye, wheel
Attributes: lit, illuminated, round, glowing, lighted
Question: What is the yellow object in the street?
Answer: hydrant
Objects: hydrant, fire hydrant, post, container, device
Attributes: yellow, different, bright yellow, banana, cold
Question: Is he wearing a blue helmet?
Answer: yes
Objects: helmet, cap, blue sky, shirt, equipment
Attributes: blue, license, written, bright blue, navy blue
Question: What sport is this man participating in?
Answer: surfing
Objects: water, material, ocean, wave, part
Attributes: calm, splashing, ocean, paddling, rippled
Question: Is this a polar bear?
Answer: no
Objects: bear, animal, hair, snow, person
Attributes: wet, warm, cold, brown, furry
Question: What room is this?
Answer: bathroom
Objects: bathroom, sink, tub, toilet, fixture
Attributes: clean, bathroom, has, wall, porcelain
Question: What is on top of the dog's head?
Answer: helmet
Objects: helmet, watch, lens, eye, camera
Attributes: city, white, black and white, white and black, red and white
Question: What is on the tower?
Answer: clock
Objects: time, clock, clock tower, number, clock face
Attributes: green, statue, gold, ornate, framed

Figure 3.4: **Interpretable inference in VQA:** Our model produces interpretable intermediate computation for region relevance and object/attribute predictions for the most relevant regions. Our region relevance explicitly grounds nouns and adjectives from the Q/A input in the image. We also show object and attribute predictions for the most relevant region identified for a few correctly answered questions. The relevance masks are generated from relevance scores projected back to their source pixels locations.

to training with (weakly supervised) VQA (Sec. 3.4.2, Fig. 3.5). We find moderate overall improvements (1.2%), with the largest improvements for classes that have few VR training examples. We also quantitatively evaluate how well our attention maps correlate with that of humans using data provided by [91] in Table 3.1.

3.4.1 Datasets

Our model is trained on two separate datasets: one for VQA supervision, one for visual recognition (attributes and object classification). We use the image-question-answer annotation triplets from Antol et al. [8] and bounding box annotations for object and attribute categories from Visual Genome [15].

3.4.2 Inductive Transfer from VR to VQA

We evaluate inductive transfer from VR to VQA in both joint training and zero-shot VQA scenarios.

Joint Training. The VR models and VQA model are simultaneously trained using object

Real-MCQ-VQA Validation Accuracies	**what color**	**what is the (wo)man/person**	**what is in/on**	**what kind/ type/animal**	**what room/sport**	can/could/ does/do/has	what does/ number/name	what brand	which/who	what is/are	why/how	how many	what time	where	is/are/was	none of the above	other	number	yes/no	overall accuracy
VQA Only	53.5	70.5	53.6	56.8	89.8	81.8	41.9	45.9	49.0	58.3	**33.8**	38.4	**53.9**	45.8	80.2	56.0	54.5	39.2	82.1	62.9
Joint Multitask	59.4	71.8	54.6	58.3	91.0	81.9	**43.8**	46.4	50.8	59.2	32.3	**39.4**	**53.9**	47.0	80.4	57.1	56.7	**39.8**	82.2	64.1
Joint SVLR	**62.1**	**74.1**	**57.9**	**60.0**	**91.1**	**82.8**	41.6	**52.9**	**52.0**	**61.1**	33.6	39.0	51.3	**48.6**	**81.4**	**58.5**	**58.8**	38.8	**83.0**	**65.3**
Zero-Shot VQA	18.8	21.0	27.4	31.4	22.0	17.1	13.9	11.6	20.6	22.9	12.7	0.7	7.2	26.1	13.5	19.2	22.4	1.2	13.3	16.4

Table 3.1: **Inductive transfer from VR to VQA through SVLR in joint training and zero-shot settings:** We evaluate the performance of our model with SVLR module trained jointly with VR and VQA supervision (provided by Genome and VQA datasets respectively) on the VQA task. We compare this *jointly-trained* model to a model trained on *only* VQA data. We also compare to a traditional multitask learning setup that is jointly trained on VQA and VR (i.e. uses same amount of data as Joint SVLR) and shares visual features but *does not* use the object and attribute word embeddings for recognition. While multitask learning outperforms VQA-only model, using the SVLR module doubles the improvement. Our model is most suited for the question types in bold that require visual recognition without specialized skills like counting or reading. Formulation of VR and *attention* in VQA in terms of inner products between word and region representations enables Zero-Shot VQA. In this setting we train on Genome VR data and apply to VQA val (Sec 3.4.2).

and attribute annotations from Genome, and Q/A annotations from the VQA dataset. The common approach to joint training is to use a common network for extracting image features (e.g. class logits from ResNet), which feeds into the task-specific networks as input. We refer to this approach in Table 3.1 as *Joint Multitask*. This baseline is implemented by replacing $g(y)$ (see Fig. 3.2) with a trainable set of vectors h_y for each of the predetermined 1000 object and 1000 attribute categories in the VR models. The embedding $g(y)$ is still in the VQA model, but is no longer shared across tasks. Our proposed *Joint SVLR* outperforms VQA-only by 2.4%, doubling the 1.2% improvement achieved by *Joint Multitask*. Our formulation of VR and VQA tasks in terms of shared word-region representations more effectively transfers recognition knowledge from VR than shared features. The gain is often larger on questions that involve recognition (in bold in Table 3.1). For example, *what color* questions improve by 8.6% due to SVLR.

Zero-Shot VQA. We evaluate on Zero-shot VQA to further highlight transfer from VR to VQA. We train on only Genome VR annotations but test on VQA val. The model has not seen any Q/A training data, but achieves an overall accuracy of 16.4% where random guessing yields 5.6% (18 choices). Our zero-shot system does not exploit language priors, which alone can score as high as 54.0% [73]. This shows that some knowledge can be directly

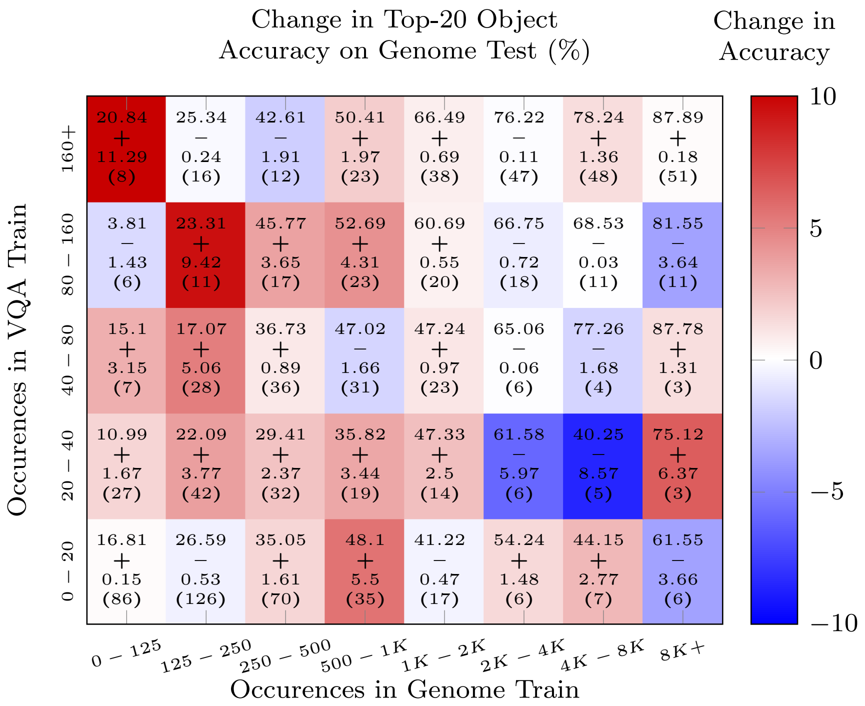


Figure 3.5: **Transfer from VQA to Object Recognition:** Each cell's color reflects the mean change in accuracy for classes within the corresponding frequency ranges of both datasets' training split. Most gains are in nouns rare in Genome but common in VQA (top left), suggesting that the weak supervision provided by training VQA attention augments recognition performance via the SVLR. The numbers in each cell show the Genome-only mean accuracy +/- the change due to SVLR multitask training, followed by the number of classes in the cell in parentheses.

applied to related tasks using SVLR without additional training.

3.4.3 Inductive Transfer from VQA to VR

We compare the performance of our SVLR based model trained jointly on VQA and VR data with a model trained only on Genome data to analyze transfer from VQA to VR. Genome *test* is used for evaluation. We observe an increase in the overall object recognition accuracy from 43.3% to 44.5%, whereas average attribute accuracy remained unchanged at 36.9%. In Fig. 3.5, we show that nouns that are rare in Genome (left columns) but have 20 or more examples in VQA (upper rows) benefit the most from weak supervision provided by VQA. On average, we measure improvement from 21% to 32% for the 8 classes that have fewer than 125 examples in Genome train but occur more than 160 times in VQA questions. We conducted the same analysis on Genome attributes, but did not observe any

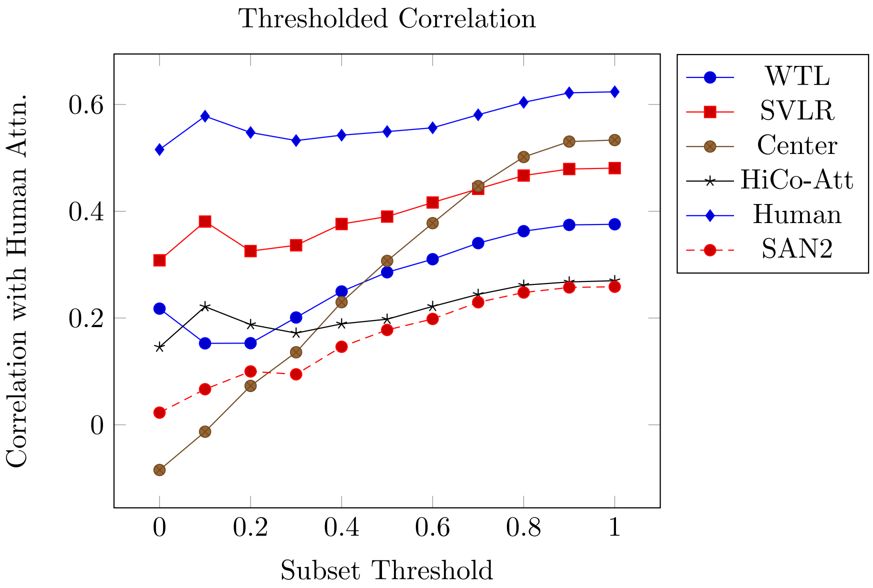


Figure 3.6: **Mean Spearman rank-correlation between model predicted and human attention at various thresholds.**Each threshold point defines a subset of the dataset for which the human attention correlation with the synthetic center heatmap is below that threshold value. For example: the first sample point of each curve is the mean correlation of each model with human attention, measured on a subset in which the human attention's correlation with the center heatmap is less than or equal to 0. As can be seen, the attention maps produced by the proposed SVLR model correlate with human attention significantly more than other models. As the threshold approaches 1, the synthetic center heatmap baseline outperforms all proposed models, confirming that the majority of the questions are about something in the center of the image. Note that due to slight differences in implementations, the subsets at ≤ 0 differ slightly from those used in [91]

notable pattern, possibly due to the inherent difficult in evaluating the multi-label attribute classification problem (the absence of attributes is not annotated in Genome).

3.4.4 Interpretable Inference for VQA

As shown in Fig. 5.5, our VQA model produces interpretable intermediate outputs such as region relevance and visual category predictions, similar to [77]. The answer choice is explained by the object and attribute predictions associated with the most relevant regions. Because relevance is posed as the explicit localization of words in the question and answer, we can qualitatively evaluate the relevance prediction by verifying that the predicted regions match said words.

We also quantitatively evaluate our attention using collected human attention maps from Das et al. [91] in Figure 3.6. We compare the correlation of our attention maps with human attention on subsets of human-attention maps. The subsets are chosen based on their correlation with center-focus heatmap. Our proposed SVLR model significantly outperforms other models we compare with. However, we note that a center-focused heatmap baseline still outperforms all models, signifying that the main topic of a question is very often located in the center of the image. Learned attention models appear to have better correlation with human attention at lower thresholds where the human attention correlates poorly with the center-focused heatmap – a result also demonstrated in [91].

3.5 CONCLUSION

Humans continuously improve their representation of the world with every new experience and use this world model to learn new skills. We attempt to achieve this behavior for the class of vision-language models using shared and aligned image and word representations. Using visual recognition and VQA tasks as examples, we demonstrate the ability of the proposed shared representations to enhance inductive transfer between tasks while simultaneously making of complex systems like VQA more interpretable.

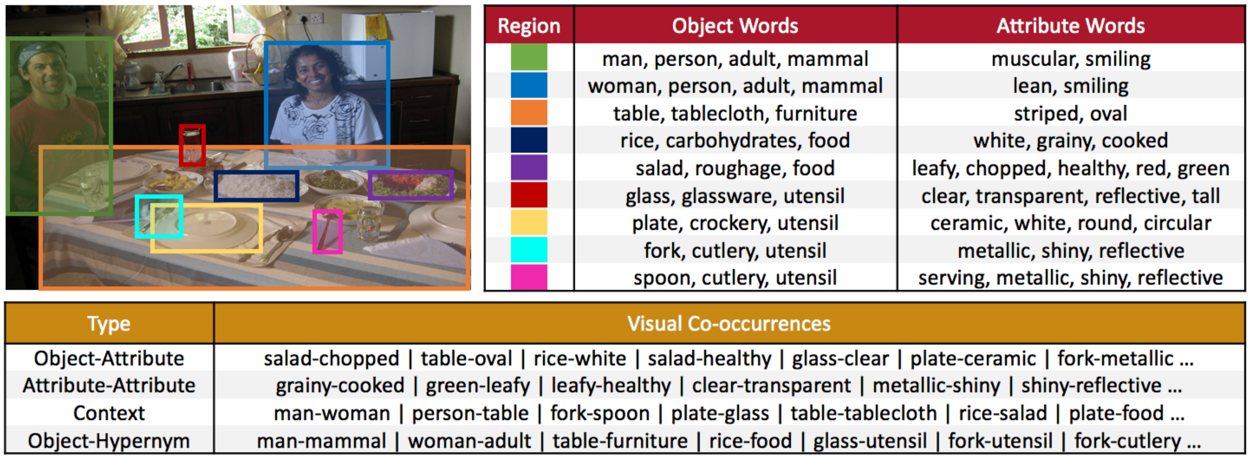
CHAPTER 4: WORD EMBEDDINGS FROM VISUAL CO-OCCURRENCES

4.1 INTRODUCTION

Word embeddings, *i.e.*, compact vector representations of words, are an integral component in many language [92, 93, 94, 95, 96, 97, 98] and vision-language models [99, 100, 101, 102, 103, 104, 105, 106, 18, 107, 17, 108, 109]. These word embeddings, *e.g.*, GloVe and word2vec, are typically learned from large-scale text corpora by modeling textual co-occurrences. However, text often consists of interpretations of concepts or events rather than a description of visual appearance. This limits the ability of text-only word embeddings to represent visual concepts.

To address this shortcoming, we propose to gather co-occurrence statistics of words based on images and learn word embeddings from these visual co-occurrences. Concretely, two words co-occur visually if both words are applicable to the same image or image region. We use four types of co-occurrences as shown in Fig. 4.1: (1) *Object-Attribute* co-occurrence between an object in an image region and the region's attributes; (2) *Attribute-Attribute* co-occurrence of a region; (3) *Context* co-occurrence which captures joint object appearance in the same image; and (4) *Object-Hypernym* co-occurrence between a visual category and its hypernym (super-class).

Ideally, for reliable visual co-occurrence modeling of a sufficiently large vocabulary (a vocabulary size of 400K is typical for text-only embeddings), a dataset with all applicable vocabulary words annotated for each region in an image is required. While no visual dataset



Region	Object Words	Attribute Words
🟩	man, person, adult, mammal	muscular, smiling
🟦	woman, person, adult, mammal	lean, smiling
🟧	table, tablecloth, furniture	striped, oval
⬛	rice, carbohydrates, food	white, grainy, cooked
🟪	salad, roughage, food	leafy, chopped, healthy, red, green
🟥	glass, glassware, utensil	clear, transparent, reflective, tall
🟨	plate, crockery, utensil	ceramic, white, round, circular
🟦	fork, cutlery, utensil	metallic, shiny, reflective
🟪	spoon, cutlery, utensil	serving, metallic, shiny, reflective

Type	Visual Co-occurrences
Object-Attribute	salad-chopped | table-oval | rice-white | salad-healthy | glass-clear | plate-ceramic | fork-metallic ...
Attribute-Attribute	grainy-cooked | green-leafy | leafy-healthy | clear-transparent | metallic-shiny | shiny-reflective ...
Context	man-woman | person-table | fork-spoon | plate-glass | table-tablecloth | rice-salad | plate-food ...
Object-Hypernym	man-mammal | woman-adult | table-furniture | rice-food | glass-utensil | fork-utensil | fork-cutlery ...

Figure 4.1: **Visual co-occurrences are a rich source of information for learning word meanings.** The figure shows regions annotated with words and attributes in an image, and the four types of visual co-occurrences used for learning ViCo embeddings.

exists with such exhaustive annotations (many non-annotated words may still be applicable to an image region), large scale datasets like VisualGenome [110] and ImageNet [2] along with their WordNet [111] *synset* annotations provide a good starting point. We use ImageNet annotations augmented with WordNet hypernyms to compute Object-Hypernym co-occurrences while the remaining types of co-occurrence are computed from VisualGenome's object and attribute annotations.

To learn ViCo, *i.e.*, word embeddings from **Vi**sual **Co**-occurrences, we could concatenate GloVe-like embeddings trained separately for each co-occurrence type via a log-bilinear model. However, in this naïve approach, the dimensionality of the learned embeddings scales linearly with the number of co-occurrence types. To avoid this linear scaling, we extend the log-bilinear model by formulating a *multi-task* problem, where learning embeddings from each co-occurrence type constitutes a different task with compact trainable embeddings shared among all tasks. In this formulation the embedding dimension can be chosen independently of the number of co-occurrence types.

To test ViCo's ability to capture similarities and differences between visual concepts, we analyze performance in an *unsupervised clustering* and a *zero-shot-like* visual generalization setting. The clustering analysis is performed on a set of most frequent words in VisualGenome which we manually label with *coarse* and *fine-grained* visual categories. For the *zero-shot-like* setting, we use CIFAR-100 with different splits of the 100 categories into seen and unseen sets. In both cases, ViCo augmented GloVe outperforms GloVe, random vectors, *vis-w2v*, or their combinations. Through a qualitative analogy question answering evaluation, we also find ViCo embedding space to better capture relations between visual concepts than GloVe.

We also evaluate ViCo on five downstream tasks – a discriminative attributes task, and four vision-language tasks. The latter includes Caption-Image Retrieval, VQA, Referring Expression Comprehension, and Image Captioning. Systems using ViCo outperform those using GloVe for almost all tasks and metrics.

While learned embeddings are typically believed to be important for vision-language tasks, somewhat surprisingly, we find random embeddings compete tightly with learned embeddings on all vision-language tasks. This suggests that either by nature of the tasks, model design, or simply training on large datasets, the current state-of-the-art vision-language models do not benefit much from learned embeddings. Random embeddings perform significantly worse than learned embeddings in our clustering, partitioning, and zero-shot analysis, as well as the discriminative attributes task, which does not involve images.

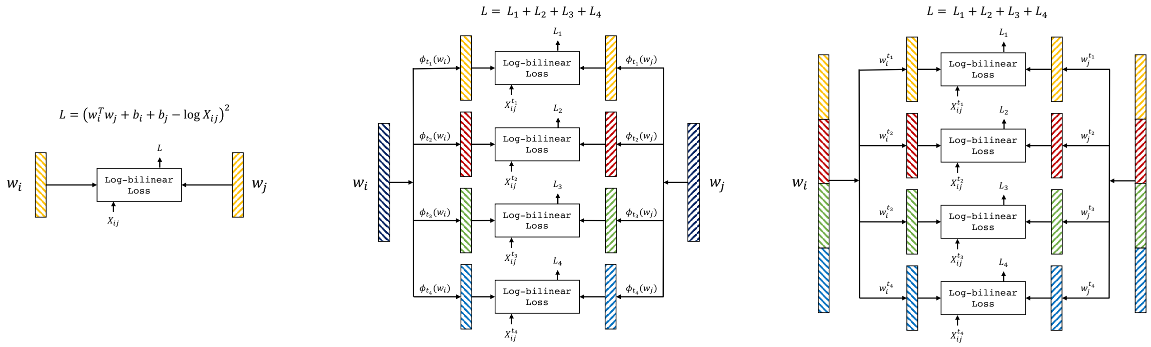
To summarize our contributions: (1) We develop a multi-task method to learn a word embedding from multiple types of co-occurrences; (2) We show that the embeddings learned from multiple visual co-occurrences, when combined with GloVe, outperform GloVe alone in unsupervised clustering and zero-shot-like analysis, as well as on multiple vision-language tasks; (3) We find that performance of supervised vision-language models is relatively insensitive to word embeddings, with even random embeddings leading to nearly the same performance as learned embeddings. To the best of our knowledge, our study provides the first empirical evidence of this unintuitive behavior for multiple vision-language tasks.

4.2 RELATED WORK

Here we describe non-associative, associative, and the most recent contextual models of word representation.

Non-Associative Models. Semantic Differential (SD) [112] is among the earliest attempts to obtain vector representations of words. SD relies on human ratings of words on 50 scales between bipolar adjectives, such as 'happy-sad' or 'slow-fast.' Osgood *et al.* [112] further reduced the 50 scales to 3 orthogonal factors. However, the scales were often vague (*e.g.*, is the word 'coffee' 'slow' or 'fast') and provided a limited representation of the word meaning. Another approach involved acquiring word similarity annotations followed by applying Multidimensional Scaling (MDS) [113] to obtain low dimensional (typically 2-4) embeddings and then identifying meaningful clusters or interpretable dimensions [114]. Like SD, the MDS approach lacked representation power, and embeddings and their interpretations varied based on words (*e.g.*, food names [114], animals [115], *etc.*) to which MDS was applied.

Associative Models. The hypothesis underlying associative models is that word-meaning may be derived by modeling a word's association with all other words. Early attempts involved factorization of word-document [24] or word-word [25] co-occurrence matrices. Since raw co-occurrence counts can span several orders of magnitude, transformations of the co-occurrence matrix based on Positive Pointwise Mutual Information (PPMI) [116] and Hellinger distance [26] have been proposed. Recent neural approaches like the Continuous Bag-of-Words (CBOW) and the Skip-Gram models [117, 118, 119] learn from co-occurrences in local context windows as opposed to global co-occurrence statistics. Unlike global matrix factorization, local context window based approaches use co-occurrence statistics rather inefficiently because of the requirement of scanning context windows in a corpus during training but performed better on word-analogy tasks. Levy *et al.* [29] later showed that Skip-Gram model with negative-sampling performs implicit matrix factorization of a PMI word-context matrix.



(i) Log-bilinear Model (GloVe)
Learning from a single co-occurrence type

(ii) Multi-task Log-bilinear
Learning a shared, more compact embedding across different co-occurrence types

(iii) Multi-task with Select Transform
Learning separate embeddings for each co-occurrence type followed by concatenation

Figure 4.2: **Log-bilinear models and our multi-task extension.** We show loss computation of different approaches for learning word embeddings w_i and w_j for words i and j. The embeddings are denoted by colored vertical bars. (i) shows GloVe's log-bilinear model. (ii) is our multi-task extension to learn from multiple co-occurrence matrices. Word embeddings w_i and w_j are projected into a dedicated space for each co-occurrence type t through transformation ϕ_t. Log-bilinear losses are computed in the projected embedding spaces. (iii) shows an approach where the different colored regions of w_i (or w_j) are allocated to learn from different co-occurrence types. This approach, equivalent to training separate embeddings followed by concatenation, can be implemented in our multi-task formulation using a *select* transform (Tab. 4.1). Tab. 4.4 shows that an appropriate choice of ϕ (*e.g.*, *linear*) in the multi-task framework leads to more compact embeddings than (iii) without sacrificing performance since the correlation between different co-occurrence types is utilized.

Our work is most closely related to GloVe [27] which combines the efficiency of global matrix factorization approaches with the performance obtained from modelling local context. We extend GloVe's log-bilinear model to simultaneously learn from multiple types of co-occurrences. We also demonstrate that visual datasets annotated with words are a rich source of co-occurrence information that complements the representations learned from text corpora alone.

Visual Word Embeddings. There is some work on incorporating image representations into word embeddings. *vis-w2v* [120] uses abstract (synthetic) scenes to learn visual relatedness. The scenes are clustered and cluster membership is used as a surrogate label in a CBOW framework. Abstract scenes have the advantage of providing good semantic features for free but are limited in their ability to match the richness and diversity of natural scenes. However, natural scenes present the challenge of extracting good semantic features. Our approach uses natural scenes but bypasses image feature extraction by only using co-occurrences of annotated words. ViEW [121] is another approach to visually enhance existing word embeddings. An autoencoder is trained on pre-trained word embeddings while matching intermediate representations to visual features extracted from a convolutional network trained on ImageNet. ViEW is also limited by the requirement of good image features.

Contextual Models. Embeddings discussed so far represent individual words. However, many language understanding applications demand representations of words in context (*e.g.*, in a phrase or sentence) which in turn requires to learn how to combine word or character level representations of neighboring words or characters. The past year has seen several advances in contextualized word representations through pre-training on language models such as ELMo [37], OpenAI GPT [122], and BERT [39]. However, building mechanisms for representing context is orthogonal to our goal of improving representations of individual words (which may be used as input to these models).

4.3 LEARNING VICO

We describe the GloVe formulation for learning embeddings from a single co-occurrence matrix in Sec. 4.3.1 and introduce our multi-task extension to learn embeddings jointly from multiple co-occurrence matrices in Sec. 4.3.2. Sec. 4.3.3 describes how co-occurrence count matrices are computed for each of the four co-occurrence types.

4.3.1 GloVe: Log-bilinear Model

Let X_{ij} denote the co-occurrence count between words i and j in a text corpus. Also let $\mathcal{N}$ be the list of word pairs with non-zero co-occurrences. GloVe learns d-dimensional embeddings $w_i \in \mathbb{R}^d$ for all words i by optimizing

$$\min_{w,b} \sum_{(i,j)\in\mathcal{N}} f(X_{ij})(w_i^T w_j + b_i + b_j - \log X_{ij})^2, \tag{4.1}$$

where $f : \mathbb{R} \rightarrow \mathbb{R}$ is a weighting function that assigns lower weight to less frequent, noisy co-occurrences and b_i is a learnable bias term for word i.

Intuitively, the program in Eq. (4.1) learns word embeddings such that for any word pair with non-zero co-occurrence, the dot product $w_i^T w_j$ approximates the log co-occurrence count up to an additive constant. The word meaning is derived by simultaneously modeling the degrees of association of a single word with a large number of other words [1]. We also refer the reader to [27] for more details.

Note the slight difference between the objective in Eq. (4.1) and the original GloVe objective: GloVe replaces w_j and b_j with $\tilde{w}_j$ (context vector) and $\tilde{b}_j$ which are also trainable. The GloVe vectors are obtained by averaging w_i and $\tilde{w}_i$. However, as also noted in [27], given

Transforms	$\boldsymbol{d}$	$\boldsymbol{d_t}$	$\boldsymbol{\phi_t}$
select (200)	200	$50 \; \forall \; t$	$\phi_t(w) = [w[i^t_0], \cdots, w[i^t_{49}]]$ where $\{i^t_0, \cdots, i^t_{49}\}$ are indices pre-allocated for t in $\{0, \cdots, 200\}$
linear (50)	50	$50 \; \forall \; t$	$\phi_t(w) = A_t w$ where $A_t \in \mathbb{R}^{50 \times 50}$
linear (100)	100	$50 \; \forall \; t$	$\phi_t(w) = A_t w$ where $A_t \in \mathbb{R}^{50 \times 100}$
linear (200)	200	$50 \; \forall \; t$	$\phi_t(w) = A_t w$ where $A_t \in \mathbb{R}^{50 \times 200}$

Table 4.1: **Description and parametrization of transforms.** $\phi_t : \mathbb{R}^d \rightarrow \mathbb{R}^{d_t}$ is a transform for co-occurrence type $t \in \mathcal{T}$. *select* corresponds to approach (iii) in Fig. 4.2 that concatenates separately trained d_t dimensional embeddings.

the symmetry in the objective, both vectors should ideally be identical. We did not observe a significant change in performance when using separate word and context vectors.

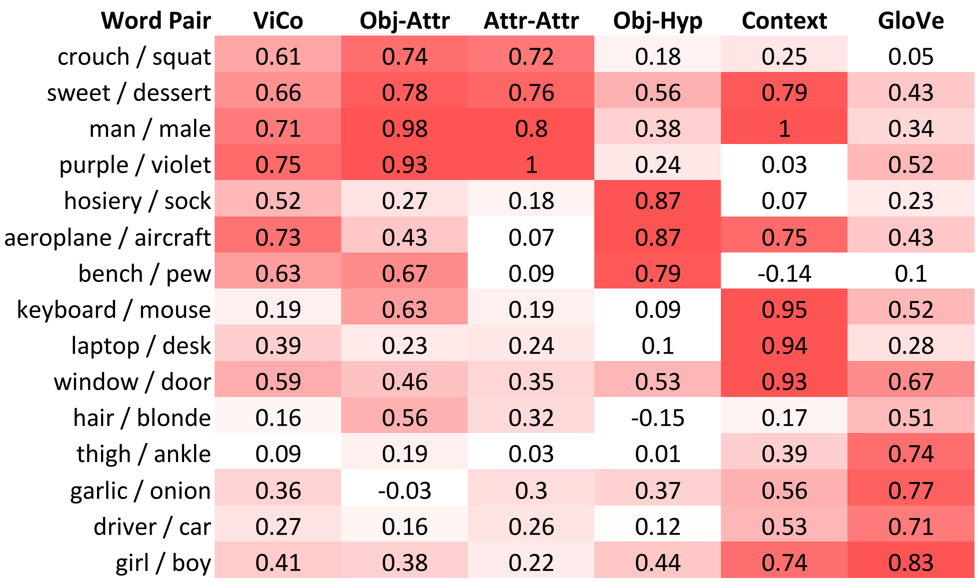
4.3.2 Multi-task Log-bilinear Model

We now extend the log-bilinear model described above to jointly learn embeddings from multiple co-occurrence count matrices X^t, where $t \in \mathcal{T}$ refers to a type from the set of types $\mathcal{T}$. Also let $\mathcal{N}_t$ and $\mathcal{Z}_t$ be the list of word pairs with non-zero and zero co-occurrences of type t respectively. We learn ViCo embeddings $w_i \in \mathbb{R}^d$ for all words i by minimizing the following loss function

$$\sum_{t \in \mathcal{T}} \sum_{(i,j) \in \mathcal{N}_t} (\phi_t(w_i)^T \phi_t(w_j) + b^t_i + b^t_j - \log X^t_{ij})^2 + \sum_{t \in \mathcal{T}} \sum_{(i',j') \in \mathcal{Z}_t} \max(0, \phi_t(w_{i'})^T \phi_t(w_{j'}) + b^t_{i'} + b^t_{j'}). \quad (4.2)$$

Here $\phi_t : \mathbb{R}^d \rightarrow \mathbb{R}^{d_t}$ is a co-occurrence type-specific transformation function that maps ViCo embeddings to a type-specialized embedding space. b^t_i is a learned bias term for word i and type t. We set function $f(X)$ in Eq. (4.1) to the constant 1 for all X. Next, we discuss the transformations ϕ_t, benefits of capturing different types of co-occurrences, use of the second term in Eq. (4.2), and training details. Fig. 4.2 illustrates (i) GloVe and versions of our model (ii,iii).

Transformations ϕ_t. To understand the role of the transformations ϕ_t in learning from multiple co-occurrence matrices, consider the naïve approach of concatenating $|\mathcal{T}|$ d_t-dimensional



Word Pair	ViCo	Obj-Attr	Attr-Attr	Obj-Hyp	Context	GloVe
crouch / squat	0.61	0.74	0.72	0.18	0.25	0.05
sweet / dessert	0.66	0.78	0.76	0.56	0.79	0.43
man / male	0.71	0.98	0.8	0.38	1	0.34
purple / violet	0.75	0.93	1	0.24	0.03	0.52
hosiery / sock	0.52	0.27	0.18	0.87	0.07	0.23
aeroplane / aircraft	0.73	0.43	0.07	0.87	0.75	0.43
bench / pew	0.63	0.67	0.09	0.79	-0.14	0.1
keyboard / mouse	0.19	0.63	0.19	0.09	0.95	0.52
laptop / desk	0.39	0.23	0.24	0.1	0.94	0.28
window / door	0.59	0.46	0.35	0.53	0.93	0.67
hair / blonde	0.16	0.56	0.32	-0.15	0.17	0.51
thigh / ankle	0.09	0.19	0.03	0.01	0.39	0.74
garlic / onion	0.36	-0.03	0.3	0.37	0.56	0.77
driver / car	0.27	0.16	0.26	0.12	0.53	0.71
girl / boy	0.41	0.38	0.22	0.44	0.74	0.83

Figure 4.3: **Rich sense of relatedness through multiple co-occurrences.** Different notions of word relatedness exist but current word embeddings do not provide a way to disentangle those. Since ViCo is learned from multiple types of co-occurrences with dedicated embedding spaces for each (obtained through transformations ϕ_t), it can provide a richer sense of relatedness. The figure shows cosine similarities computed in GloVe, ViCo(linear) and embedding spaces dedicated to different co-occurrence types (components of ViCo(select)). For example, 'hosiery' and 'sock' are related through an object-hypernym relation but not related through object-attribute or a contextual relation. 'laptop' and 'desk' on the other hand are related through context.

word embeddings learned separately for each type t using Eq. (4.1). Such an approach would yield an embedding with $d \geq |\mathcal{T}| \min_t d_t$ dimensions. For instance, 4 co-occurrence types, each producing embeddings of size $d_t = 50$, leads to $d = 200$ dimensional final embeddings. Thus, a natural question arises – *Is it possible to learn a more compact representation by utilizing the correlations between different co-occurrence types?*

Eq. (4.2) is a multi-task learning formulation where learning from each type of co-occurrence constitutes a different task. Hence, ϕ_t is equivalent to a task-specific head that projects the shared word embedding $w \in \mathbb{R}^d$ to a type-specialized embedding space $\phi_t(w) \in \mathbb{R}^{d_t}$. A log-bilinear model equivalent to Eq. (4.1) is then applied for each co-occurrence type in the corresponding specialized embedding space. We learn the embeddings w and parameters of ϕ_t simultaneously for all t in an end-to-end manner.

With this multi-task formulation the dimensions of w can be chosen independently of $|\mathcal{T}|$ or d_t. Also note that the new formulation encompasses the naïve approach which is implemented in this framework by setting $d = \sum_t d_t$, and ϕ_t as a slicing operation that 'selects' d_t non-overlapping indices allocated for type t. In our experiments, we evaluate this naïve

	Obj-Attr	**Attr-Attr**	**Obj-Hyp**	**Context**	**Overall**
Unique Words	15,548	11,893	11,981	25,451	35,476
Non-zero entries (in millions)	1.37	1.37	0.61	8.12	11.48

Table 4.2: **Co-occurrence statistics** showing the number of words and millions of non-zero entries in each co-occurrence matrix. For reference, GloVe uses a vocabulary of 400,000 words with 8-40 billion non-zero entries.

approach and refer to it as the *select* transformation. We also assess *linear* transformations of different dimensions as described in Tab. 4.1. We find that 100 dimensional ViCo embeddings learned with *linear* transform achieve the best performance *vs*. compactness trade-off.

Role of max **term.** Optimizing only the first term given in Eq. (4.2) can lead to accidentally embedding a word pair from $\mathcal{Z}_t$ (zero co-occurrences) close together (high dot product). To suppress such spurious similarities, we include the max term which encourages all word pairs $(i', j') \in \mathcal{Z}_t$ to have a small predicted log co-occurrence

$$\log \tilde{X}^t_{i'j'} = \phi_t(w_{i'})^T \phi_t(w_{j'}) + b^t_{i'} + b^t_{j'}. \tag{4.3}$$

In particular, the second term in the objective linearly penalizes positive predicted log co-occurences of word-pairs that do not co-occur.

Training details. Pennington *et al.* [27] report Adagrad to work best for GloVe. We found that Adam leads to faster initial convergence. However, fine-tuning with Adagrad further decreases the loss. For both optimizers, we use a learning rate of 0.01, a batch size of 1000 word pairs sampled from $\mathcal{N}_t$ and $\mathcal{Z}_t$ each for all t, and no weight decay.

Multiple notions of relatedness. Learning from multiple co-occurrence types leads to a richer sense of relatedness between words. Fig. 4.3 shows that the relationship between two words may be better understood through similarities in multiple embedding spaces than just one. For example, 'window' and 'door' are related because they occur in context in scenes, 'hair' and 'blonde' are related through an object-attribute relation, 'crouch' and 'squat' are related because both attributes apply to similar objects, *etc*.

4.3.3 Computing Visual Co-occurrence Counts

To learn meaningful word embeddings from visual co-occurrences, reliable co-occurrence count estimates are crucial. We use Visual Genome and ImageNet for estimating visual co-occurrence counts. Specifically, we use object and attribute *synset* (set of words with the same meaning) annotations in VisualGenome to get *Object-Attribute* (oa), *Attribute-Attribute* (aa), and *Context* (c) co-occurrence counts. ImageNet *synsets* and their ancestors in WordNet are used to compute *Object-Hypernym* (oh) counts. Tab. 4.2 shows the number of unique words and non-zero entries in each co-occurrence matrix.

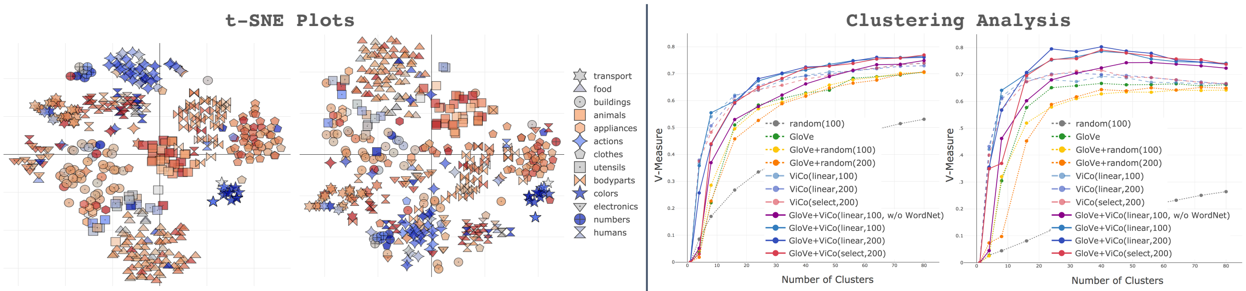
Let $\mathcal{T} = \{oa, aa, c, oh\}$ denote the set of four co-occurrence types and X_{ij}^{t} denote the number of co-occurrences of type $t \in \mathcal{T}$ between words i and j. We denote a *synset* and its associated set of words as $\mathcal{S}$. All co-occurrences are initialized to 0. We now describe how each co-occurrence matrix X^t is computed.

- Let $\mathcal{O}$ and $\mathcal{A}$ be the sets of object and attribute synsets annotated for an image region. For each region in VisualGenome, we increment X_{ij}^{oa} by 1, for each word pair $(i, j) \in \mathcal{S}_o \times \mathcal{S}_a$, and for all *synset* pairs $(\mathcal{S}_o, \mathcal{S}_a) \in \mathcal{O} \times \mathcal{A}$. X_{ji}^{oa} is also incremented unless $i = j$.
- For each region in VisualGenome, we increment X_{ij}^{aa} by 1, for each word pair $(i, j) \in \mathcal{S}_{a_1} \times \mathcal{S}_{a_2}$, and for all *synset* pairs $(\mathcal{S}_{a_1}, \mathcal{S}_{a_2}) \in \mathcal{A} \times \mathcal{A}$.
- Let $\mathcal{C}$ be the union of all object *synsets* annotated in an image. For each image in VisualGenome, X_{ij}^{c} is incremented by 1, for each word pair $(i, j) \in \mathcal{S}_{c_1} \times \mathcal{S}_{c_2}$, and for all *synset* pairs $(\mathcal{S}_{c_1}, \mathcal{S}_{c_2}) \in \mathcal{C} \times \mathcal{C}$.
- Let $\mathcal{H}$ be a set of object synsets annotated for an image in ImageNet and its ancestors in WordNet. For each each image in ImageNet, X_{ij}^{oh} is incremented by 1, for each word pair $(i, j) \in \mathcal{S}_{h_1} \times \mathcal{S}_{h_2}$, and for all *synset* pairs $(\mathcal{S}_{h_1}, \mathcal{S}_{h_2}) \in \mathcal{H} \times \mathcal{H}$.

4.4 EXPERIMENTS

We analyze ViCo embeddings with respect to the following properties: (1) Does unsupervised clustering result in a natural grouping of words by visual concepts? (Sec. 4.4.1); (2) Do the word embeddings enable transfer of visual learning (*e.g.*, visual recognition) to classes not seen during training? (Sec. 4.4.2); (3) How well do the embeddings perform on downstream applications? (Sec. 4.4.3); (4) Does the embedding space show word arithmetic properties ($land - car + aeroplane = sky$)? (Sec. 4.4.4).

Data for clustering analysis. To answer (1) we manually annotate 495 frequent words in VisualGenome with 13 coarse (see legend in the t-SNE plots in Fig. 4.4) and 65 fine categories (see appendix for the list of categories).



(a) GloVe+ViCo(linear) (b) GloVe (c) Fine Categories (d) Coarse Categories

Figure 4.4: **Unsupervised Clustering Analysis.** (a,b) **Qualitative evaluation with t-SNE:** Plots show that ViCo augmented GloVe results in tighter, more homogenous clusters than GloVe. Marker shape encodes the annotated coarse category and color denotes if the word is used more frequently as an object or an attribute; (c,d) **Quantitative evaluation:** Plots show clustering performance of different embeddings measured through V-Measure at different number of clusters. All ViCo based embeddings outperform GloVe for both fine and coarse annotations (Sec. 4.4.1). See Tab. 4.3 and Tab. 4.4 for average performance across cluster numbers. Best viewed in color on a screen.

Data for zero-shot-like analysis. To answer (2), we use CIFAR-100 [123]. We generate 4 splits of the 100 categories into disjoint Seen (categories used for training visual classifiers) and Unseen (categories used for evaluation) sets. We use the following scheme for splitting: The list of 5 sub-categories in each of the 20 coarse categories (provided by CIFAR) is sorted alphabetically and the first k categories are added to Seen and the remaining to Unseen for $k \in \{1, 2, 3, 4\}$.

4.4.1 Unsupervised Clustering Analysis

The main benefit of word vectors over one-hot or random vectors is the meaningful structure captured in the embedding space: words that are closer in the embedding space are semantically similar. We hypothesize that ViCo represents similarities and differences between visual categories that are missing from GloVe.

Qualitative evidence to support this hypothesis can be found in t-SNE plots shown in Fig. 4.4, where concatenation of GloVe and ViCo embeddings leads to tighter, more homogenous clusters of the 13 coarse categories than GloVe.

To test the hypothesis quantitatively, we cluster word embeddings with agglomerative clustering (cosine affinity and average linkage) and compare to the coarse and fine ground truth annotations using *V-Measure* which is the harmonic mean of *Homogeneity* and *Completeness* scores. *Homogeneity* is a measure of cluster purity, assessing whether all points in the same cluster have the same ground truth label. *Completeness* measures whether all points with the same label belong to the same cluster

Plots (c,d) in Fig. 4.4 compare random vectors, GloVe, variants of ViCo and their combinations (concatenation) for different number of clusters using V-Measure. Average performance across different cluster numbers is shown in Tab. 4.3 and Tab. 4.4. The main conclusions are as follows:

ViCo clusters better than other embeddings. Tab. 4.3 shows that *ViCo* alone outperforms *GloVe*, *random*, and *vis-w2v* based embeddings. *GloVe+ViCo* improves performance further, especially for coarse categories.

WordNet is not the sole contributor to strong performance of ViCo. To verify that ViCo's gains are not simply due to the hierarchical nature of WordNet, we evaluate a version of ViCo trained on co-occurrences computed without using WordNet, *i.e.*, using raw *word* annotations in VisualGenome instead of *synset* annotations and without Object-Hypernym co-occurrences. Tab. 4.3 shows that *GloVe+ViCo(linear,100,w/o WordNet)* outperforms *GloVe* for both coarse and fine categories on both metrics. However, *GloVe+ViCo(linear,100)* does see healthy gains over *GloVe+ViCo(linear,100,w/o WordNet).*

ViCo outperforms existing visual word embeddings. Tab. 4.3 evaluates performance of existing visual word embeddings which are learned from abstract scenes [120]. *wiki* and *coco* are different versions of *vis-w2v* depending on the dataset (Wikipedia or MS-COCO [124, 125]) used for training word2vec for initialization. After initialization, both models are trained on an abstract scenes (clipart images) dataset [126]. *ViCo(linear,100)* outperforms both of these embeddings. *GloVe+vis-w2v-wiki* performs similarly to *GloVe* and *GloVe+vis-w2v-wiki-coco* performs only slightly better than *GloVe*, showing that the majority of the information captured by *vis-w2v* may already be present in *GloVe.*

Learned embeddings significantly outperform random vectors. Tab. 4.3 shows that random vectors perform poorly in comparison to learned embeddings. *GloVe+random* performs similarly to *GloVe* or worse. This implies that gains of *GloVe+ViCo* over *GloVe* are not just an artifact of increased dimensionality.

***Linear* achieves similar performance as *Select* with fewer dimensions.** Tab. **??** illustrates the ability of the multi-task formulation to learn a more compact representatio than *select* (concatenating embeddings learned from each co-occurrence type separately) without sacrificing performance. 50, 100, and 200 dimensional ViCo embeddings learned with linear transformations, all achieve performance similar to *select.*

Embeddings	**Dim.**	**Fine**	**Coarse**
random(100)	100	0.34	0.15
GloVe	300	0.50	0.52
GloVe+random(100)	300+100	0.50	0.49
vis-w2v-wiki [120]	200	0.41	0.43
vis-w2v-coco [120]	200	0.45	0.4
GloVe+vis-w2v-wiki	300+200	0.5	0.52
GloVe+vis-w2v-coco	300+200	0.52	0.55
ViCo(linear,100)	100	**0.60**	**0.59**
GloVe+ViCo(linear,100)	300+100	**0.61**	**0.65**
GloVe+ViCo(linear,100, w/o WN)	300+100	0.54	0.58

Table 4.3: **Comparing ViCo to other embeddings.** All ViCo based embeddings outperform GloVe and random vectors. *ViCo(linear,100)* also outperforms *vis-w2v*. *GloVe+vis-w2v* performs similarly to *GloVe* while *GloVe+ViCo* outperforms both *GloVe* and ViCo. Using WordNet yields healthy performance gains but is not the only contributor to performance since *GloVe+ViCo(linear,100, w/o WN)* also outperforms *GloVe*. **Best** and **second best** numbers are highlighted in each column.

4.4.2 Zero-Shot-like Analysis

The ability of word embeddings to capture relations between visual categories enables visual models trained on limited visual categories to generalize to larger sets unseen during training. To assess this ability, we evaluate embeddings on their zero-shot-like object classification performance using the CIFAR-100 dataset. Note that our *zero-shot-like* setup is slightly different from a typical zero-shot setup because even though the visual classifier is not trained on unseen class images in CIFAR, annotations associated with images of unseen categories in VisualGenome or ImageNet may be used to compute word co-occurrences while learning word embeddings.

Model. Let $f(I) \in \mathbb{R}^n$ be the features extracted from image I using a CNN and let $w_c \in \mathbb{R}^m$ denote the word embedding for class $c \in \mathcal{C}$. Let $g : \mathbb{R}^m \rightarrow \mathbb{R}^n$ denote a function that projects word embeddings into the space of image features. We define the score $s_c(I)$ for class c as $\text{cosine}(f(I), g(w_c))$,

where $\text{cosine}(\cdot)$ is the cosine similarity. The class probabilities are defined as

$$p_c(I) = \frac{\exp(s_c(I)/\epsilon)}{\sum_{c' \in \mathcal{C}} \exp(s_{c'}(I)/\epsilon)}, \tag{4.4}$$

where ϵ is a learnable temperature parameter. In our experiments, $f(I)$ is a 64-dimensional feature vector produced by the last linear layer of a 34-layer ResNet (modified to accept

Embeddings	Dim.	Fine	Coarse
ViCo(linear,50)	50	0.57	0.56
ViCo(linear,100)	100	**0.60**	0.59
ViCo(linear,200)	200	0.59	0.60
ViCo(select,200)	200	0.59	0.60
GloVe	300	0.50	0.52
GloVe+ViCo(linear,50)	300+50	0.60	**0.66**
GloVe+ViCo(linear,100)	300+100	**0.61**	**0.65**
GloVe+ViCo(linear,200)	300+200	**0.60**	**0.65**
GloVe+ViCo(select,200)	300+200	0.57	0.63

Table 4.4: **Effect of transformations on clustering performance.** The table compares average performance across number of clusters. The *linear* variants achieve performance similar to *select* with fewer dimensions. In fact, when used in combination with GloVe, *linear* variants outperform *select*. **Best** and **second best** numbers are highlighted in each column.

32×32 CIFAR images) and g is a linear transformation.

Learning. The model (parameters of f, g, and ϵ) is trained on images from the set of seen classes $\mathcal{S} \subset \mathcal{C}$. We use the Adam [110] optimizer with a learning rate of 0.01. The model is trained with a batch size of 0.01 for 50 epochs.

Model Selection and Evaluation. The best model (among iteration checkpoints) is selected based on seen class accuracy (classifying only among classes in $\mathcal{S}$) on the test set. The selected model is evaluated on unseen category ($\mathcal{U} = \mathcal{C} \setminus \mathcal{S}$) prediction accuracy computed on the test set.

Fig. 4.5 compares chance performance ($1/|\mathcal{U}|$), random vectors, *GloVe*, and *GloVe+ViCo* on four seen/unseen splits. We show mean and standard deviation computed across four runs ($7 \times 4 \times 4 = 112$ models trained in all). The key conclusions are as follows:

ViCo generalizes to unseen classes better than GloVe. ViCo based embeddings, especially 200-dim. select and linear variants show healthy gains over *GloVe*. Note that this is not just due to higher dimensions of the embeddings since *GloVe+random(200)* performs worse than *GloVe*.

Learned embeddings significantly outperform random vectors. Random vectors alone achieve close to chance performance, while concatenating random vectors to *GloVe* degrades performance.

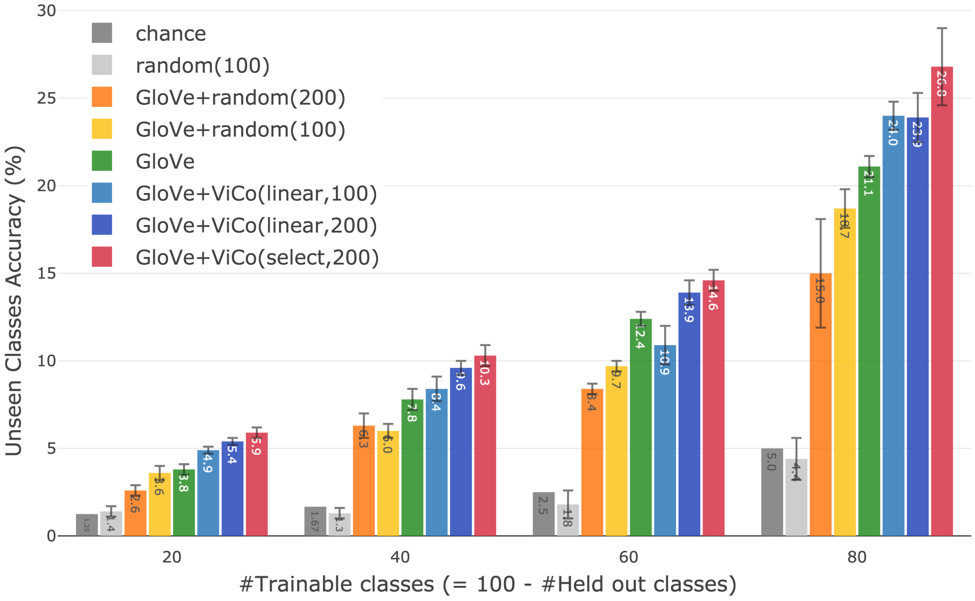


Figure 4.5: **Zero-Shot Analysis.** The histogram compares the transfer learning ability of a simple word embedding based object classification model. The x-axis denotes the number of CIFAR-100 classes (m) used during training. During test, we evaluate the classifier on its ability to correctly classify among the remaining $(100 - m)$ unseen classes. Results show that *GloVe+ViCo* leads to better transfer to unseen classes than GloVe alone (Sec. 4.4.2).

Select performs better than Linear. Compression to 100-dimensional embeddings using linear transformation shows a more noticeable drop in performance as compared to the *select* setting. However, *GloVe+ViCo(linear,100)* still outperforms *GloVe* in 3 out of 4 splits.

We compare *random* (chance performance), *GloVe*, *GloVe+ViCo(linear)*, and *GloVe+ViCo(select)* in Fig. 4.5. *GloVe+ViCo* variants yield significant performance gains over *GloVe*, and *select* consistently outperforms *linear* across all 4 seen-unseen splits. As expected, learned embeddings (GloVe or ViCo based) perform significantly better than chance performance.

4.4.3 Downstream Task Evaluation

We now evaluate ViCo embeddings on a range of downstream tasks. Generally, we expect tasks requiring better word representations of objects and attributes to benefit from our embeddings. When using existing models, we initialize and freeze word embeddings so that performance changes are not due to fine-tuning embeddings of different dimensions. The rest of the model is left untouched except for the dimensions of the input layer where the size of the input features needs to match the embedding dimension.

		Discr. Attr.	Im-Cap Retr.		VQA				Ref. Exp.			Image Captioning			
		Avg. F1	Recall@1		Accuracy				Loc. Accuracy			Captioning Metrics			
Embeddings	**Dim.**	$m \pm \sigma$	Im2Cap	Cap2Im	Overall	Y/N	Num.	Other	Val	TestA	TestB	B1	B4	C	S
random	300	50.03 ± 2.26	43.1	30.6	66.1	82.0	44.8	57.5	71.3	73.5	66.3	**0.71**	**0.30**	**0.91**	**0.17**
GloVe	300	63.85 ± 0.04	**44.8**	33.5	**67.5**	83.8	**46.5**	**58.3**	72.2	**75.3**	66.8	0.71	0.29	0.89	0.17
GloVe+random	300+100	**63.88 ± 0.03**	44.3	**34.4**	**67.5**	**84.1**	45.9	58.2	**72.5**	75.1	**67.5**	0.71	0.29	0.88	0.17
GloVe+ViCo(linear)	300+100	**64.46 ± 0.17**	**46.3**	**34.2**	**67.7**	**84.4**	**46.6**	**58.4**	**72.7**	**75.5**	**67.5**	**0.71**	**0.29**	**0.89**	**0.17**

Table 4.5: **Comparing ViCo to GloVe and random vectors.** *GloVe+ViCo(linear)* outperforms *GloVe* and *GloVe+random* for all tasks and outperforms *random* for all tasks except Image Captioning. While random vectors perform close to chance on the **word-only** task, they compete tightly with learned embeddings on **vision-language** tasks. This suggests that vision-language models are relatively insensitive to the choice of word embeddings. **Best** and **second best** numbers in each column are highlighted.

Tab. 4.5 compares performance of embeddings on a word-only discriminative attributes task and 4 vision-language tasks. On all tasks *GloVe+ViCo* outpeforms *GloVe* and *GloVe+random*. Unlike the word-only task which depends solely on word representations, vision-language tasks are less sensitive to word embeddings, with performance of random embeddings approaching learned embeddings.

Discriminative Attributes [127] is one of the SemEval 2018 challenges. The task requires to identify whether an attribute word discriminates between two concept words. For example, the word "red" is a discriminative attribute for word pair ("apple", "banana") but not for ("apple", "cherry"). Samples are presented as tuples of attribute and concept words and the model makes a binary prediction. Performance is evaluated using class averaged F1 scores.

Let w_1, w_2, and a be the word embeddings (GloVe or ViCo) for the two concept words and the attribute word. We compute the scores s_g and s_v for GloVe and ViCo using function $s(a, w_1, w_2) = \text{cosine}(a, w_1) - \text{cosine}(a, w_2)$, where cosine$(\cdot)$ is the cosine similarity. We then learn a linear SVM over s_g for the *GloVe* only model and over s_g and s_v for the *GloVe+ViCo* model.

Caption-Image Retrieval is a classic vision-language task requiring a model to retrieve images given a caption or vice versa. We use the open source VSE++ [30] implementation which learns a joint embedding of images and captions using a *Max of Hinges* loss that encourages attending to hard negatives and is geared towards improving top-1 Recall. We evaluate the model using Recall@1 on MS-COCO.

Visual Question Answering [8, 128] systems are required to answer questions about an image. We compare the performance of embeddings using Pythia [17, 129] which uses bottom-up top-down attention for computing a question-relevant image representation. Image features are then fused with a question representation using a GRU operating on word embeddings and fed into an answer classifier. Performance is evaluated using overall and by-question-type accuracy on the test-dev split of the VQA v2.0 dataset.

Referring Expression Comprehension consists of localizing an image region based on a natural language description. We use the open source implementation of MAttNet [130] to compare localization accuracy with different embeddings on the RefCOCO+ dataset using the UNC split. MAttNet uses an attention mechanism to parse the referring expression into phrases that inform the subject's appearance, location, and relationship to other objects. These phrases are processed by corresponding specialized localization modules. The final region scores are

Image Captioning involves generating a caption given an image. We use the Show and Tell model of Vinyals *et al*. [131] which feeds CNN extracted image features into an LSTM followed by beam search to sample captions. We report BLEU1 (B1), BLEU4 (B4), CIDEr (C), and SPICE (S) metrics [132, 133, 134] on the MS-COCO test set.

4.4.3.1 Why are random vectors competitive with learned embeddings?

Tab. 4.5 shows that while *GloVe+ViCo* outperforms *GloVe* and *GloVe+random*, Random vectors are surprisingly competitive with learned embeddings (both GloVe and ViCo) on vision-language tasks. Below, we present a hypothesis for this behavior and test the hypothesis on image to caption retrieval task.

Hypothesis: *Given enough data, vision-language models learn to transform random vectors to get useful intermediate word representations.*

Test: Fig. 4.4.3.1 shows the performance of random and learned embeddings when trained on different amounts of training data. We see that learned embeddings have a significant advantage over random ones when the model is trained with only 1-2% of the available training data but diminishing gains (green line) are observed with more data.

Reason for limited improvement of ViCo over Random and GloVe on VQA and Captioning. Because of the above hypothesis and availability of sufficient training data

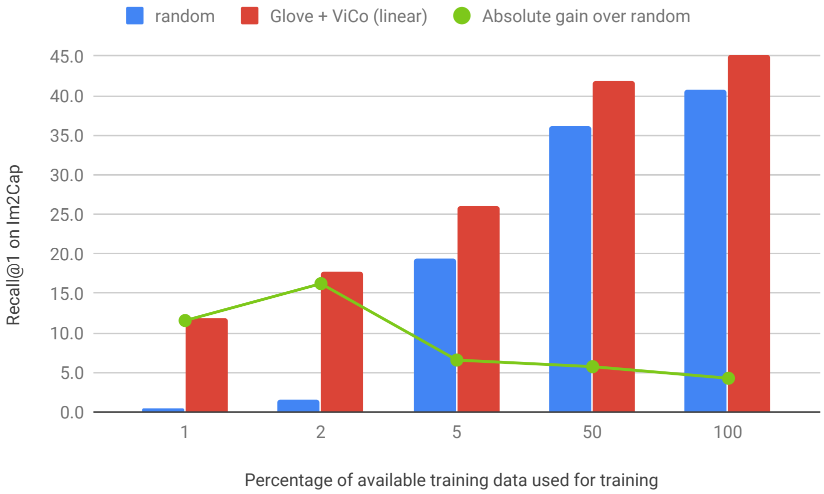


Figure 4.6: Comparing random and learned embeddings for Im2Cap model trained with varying amounts of data. We report average recall across 3 runs because of variance observed during training.

for tasks like VQA and Image Captioning, gains due to learned embeddings (for both GloVe and ViCo) are relatively small in comparison to random vectors.

However, we want to emphasize that our clustering, partitioning, and zero-shot analysis, as well as the discriminative attributes task highlight the advantages of learned embeddings over random embeddings, and ViCo over existing word embeddings. Finally, the ability to represent multiple senses of relatedness (Fig. 3 in the main submission) also distinguishes ViCo from existing word embeddings.

4.4.4 Exploring Embedding Space Structure

Previous work [118] has demonstrated linguistic regularities in word embedding spaces through analogy tasks solved using simple vector arithmetics. Fig. 4.6 shows qualitatively that ViCo embeddings possess similar properties, capturing relations between visual concepts well.

4.5 CONCLUSION

This work shows that in addition to textual co-occurrences, visual co-occurrences are a surprisingly effective source of information for learning word representations. The resulting embeddings outperform text-only embeddings on unsupervised clustering, zero-shot gener-

Analogy	Answer Candidates	GloVe	ViCo
car:land::aeroplane:?	ocean, sky, road, railway	ocean	**sky**
clock:circle::tv:?	triangle, square, octagon, round	triangle	**square**
park:bench::church:?	door, sofa, cabinet, pew	door	**pew**
sheep:fur::person:?	hair, horn, coat, tail	coat	**hair**
monkey:zoo::cat:?	park, house, church, forest	park	**house**
leg:trouser::wrist:?	watch, shoe, tie, bandana	bandana	**watch**
yellow:banana::red:?	strawberry, lemon, mango, orange	mango	**strawberry**
rice:white::spinach:?	blue, green, red, yellow	blue	**green**
train:railway::car:?	land, desert, ocean, sky	land	**land**
can:metallic::bottle:?	wood, glass, cloth, paper	glass	**glass**
man:king::woman:?	queen, girl, female, adult	**queen**	girl
can:metallic::bottle:?	wood, plastic, cloth, paper	**plastic**	wood
train:railway::car:?	road, desert, ocean, sky	**road**	ocean

Table 4.6: **Answering Analogy Questions.** Out of 30 analogy pairings tested, we found both GloVe and ViCo to be correct 19 times, only ViCo was correct 8 times, and only Glove was correct 3 times. Correct answers are **highlighted**.

alization, and various supervised downstream tasks. We also develop a multi-task extension of *GloVe*'s log-bilinear model to learn a compact shared embedding from multiple types of co-occurrences. Type-specific embedding spaces learned as part of the model help provide a richer sense of relatedness between words.

CHAPTER 5: CONTRASTIVE LEARNING FOR WEAKLY SUPERVISED PHRASE GROUNDING

5.1 INTRODUCTION

Humans can learn from captioned images because of their ability to associate words to image regions. For instance, humans perform such word-region associations while acquiring facts from news photos, making a diagnosis from MRI scans and radiologist reports, or enjoying a movie with subtitles. This word-region association problem is called word or phrase *grounding* and is a crucial capability needed for downstream applications like visual question answering, image captioning, and text-image retrieval.

Existing object detectors can detect and represent object regions in an image, and language models can provide contextualized representations for noun phrases in the caption. However, learning a mapping between these continuous, independently trained visual and textual representations is challenging in the absence of explicit region-word annotations. We focus on learning this mapping from weak supervision in the form of paired image-caption data without requiring laborious grounding annotations.

Current state-of-the-art approaches [135, 136, 137] formulate weakly supervised phrase grounding as a multiple instance learning (MIL) problem [138, 139]. The image can be viewed as a bag of regions. For a given phrase, all images with captions containing the phrase are treated as positive bags while remaining images are treated as negatives. Models aggregate per region features or phrase scores to construct image-level predictions that can be supervised with image-level labels in the form of phrases or captions. Common aggregation approaches include max or mean pooling, noisy-OR [140], and attention [135, 139]. Popular training objectives include binary classification loss [140] (whether the image contain the phrase) or caption reconstruction loss [137] (generalization of binary classification to caption prediction) or ranking objectives [136, 135] (do true image-caption or image-phrase pairs score higher than negative pairs).

Fig. 5.1 provides an overview of our proposed contrastive training. We propose a novel formulation of the weakly supervised phrase grounding problem as that of maximizing a lower bound on mutual information between set of region features extracted from an image and contextualized word representations. We use pretrained region and word representations from an object detector and a language model and perform optimization over parameters of word-region attention instead of optimizing the region and word representations themselves. Intuitively, to compute mutual information with a word's representation, attention must discard nuisance regions in the word-conditional attended visual representation, thereby

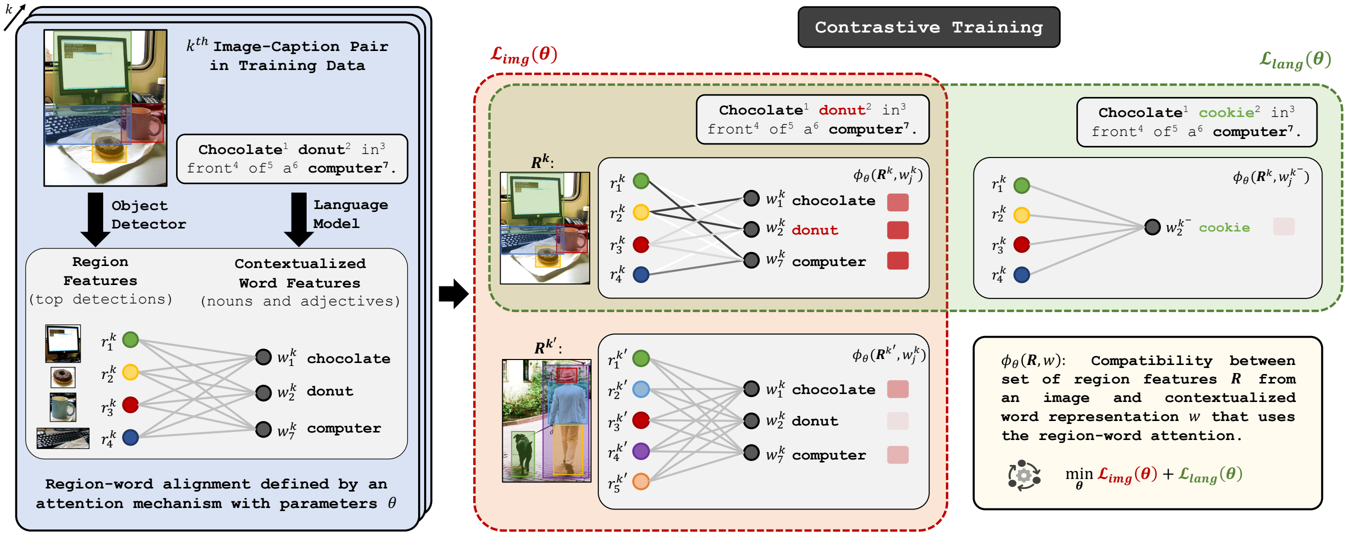

k
k^{th} Image-Caption Pair in Training Data
Chocolate^1 donut^2 in^3 front^4 of^5 a^6 computer^7.
Object Detector
Language Model
Region Features (top detections)
Contextualized Word Features (nouns and adjectives)
r_1^k
r_2^k
r_3^k
r_4^k
w_1^k chocolate
w_2^k donut
w_7^k computer
Region-word alignment defined by an attention mechanism with parameters θ
Contrastive Training
$\mathcal{L}_{img}(\theta)$
$\mathcal{L}_{lang}(\theta)$
Chocolate^1 donut^2 in^3 front^4 of^5 a^6 computer^7.
R^k:
$\phi_\theta(R^k, w_j^k)$
r_1^k
r_2^k
r_3^k
r_4^k
w_1^k chocolate
w_2^k donut
w_7^k computer
Chocolate^1 cookie^2 in^3 front^4 of^5 a^6 computer^7.
$\phi_\theta(R^k, w_j^{k^-})$
r_1^k
r_2^k
r_3^k
r_4^k
$w_2^{k^-}$ cookie
$R^{k'}$:
$\phi_\theta(R^{k'}, w_j^k)$
$r_1^{k'}$
$r_2^{k'}$
$r_3^{k'}$
$r_4^{k'}$
$r_5^{k'}$
w_1^k chocolate
w_2^k donut
w_7^k computer
$\phi_\theta(R, w)$: Compatibility between set of region features R from an image and contextualized word representation w that uses the region-word attention.
$\min_\theta \mathcal{L}_{img}(\theta) + \mathcal{L}_{lang}(\theta)$

Figure 5.1: **Overview of our contrastive learning framework.** We begin by extracting region and word features using an object detector and a language model respectively. Contrastive learning trains a word-region attention mechanism as part of a compatibility function ϕ_θ between the set of region features from an image and individual contextualized word representations. The compatibility function is trained to maximize a lower bound on mutual information with two losses. For a given caption word, $\mathcal{L}_{\texttt{img}}$ learns to produce a higher compatibility for the true image than a negative image in the mini-batch. $\mathcal{L}_{\texttt{lang}}$ learns to produce a higher compatibility of an image with a true caption-word than with a word in a negative caption. We construct negative captions by substituting a noun word like "donut" in the true caption with contextually plausible but untrue words like "cookie" using a language model.

selecting regions that match the word. For any given word, the learned attention thus functions as a soft selection or grounding mechanism over regions.

Since computing MI is intractable, we maximize the recently introduced InfoNCE lower bound [141] on mutual information. The InfoNCE bound requires a compatibility score between each caption word and the image to contrast positive image and caption word pairs with negative pairs in a minibatch. We use two objectives. The first objective ($\mathcal{L}_{\texttt{img}}$ in Fig. 5.1) contrasts a positive pair with negative pairs with the same caption word but different image regions. The second objective ($\mathcal{L}_{\texttt{lang}}$ in Fig. 5.1) contrasts a positive pair with negative pairs with the same image but different captions. We show empirically that sampling negative captions randomly from the training data to optimize $\mathcal{L}_{\texttt{lang}}$ does not yield any gains over optimizing $\mathcal{L}_{\texttt{img}}$ only. Instead of random sampling, we propose to use a language model to construct context-preserving negative captions by substituting a single noun word in the caption.

We design the compatibility function using a `query-key-value` attention mechanism. The `queries` and `keys`, computed from words and regions respectively, are used to compute a word-specific attention over each region which acts as a soft alignment or grounding between words and regions. The compatibility score between regions and word is computed by comparing attended visual representation and the word representation.

Our key contributions are: (i) a novel MI based contrastive training framework for weakly supervised phrase grounding; (ii) an InfoNCE compatibility function between a set of regions and a caption word designed for phrase grounding; and (iii) a procedure for constructing context-preserving negative captions that provides $\approx 10\%$ absolute gain in grounding performance.

5.1.1 Related Work

Our work is closely related to three active areas of research. We now provide an overview of prior arts in each.

Weakly Supervised Phrase Grounding. Weakly supervised phrase localization is typically posed as a multiple instance learning (MIL) problem [138, 139] where each image is considered as a bag of region proposals. Images whose captions mention a word or a phrase are treated as positive bags while rest of the images are treated as negatives for that word or phrase. Features or scores for a phrase or the entire caption are aggregated across all regions to make a prediction for the image. Common methods of aggregation are max or average pooling, noisy-OR [140], or attention [137, 139]. With the ability to produce image-level scores for pairs of images and phrases or captions, the problem becomes an image-level fully-supervised phrase classification problem [140] or an image-caption retrieval problem [136, 135]. An alternatives to the MIL formulations is the approach of Ye *et al.* [142] which uses statistical hypothesis testing approach to link concepts detected in an image and words mentioned in the sentence. While all the above approaches assume paired image-caption data, Wang *et al.* [143] recently address the problem of phrase grounding without access to image-caption pairs. Instead they assume access to a set of scene and color classifiers, and object detectors to detect concepts in the scene and use word2vec [144] similarity between concept labels and caption words to achieve grounding.

MI-based Representation Learning. Recently MI-based approaches have shown promising results on a variety representation learning problems. Computing the MI between two representations is challenging as we often have access to samples but not the underlying joint distribution that generated the samples. Thus, recent efforts rely on variational estimation of MI [145, 146, 147, 141]. An overview of such estimators is discussed in [148, 149] while the statistical limitations are reviewed in [150, 151].

In practice, MI-based representation learning models are often trained by maximizing an estimation of MI across different *transformations* of data. For example, deep InfoMax [19] maximizes MI between local and global representation using MINE [147]. Contrastive predictive coding [141, 152] inspired by noise contrastive estimation [153, 154] assumes an order in the features extracted from an image and uses summary features to predict future features. Contrastive multiview coding [155] maximizes MI between different color channels or data modalities while augmented multiscale Deep InfoMax [20] and SimCLR [23] extract views using different augmentations of data points. Since the infoNCE loss is limited by the batch size, several previous work rely on memory banks [156, 157, 22] to increase the set of negative instances.

Joint Image-Text Representation Learning. With the advances in both visual analysis and natural language understanding, there has been a recent shift towards learning representation jointly from both visual and textual domains [158, 159, 160, 161, 162, 163, 164, 165, 166, 167]. Among these efforts, ViLBERT [160] and LXMERT [162] learn representation from both modalities using two-stream transformers, applied to image and text independently. In contrast, UNITER [165], VisualBERT [158], Unicoder-VL [164], VL-BERT [159] and B2T2 [166] propose a unified single architecture that learns representation jointly from both domains. Our method is similar to the first group, but differs in its fundamental goal. Instead of focusing on learning a task-agnostic representation for a range of downstream tasks, we are interested in the quality of region-phrase grounding emerged by maximizing mutual information. Moreover, we rely on the language modality as a weak training signal for grounding, and we perform phrase-grounding without any further finetuning.

5.2 METHOD

Consider the set of region features and contextualized word representation as two multivariate random variables. Intuitively, estimating MI between them requires extracting the information content shared by these two variables. We model this MI estimation as maximizing a lower bound on MI with respect to parameters of a word-region attention model. This maximization forces the attention model to downweight regions from the image that do not match the word, and to attend to the image regions that contain the most shared information with the word representation.

Sec. 5.2.1 describes MI and the InfoNCE lower bound. Sec. 5.2.2 introduces notation and

InfoNCE based objective for learning phrase grounding from paired image caption data. Sec. 5.2.3 presents the design of a word-region attention based compatibility function which is part of the InfoNCE objective.

5.2.1 InfoNCE Lower Bound on Mutual Information

Let $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ be random variables drawn from a joint distribution with density $p(x, y)$. The MI between x and y measures the amount of information that these two variables share:

$$\mathrm{MI}(x, y) = \mathbb{E}_{(x,y)\sim p(x,y)}\left[\log \frac{p(x,y)}{p(x)p(y)}\right], \tag{5.1}$$

which is also the KullbackLeibler Divergence from $p(x, y)$ to $p(x)p(y)$.

However, computing MI is intractable in general because it requires a complete knowledge of the joint and marginal distributions. Among the existing MI estimators, the InfoNCE [141] lower bound provides a low-variance estimation of MI for high dimensional data, albeit being biased [148]. The appealing variance properties of this estimator may explain its recent success in representation learning [23, 141, 152, 167]. InfoNCE defines a lower bound on MI by:

$$\mathrm{MI}(x, y) \geq \log(k) - \mathcal{L}_k(\theta). \tag{5.2}$$

Here, $\mathcal{L}_k$ is the InfoNCE objective defined in terms of a compatibility function ϕ parametrized by θ: $\phi_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. The lower bound is computed over a mini-batch $\mathcal{B}$ of size k, consisting of one positive pair $(x, y) \sim p(x, y)$ and $k - 1$ negative pairs $\{(x'_i, y)\}_{i=1}^{k-1}$ where $x' \sim p(x)$:

$$\mathcal{L}_k(\theta) = \mathbb{E}_{\mathcal{B}}\left[-\log\left(\frac{e^{\phi_\theta(x,y)}}{e^{\phi_\theta(x,y)} + \sum_{i=1}^{k-1} e^{\phi_\theta(x'_i,y)}}\right)\right]. \tag{5.3}$$

Oord *et al*. [141] showed that maximizing the lower bound on MI by minimizing $\mathcal{L}_k$ with respect to θ leads to a compatibility function ϕ_{θ^*} that obeys

$$e^{\phi_{\theta^*}(x,y)} \propto \frac{p(x|y)}{p(x)} = \frac{p(x,y)}{p(x)p(y)}, \tag{5.4}$$

where θ^* is the optimal θ obtained by minimizing $\mathcal{L}_k$.

5.2.2 InfoNCE for Phrase Grounding

Recent work [135] has shown that pre-trained object detectors such as FasterRCNN [4] and language models such as BERT [39] provide rich representations in the visual and textual domains for the phrase grounding problem. Inspired by this, we aim to maximize mutual information between region features generated by an object detector and contextualized word representation extracted by a language model.

Let us denote image region features for an image by $\mathbf{R} = \{r_i\}_{i=1}^{m}$ where m is the number of regions in the image with each $r_i \in \mathbb{R}^{d_r}$. Similarly, caption word representations are denoted as $\mathbf{W} = \{w_j\}_{j=1}^{n}$ where n is the number of words in the caption with each word represented as $w_j \in \mathbb{R}^{d_w}$.

We maximize the InfoNCE lower bound on MI between image regions and each individual word representation denoted by $\text{MI}(\mathbf{R}, w_j)$. Thus using Eq. 5.2 we maximize the following lower bound:

$$\sum_{j=1}^{n} \text{MI}(\mathbf{R}, w_j) \geq n \log(k) - \sum_{j=1}^{n} \mathcal{L}_{kj}(\theta). \tag{5.5}$$

We empirically show that maximizing the lower bound in Eq. 5.5 with an appropriate choice of compatibility function ϕ_θ results in learning phrase grounding without strong grounding supervision. The following section details the design of the compatibility function.

5.2.3 Compatibility Function with Attention

The InfoNCE loss in our phrase grounding formulation requires a compatibility function between the *set* of region feature vectors $\mathbf{R}$ and the contextualized word representation w_j. To define the compatibility function, we propose a `query-key-value` attention mechanism. Specifically, we define neural modules $k_r, v_r : \mathbb{R}^{d_r} \rightarrow \mathbb{R}^d$ to map each image region to `keys` and `values` and $q_w, v_w : \mathbb{R}^{d_w} \rightarrow \mathbb{R}^d$ to compute `query` and `values` for the words. The `query` vectors for each word are used to compute the attention score for every region given a word using

$$a(r_i, w_j) = \frac{e^{s(r_i, w_j)}}{\sum_{i'=1}^{m} e^{s(r_{i'}, w_j)}}, \tag{5.6}$$

where $s(r_i, w_j) = q_w(w_j)^T k_r(r_i)/\sqrt{d}$. The attention scores are used as a soft selection mechanism to compute a word-specific visual representation using a linear combination of region `values`

$$v_{att}(\mathbf{R}, w_j) = \sum_{i=1}^{m} a(r_i, w_j) v_r(r_i). \tag{5.7}$$

Finally, the compatibility function is defined as $\phi_\theta(\mathbf{R}, w_j) = v_w^T(w_j) v_{att}(\mathbf{R}, w_j)$, where θ refers to the parameters of neural modules k_r, v_r, q_w, and v_w, implemented using simple feed-forward MLPs. Following Eqs. 5.3 & 5.5, the InfoNCE loss for phrase grounding is defined

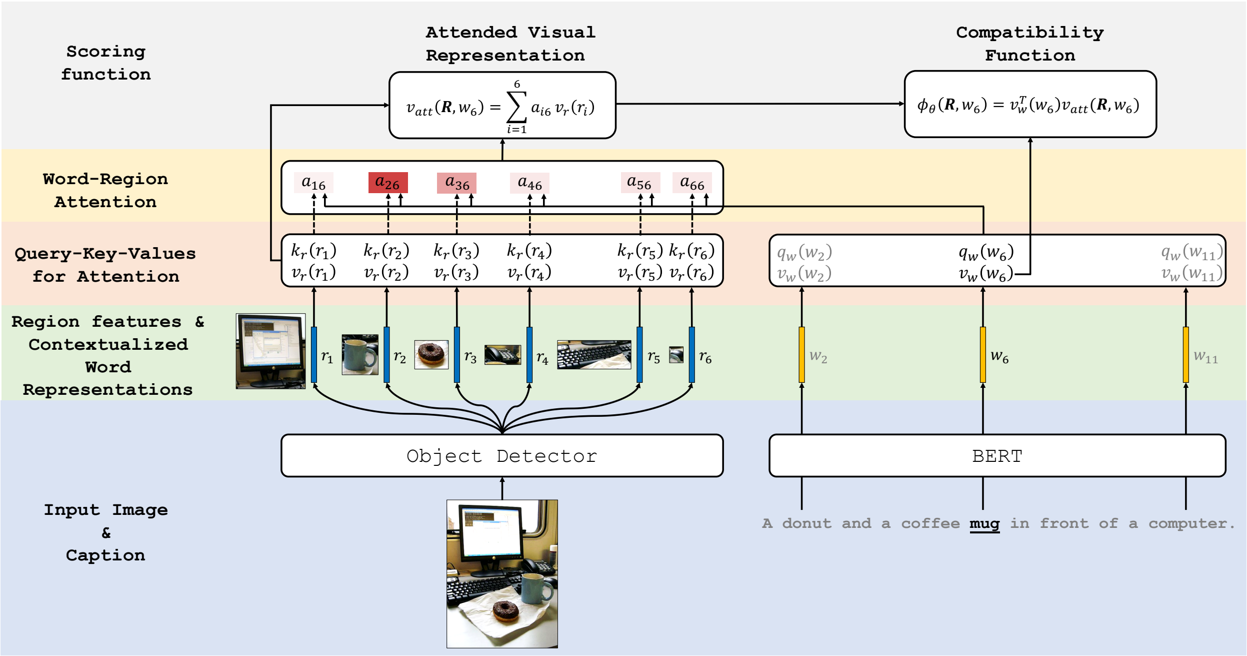

Scoring function
Attended Visual Representation
$v_{att}(\boldsymbol{R}, w_6) = \sum_{i=1}^{6} a_{i6}\, v_r(r_i)$
Compatibility Function
$\phi_\theta(\boldsymbol{R}, w_6) = v_w^T(w_6) v_{att}(\boldsymbol{R}, w_6)$
Word-Region Attention
a_{16} a_{26} a_{36} a_{46} a_{56} a_{66}
Query-Key-Values for Attention
$k_r(r_1)$ $v_r(r_1)$ $k_r(r_2)$ $v_r(r_2)$ $k_r(r_3)$ $v_r(r_3)$ $k_r(r_4)$ $v_r(r_4)$ $k_r(r_5)$ $v_r(r_5)$ $k_r(r_6)$ $v_r(r_6)$
$q_w(w_2)$ $v_w(w_2)$ $q_w(w_6)$ $v_w(w_6)$ $q_w(w_{11})$ $v_w(w_{11})$
Region features & Contextualized Word Representations
r_1 r_2 r_3 r_4 r_5 r_6
w_2 w_6 w_{11}
Object Detector
BERT
Input Image & Caption
A donut and a coffee mug in front of a computer.

Figure 5.2: **Compatibility function ϕ_θ with word-region attention.** The figure shows compatibility computation between the set of image regions and the word "mug" in the caption. The compatibility function consists of learnable `query-key-value` functions k_r, v_r, q_w, v_w. The `query` constructed from contextualized representation of the word "mug" is compared to `keys` created from region features to compute attention scores. The attention scores are used as weights to linearly combine `values` created from region features to construct an attended visual representation for "mug". The compatibility is defined by the dot product of the attended visual representation and `value` representation for "mug".

as

$$\mathcal{L}_{\texttt{img}}(\theta) = \mathbb{E}_{\mathcal{B}}\left[-\sum_{j=1}^{n} \log\left(\frac{e^{\phi_\theta(\mathbf{R}, w_j)}}{e^{\phi_\theta(\mathbf{R}, w_j)} + \sum_{i=1}^{k-1} e^{\phi_\theta(\mathbf{R}'_i, w_j)}}\right)\right]. \tag{5.8}$$

which is marked using subscript `img` as negative pairs are created by replacing image regions from a positive pair with regions extracted from negative instance in the mini-batch.

Remark: We enforce compatibility between each word and all image regions using $\text{MI}(\mathbf{R}, w_j)$ in Eq. 5.5, but not between a region and all caption words ($\text{MI}(r_i, \mathbf{W})$). This is because the words only describe part of the image, so there will be regions with no corresponding word in the caption.

Caption	Negatives Selected After Reranking	Candidates Rejected After Reranking
A man is seated at a counter with all types of delicious looking foods, yet is completely unaffected, casually reading his newspaper.	menu, books, phone, scripts, email, messages, bible, tablet	newspaper, paper, journal, article, magazine
A BMX bike rider in red clothing and a helmet is riding his bike next to a wooden fence.	bench, pole, statue, door, table, chair, sign, platform, piano	fence, gate, wall, railing, screen
A man in a blue jumpsuit stands next to a red van pulling a trailer.	bike, sedan, horse, jeep, cart, car, tractor, bull, engine, motorcycle	van, trailer, vehicle, light, truck
A man and a boy are playing with a dog in the evening.	girl, lady, mother, woman, teenager, child, teacher, mom	man, boy, guy, couple, youth
A woman in a brown sweater sits at a table covered with food.	boy, guy, gentleman, kid, nurse, soldier, waiter, priest, child	woman, female, person, face, lady
A man with shorts and a hat is holding onto a little boy and a dog.	gloves, glasses, coat, trousers, bags, apron, moustache, beard	shorts, ties, pants, stripes, jeans

Figure 5.3: **Context-preserving negative captions.** We construct negative captions which share the same context as the true caption but substitute a noun word. We choose the substitute using a language model such that it is plausible in the context but we reject potential synonyms or hypernyms of the original word by a re-ranking procedure.

5.2.4 Context-Preserving Negative Captions

The objective in Eq. 5.8 trains the compatibility function by contrasting positive regions-word pairs against pairs with replaced image regions. We now propose a complementary objective function that contrasts the positive pairs against negative pairs whose captions are replaced with plausible negative captions. However, extracting negative captions that are related to a captions is challenging as it requires semantic understanding of words in a caption. Here, we leverage BERT as a pretrained bidirectional language model to extract such negative captions.

For a caption with a noun word s and context c, we define a context-preserving negative caption as one which has the same context c but a different noun s' with the following properties: (i) s' should be plausible in the context; and (ii) the new caption defined by the pair (s', c) should be untrue for the image. For example, consider the caption `"A man is walking on a beach"` where s is chosen as `"man"` and c is defined by `"A [MASK] is walking on a beach"` where `[MASK]` is the token that denotes a missing word. A potential candidate for a context-preserving negative caption might be `"A woman is walking on a beach"` where s' is `woman`. However, `"A car is walking on a beach"` and `"A person is walking on a beach"` are not negative captions because `car` is not plausible given the context, and the statement with `person` is still true given that the original caption is true for the image.

Constructing context-preserving negative captions. We propose to use a pre-trained

BERT language model to construct context-preserving negative captions for a given true caption. Our approach for extracting such words consists of two steps: First, we feed the context c into the language model to extract 30 most likely candidates $\{s'_l\}_{l=1}^{30}$ for the masked word using probabilities $p(s'|c)$ predicted by BERT. Intuitively, these words correspond to those that fill in the masked word in caption according to BERT. However, the original masked word or its synonyms may be present in the set as well. Thus, in the second step, we pass the original caption into BERT to compute $q(s'_l|s,c)$ which we use as a proxy for how true (s'_l, c) is given that (s,c) is true. We re-rank the candidates using the score $\frac{p(s'|c)}{q(s'|s,c)}$ and we keep the top 25 captions $\{(s'_l, c)\}_{l=1}^{25}$ as negatives for the original caption (s,c).

We empirically find that the proposed approach is effective in extracting context-preserving negative captions. Fig. 5.3 shows a context-preserving negatives for a set of captions along with candidates that were rejected after re-ranking. Note that the selected candidates match the context and the rejected candidates are often synonyms or hypernyms of the true noun.

Training with context-preserving negative captions. Given the context-preserving negative captions, we can train our compatibility function by contrasting the positive pairs against negative pairs with plausible negative captions. We use a loss function similar to InfoNCE to encourage higher compatibility score of an image with the true caption than any negative caption. Let w and $\{w'_l\}_{l=1}^{25}$ denote the contextualized representation of the positive word s and the corresponding negative noun words $\{s'_l\}_{l=1}^{25}$. The language loss is defined as

$$\mathcal{L}_{\texttt{lang}}(\theta) = \mathbb{E}_{\mathcal{B}}\left[-\log\left(\frac{e^{\phi_\theta(\mathbf{R},w)}}{e^{\phi_\theta(\mathbf{R},w)} + \sum_{l=1}^{25} e^{\phi_\theta(\mathbf{R},w'_l)}}\right)\right]. \tag{5.9}$$

For captions with multiple noun words, we randomly select s from the noun words for simplicity.

5.2.5 Implementation Details

Regions and Visual Features. We use the Faster-RCNN object detector provided by Anderson *et al.* [16] and used for extracting visual features in the current state-of-the-art phrase grounding approach Align2Ground [135]. The detector is trained jointly on Visual Genome object and attribute annotations and yields $\sim$ 10 to 50 top scoring bounding boxes per image with 2048 dimensional ROI-pooled region features.

Contextualized Word Representations. We use a pretrained BERT language model to extract 768 dimensional contextualized word representations for each caption word. Note that BERT is trained on a text corpora using masked language model training where words are randomly replaced by a `[MASK]` token in the input and the likelihood of the masked word is maximized in the distribution over vocabulary words predicted at the output. Thus, BERT is trained to model distribution over words given context and hence suitable for modeling $p(s|c)$ defined in Sec. 5.2.4 for constructing context-preserving negative captions.

Query-Key-Value Networks. We use an MLP with 1 hidden layer for each of k_r, v_r, q_w, v_w for all experiments except the ablation in Fig. 5.4. We use BatchNorm [168] and ReLU activations after the first linear layer. The hidden layer has the same number of neurons as the input dimensions of these networks which are 2048 for (k_r, v_r), and 768 for (q_w, v_w). The output layer is 384 $(= 768/2)$ for all networks.

Losses. Since we only care about grounding noun phrases, we compute $\mathcal{L}_{\text{img}}$ only for noun and adjective words in the captions as identified by a POS tagger instead of all caption words for computation efficiency.

Optimization. We optimize the losses computed over batches consisting of 50 image-caption pairs using the ADAM optimizer [110] with a learning rate of 10^{-5}. We compute $\mathcal{L}_{\texttt{img}}$ for each image using other images in the batch as negatives.

Attention to phrase grounding. We use the BERT tokenizer to convert captions into individual word or sub-word tokens. Attention is computed per token. For evaluation, the phrase-level attention score for each region is computed as the maximum attention score assigned to the region by any of the tokens in the phrase. The regions are then ranked according to this phrase level score.

5.3 EXPERIMENTS

Our experiments compare our approach to state-of-the-art on weakly supervised phrase localization (Sec. 5.3.2), ablate gains due to pretrained language representations and context-preserving negative sampling using a language model (Sec. 5.3.3), and analyse the relation between phrase grounding performance and the InfoNCE bound that we optimize as a proxy for phrase grounding (Sec. 5.3.4).

Method	Training Data	Visual Features	R@1	R@5	R@10	Accuracy
GroundeR (2015) [137]	Flickr30K Entities	VGG-det (VOC)	28.94	-	-	-
Yeh *et al.* (2018) [142]	Flickr30K Entities	VGG-cls (IN)	22.31	-	-	-
Yeh *et al.* (2018) [142]	Flickr30K Entities	VGG-det (VOC)	35.90	-	-	-
Yeh *et al.* (2018) [142]	Flickr30K Entities	YOLO (COCO)	36.93	-	-	-
KAC Net+Soft KBP (2018) [169]	Flickr30K Entities	VGG-det (VOC)	38.71	-	-	-
Fang *et al.* (2015) [140]	COCO	VGG-cls (IN)	-	-	-	29.00
Akbari *et al.* (2019) [136]	COCO	VGG-cls (IN)	-	-	-	61.66
Akbari *et al.* (2019) [136]	COCO	PNAS Net (IN)	-	-	-	69.19
Align2Ground (2019) [135]	COCO	Faster-RCNN (VG)	-	-	-	71.00
Ours	Flickr30K Entities	Faster-RCNN (VG)	**47.88**	**76.63**	**82.91**	**74.94**
Ours	COCO	Faster-RCNN (VG)	**51.67**	**77.69**	**83.25**	**76.74**

Table 5.1: **Grounding performance on Flickr30K Entities test set**. We make our approach directly comparable to the current state-of-the-art, Align2Ground [135]. The performance of older methods are reported for completeness but the use of different visual features makes direct comparison difficult.

5.3.1 Datasets and Metrics

We train our models on image-caption pairs from COCO training set which consists of $\sim$ 83K training images. We use the validation set with $\sim$ 41K images for part of our analysis. Each image is accompanied with 5 captions. For evaluation, we use the Flickr30K Entities validation set for model selection (early stopping) and test set for reporting final performance. Both sets consist of 1K images with 5 captions each. We report two metrics:

Recall@k which is the fraction of phrases for which the ground truth bounding box has an IOU ≥ 0.5 with any of the top-k predicted boxes.

Pointing accuracy which requires the model to predict a single point location per phrase and the prediction is counted as correct if it falls within the ground truth bounding box for the phrase. Unlike recall@k, pointing accuracy does not require identifying the extent of the object. Since our model selects one of the detected regions in the image, we use use center of the selected bounding box as the prediction for each phrase for computing pointing accuracy.

5.3.2 Performance on Flickr30K Entities

Tab. 5.3.2 compares performance of our method to existing weakly supervised phrase grounding approaches on the Flickr30K Entities test set. A few existing approaches train on

Negative Captions	**Language Model**	**R@1**	**R@5**	**R@10**	**Accuracy**
None	`BERT (Random)`	25.66	59.57	75.16	57.37
None	`BERT (Pretrained)`	35.74	72.91	82.07	66.89
Random	`BERT (Pretrained)`	36.32	72.42	81.81	66.92
Contextually plausible	`BERT (Pretrained)`	**48.05**	**76.78**	**82.97**	**74.91**
Excluding near-synonyms & hypernyms	`BERT (Pretrained)`	**51.67**	**77.69**	**83.25**	**76.74**

Table 5.2: **Benefits of language modeling.** The first two rows show the gains due to pretrained language representations. The next three rows show gains from each step in our proposed context-preserving negative caption construction.

Flickr30K Entities train set and report recall@1 while recent methods use COCO train set and report pointing accuracy. Further, all approaches use different visual features making direct comparison difficult. For a fair comparison to state-of-the-art, we use Faster-RCNN trained on Visual Genome object and attribute annotations used in Align2Ground [135] and report performance for models trained on either datasets on both recall and pointing accuracy metrics.

Using the same training data and visual feature architecture, our model shows a 5.7% absolute gain in pointing accuracy over Align2Ground. Learning using our contrastive formulation is also quite sample efficient as can be seen by only a 2 to 3 points drop in performance when the model is trained on the much smaller Flickr30K Entities train set which has approximately one-third as many image-caption pairs as COCO.

5.3.3 Benefits of Language Modeling

Our approach benefits from language modeling in two ways: (i) using the pretrained language model to extract contextualized word representations, and (ii) using the language model to sample context-preserving negative captions. Tab. 5.3.3 evaluates along both of these dimensions.

Gains from pretrained word representations. In Tab. 5.3.3, `BERT (Random)` refers to the BERT architecture initialized with random weights and finetuned on COCO image-caption data along with parameters of the attention mechanism. `BERT (Pretrained)` refers to the off-the-shelf pretrained BERT model which is used as a contextualized word feature extractor during contrastive learning without finetuning. We observe a ~10% absolute gain in both recall@1 and pointing accuracy by using pretrained word representations from BERT.

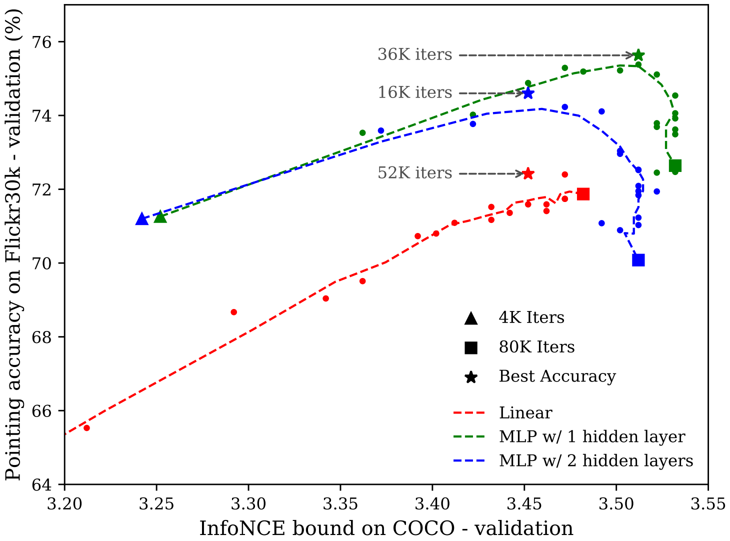


Figure 5.4: Relation between `InfoNCE` lower bound and phrase grounding performance with training iterations for 3 different choices of `key-value` modules in the compatibility function ϕ_θ. Each epoch is $\sim$ 8K iterations. The scattered points visualize the measured quantities during training. The dashed lines are created by applying moving average to highlight the trend.

Gains from context-preserving negative caption sampling. Our context-preserving negative sampling has two steps. The first step is drawing negative noun candidates given the context provided by the true caption. The second step is re-ranking the candidates to filter out likely synonyms or hypernyms that are also true for the image.

First, note that randomly sampling negative captions from training data for computing $\mathcal{L}_{\text{lang}}$ performs similarly to only training using $\mathcal{L}_{\text{img}}$. Model trained with contextually plausible negatives significantly outperforms random sampling by $\geq$8% gain in recall@1 and pointing accuracy. Excluding near-synonyms and hypernyms yields another $\sim$3 points gain in recall@1 and accuracy.

5.3.4 Is InfoNCE a good proxy for learning phrase grounding?

The fact that optimizing our InfoNCE objective results in learning phrase grounding is intuitive but not trivial. Fig. 5.4 shows how maximizing the InfoNCE lower bound correlates well with phrase grounding performance on a heldout dataset. We make several interesting observations: **(i)** As training progresses (from left to right), InfoNCE lower bound (Eq. 5.5) mostly keeps increasing on the validation set. This indicates that there is no overfitting in terms of the InfoNCE bound. **(ii)** With the increase in InfoNCE lower bound, phrase grounding performance first increases until peak performance and then starts decreasing.

This shows that the InfoNCE bound is correlated with the grounding performance but maximizing it fully does not necessarily yields at the best grounding. Similar observation has been made in [155] for representation learning. **(iii)** The peak performance and the number of iterations needed for the best performance depends on the choice of `key-value-query` modules. One and two layer MLPs hit the peak faster and perform better than linear functions.

5.3.5 Qualitative Results

Fig. 5.5 visualizes the word-region attention learned by our model. The qualitative results demonstrate the following abilities: **(i)** localizing different objects mentioned in the same caption with varying degrees of semantic relatedness, e.g., `man` and `canine` in row 1 vs. `man` and `woman` in row 3; **(ii)** disambiguation between two instances of the same object category using caption context. For example, `boy` and `another` in row 4 and bride and groom from other men and women in row 3; **(iii)** localizing object parts such as toddler's `shirt` in row 2 and instrument's `mouthpiece` in row 5; **(iv)** handling occlusion, e.g., `table` covered with toys in row 6; **(v)** handling uncommon words or categories like `ponytail` and `mouthpiece` in row 5 and `hose` in row 7.

These results show that given rich visual and contextualized word representations, contrastive learning causes our attention mode to learn phrase grounding.

5.4 LIMITATIONS AND FUTURE WORKS

The empirical examination of our framework reveals the following limitations:

Pretrained representations. Like prior arts, our approach relies on pretrained object detector and a language model to represent regions and caption-words. Ideally, we would expect to learn from scratch or improve existing region and word representations directly from image-caption data.

Need for fully-labeled validation set. In Fig. 5.4, we observe that an early stopping based on the validation performance is required to choose the best model for phrase grounding. While this is common practice for weakly supervised learning [170] and the Flickr30K Entities validation set we use is 80× smaller than the COCO training set, this translates to using full supervision for a small set of images.

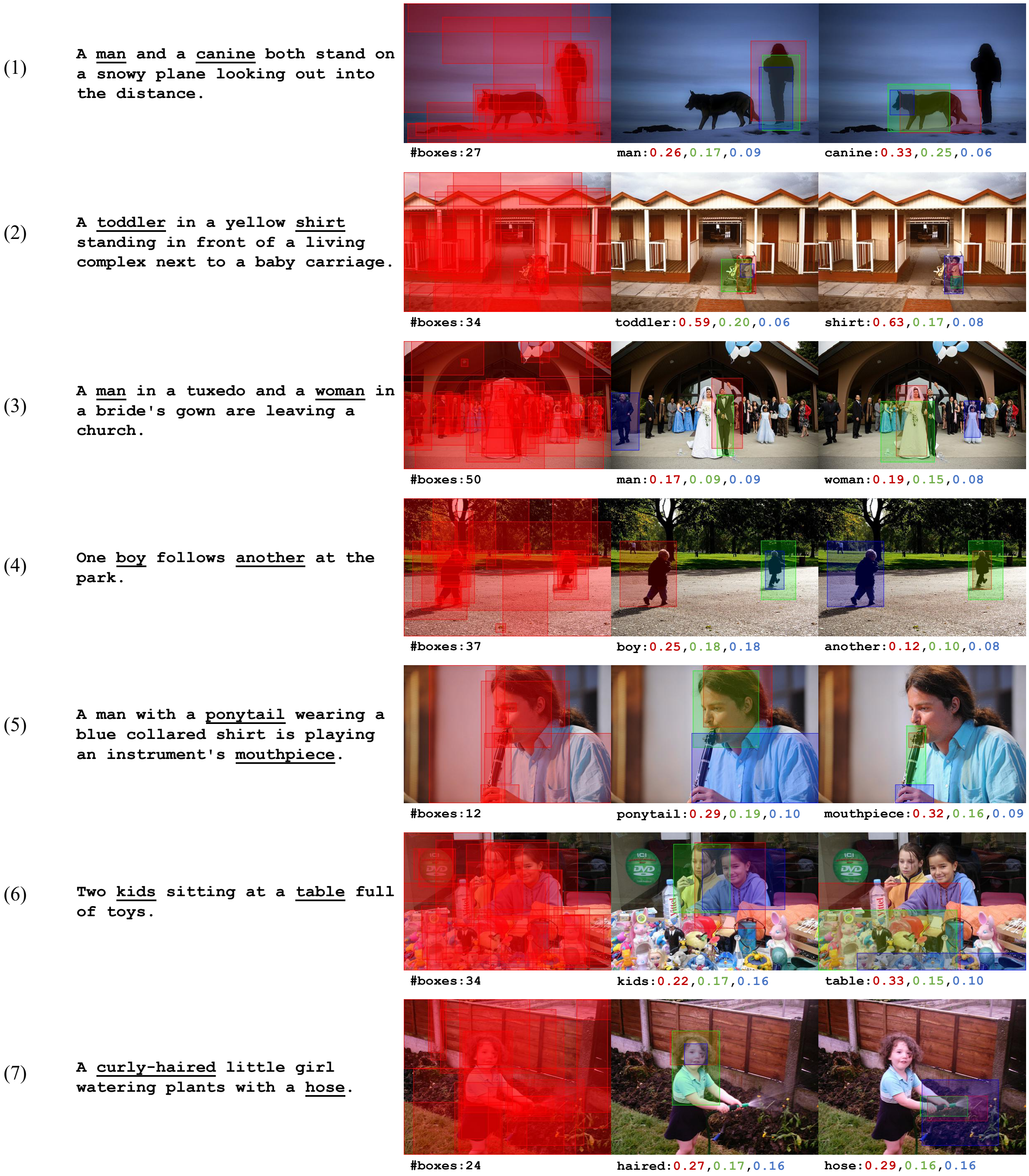

(1)
A man and a canine both stand on a snowy plane looking out into the distance.
#boxes:27
man:0.26,0.17,0.09
canine:0.33,0.25,0.06
(2)
A toddler in a yellow shirt standing in front of a living complex next to a baby carriage.
#boxes:34
toddler:0.59,0.20,0.06
shirt:0.63,0.17,0.08
(3)
A man in a tuxedo and a woman in a bride's gown are leaving a church.
#boxes:50
man:0.17,0.09,0.09
woman:0.19,0.15,0.08
(4)
One boy follows another at the park.
#boxes:37
boy:0.25,0.18,0.18
another:0.12,0.10,0.08
(5)
A man with a ponytail wearing a blue collared shirt is playing an instrument's mouthpiece.
#boxes:12
ponytail:0.29,0.19,0.10
mouthpiece:0.32,0.16,0.09
(6)
Two kids sitting at a table full of toys.
#boxes:34
kids:0.22,0.17,0.16
table:0.33,0.15,0.10
(7)
A curly-haired little girl watering plants with a hose.
#boxes:24
haired:0.27,0.17,0.16
hose:0.29,0.16,0.16

Figure 5.5: **Visualization of attention.** We show all detected regions and top-3 attended regions with attention scores for two words highlighted in each caption.

Bounds on MI. While $\log(K) - \mathcal{L}_{\texttt{img}}$ in Eq. 5.8 is a valid lower bound on MI, our $\log(K) - \mathcal{L}_{\texttt{lang}}$ in Eq. 5.9 is no longer a lower bound on MI as it oversamples negative words related to a caption. A valid bound would involve random sampling of captions from the training data however our context-preserving negative captions lead to much better performance.

5.5 CONCLUSION

In this work, we offer a novel perspective on weakly supervised phrase grounding from paired image-caption data which has traditionally been cast as a multiple instance learning problem. We formulate the problem as that of estimating mutual information between image regions and caption words. We demonstrate that maximizing a lower bound on mutual information with respect to parameters of a region-word attention mechanism results in learning to ground words in images. We also show that language models can be used to generate context-preserving negative captions which greatly improve learning in comparison to randomly sampling negatives from training data.

CHAPTER 6: SEMANTIC SCENE GENERATION

6.1 INTRODUCTION

In the previous chapter we designed a representation space for image regions and words which generalizes across multiple vision-language tasks. However, this model treats images as a bag of non-interacting regions. On the other hand, it is often the interactions between objects that makes an image or a natural language description of a scene or an event interesting. The current chapter and the next focus on modeling such interactions in two distinct applications: (i) Semantic Scene Generation (this chapter), and (ii) Human-Object Interaction Detection (Chapter 4).

Consider the scene description: *Fred is wearing a blue hat and talking to Wilma in the living room. Wilma then sits down on a couch.* Picturing the scene in our mind requires the knowledge of plausible locations, appearances, actions, and interactions of characters and objects being described, as well as an ability to understand and translate the natural language description into a plausible visual instantiation. In this work, we introduce Semantic Scene Generation (SSG), the task of generating complex scene videos from rich natural language descriptions which requires jointly modeling the layout and appearances of entities mentioned in the description. SSG models are trained using a densely annotated video dataset with scene descriptions and entity bounding boxes. During inference, the models must generate videos for novel descriptions (unseen during training).

Modelling the layout and appearances of entities for descriptions like the one above poses several challenges:

- **Entity Recall** - the video must contain the relevant characters (Fred, Wilma), objects (blue hat, couch) and background (setting that resembles a living room)
- **Layout Feasibility** - characters and objects must be placed at plausible locations and scales (Fred, Wilma and the couch should be placed on the ground plane, the hat must lie on top of Fred's head)
- **Appearance Fidelity** - entity appearance, which may be affected by identity, pose, action, attributes and layout, should respect the scene description
- **Interaction Consistency** - appearance of characters and objects must be *consistent with each other* given the described, sometimes implicit, interaction (Fred and Wilma should face each other as do people when they talk to each other)

- **Language Understanding** - the system must be able to understand and translate a natural language description into a plausible visual instantiation.

Thus, one of the main considerations while designing a model for SSG is to come up with contextualized representations of different entities in a given description and a video that can address the above challenges.

Towards the goal of SSG, we introduce FLINTSTONES, a densely annotated dataset based on *The Flintstones* animated series, consisting of over 25000 videos, each 75 frames long. Each clip has been annotated with a caption with entities and background word annotated in each caption. Bounding box tracks and segmentation masks (using SLIC [171] and GrabCut [172]) are annotated for each entity in the clip. A clean background is also obtained for each clip through PatchMatch [173] hole filling.

FLINTSTONES has several advantages over using a random sample of internet videos. First, in a closed world setting such as a television series, the most frequent characters are present in a wide variety of settings, which serves as a more manageable learning problem than a sparse set obtained in an open world setting. Second, the flat textures in animations are easier to model than real world videos. Third, in comparison to other animated series, The Flintstones has a good balance between having fairly complex interactions between characters and objects while not having overly complicated, cluttered scenes. For these reasons, we believe that the FLINTSTONES dataset is semantically rich, preserves all the challenges of text to scene generation and is a good stepping stone towards real videos. FLINTSTONES consists of an 80-10-10 train-val-test split. The train and val sets are used for learning and model selection respectively. Test captions serve as novel descriptions to generate videos at test time. To quantitatively evaluate our model, we use two sets of metrics. The first measures *semantic fidelity* of the generated video to the desired description using entity noun, adjective, and verb recalls. The second measures *composition consistency*, *i.e.* the consistency of the appearances, poses and layouts of entities with respect to other entities in the video and the background.

6.2 SEMI-PARAMETRIC VIDEO SYNTHESIS APPROACH OVERVIEW

Currently, the dominant approaches to conditional generation of visual data from text rely on directly learning distributions in a *high dimensional pixel space*. While these approaches have shown impressive results for aligned images of objects (faces, birds, flowers, etc.), they are often inadequate for addressing the above challenges, due to the *combinatorial explosion* of the image space arising from multiple characters and objects with sig-

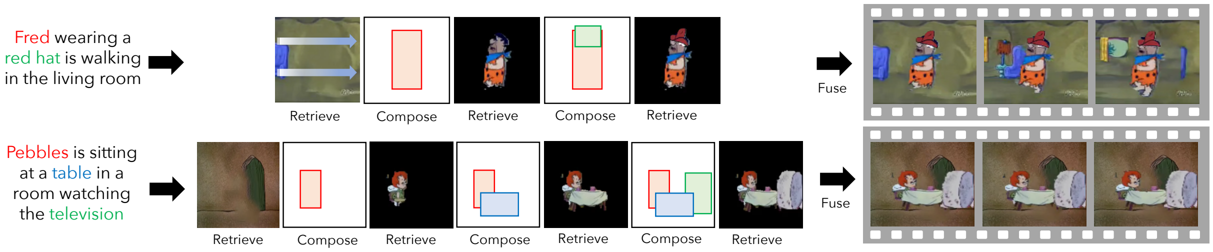


Figure 6.1: Given a novel description, Craft sequentially composes a scene layout and retrieves entities from a video database to create complex scene videos.

nificant appearance variations arranged in a large number of possible layouts. In contrast, our proposed **C**omposition, **R**etrieval **a**nd **F**usion Ne**t**work (Craft) explicitly models the spatio-temporal layout of characters and objects in the scene jointly with entity appearances. Unlike pixel generation approaches, our appearance model is based on text to entity segment retrieval from a video database. Spatio-temporal segments are extracted from the retrieved videos and fused together to generate the final video. The layout composition and entity retrieval work in a sequential manner which is determined by the language input. Factorization of our model into composition and retrieval stages alleviates the need to directly model pixel spaces, results in an architecture that exploits location and appearance contextual cues, and renders an interpretable output.

We use Flintstones to evaluate Craft and provide a detailed ablation analysis. Craft outperforms baselines that generate pixels directly from captions as well as a whole video retrieval approach (as opposed to modeling entities). It generalizes well to unseen captions as well as unseen videos in the target database. Our quantitative and qualitative results show that for simpler descriptions, Craft exploits location and appearance contextual cues and outputs videos that have consistent layouts and appearances of described entities. However, there is tremendous scope for improvement. Craft can fail catastrophically for complex descriptions (containing large number of entities, specially infrequent ones). The adjective and verb recalls are also fairly low. We believe SSG on Flintstones presents a challenging problem for future research. (See Fig 6.6 for qualitative results).

6.3 RELATED WORK

Generative models. Following pioneering work on Variational Autoencoders [174] and Generative Adversarial Networks [175], there has been tremendous interest in generative modelling of visual data in a high dimensional pixel space. Early approaches focused on unconditional generation [176, 177, 178, 179], whereas recent works have explored models

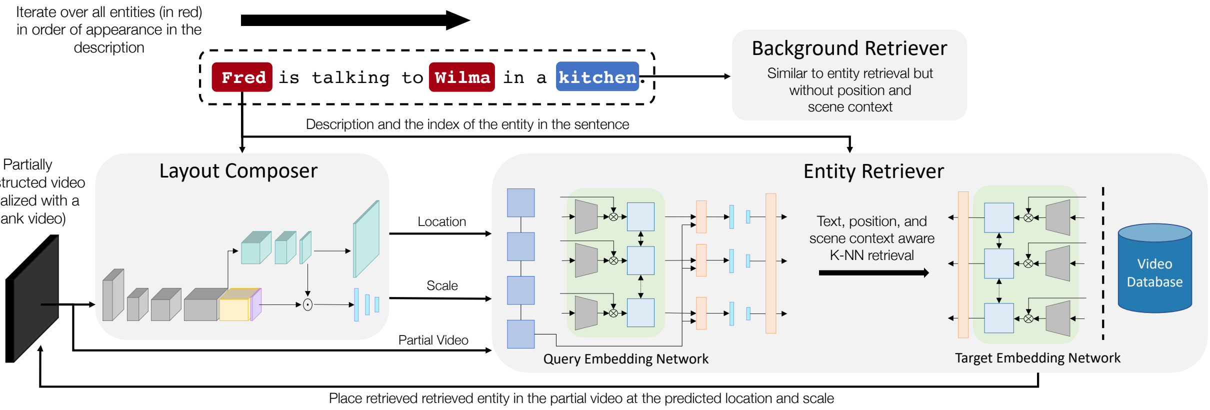

Iterate over all entities (in red) in order of appearance in the description
Fred is talking to Wilma in a kitchen.
Background Retriever
Similar to entity retrieval but without position and scene context
Description and the index of the entity in the sentence
Partially constructed video (initialized with a blank video)
Layout Composer
Location
Scale
Partial Video
Entity Retriever
Text, position, and scene context aware K-NN retrieval
Video Database
Query Embedding Network
Target Embedding Network
Place retrieved retrieved entity in the partial video at the predicted location and scale

Figure 6.2: **Overview** of **C**omposition, **R**etrieval **a**nd **F**usion Ne**t**work (CRAFT), consisting of three parts: *Layout Composer*, *Entity Retriever* and *Background Retriever*. CRAFT begins with an empty video and sequentially adds entities mentioned in the input description at locations and scales predicted by the Layout Composer.

conditioned on simple textual inputs describing objects [180, 181, 182, 183, 184]. While the visual quality of images generated by these models has been steadily improving [185, 186], success stories have been limited to generating images of aligned objects (e.g. faces, birds, flowers), often training one model per object class. In contrast, our work deals with generating complex scenes which requires modelling the layout and appearances of multiple entities in the scene.

Of particular relevance is the work by Hong *et al.* [187] who first generate a coarse semantic layout of bounding boxes, refine that to segmentation masks and then generate an image using an image-to-image translation model [188, 189]. A limitation of this approach is that it assumes a fixed number of object classes (80 in their experiments) and struggles with the usual challenge of modeling high dimensional pixel spaces such as generating coherent entities. Formulating appearance generation in terms of entity retrieval from a database allows our model to scale to a large number of entity categories, guarantee intra-entity coherence and allows us to focus on the semantic aspects of scene generation and inter-entity consistency. The retrieval approach also lends itself to generating videos without significant modification. There have been attempts at extending GANs for unconditional [190, 191] as well as text conditional [192, 193] video generation, but quality of generated videos is usually worse than that of GAN generated images unless used in very restrictive settings. A relevant generative modelling approach is by Kwak *et al.* [194] who proposed a model in which parts of the image are generated sequentially and combined using alpha blending. However, this work does not condition on text and has not been demonstrated on complex scenes. Another relevant body of work is by Zitnick *et al.* [195, 196, 197] who compose static images from descriptions with clipart images using a Conditional Random Field formulation.

To control the structure of the output image, a growing body of literature conditions image generation on a wide variety of inputs ranging from keypoints [198] and sketches [199] to semantic segmentation maps [188]. In contrast to these approaches which condition on *provided* location, our model *generates* a plausible scene layout and then conditions entity retrieval on this layout.

Phrase Grounding and Caption-Image Retrieval. The entity retriever in CRAFT is related to caption based image retrieval models. The caption-image embedding space is typically learned by minimizing a ranking loss such as a triplet loss [200, 31, 200, 201, 30]. Phrase grounding [61] is another closely related task where the goal is to localize a region in an image described by a phrase.

One of our contributions is enriching the semantics of embeddings learned through triplet loss by simultaneously minimizing an auxiliary classification loss based on noun, adjective and verb words associated with an entity in the text description. This is similar in principle to [202] where auxiliary autoencoding losses were used in addition to a primary binary prediction loss to learn robust visual semantic embeddings. Learning shared representations across multiple related tasks is a key concept in multitask learning [40, 203].

6.4 METHOD

Figure 6.2 presents an overview of **C**omposition, **R**etrieval **a**nd **F**usion Ne**t**work which consists of three parts: *Layout Composer*, *Entity Retriever*, and *Background Retriever*. Each is a neural network that is trained independently using ground truth supervision. During inference, CRAFT begins with an empty video and adds entities in the scene sequentially based on the order of appearance in the description. At each step, the *Layout Composer* predicts a location and scale for an entity given the text and the video constructed so far. Then, conditioned on the predicted location, text, and the partially constructed video, the *Entity Retriever* produces a query embedding that is looked up against the embeddings of entities in the target video database. The entity is cropped from the retrieved video and placed at the predicted location and scale in the video being generated. Alternating between the *Layout Composer* and *Entity Retriever* allows the model to condition the layout of entities on the appearance and vice versa. Similar to *Entity Retriever*, the *Background Retriever* produces a query embedding for the desired scene from text and retrieves the

Caption	
T	Caption with length $\lvert T \rvert$
$\{E_i\}_{i=1}^n$	n entities in T in order of appearance
$\{e_i\}_{i=1}^n$	entity noun positions in T
Video	
F	number of frames in a video
$\{(l_i, s_i)\}_{i=1}^n$	position of entities in the video
l_i	entity bounding box at each frame $(\{(x_{if}, y_{if}, w_{if}, h_{if})\}_{f=1}^F)$
s_i	entity pixel segmentation mask at each frame
V_{i-1}	partially constructed video with entities $\{E_j\}_{j=1}^{i-1}$
$V\ (= V_n)$	full video containing all entities
$\{(V^{[m]}, T^{[m]})\}_{m=1}^M$	training data points, where M = number of data points

Figure 6.3: **Notations.**

closest background video from the target database. The retrieved spatio-temporal entity segments and background are fused to generate the final video. We now present the notation used in the rest of the paper, followed by architecture and training details for the three components.

6.4.1 Layout Composer

The layout composer is responsible for generating a plausible layout of the scene consisting of the locations and scales of each character and object mentioned in the scene description. Jointly modeling the locations of all entities in a scene presents fundamentally unique challenges for spatial knowledge representation beyond existing language-guided localization tasks. Predicting plausible locations and scales for objects not yet in an image requires a significant amount of *spatial knowledge* about people and objects, in contrast to text based object localization which relies heavily on appearance cues. This includes knowledge like – a hat goes on top of a person's head, a couch goes under the person sitting on it, a person being talked to faces the person speaking instead of facing away, tables are short and wide while standing people are tall and thin, etc.

Figure 6.4 presents a schematic for the layout composer. Given the varying number of entities across videos, the layout composer is setup to run in a sequential manner over the set of distinct entities mentioned in a given description. At each step, a text embedding of the desired entity along with a partially constructed video (consisting of entities fused into the video at previous steps) are input to the model which predicts distributions for the location and scale of the desired entity.

The layout composer models $P(l_i|V_{i-1}, T, e_i; \theta_{loc}, \theta_{sc})$, the conditional distribution of the location and scale (width and height normalized by image size) of the i^{th} entity given the

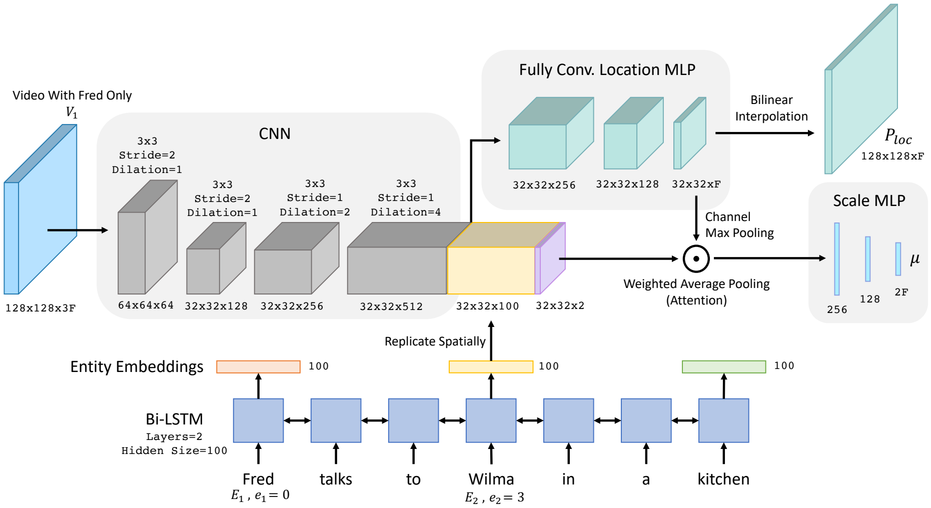


Figure 6.4: **Layout Composer** is run sequentially through the set of entities in the description, predicting the distributions for the location and scale of the desired entity.

text, entity noun position in tokenized text, and the partial video with previous entities. Let C_i denote the conditioning information, (V_{i-1}, T, e_i). We factorize the position distribution into location and scale components as follows:

$$P(l_i|C_i;\theta_{loc},\theta_{sc}) = \prod_{f=1}^{F} P^f_{loc}(x_{if},y_{if}|C_i;\theta^f_{loc}) \cdot P^f_{sc}(w_{if},h_{if}|x_{if},y_{if},C_i;\theta^f_{sc}) \tag{6.1}$$

$\theta_{loc} = \{\theta^f_{loc}\}^F_{f=1}$ and $\theta_{sc} = \{\theta^f_{sc}\}^F_{f=1}$ are learnable parameters. P^f_{loc} is modelled using a network that takes C_i as input and produces a distribution over all pixel locations for the f^{th} image frame. We model P^f_{sc} using a Gaussian distribution whose mean μ_f and covariance Σ_f are predicted by a network given (x_i, y_i, C_i). Parameters θ_{loc} and θ_{sc} are learned from ground truth position annotations by minimizing the following maximum likelihood estimation loss:

$$\sum_{m=1}^{M}\sum_{i=1}^{n^{[m]}}\sum_{f=1}^{F}\Big[-\log(P^f_{loc}(x^{[m]}_{if},y^{[m]}_{if}|C^{[m]}_i;\theta^f_{loc})) + 0.5\cdot\log(\det(\Sigma(x_{if},y_{if},C_i;\theta^f_{sc}))) +$$
$$0.5\cdot(z^{[m]}_{if} - \mu_f(D^{[m]}_i;\theta^f_{sc}))^T {\Sigma_f}^{-1}(z^{[m]}_{if} - \mu_f(D^{[m]}_i;\theta^f_{sc})) + \log(2\pi)\Big] \tag{6.2}$$

where $z_{if} = [w_{if}; h_{if}]$ & $D^{[m]}_i = (x^{[m]}_i, y^{[m]}_i, C^{[m]}_i)$. For simplicity, we manually set and freeze Σ to an isometric diagonal covariance matrix with variance of 0.005.

Feature Computation Backbone. The location and scale predictors have an identical feature computation backbone comprising of a CNN and a bidirectional LSTM. The CNN encodes V_{i-1} (8 sub-sampled frames concatenated along the channel dimension) as a set of convolutional feature maps which capture appearance and positions of previous entities in the scene. The LSTM is used to encode the entity E_i for which the prediction is to be made along with semantic context available in the caption. The caption is fed into the LSTM and the hidden output at e_i^{th} word position is extracted as the entity text encoding. The text encoding is replicated spatially and concatenated with convolutional features and 2-D grid coordinates to create a representation for each location in the convolutional feature grid that is aware of visual, spatial, temporal, and semantic context.

Location Predictor. P_{loc}^f is modelled using a Multi Layer Perceptron (MLP) that produces a score for each location. This map is bilinearly upsampled to the size of input video frames. Then, a softmax layer over all pixels produces $P_{loc}^f(x, y|C; \theta_{loc}^f)$ for every pixel location (x, y) in the f^{th} video frame.

Scale Predictor. Features computed by the backbone at a particular (x, y) location are selected and fed into the scale MLP that produces $\mu_f(x_i, y_i, C_i; \theta_{sc}^f)$.

Feature sharing and multitask training. While it is possible to train a separate network for each $\{P_{loc}^f, \mu_f\}_{f=1}^F$, we present a pragmatic way of sharing features and computation for different frames and also between the location and scale networks. To share features and computation across frames, the location network produces F probability maps in a single forward pass. This is equivalent to sharing all layers across all P_{loc}^f nets except for the last layer of the MLP that produces location scores. Similarly, all the μ_f nets are also combined into a single network. We refer to the combined networks by P_{loc} and μ.

In addition, we also share features across the location and scale networks. First, we share the feature computation backbone, the output from which is then passed into location and scale specific layers. Second, we use a soft-attention mechanism to select likely positions for feeding into the scale layers. This conditions the scale prediction on the plausible locations of the entity. We combine the F spatial maps into a single attention map through max pooling. This attention map is used to perform weighted average pooling on backbone features and then fed into the scale MLP. Note that this is a differentiable greedy approximation to find the most likely location (by taking argmax of spatial probability maps) and scale (directly using output of μ, the mode for a gaussian distribution) in a single forward pass. To keep training consistent with inference, we use the soft-attention mechanism instead of feeding ground-truth locations into μ.

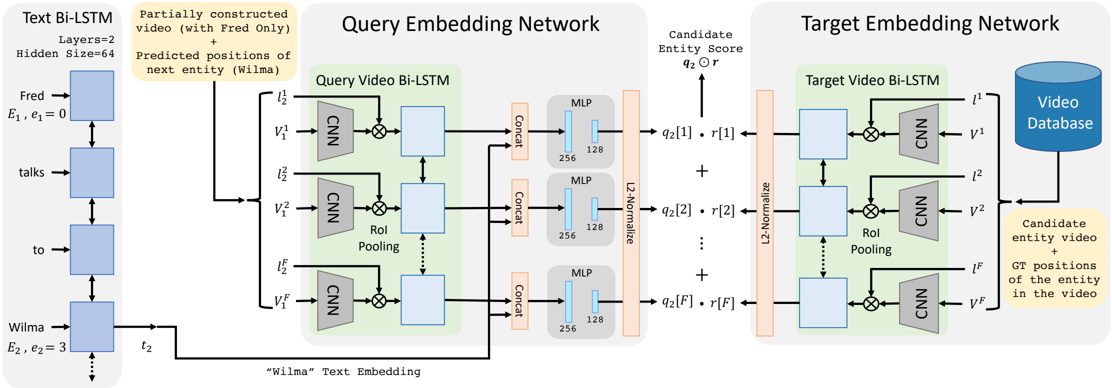


Figure 6.5: **Entity Retriever** retrieves spatio-temporal patches from a target database that match entity description as encoded by the query embedding network.

6.4.2 Entity Retriever

The task of the entity retriever is to find a spatio-temporal patch within a target database that matches an entity in the description *and* is consistent with the video constructed thus far – the video with all previous entities retrieved and placed in the locations predicted by the layout network. We adopt an embedding based lookup approach for entity retrieval. This presents several challenges beyond traditional image retrieval tasks. Not only does the retrieved entity need to match the semantics of the description but it also needs to respect the implicit relational constraints or context imposed by the appearance and locations of other entities. E.g. for *Fred is talking to Wilma*, it is not sufficient to retrieve *a Wilma*, but one who is also facing in the right direction, i.e. towards *Fred*.

The Entity Retriever is shown in Figure 6.5 and consists of two parts: (i) query embedding network Q, and (ii) target embedding network R. Q and R are learned using the query-target pairs $\langle(T^{[m]}, e_i^{[m]}, l_i^{[m]}, V_{i-1}^{[m]}), (V^{[m]}, l_i^{[m]}, s_i^{[m]})\rangle_{i,m}$ in the training data. For clarity, we abbreviate $Q(T^{[m]}, e_i^{[m]}, l_i^{[m]}, V_{i-1}^{[m]})$ as $q_i^{[m]}$ and $R(V^{[m]}, l_i^{[m]}, s_i^{[m]})$ as $r_i^{[m]}$. At each training iteration, we sample a mini-batch of B pairs without replacement and compute embeddings $\{(q_{i_b}^{[m_b]}, r_{i_b}^{[m_b]})\}_{b=1}^{B}$ where q and r are each sequence of F embeddings corresponding to F video frames. The model is trained using a triplet loss computed on *all possible* triplets in the mini-batch. Let δ_b denote the set of all indices from 1 to B except b. The loss can then be defined as

$$\mathcal{L}_{triplet} = \frac{1}{B \cdot (B-1)} \sum_{b=1}^{B} \sum_{b^- \in \delta_b} \Big[\max(0, \gamma + q_{i_b}^{[m_b]} \odot r_{i_{b^-}}^{[m_{b^-}]} - q_{i_b}^{[m_b]} \odot r_{i_b}^{[m_b]}) + \max(0, \gamma + q_{i_{b^-}}^{[m_{b^-}]} \odot r_{i_b}^{[m_b]} - q_{i_b}^{[m_b]} \odot r_{i_b}^{[m_b]}) \Big] \quad (6.3)$$

where $q \odot r = \frac{1}{F} \sum_{f=1}^{F} q[f] \cdot r[f]$ is the average dot product between corresponding query and target frame embeddings. We use a margin of $\gamma = 0.1$.

Auxiliary Multi-label Classification Loss We found models trained using triplet loss alone could simply learn a one-to-one mapping between ground truth text and entity video segments with poor generalization to unseen captions and database videos. To guide the learning to utilize the compositional nature of text and improve generalization, we add an auxiliary classification loss on the embeddings. The idea is to enrich the semantics of the embedding vectors by predicting the noun, adjectives, and action words directly associated with the entity in the description. For example, in the sentence *Fred is talking to a happy Wilma who is sitting on a chair*, *Wilma*'s embedding produced by the query and target embedding networks are forced to predict *Wilma*, *happy* and *sitting* ensuring their representation in the embeddings. A vocabulary $\mathcal{W}$ is constructed of all nouns, adjectives and verbs in the training data. Then for each sample in the mini-batch, an MLP is used as a multi-label classifier to predict associated words from the query and target embeddings. Note that a single MLP is used to make these noun, adjective and verb predictions on *both* query and target embeddings.

Query Embedding Network (Q). Similar to the layout composer's feature computation backbone, Q consists of a CNN to independently encode every frame of V_{i-1} and an LSTM to encode (T, e_i) which are concatenated together along with a 2-D coordinate grid to get per-frame feature maps. However, unlike layout composer, the query embedding network also needs to be conditioned on the position l_i where entity E_i is to be inserted in V_{i-1}. To get location and scale specific query embeddings, we use a simplified RoIAlign (RoIPool with RoI quantization and bilinear interpolation) mechanism to crop out the per-frame feature maps using the corresponding bounding box l_i^f and scaling it to a 7×7 receptive field. The RoIAlign features are then averaged along the spatial dimensions to get the vector representations for each time step independently. An LSTM applied over the sequence of these embeddings is used to capture temporal context. The hidden output of the LSTM at each time step is normalized and used as the frame query embedding $q[f]$.

Target Embedding Network (R). Since during inference, R needs to embed entities in the target database which do not have text annotations, it does not use T as an input. Thus, R is similar to Q but without the LSTM to encode the text. In our experiments we found that using 2-D coordinate features in both query and target networks made the network susceptible to ignoring all other features since it provides an easy signal for matching ground truth query-target pairs during training. This in turn leads to poor generalization. Thus, R has no 2-D coordinate features.

6.4.3 Background Retriever

The task of the background retriever is to find a background scene that matches the setting described in the description. To construct a database of backgrounds without characters in them, we remove characters from videos (given bounding boxes) and perform hole filling using PatchMatch [173]. The background retriever model is similar to the entity retriever with two main differences. First, since the whole background scene is retrieved instead of entity segments, the conditioning on position is removed from both query and database embedding networks replacing RoI pooling with global average pooling. Second, while ideally we would like scene and entity retrieval to be conditioned on each other, for simplicity we leave this to future work and currently treat them independently. These modifications essentially reduce the query embedding network to a text Bi-LSTM whose output at the background word location in the description is used as the query embedding, and the target embedding network to a video Bi-LSTM without RoI pooling. The model is trained using just the triplet loss.

6.5 EXPERIMENTS

6.5.1 Layout Composer Evaluation

Metrics. We evaluate layout composer using 2 metrics: (a) negative log-likelihood (NLL) of ground truth (GT) entity positions under the predicted distribution, and (b) average normalized pixel distance (coordinates normalized by image height and width) of the ground truth from the most likely predicted entity location. While NLL captures both location and scale, pixel distance only measures location accuracy. We report metrics on unseen test descriptions using ground truth locations and appearances for previous entities in the partial video.

Feature Ablation. The ablation study in Table 6.1 shows that the layout composer benefits from each of the 3 input features – text, scene context (partial video), and 2D coordinate grid. The significant drop in NLL without text features indicates the importance of entity identity, especially in predicting scale. The lack of spatial awareness in convolutional feature maps without the 2D coordinate grid causes pixel distance to approximately double. The

Text	Scene Context	2D Coord. Grid	Dil. Conv	NLL	Pixel Dist.
Uniform Distribution				>9.704	>0.382
✗	✓	✓	✓	9.845	0.180
✓	✗	✓	✓	8.167	0.185
✓	✓	✗	✓	8.250	0.287
✓	✓	✓	✗	7.780	0.156
✓	✓	✓	✓	**7.636**	**0.148**

Table 6.1: **Layout Composer Analysis.** Evaluation of our model (last row) and ablations on test set. First row provides theoretically computed values assuming a uniform location distribution while making no assumptions about the scale distribution.

performance drop on removing scene context is indicative of the relevance of knowing *what* entities are *where* in the scene in predicting the location of next entity. Finally, replacing vanilla convolutions by dilated convolutions improves performance by increasing the spatial receptive field without increasing the number of parameters. This corroborates the usefulness of scene context in layout prediction.

6.5.2 Entity Retriever Evaluation.

Metrics. To evaluate semantic fidelity of retrieved entities to the query caption, we measure noun, adjective, and verb recalls (@1 and @10) averaged across entities in the test set. The captions are automatically parsed to identify nouns, adjectives and verbs associated with each entity both in the query captions and target database (using GT database captions for evaluation only). Note that captions often contain limited adjective and verb information. For example, a *red hat* in the video may only be referred to as a *hat* in the caption, and *Fred standing and talking* may be described as *Fred is talking*. We also do not take synonyms (*talking-speaking*) and hypernyms (*person-woman*) into account. Thus the proposed metric underestimates performance of the entity retriever.

Feature Ablation. Table 6.2 shows that text and location features are critical to noun, adjective and verb recall. Scene context only marginally affects noun recall but causes significant drop in adjective and verb recalls.

Effect of Auxiliary Loss. Table 6.3 shows that triplet loss alone does significantly worse than in combination with auxiliary classification loss. Adding the auxiliary classification loss on either query or target embeddings improves over triplet only but is worse than using all three. Interestingly, using both auxiliary losses outperforms triplet loss with a single auxiliary loss (and triplet only) on adjective and verb recall. This strongly suggests the benefits

Query Features			Recall@1			Recall@10		
Text	Context	Location	Noun	Adj.	Verb	Noun	Adj.	Verb
✗	✓	✓	24.88	3.04	9.48	55.22	19.39	37.18
✓	✗	✓	60.54	9.5	11.2	**77.71**	39.92	43.58
✓	✓	✗	56.14	8.56	11.34	73.03	39.35	41.48
✓	✓	✓	**61.19**	**12.36**	**14.77**	75.98	**47.72**	**46.86**

Table 6.2: **Entity retriever feature ablation.** Top-1 and top-10 recalls of our model (last row) and ablations while generating videos for unseen test captions.

of multi-task training in entity retrieval.

Generalization to unseen videos. A key advantage of the embedding based text to entity video retrieval approach over text only methods is that the embedding approach can use any unseen video databases without any text annotations, potentially in entirely new domains (eg. learning from synthetic video caption datasets and applying the knowledge to generate real videos). However, this requires a model that generalizes well to unseen captions as well as unseen videos. In Table 6.4 we compare entity recall when using the train set (seen) videos as the target database vs using the test set (unseen) video as the target database.

	Auxiliary Loss		Recall@1			Recall@10		
Triplet	Query	Target	Noun	Adj.	Verb	Noun	Adj.	Verb
✗	✓	✓	35.75	7.79	8.83	63.62	43.35	33.12
✓	✗	✓	51.68	3.8	8.66	67.86	25.28	39.46
✓	✓	✗	50.54	4.94	9.94	66.36	28.52	39.5
✓	✗	✗	48.59	3.04	9.34	65.64	20.15	37.95
✓	✓	✓	**61.19**	**12.36**	**14.77**	**75.98**	**47.72**	**46.86**

Table 6.3: **Entity retriever loss ablation.** Top-1 and top-10 recalls of our model (last row) and ablations while generating videos for unseen test captions.

Video Database	Recall@1			Recall@10		
	Noun	Adj.	Verb	Noun	Adj.	Verb
Seen (Train)	61.19	12.36	14.77	75.98	47.72	46.86
Unseen (Test)	50.52	11.98	10.4	69.1	41.25	42.57

Table 6.4: **Generalization to Unseen Database Videos.** Retrieval results for Craft when queried against seen videos vs unseen videos.

Modelling Whole Video vs Entities. A key motivation to composing a scene from entities is the combinatorial nature of complex scenes. To illustrate this point we compare Craft to a text-to-text based whole video retrieval baseline. For a given test caption, we

	Composition Consistency			Visual Quality		
	Position	**Rel. Size**	**Interact.**	**FG**	**BG**	**Sharpness**
Pixel Generation L1	0.69	0.65	0.55	0.96	1.44	1.07
Ours (GT Position)	1.69	1.69	1.34	1.49	1.65	**2.16**
Ours	**1.78**	**1.86**	**1.46**	**1.98**	**1.95**	1.82

Table 6.5: **Human evaluation** to estimate consistency and quality of generated videos.

return a video in the database whose caption has the highest BLEU-1 score. This approach performs much worse than our model except on verb recall (BLEU: $49.57, 5.18, 26.64$; Ours: $62.3, 21.7, 16.0$). This indicates that novel captions often do not find a match in the target database with all entities and their attributes present in the same video. However, it is more likely that each entity and attribute combination appears in some video in the database. Note that text-to-text matching also prevents extension to unseen video databases without text annotations.

6.5.3 Human Evaluation

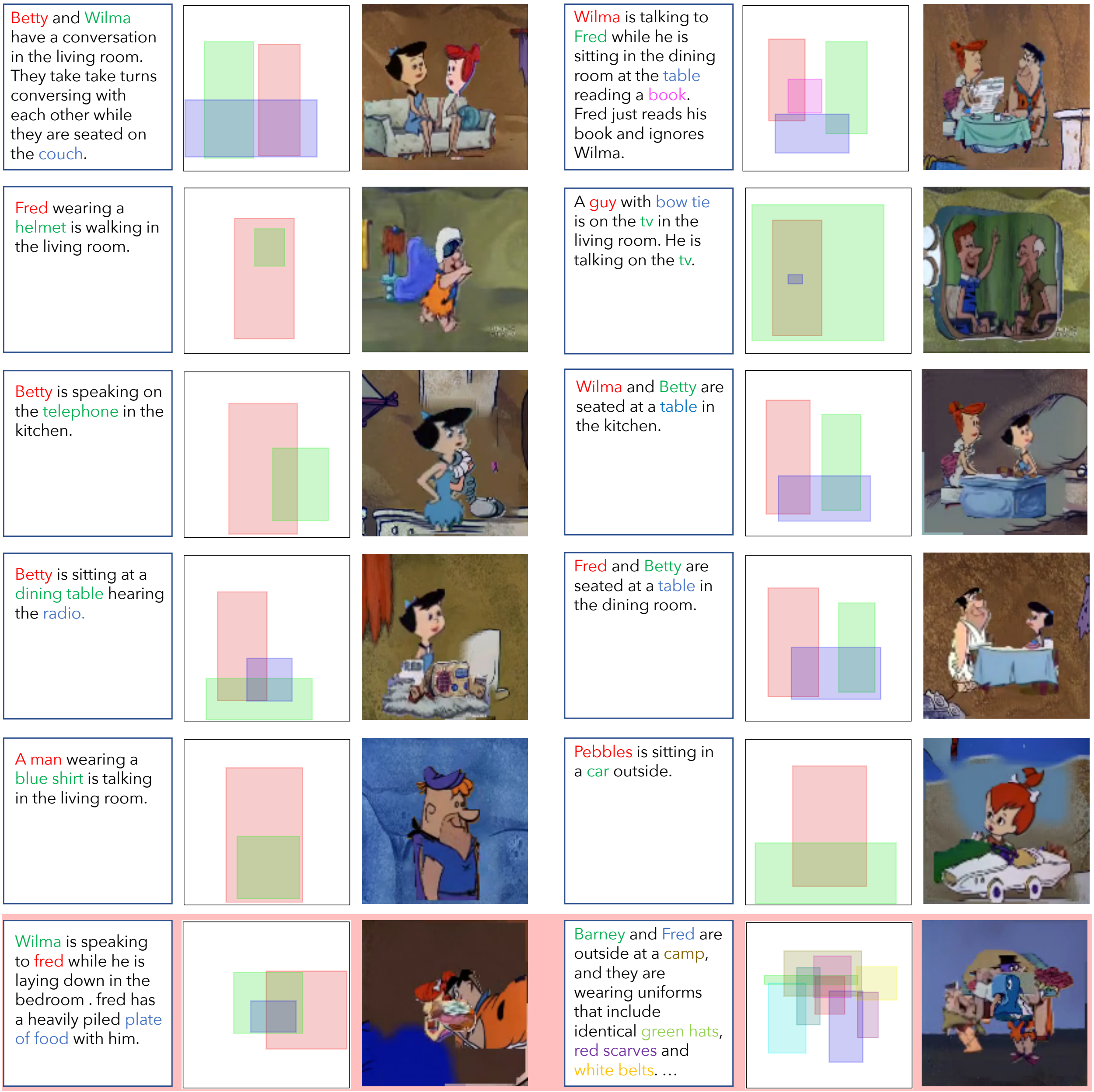
Metrics. In addition to the automated recall metrics which capture semantic fidelity of the generated videos to the captions, we run a human evaluation study to estimate the *compositional consistency* of entities in the scene (given the description) and the overall *visual quality* (independent of the description). The consistency metric requires humans to rate each entity in the video on a 0-4 scale on three aspects: (a) *position* in the scene, (b) *size* relative to other entities or the background, and (c) appearance and consistency of described *interactions* with other entities in the scene. The visual quality metric measures the aesthetic and realism of the generated scenes on a 0-4 scale along three axes: (a) *foreground quality*, (b) *background quality*, and (c) *sharpness*. See supplementary material for the design of these experiments.

Generating Pixels (Parametric) vs Retrieval (Semi-Parametric). We experimented extensively with text conditioned whole video generation using models with and without adversarial losses and obtained poor results. Since generative models tend to work better on images with single entities, we swapped out the target embedding network in the entity retriever by a generator. Given the query embedding at each of the F time steps, the generator produces an appearance image and a segmentation mask. The model is trained using an $L1$ loss between the masked appearance image and the masked ground truth image, and an $L1$ loss between the generated and ground truth masks. See supplementary material for more details. This baseline produced blurry results with recognizable colors and shapes for most common characters like *Fred, Wilma, Barney*, and *Betty* at best. We also tried GAN and VAE based approaches and got only slightly less blur. Table 6.5 shows that this model performs poorly on the Visual Quality metric compared to CRAFT. Moreover, since the visual quality of the generated previous entities affects the performance of the layout composer, this also translates into poor ratings on the composition consistency metric. Since the semantic fidelity metrics can not be computed for this pixel generation approach, we ran a human evaluation to compare this model to ours. Humans were asked to mark nouns, adjectives and verbs in the sentence missing in the generated video. CRAFT significantly outperformed the pixel generation approach on noun, adjective, and verb recall (CRAFT : $61.0, 54.5, 67.8$, L1: $37.8, 45.9, 48.1$).

Joint vs Independent Modelling of Layout. We compare CRAFT to a model that uses the same entity retriever but with ground truth (GT) positions. Using GT positions performed worse than CRAFT (GT: $62.2, 18.1, 12.4$; Full: $62.3, 21.7, 16.0$ Recall@1). This is also reflected in the composition consistency metric (GT: $1.69, 1.69, 1.34$; Full: $1.78, 1.89, 1.46$). This emphasizes the need to model layout composition and entity retrieval jointly. When using GT layouts, the retrieval gets conditioned on the layout but not vice versa.

6.6 CONCLUSION

In this chapter we showed how to model interactions between entities described in a scene to predict a feasible layout and appearances of entities. Our proposed approach is a semi-parametric alternative to fully parametric GAN and VAE based generative approaches commonly used in the literature. We perform a thorough ablation study to illustrate the ability of our model to understand natural language scene description, use context (in the form of position and appearance of other entities in the scene), model appearance and layout jointly, and generalize to both unseen query captions and target video databases. Our contributions from a representation perspective include contextual representations of different entities in a given sentence and video, and an auxiliary multi-label classification loss that encourages compositional representations.



Betty and Wilma have a conversation in the living room. They take take turns conversing with each other while they are seated on the couch.

Wilma is talking to Fred while he is sitting in the dining room at the table reading a book. Fred just reads his book and ignores Wilma.

Fred wearing a helmet is walking in the living room.

A guy with bow tie is on the tv in the living room. He is talking on the tv.

Betty is speaking on the telephone in the kitchen.

Wilma and Betty are seated at a table in the kitchen.

Betty is sitting at a dining table hearing the radio.

Fred and Betty are seated at a table in the dining room.

A man wearing a blue shirt is talking in the living room.

Pebbles is sitting in a car outside.

Wilma is speaking to fred while he is laying down in the bedroom . fred has a heavily piled plate of food with him.

Barney and Fred are outside at a camp, and they are wearing uniforms that include identical green hats, red scarves and white belts. ...

Figure 6.6: **Qualitative results** for Craft. Last row shows failures of the layout composer (left) and the entire system (right). See `https://youtu.be/688Vv86n0z8` for video examples, failure cases, and visualization of predicted location and scale distributions

CHAPTER 7: HUMAN-OBJECT INTERACTION DETECTION: FACTORIZATION, LAYOUT ENCODINGS, AND TRAINING TECHNIQUES

7.1 INTRODUCTION

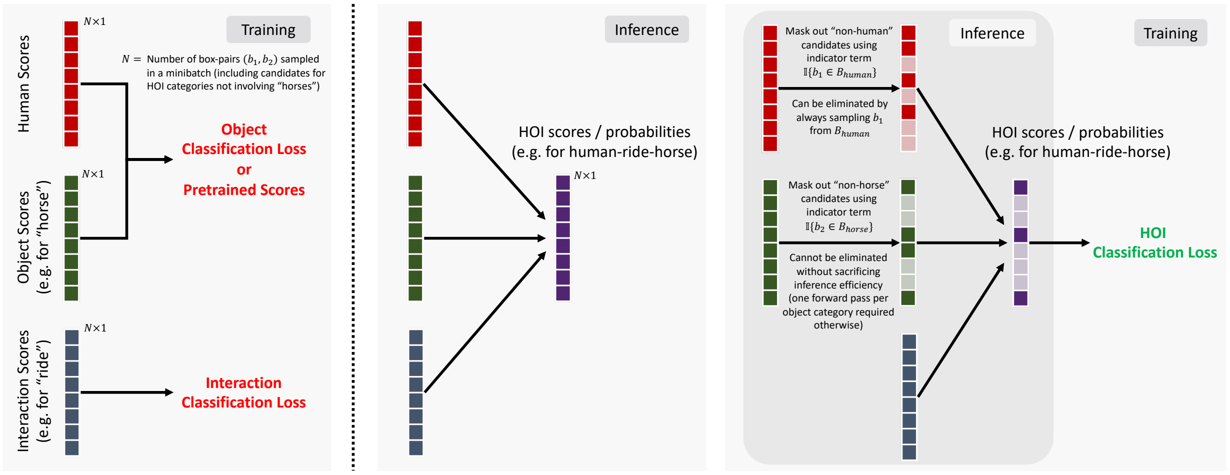
Continuing with the idea of modeling interactions between entities in an image, in this chapter, we focus on the task of human-object interaction (HOI) detection. Given an image, the task is to localize and recognize a predetermined set of human-object interactions. For instance, detecting the HOI "human-row-boat" refers to localizing a "human", a "boat", and predict the interaction "row" for this human-object pair. Note that an image may contain multiple people rowing boats (or even the same boat), and the same person could simultaneously be interacting with the same or a different object in different ways. For example, a person can simultaneously "sit on" and "row" a boat while "wearing" a backpack.

Recently, increasingly sophisticated techniques have been proposed for encoding position and appearance for HOI detection. For instance, Chao *et al*. [204] encode the configuration of human-object box pairs using a CNN operating on a two channel binary image called the *interaction pattern*. Gkioxari *et al*. [205] predict a distribution over target object locations based on human appearance using a mixture density network [206]. For encoding appearance, approaches range from multitask training of a human-centric branch [205] alongside object classification, to using an attention mechanism which gathers contextual information from the image [207].

In this work, we propose a no-frills model for HOI detection. In contrast to sophisticated end-to-end models, we use appearance features from pretrained object detectors, and encode layout using hand-crafted bounding-box coordinate features (optionally human pose keypoints). Our network architecture is also modest, comprising of light-weight multi-layer perceptrons (MLPs) that operate on these appearance and layout features. In spite of these simplifications, our model achieves state-of-the-art performance on the challenging HICO-Det dataset.

Our gains are due to the choice of factorization, direct encoding and scoring of layout, and improved training techniques. Our model consists of human/object detection terms and an interaction term. The interaction term further consists of human and object appearance, box-configuration, and pose or fine-layout factors. We perform a thorough ablation study to evaluate the effect of each factor.

In contrast to existing work, which needs to train a CNN [204] or a mixture density network [205] to encode layout, we use hand-crafted absolute and relative position features


Training
$N\times1$
Human Scores
$N =$ Number of box-pairs (b_1, b_2) sampled in a minibatch (including candidates for HOI categories not involving "horses")
Object Classification Loss or Pretrained Scores
$N\times1$
Object Scores (e.g. for "horse")
$N\times1$
Interaction Scores (e.g. for "ride")
Interaction Classification Loss
Inference
HOI scores / probabilities (e.g. for human-ride-horse)
$N\times1$
Mask out "non-human" candidates using indicator term $\mathbb{I}\{b_1 \in B_{human}\}$
Can be eliminated by always sampling b_1 from B_{human}
Inference
Training
HOI scores / probabilities (e.g. for human-ride-horse)
Mask out "non-horse" candidates using indicator term $\mathbb{I}\{b_2 \in B_{horse}\}$
Cannot be eliminated without sacrificing inference efficiency (one forward pass per object category required otherwise)
HOI Classification Loss

(a) Training and inference mismatch in previous state-of-the-art models

(b) Proposed approach with easy negative rejection using indicator terms

Figure 7.1: **Fixing training-inference mismatch and rejecting easy negatives.** The figure illustrates training and inference on a single HOI class ("human-ride-horse") for simplicity. As shown in (a), existing models [205, 207] often train human/object and interaction branches using object and interaction classification losses respectively. The scores produced by these branches are combined during inference to produce HOI scores. Hence, training does not reflect the inference objective. Our model, shown in (b), fixes this mismatch by optimizing the combined scores using a multi-label HOI classification loss. Our model also rejects easy negative box-pairs (or keeps only "human-horse" box pairs) during training and inference using the sets of detections selected for human and object categories (B_{human}, B_{horse}). While existing approaches also select detection candidates, the models are typically trained using minibatches containing candidates for different HOI/object categories.

computed from bounding boxes or human pose keypoints. Our choice is motivated by the observation illustrated in Fig. 7.1: pretrained object and pose detectors provide strong geometric cues for interaction prediction.

We also develop the following training techniques for improving learning efficiency of our factored model:

(1) **Eliminating train-inference mismatch.** [205, 207] learn detection and interaction terms via separate detection and interaction losses. During inference, the scores of all factors are simply multiplied to get final HOI class probabilities. Instead, we directly optimizing the HOI class probabilities using a multi-label HOI classification loss (Fig. 7.1) (Interaction Loss: 15.89 mAP *vs*. HOI Loss: 16.96 mAP).

(2) **Rejecting easy negatives using indicator terms.** Rejecting easy negatives is beneficial not only during test but also during training because it allows the model to focus

on learning to score hard negatives. We generate a candidate box-pair (b_1, b_2) using a pre-trained object detector which is then scored by the factor model. If either b_1 is not a "human" candidate (category h) or b_2 is not an object candidate o, then the factor model should predict a 0 probability of (b_1, b_2) belonging to HOI category (h, o, i) for *any* interactions i. This is achieved by including *indicator* terms in our object detection factors and can be implemented efficiently by applying a mask on predicted probabilities constructed from labels predicted by the object detector (Fig. 7.1) (w/o indicators: 15.93 mAP *vs.* w indicators: 16.96 mAP).

(3) **Training with large negative to positive ratio.** We construct training mini-batches by sampling a two orders of magnitude larger number of negative box-pairs per positive pair than related work (1000 *vs.* < 10). Higher ratios compared to *object* detector training are expected since the number of negative pairs is quadratic in the number of object proposals as opposed to being linear for object detectors (neg. to pos. ratio 10: 13.40 mAP *vs.* 1000: 16.96 mAP).

In summary, our key contributions are: (1) a simple but competitive model for HOI detection that takes advantage of appearance and layout encodings from a pre-trained object detector (and optionally a pose detector); (2) a comparison of coarse and fine-grained layout encodings; and (3) techniques for enhancing learning efficiency of our model.

7.2 RELATED WORK

Assessing interactions between humans and objects in images is a challenging problem which has received a considerable amount of attention from the machine learning, computer vision and robotics community in the last decade [208, 209, 210, 211, 212, 213].

Human activity recognition is among the early efforts to analyze human actions in images or videos. Benchmarks such as UCF101 [214] and THUMOS [215] focused on classifying a video sequence into one of 101 action categories. While UCF101 only dealt with carefully trimmed videos, an artificial setting, the THUMOS challenge additionally introduced the task of temporal localization of activities in untrimmed videos. Image action recognition benchmarks such as Stanford 40 Actions [216] and PASCAL VOC 2010 [213] have also been used in the literature. While similar in intent, these action recognition challenges differ from human-object interaction detection in three ways – (1) the tasks are limited to images or videos containing a *single* human-centric action, such as bowling, diving, fencing, *etc.*; (2)

the action classes are *disjoint* and often involve interaction with an object unique to the activity (allowing models to cheat by simply recognizing the object); and (3) spatial localization of neither the person nor the object being interacted with is required.

Moving from actions to interactions, Chao *et al.* [217, 204] introduce the HICO and HICO-DET datasets to address the above limitations. The HICO dataset consists of a large collection of images annotated with 600 human-object interactions with a diverse set of 117 interactions with 80 COCO [218] object categories. Unlike previous tasks, HOI classification is multi-label in nature since each image may contain multiple humans interacting with same or different objects. Recently, Chao *et al.* extended the HICO dataset with exhaustive bounding box annotations for each of the HOI classes to create HICO-DET. Due to the human-centric nature of the annotation task and predefined set of objects and interactions, HICO-DET does not suffer from the missing annotation problem (at least to the same extent) that plagues datasets such as Visual Genome [219] and VRD [220] that are used for the general visual relationship (object-object interaction) detection task.

In a similar effort, Gupta *et al.* [221] augment the COCO dataset [218] by annotating people (agents) with one of 26 action labels along with location and labels of objects fulfilling various semantic roles for the action. In another visual equivalent of the semantic role labelling (SRL) task studied in NLP, Yatskar *et al.* [222] create an image dataset for situation recognition, which is defined to subsume recognition of activity, participating objects and their roles.

In this work, we choose HICO-DET as a test bed for experimentation due to its large, diverse, and exhaustively annotated set of human-object interactions which allows for an accurate and meaningful evaluation. The task is also a natural extension of classical object detection to detection of human-object pairs with interaction labels. In contrast, the visual-SRL task is further complicated by varying number of semantic roles for each action.

Existing models for HOI detection. In [204] Chao *et al.* propose HO-RCNN, a 3-stream architecture with one stream each for a human candidate, an object candidate, and a geometric encoding of the pair of boxes using the proposed *interaction pattern*. Each stream produces scores for every possible object-interaction category (600 for HICO-DET). The 3 set of scores are combined using late-fusion to make the final prediction. Note that this approach treats “ride bicycle” and “ride horse” as independent visual entities and does not use the knowledge of “ride” being a common component. In contrast, our approach exploits this compositionality to learn shared visual appearance and geometric representations (*e.g.*, “ride” typically involves a human box above an object box). In other words, weight sharing between different HOI classes in our factored model makes it more data efficient than [204] which predicts scores for 600 HOI categories using independent weights in the last 600-way fully connected layer in each of the 3 streams.

Gkioxari *et al.* [205] take a multitask learning [40] perspective on this problem. The idea is to augment the Faster-RCNN [223] object detection framework with a human-centric branch and an interaction branch that are trained jointly alongside the original object recognition branch. To incorporate geometric cues, a Mixture Density Network (MDN) [206] is used to produce parameters of the object location distribution given the location of the human box. This distribution is used to score candidate objects for a given human box. The model is trained using object classification loss for the object branch, interaction classification losses for the human centric action classification branch and the optional interaction branch, and a smooth L1 loss between the ground truth box-pair encoding and mean predicted by the localization MDN. During inference, predictions from these branches are fused heuristically. In contrast, we optimize the final HOI score obtained after fusing the individual factor scores. We also more directly encode box-pair layout using absolute and relative bounding box features which are then scored using a dedicated factor.

7.3 METHOD

In the following, we first present an overview of the proposed factor model, followed by details of the potentials which encode appearance, box configuration, and optionally human pose. Finally, we discuss our strategy for learning these factors from annotated box pairs.

7.3.1 Overview

Given an image x and a set of object-interaction categories of interest, human-object interaction (HOI) detection is the task of localizing all human-object pairs participating in one of the said interactions. The combinatorial search over human and object bounding-box locations and scales, as well as object labels, $\mathcal{O}$, and interaction labels, $\mathcal{I}$, makes both learning and inference challenging. To deal with this complexity, we decompose inference into two stages. In the **first stage**, object category specific bounding box candidates $B_o \; \forall o \in \mathcal{O}$ are selected using a *pre-trained* object detector such as Faster-RCNN. For each HOI category, *i.e.*, for each triplet $(h, o, i) \in \mathcal{H}$, a set of candidate human-object box pairs is constructed by pairing every human box candidate $b_h \in B_h$ with every object box candidate $b_o \in B_o$ of the corresponding object class $o \in \mathcal{O}$. In the **second stage**, an HOI *category specific* factored model is used to score and rank candidate box pairs $(b_h, b_o) \in B_h \times B_o$ for each HOI category.

Algorithm 7.1: Inference on a single image

```
   Input  : Image $x$,
            Set of object ($\mathcal{O}$), interaction ($\mathcal{I}$), and HOI ($\mathcal{H} \subseteq \{\text{human}\} \times \mathcal{O} \times \mathcal{I}$) classes of interest,
            Pretrained Faster-RCNN object detector and OpenPose human keypoints detector
   // Stage 1: Create a set of box candidates for each object (including human)
1  Run Faster-RCNN on $x$ to get $\forall\, o \in \mathcal{O}$, 300 region proposals ($R_o$) with ROI appearance features and
     detection probabilities for class $o$
2  foreach $o \in \mathcal{O}$ do
3      Construct $B_o = \{b \in R_o | \, b$ survives NMS (threshold 0.3) and $P_{det}(l_{det} = o|b, x) > 0.01\}$
4      Update $B_o$ to keep at most 10 highest ranking detections.
5  end
6  Run OpenPose on $x$ to get skeletal-keypoints $k(b) \;\forall\; b \in B_h$ (set of human boxes)
   // Stage 2: Score candidate pairs using the proposed factored model
7  foreach $(h, o, i) \in \mathcal{H}$ do
8      foreach $b_h \in B_h$ do
9          foreach $b_o \in B_o$ do
10             Compute box configuration features for $(b_h, b_o)$
11             Compute fine grained pose features for $(k(b_h), b_h, b_o)$
12             Compute $P(y_{(h,o,i)} = 1|b_1, b_2, x)$ using equations 7.1,  7.2, and 7.3
13         end
14     end
       Output: Ranked list of $(b_h, b_o) \in B_h \times B_o$ as detections with probabilities for class $(h, o, i)$
15 end
   // Steps 7-15 are implemented with a single forward pass on a mini-batch of
      precomputed features
```

Our factor graph consists of human and object appearance, box pair configuration and human pose potentials that encode visual and spatial knowledge useful for understanding human-object interactions. The model is parameterized to share representations and computation across different object and interaction categories to efficiently score candidate box pairs for all HOI categories of interest in a single forward pass. See Alg. 7.1 for a detailed description of the inference procedure.

7.3.2 Factored Model

For an image x, given a human-object candidate box pair (b_1, b_2), human pose keypoints $k(b_1)$ detected inside b_1 (if any), and the set of box candidates for each object category, the factored model computes the probability of occurrence of human-object interaction (h, o, i) in (b_1, b_2) as follows:

$$\begin{aligned}
&P(y_{(h,o,i)} = 1|b_1, b_2, x, o, k(b_1), B_h, B_o) \\
&= P(y_h = 1, y_o = 1, y_i = 1|b_1, b_2, x, o, k(b_1), B_h, B_o) \\
&= P(y_h = 1|b_1, x, B_h) \cdot P(y_o = 1|b_2, x, B_o) \cdot P(y_i = 1|b_1, b_2, k(b_1), o, x)
\end{aligned} \tag{7.1}$$

Here, $y_h \in \{0, 1\}$ is a random variable denoting if b_1 is labeled as a human, $y_o \in \{0, 1\}$ denotes if b_2 is labeled as object category o, and $y_i \in \{0, 1\}$ denotes if the interaction assigned to the box-pair is i. The above factorization assumes that human and object class labels depend on the individual boxes and the image, while the interaction label depends on the box-pair, pose, object label under consideration, and the image. For brevity, we will refer to the left hand side of the above equation as $P(y_{(h,o,i)} = 1|b_1, b_2, x)$. We now describe how the 3 terms are modelled.

7.3.2.1 Detector Terms

The first two terms in Eq. 7.1 are modelled using the set of candidate bounding boxes for each object class and classification probabilities produced by a pretrained object detector. For any object category c (including h), the detector term can be computed as

$$P(y_c = 1|b, x, B_c) = \mathbb{1}(b \in B_c) \cdot P_{det}(l_{det} = c|b, x), \tag{7.2}$$

where the P_{det} term corresponds to the probability of assigning object class c to region b in image x by the object detector. The indicator term checks if the region belongs to the set of candidate bounding boxes for c selected from the set of all region proposals using non-maximum suppression and thresholding on class probabilities.

7.3.2.2 Interaction Term

Interaction term refers to the probability of entities in b_1 and b_2 engaging in interaction $i \in \mathcal{I}$. Note that the interaction term is conditioned on the object label o. This allows the model to learn that only certain interactions are feasible for a given object. For example, it is possible to "clean" or "eat at" a "dinning table" but not to "drive" or "greet" it. In practice, we found conditioning on o did not affect results significantly. To capture visual and spatial knowledge required for predicting interactions given human box, object box, human pose and the object label, the interaction term $P_{int}(y_i = 1|b_1, b_2, k(b_1), o, x)$ is written as

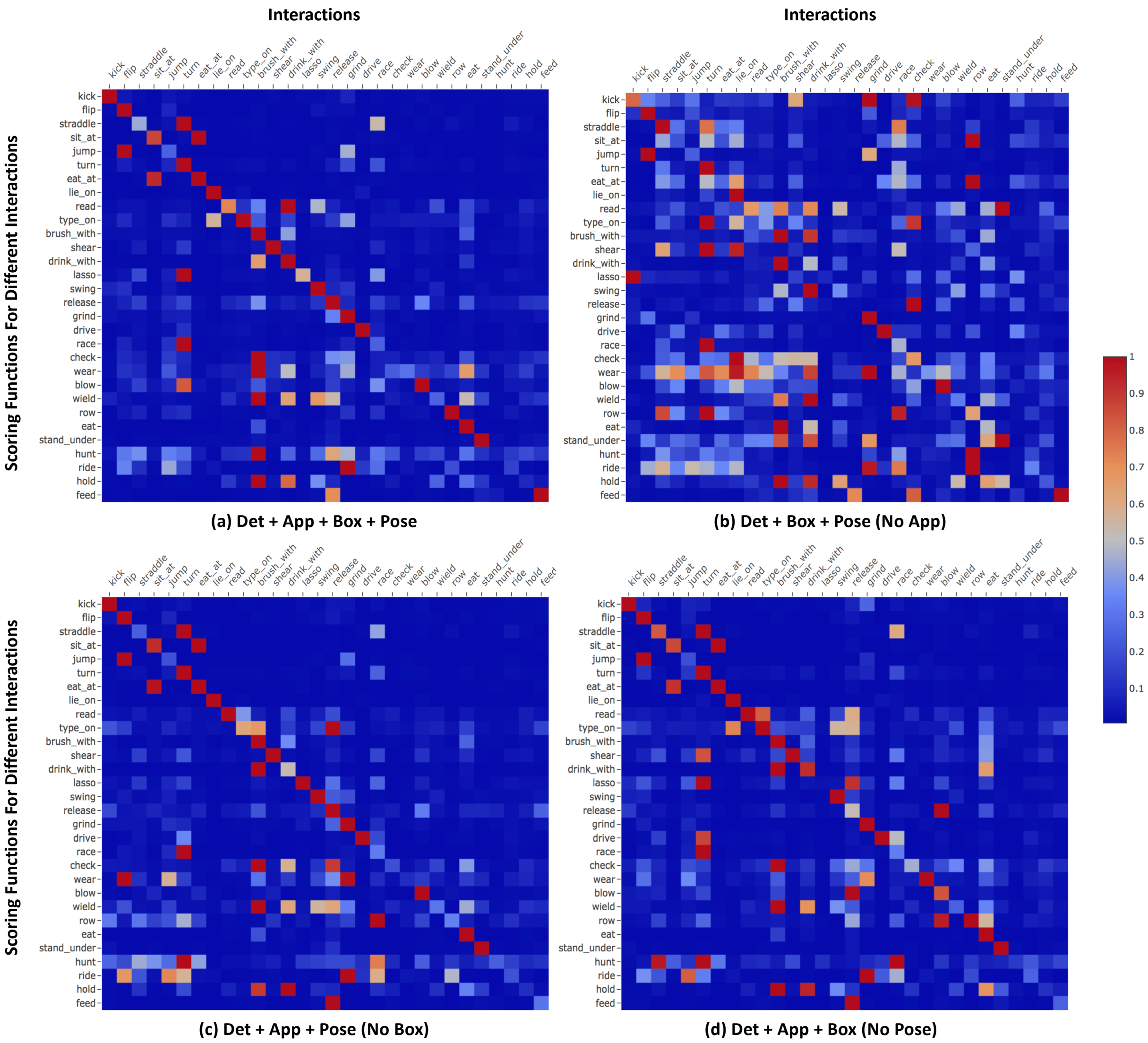
$$\sigma\left(\phi_{human}(i|b_1, x) + \phi_{object}(i|b_2, x) + \phi_{boxes}(i|b_1, b_2, o) + \phi_{pose}(i|b_1, b_2, k(b_1), o)\right), \tag{7.3}$$

where σ is the Sigmoid function and each of the feature functions ϕ is a learnable deep net factor. The information captured by each factor along with input data and network architecture are as follows:

Appearance. Factors ϕ_{human} and ϕ_{object} predict the interaction that the human and the object are engaged in, based on visual appearance alone. The appearance of a box in an image is encoded using Faster-RCNN [223] *fc7* features extracted from the RoI. By design, this representation captures context in addition to content within the box. The 2048 dimensional *fc7* features are fed into a multi-layer perceptron (MLP) with a single 2048 dimensional hidden layer with Batch Normalization [168] and ReLU [224]. The output layer has 117 neurons, one per interaction category in $\mathcal{I}$.

Box Configuration. Object label and the absolute and relative positions and scales of the human and object boxes are often indicative of the interaction, without even knowing the appearance (*e.g.*, a human box above and overlapping with a 'horse' box strongly suggests a 'riding' interaction). ϕ_{boxes} encodes this intuition by predicting a score for each interaction given an encoding of the bounding boxes and the object label. The object label is encoded as a $|\mathcal{O}|$ $(= 80)$ dimensional one hot vector. The bounding boxes are represented using a 21 dimensional feature vector. We encode the *absolute position and scale* of both the human and object boxes using box width, height, center position, aspect ratio, and area. We also encode *relative configuration* of the human and object boxes using relative position of their centers, ratio of box areas and their intersection over union. These 21 dimensional features are concatenated with their log absolute values and the object label encoding and passed through an MLP with 2 hidden layers, 122 $(= 2 \times 21 + 80)$ dimensional each (same as the input feature dimension), with Batch Normalization and ReLU.

Human Pose. We supplement the coarse layout encoded by bounding boxes with more fine-grained layout information provided by human pose keypoints. We use OpenPose [225, 226, 227] to detect 18 keypoints for each person in the image. A human candidate box is assigned a keypoints-skeleton if the smallest bounding box around the keypoints has 70% or more of its area inside the human box. Similar to box features, we encode both absolute human pose and the relative location with respect to the object candidate box. The absolute pose features $(18 \times 3 = 54)$ consist of keypoint coordinates normalized to the human bounding box frame of reference and confidence of each keypoint predicted by OpenPose. The relative pose features $(18 \times 5 = 90)$ consists of offset of the top left and bottom right corners of the object box relative to each keypoint and keypoint confidences. The absolute and relative pose features and their log values are concatenated along with one hot object label encoding before feeding into ϕ_{pose}. ϕ_{pose} is also an MLP with 2 hidden layers with 368 $(= 2 \times (54 + 90) + 80)$ neurons each. Both hidden layers are equipped with Batch Normalization and ReLU. The output layer, like the other factors, has 117 neurons.



(a) Det + App + Box + Pose

(b) Det + Box + Pose (No App)

(c) Det + App + Pose (No Box)

(d) Det + App + Box (No Pose)

Figure 7.2: **Interaction confusions.** Element (m, n) in each heatmap visualizes $P(y_{i_m} = 1|b_1, b_2, k(b_1), o, x)$, the probability of interaction $i_m \in \mathcal{I}$ for box-pair (b_1, b_2), averaged across all box pairs with ground truth interaction $i_n \in \mathcal{I}$. Each row m is independently normalized and exponentiated to highlight the interactions most confused with interaction i_m. Only 30 of the 117 classes with the highest median AP across objects (see Fig. 7.4) are shown for clarity.

7.3.3 Training

Since more than one HOI label might be assigned to a pair of boxes, the model is trained in a fully supervised fashion using the multi-label binary cross-entropy loss. For each image in the training set, all candidate boxes for all HOI classes ($B_h \times B_o$ for class (h, o, i)) are assigned a binary label based on whether both the human and object candidate boxes have

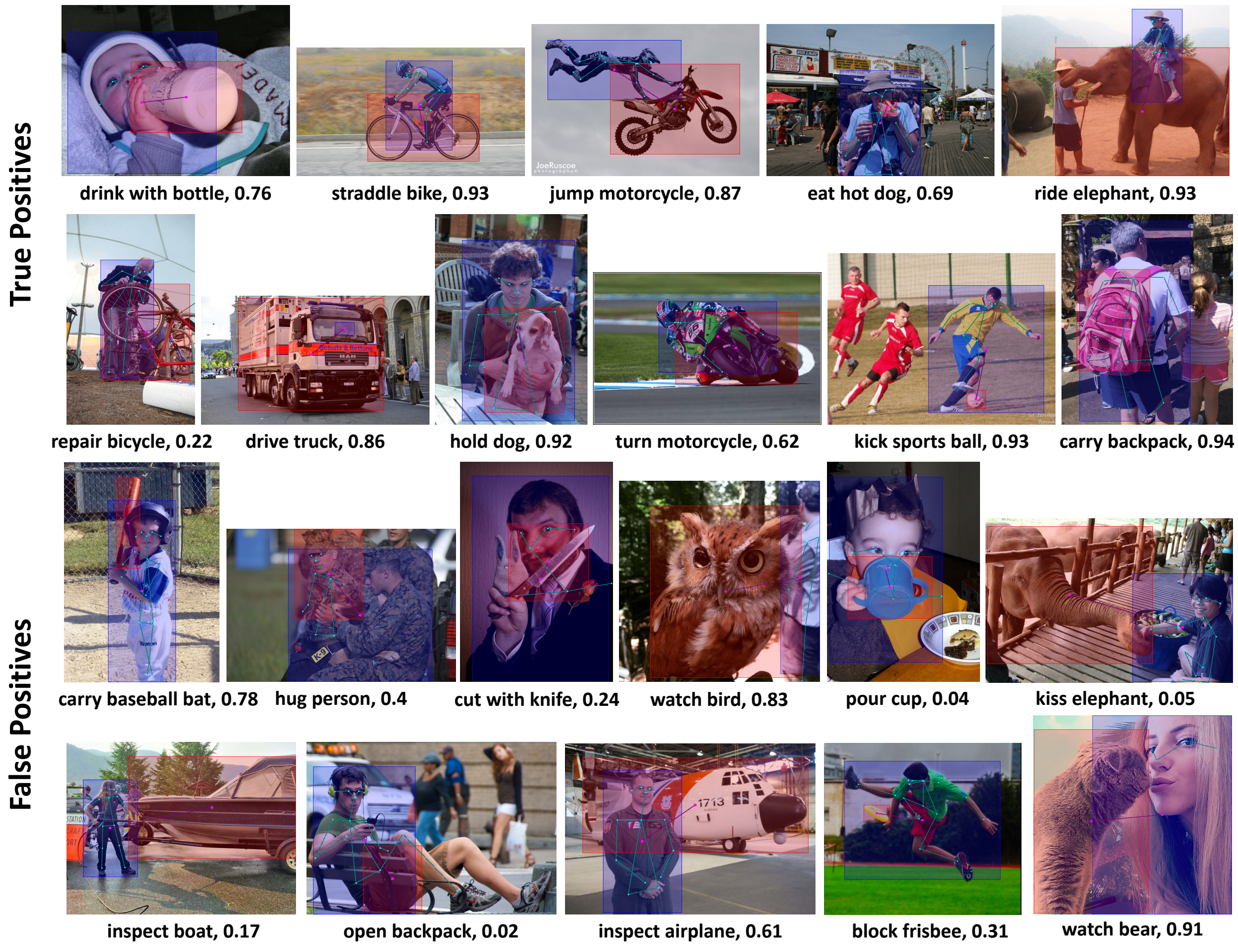


Figure 7.3: **Qualitative results** with top ranking true and false positives with predicted probability.

an intersection-over-union (IoU) score greater than 0.5 with a ground truth box-pair of the corresponding HOI category. During training, the j^{th} sample in a mini-batch consists of a box pair (b_1^j, b_2^j), HOI category $l_j \in \mathcal{H}$ for which the box pair is a candidate (the box pair is a candidate for HOI class (h, o, i) iff $b_1^j \in B_h$ and $b_2^j \in B_o$), binary label y^j to indicate match (or not) with a ground truth box pair of class l_j, detection scores for human and object category corresponding to class l_j, and input features for each factor ϕ. Pair of boxes which are candidates for more than one HOI category are treated as multiple samples during training. Since the number of samples per image is 3 orders of magnitude (typically > 1000) larger than the number of positive samples (typically < 3), random sampling would leave most mini-batches with no positives. We therefore select all positive samples per image and then randomly sample 1000 negatives per positive. Given a mini-batch of size N constructed

	Full	**Rare**	**Non-Rare**	**Number of training instances per HOI class**					
				0-9	**10-49**	**50-99**	**100-499**	**500-999**	**1000+**
HO-RCNN [204]	7.81	5.37	8.54	-	-	-	-	-	-
VSRL [221] (impl. by [205])	9.09	7.02	9.71	-	-	-	-	-	-
InteractNet [205]	9.94	7.16	10.77	-	-	-	-	-	-
GPNN [228]	13.11	9.34	14.23	-	-	-	-	-	-
iCAN [207]	14.84	10.45	16.15	-	-	-	-	-	-
Det	8.32	6.84	8.76	6.84	4.85	6.05	10.18	14.40	21.46
Det + Box	12.54	10.40	13.18	10.40	7.46	9.99	14.62	20.12	35.98
Det + Human App	11.12	8.82	11.80	8.82	7.73	9.19	13.41	15.85	26.42
Det + Object App	11.05	7.41	12.13	7.41	7.68	9.72	14.61	15.58	23.27
Det + App	15.74	11.35	17.05	11.35	10.58	13.96	20.11	22.76	34.75
Det + Human App + Box	15.63	**12.45**	16.58	**12.45**	9.94	12.69	19.05	23.60	39.63
Det + Object App + Box	15.68	10.47	17.24	10.47	9.97	12.84	20.48	23.88	40.87
Det + App + Box	**16.96**	11.95	**18.46**	11.95	**11.02**	**14.00**	**22.02**	**25.01**	**41.13**
Det + Pose	11.09	8.04	12.00	8.04	7.26	8.47	13.08	18.81	32.66
Det + Box + Pose	14.49	11.86	15.27	11.86	9.73	12.21	16.51	21.72	38.81
Det + App + Pose	15.50	10.14	17.10	10.14	10.40	13.11	20.40	23.45	36.08
Det + App + Box + Pose	**17.18**	**12.17**	**18.68**	**12.17**	**11.28**	**14.49**	**22.08**	**25.27**	**41.47**

Table 7.1: **Results on HICO-Det test set.** Det, Box, App, and Pose correspond to object detector terms, appearance, box configuration, and pose factors respectively. Each row was both trained and evaluated with specified factors. **Best** and **second best** numbers are highlighted in color.

from a single image x, the loss is computed as

$$\mathcal{L}_{\text{mini-batch}} = -\frac{1}{N|\mathcal{H}|}\sum_{j=1}^{N}\sum_{l\in\mathcal{H}} \mathbb{1}(l = l^j)\cdot \text{BCE}(y^j, P(y_l = 1|b_1^j, b_2^j, k(b_1^j), o_l, x)), \tag{7.4}$$

where $\text{BCE}(y,p) = y\log(p) + (1-y)\log(1-p)$ is the binary cross entropy loss and the probability is computed using Eq. 7.1. In our experiments, we only learn parameters of the interaction term (*i.e.* MLPs used to compute factors ϕ_{human}, ϕ_{box}, and ϕ_{pose})

7.4 EXPERIMENTS

We use the HICO-Det dataset to evaluate the proposed approach. In addition to demonstrating our model's mAP to be more than 1.7× that of the current state-of-the-art, our experiments evaluate the contribution of different factors in our model through an ablation study (Tab. 7.1) that shows the effect of factors on HOI categories with different number of training samples. In Tab. 7.2 we evaluate the impact of several training procedure design choices. Our analysis also includes visualization of distribution of performance across object and interaction categories (Fig. 7.4), inter-interaction confusions (Fig. 7.2), and examples of top ranking detections and failure cases (Fig. 7.3).

HICO-Det dataset contains 38118 training and 9658 test images annotated with 600 HOI categories. We further use an 80-20 split of the training images to generate our actual training and validation sets. For all experiments we train on this smaller training set and use the validation set for model selection. HOI categories consist of 80 object categories (same as COCO classes) and 117 interactions. Each image on average contains 1.67 HOI detections.

7.4.1 Comparison to State-of-the-art

Tab. 7.1 shows mAP of our final models *Det+App+Box* and *Det+App+Box+Pose*, (and ablations) in comparison to existing models in the literature on various sets of HOI categories – *Full* is mAP across all 600 classes, *Rare* on classes with less than 10 training instances, and *Non-Rare* on the rest. To present a clearer picture, in addition to this *Rare-Non-Rare* split specified in [204], we show results for a more fine-grained grouping of classes based on number of training instances.

The model most similar to ours is *InteractNet* [205] which extends Faster-RCNN with a human-centric branch that produces interaction scores based on human (and optionally object) appearance and a distribution over target object location. There are 4 factors contributing to the improved performance of our model over *InteractNet*: (i) use of significantly large ratio of negative to positive box-pairs during minibatch training (our model uses 1000 whereas [205] uses 3 for the detection branch and no negatives for the interaction branch); (ii) box configuration term in our model directly scores box-pair features, a formulation that maybe easier to learn than predicting distribution over target object locations using human appearance features alone; (iii) fixing training-inference mismatch (Fig 7.1); (iv) easy negative rejection that allows our model to focus on learning to rank only hard candidate pairs for a particular HOI category, namely all combinations of human and object detections of the relevant category. Effect of factors (i), (iii), and (iv) towards our model's performance are further investigated in Tab. 7.2 and Sec. 7.4.2.

HO-RCNN [204] takes human appearance, object appearance, and box configuration encoded as an interaction pattern as inputs and processes them with 3 separate branches, each of which produces a score for each HOI category. The scores are combined along with object detection scores to produce HOI probabilities and the model is trained using multi-label binary classification loss. Our model improves over *HO-RCNN* in two ways: (i) weight sharing in our factored model (and also in *InteractNet* and *iCAN*) makes it more data efficient

Neg./Pos.	**Indicators**	**HOI Loss**	**Interaction Loss**	**mAP**
10	✓	✓	✗	13.40
50	✓	✓	✗	15.51
100	✓	✓	✗	16.30
500	✓	✓	✗	**17.06**
1000	✓	✓	✗	**16.96**
1500	✓	✓	✗	16.62
1000	✗	✓	✗	15.93
1000	✓	✗	✓	15.89

Table 7.2: **Training procedure choices evaluated using Det + App + Box model.** The results highlight the importance of: (i) large *negative to positive ratio* in mini-batches; (ii) using *indicators* during training to only learn to rank candidates selected specifically for a given HOI category instead of all detection pairs; (iii) directly optimizing the *HOI classification* loss instead of training with an *interaction classification* loss and then combining with object detector scores heuristically. **Best** and **second best** numbers are highlighted in color.

than [204] which predicts scores for 600 HOI categories using independent weights in the last 600-way fully connected layer; and (ii) we explicitly encode spatial layout as opposed to [204] which has to learn such a representation via a CNN.

Like *iCAN* [207], we also observe that object appearance provides useful information complementary to human appearance for HOI detection (*Det + Human App*: 11.12, *Det + Object App*: 11.05 *vs. Det + App*: 15.74). While our model only uses human and object appearance encoded in pretrained detector features, [207] further proposes an attention mechanism to augment human and object appearance with contextual information from the image, a contribution complementary to ours. *iCAN* models its training after *InteractNet* and uses interaction pattern from [204] to encode spatial layout, and hence can benefit from our training procedure design choices and spatial encoding.

7.4.2 Significant Training Procedure Design Decisions

As shown in Tab. 7.2, increasing the ratio of negative box-pairs sampled per positive in a mini-batch during training leads to a dramatic increase in performance. This is in contrast to low ratios (typically < 10) used for training object detectors and hence also in related work [204, 205]. We believe this is because seeing a large number of negative pairs is important for learning to reject false positives. Also higher ratios are expected since the number of negative pairs is quadratic in the number of region proposals as opposed to linear for object detectors.

A distinguishing feature of our training and inference procedures is the use of indicator variables in interaction terms (Eq. 7.3) and training objective (Eq. 7.4). The observation behind this choice is that with state-of-the-art object detectors like Faster-RCNN with only 6 human and 1.2 object detections per image on average (after NMS and score thresholding), the recall of ground truth HOI candidates in HOI category specific candidate box-pairs stands at 59% (much higher than mAP of existing approaches). This suggests that object detectors are effective at rejecting easy negative pairs. Thus, using the indicator variables increases learning efficiency by allowing the model to focus on learning to reject hard negatives, namely candidate pairs which contain a human and object of interest but not engaging in the desired interaction. Tab. 7.2 shows that even while using the indicators during inference, not using them during training causes a drop in mAP from 16.96 to 15.93.

Finally, training the model using interaction classification loss on the probabilities predicted by the interaction term, as done in [205], is suboptimal in comparison to training using HOI classification loss (15.89 vs 16.96 mAP) even though the same set of parameters are optimized by both losses. This is because the latter provides an opportunity for the interaction term to calibrate itself relative to the detection terms. This approach is also used in [204] but without strong weight sharing assumptions made by our factor model.

7.4.3 Factor Ablation Study

To identify the role of different sources of appearance and spatial information in our model we train models with subsets of available factors.

The role of individual factors can be assessed by comparing *Det*, *Det+Box*, *Det+App*, and *Det+Pose*. Note that appearance terms lead to largest gains over *Det* followed by *Box* and *Pose*. We further analyse the contribution of human and object appearance towards predicting interactions. Interestingly, while *Det+Human App* and *Det+Object App* perform comparably (11.12 and 11.05), the combination outperforms either of them with an mAP of 15.74 showing that the human and object appearance provide some complementary information. Note that an mAP of 11.12 ($= \max(11.12, 11.05)$) or less would indicate completely redundant or noisy signals. Similar sense of complementary information can be assessed from Table 7.1 for *App-Box*, *App-Pose*, and *Box-Pose* pairs. While *Det+Box+Pose* improves over *Det+Box*, *Det+App+Pose* and *Det+App* perform comparably. Similarly *Det+App+Box+Pose* only slightly improves the performance of *Det+App+Box*. This suggests that while explicit pose features may provide useful information in absence of appearance information, they are somewhat redundant otherwise.

Another way of understanding the role of factors is to consider the drop in performance when a particular factor is removed from the final model. Relative to *Det+App+Box+Pose*,

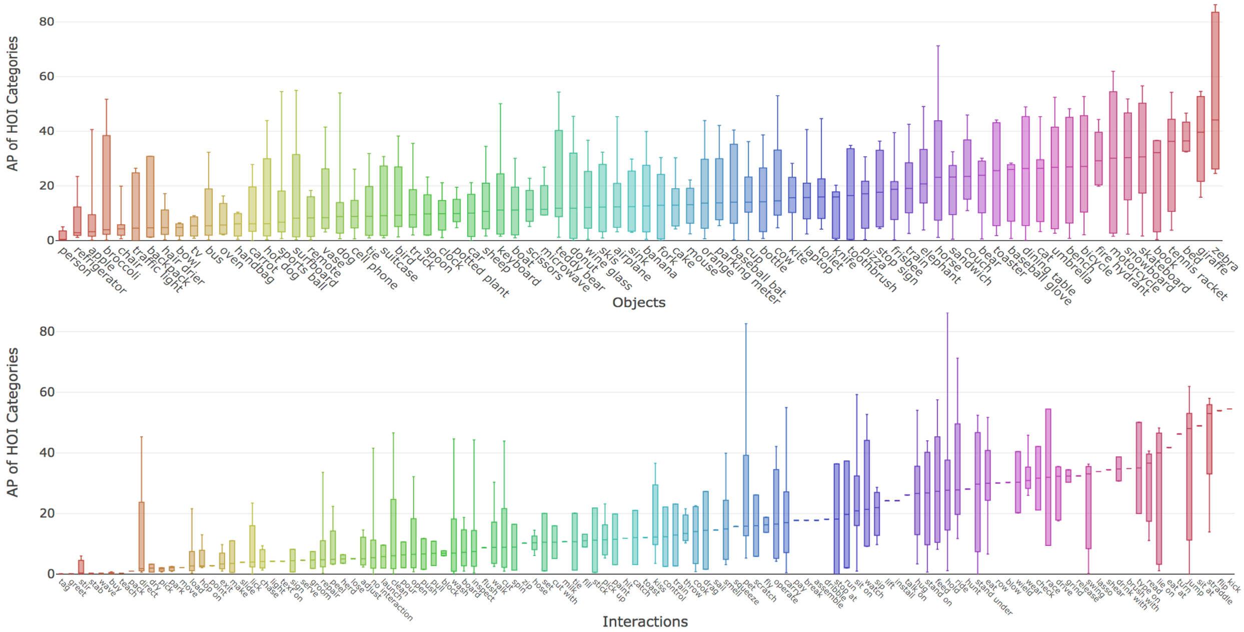


Figure 7.4: **Spread of performance** (range and quartiles) across interactions with the same object (top) and across objects for a given interaction (bottom). The horizontal axis is sorted by median AP.

performance drops are 2.69, 1.68, and 0.22 mAP for *App*, *Box* and *Pose* factors respectively.

7.4.4 How is the performance distributed across objects and interactions?

Fig. 7.4 visualizes the spread of performance of our final model across interactions with a given object and across objects for a given interaction. The figure shows that for most objects certain interactions are much easier to detect than others (with the caveat that AP computation for any class is sensitive to the number of positives for that class in the test set). Similar observation is true for different objects given an interaction. In addition, we observe that interactions which can occur with only a specific object category (as indicated by absence of box) such as "kick-ball" and "flip-skateboard" are easier to detect than those that tend to occur with more than one object such as "cut" and "clean" and could have drastically different visual and spatial appearance depending on the object. Heatmaps in Fig. 7.2 show the interactions that are confused by different models. Comparing heatmap **b** with **a** shows the role of the appearance factor in reducing confusion between interactions. For instance, without *App* "eat" is confused with "brush with" and "drink with", but not in the final model. Similarly, compare **c** and **d** with **a** for the effect of *Box* and *Pose* factors respectively.

7.4.5 Qualitative Results

Qualitative results (Fig. 7.3) demonstrate the advantages of building HOI detectors on the strong foundation of object detectors. False positives are more commonly due to incorrect interaction than object. Interaction errors are often due to fine grained differences between classes: *e.g.*, “carry” *vs*. “wield” “baseball bat” and “inspect” *vs*. “repair” “boat.” Notice in some examples like “inspect airplane” and “watch bear,” cues for preventing false positives are as subtle as gaze direction.

7.4.6 Conclusion

In this chapter we proposed a simplified yet powerful factored model for detecting human-object interactions. We analyse the model thoroughly to provide insight into the relative importance of appearance, box configuration, and pose factors towards HOI detection. We also highlight training procedures that demonstrably improve model performance.

CHAPTER 8: CONCLUSION

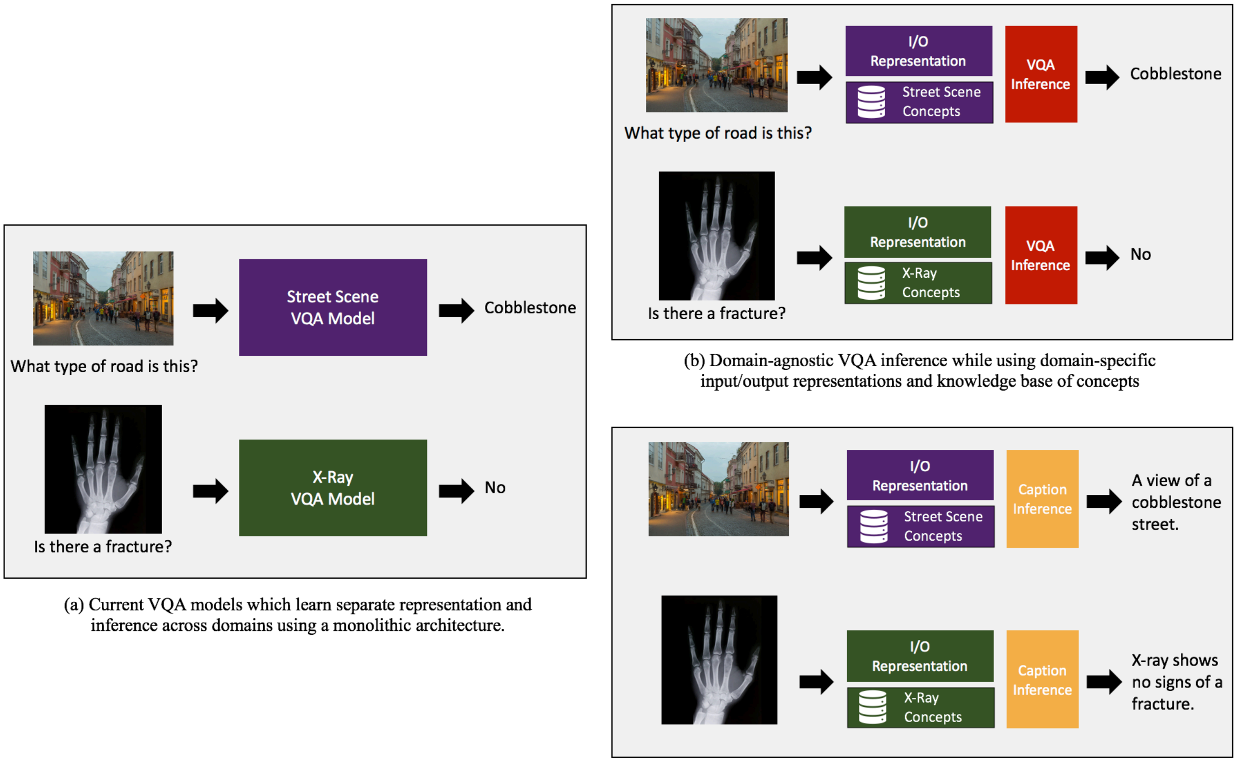
Vision and Language research is not just about a collection of tasks that involve text and visual data such as VQA or Image Captioning. This field of study allows us to ask fundamental questions about how an artificially intelligent system may acquire information about the world, represent that information efficiently and in an extensible manner, and use the representation to perform tasks involving making predictions, communicating with humans in natural language, or taking actions.

In this dissertation, we have only scratched the surface by focusing on 3 challenges: (i) learning generalizable representation of images and words; (ii) modeling interactions between objects; and (iii) learning to map words to image regions without word-region grounding supervision. The key guiding principles behind the questions asked and solutions provided in this thesis and that, we believe, should continue to guide the direction of future work are as follows:

Improve generalization and extensibility: Current approaches excel at learning tasks like VQA or Image Captioning from direct task supervision. However, supervised learning assumes similar distributions for training and test data. This assumption jeopardizes generalization of learned representations and inference not only to new tasks but also to new domains and novel concepts for the same task that may not be seen during task training. Today, it is difficult to generate captions about a "tiger" detected in an image if none of the training captions mention "tiger". Future research should make it easy to incorporate novel concepts or domains into an existing VQA or captioning systems.

Minimize supervision and increase learning efficiency: Humans are constantly taking in visual and textual information, building representations and learning skills. Much of this process takes place through sparse supervision. It is therefore quite unsatisfying that vision-language models need hundreds of thousands of question-answers or captions to learn to perform question-answering or captioning. It is crucial for future work to build representations in an unsupervised manner, and develop sample efficient and flexible mechanisms that can take advantage of unsupervised representations and sparse supervision available across multiple tasks to learn task-inference.

In the next section, we present our vision for what models for vision-language tasks may look like in the future and highlight their desirable properties.



(a) Current VQA models which learn separate representation and inference across domains using a monolithic architecture.

(b) Domain-agnostic VQA inference while using domain-specific input/output representations and knowledge base of concepts

(c) Sharing representations and knowledge across tasks while using tasks-specific inference modules

Figure 8.1: Current vision-language models *vs*. recommended compositional models with representation and inference sharing schemes.

8.1 A VISION FOR THE FUTURE

Consider VQA and Image Captioning tasks to be performed on 2 domains – street scenes and medical X-ray images. Current vision-language models train end-to-end models that simultaneously learn representations and inference in a monolithic architecture. This means that even though one might have a trained VQA model for street scenes, applying it to answer questions about medical images or produce captions for street scenes requires retraining the model, perhaps with parameters initialized from the street-scene VQA model. This is in stark contrast to traditional algorithms in computer science.

A sorting analogy. VQA models are learned algorithms for answering questions. However, these learned algorithms behave quite differently from traditional algorithms such as Merge Sort. Whether one wants to sort a sequence of numbers or a sequence of words, the exact same algorithm – recursively dividing the sequence, sorting, and merging – is able to achieve the desired result. What does change is the comparison operation or the input representation – order of numbers *vs.* words. In contrast, for current VQA models, representations and inference are so intertwined (what layers learn representations and what layers perform inference remains unclear). Therefore, retraining the model on a new domain changes the image, question and answer representations as well as the implicit inference captured by the model parameters (*e.g.* behavior of the attention mechanism). Hence, the current monolithic approaches to vision-language tasks introduce unnecessary duplication of effort in the learning process.

A compositional solution. We believe a potential solution is to disentangle representation from inference for vision-language tasks. The key idea, as shown in Fig. 8.1, is to compose a model for a task from 3 modules: (i) a domain-specific but task-agnostic input/output (I/O) representation module such as for representing images and words; (ii) domain-specific knowledge base such as a dictionary of medical conditions and affected body parts for medical-image question answering; and (iii) task-specific but domain-agnostic inference module. When training on multiple tasks and domains, the modules are composed on the fly for each task sharing representation and knowledge bases across similar domains across different tasks, and sharing inference across domains for the same task.

There exists some work on creating inference for VQA depending on the question by composing neural modules [74, 75, 229]. However, the key difference between our approach and these methods is the level of abstraction at which compositionality is enforced. While these methods are trying to compose inference for a single task and domain (a given VQA dataset) from neural modules, we are proposing to compose representation, knowledge and inference across multiple tasks (VQA, Captioning *etc.*) and domains (street scenes, medical images *etc.*).

8.2 CONCLUDING REMARKS

It is a testament to the progress made by the AI research community that complex problems like visual question answering, image captioning, or semantic scene generation are within reach of current computational tools and techniques. However, today's state-of-the-art is far from the human ability to collect information, synthesize a consistent world view, make well reasoned decisions, and act to achieve complex goals. End-to-end, task-specific learning from massive datasets has been a foundational stepping stone towards general intelligence. But to keep making progress, we must continue our search for more efficient, scalable, generalizable, and extensible learning solutions.

REFERENCES

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

[2] 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.

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.

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

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn. arxiv preprint arxiv: 170306870," 2017.

[6] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015.

[7] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *TPAMI*, 2017.

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

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

[10] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," *arXiv preprint arXiv:1606.00061*, 2016.

[11] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," *arXiv preprint arXiv:1511.02274*, 2015.

[12] 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.

[13] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *European Conference on Computer Vision.* Springer, 2014, pp. 391–405.

[14] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *IJCV*, 2013.

[15] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma et al., "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *arXiv preprint arXiv:1602.07332*, 2016.

[16] 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," *CVPR*, 2017.

[17] Yu Jiang*, Vivek Natarajan*, Xinlei 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.

[18] K. J. Shih, S. Singh, and D. Hoiem, "Where to look: Focus regions for visual question answering," in *CVPR*, 2016.

[19] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," in *ICLR*, 2019.

[20] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," *arXiv preprint arXiv:1906.00910*, 2019.

[21] F. Yan and K. Mikolajczyk, "Deep correlation for matching images and text," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3441–3450, 2015.

[22] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," *arXiv preprint arXiv:1911.05722*, 2019.

[23] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," *arXiv preprint arXiv:2002.05709*, 2020.

[24] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by latent semantic analysis," *JASIS*, 1990.

[25] K. Lund and C. Burgess, "Producing high-dimensional semantic spaces from lexical co-ocurrence," 1996.

[26] R. Lebret and R. Collobert, "Word embeddings through hellinger pca," in *EACL*, 2014.

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

[28] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013.

[29] O. Levy and Y. Goldberg, "Neural word embedding as implicit matrix factorization," in *NIPS*, 2014.

[30] F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler, "Vse++: Improved visual-semantic embeddings," *BMVC*, 2018.

[31] R. Kiros, R. Salakhutdinov, and R. S. Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," *arXiv preprint arXiv:1411.2539*, 2014.

[32] I. Vendrov, R. Kiros, S. Fidler, and R. Urtasun, "Order-embeddings of images and language," *arXiv preprint arXiv:1511.06361*, 2015.

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

[34] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Gated feedback recurrent neural networks," in *International Conference on Machine Learning*, 2015, pp. 2067–2075.

[35] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler, "Skip-thought vectors," in *NeurIPS*, 2015.

[36] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, 1990.

[37] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. S. Zettlemoyer, "Deep contextualized word representations," in *NAACL-HLT*, 2018.

[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017.

[39] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *NAACL-HLT*, 2018.

[40] R. Caruana, "Multitask learning," in *Learning to learn.* Springer, 1998, pp. 95–133.

[41] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, 2009.

[42] Z. Li and D. Hoiem, "Learning without forgetting," in *European Conference on Computer Vision.* Springer, 2016, pp. 614–629.

[43] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *CVPR*, 2017.

[44] H. Yin, P. Molchanov, Z. Li, J. M. Alvarez, A. Mallya, D. Hoiem, N. K. Jha, and J. Kautz, "Dreaming to distill: Data-free knowledge transfer via deepinversion," *arXiv preprint arXiv:1912.08795*, 2019.

[45] A. Pentina, V. Sharmanska, and C. H. Lampert, "Curriculum learning of multiple tasks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5492–5500.

[46] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *ICML*, 2009.

[47] 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.

[48] T. Mitchell, "Never-ending learning," DTIC Document, Tech. Rep., 2010.

[49] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr, and T. M. Mitchell, "Toward an architecture for never-ending language learning." in *AAAI*, vol. 5, 2010, p. 3.

[50] S. Thrun, "Lifelong learning algorithms," in *Learning to learn.* Springer, 1998, pp. 181–209.

[51] D. L. Silver, Q. Yang, and L. Li, "Lifelong machine learning systems: Beyond learning algorithms." in *AAAI Spring Symposium: Lifelong Machine Learning.* Citeseer, 2013, pp. 49–55.

[52] X. Chen, A. Shrivastava, and A. Gupta, "Neil: Extracting visual knowledge from web data," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1409–1416.

[53] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[55] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[56] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[57] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[58] L. Wang, Y. Li, and S. Lazebnik, "Learning deep structure-preserving image-text embeddings," in *Computer Vision and Pattern Recognition*, 2016.

[59] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision.* Springer, 2014, pp. 740–755.

[60] M. Hodosh, P. Young, and J. Hockenmaier, "Framing image description as a ranking task: Data, models and evaluation metrics," *Journal of Artificial Intelligence Research*, vol. 47, pp. 853–899, 2013.

[61] 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 *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2641–2649.

[62] S. Kazemzadeh, V. Ordonez, M. Matten, and T. L. Berg, “Referit game: Referring to objects in photographs of natural scenes,” in *EMNLP*, 2014.

[63] J. Mao, J. Huang, A. Toshev, O. Camburu, and K. Murphy, “Generation and comprehension of unambiguous object descriptions,” in *Computer Vision and Pattern Recognition*, 2016.

[64] M. Ren, R. Kiros, and R. S. Zemel, “Exploring models and data for image question answering,” in *Proceedings of the 28th International Conference on Neural Information Processing Systems*, ser. NIPS’15. Cambridge, MA, USA: MIT Press, 2015. [Online]. Available: http://dl.acm.org/citation.cfm?id=2969442.2969570 pp. 2953–2961.

[65] L. Yu, E. Park, A. C. Berg, and T. L. Berg, “Visual madlibs: Fill in the blank description generation and question answering,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2461–2469.

[66] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele, “Grounding of textual phrases in images by reconstruction,” in *European Conference on Computer Vision (ECCV)*, 2016.

[67] Y. Gong, L. Wang, M. Hodosh, J. Hockenmaier, and S. Lazebnik, “Improving image-sentence embeddings using large weakly annotated photo collections,” in *European Conference on Computer Vision.* Springer International Publishing, 2014, pp. 529–545.

[68] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt et al., “From captions to visual concepts and back,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1473–1482.

[69] C. Liu, J. Mao, F. Sha, and A. Yuille, “Attention correctness in neural image captioning,” *arXiv preprint arXiv:1605.09553*, 2016.

[70] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach, “Multimodal compact bilinear pooling for visual question answering and visual grounding,” *arXiv preprint arXiv:1606.01847*, 2016.

[71] I. Ilievski, S. Yan, and J. Feng, “A focused dynamic attention model for visual question answering,” *arXiv preprint arXiv:1604.01485*, 2016.

[72] H. Xu and K. Saenko, “Ask, attend and answer: Exploring question-guided spatial attention for visual question answering,” in *European Conference on Computer Vision (ECCV)*, 2016.

[73] K. J. Shih, S. Singh, and D. Hoiem, "Where to look: Focus regions for visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[74] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, "Neural module networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 39–48.

[75] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, "Learning to compose neural networks for question answering," *arXiv preprint arXiv:1601.01705*, 2016.

[76] A. Kumar, O. Irsoy, J. Su, J. Bradbury, R. English, B. Pierce, P. Ondruska, I. Gulrajani, and R. Socher, "Ask me anything: Dynamic memory networks for natural language processing," *arXiv preprint arXiv:1506.07285*, 2015.

[77] T. Tommasi, A. Mallya, B. Plummer, S. Lazebnik, A. C. Berg, and T. L. Berg, "Solving visual madlibs with multiple cues," in *Proceedings of the British Machine Vision Conference 2016*, 2016.

[78] V. Mnih, N. Heess, A. Graves et al., "Recurrent models of visual attention," in *Advances in Neural Information Processing Systems*, 2014, pp. 2204–2212.

[79] M. Jaderberg, K. Simonyan, A. Zisserman et al., "Spatial transformer networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 2017–2025.

[80] J. Johnson, A. Karpathy, and L. Fei-Fei, "Densecap: Fully convolutional localization networks for dense captioning," *arXiv preprint arXiv:1511.07571*, 2015.

[81] J. Weston, S. Chopra, and A. Bordes, "Memory networks," *arXiv preprint arXiv:1410.3916*, 2014.

[82] S. Sukhbaatar, J. Weston, R. Fergus et al., "End-to-end memory networks," in *Advances in neural information processing systems*, 2015, pp. 2440–2448.

[83] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *Advances in Neural Information Processing Systems*, 2015, pp. 1693–1701.

[84] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[85] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey et al., "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[86] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," *arXiv preprint arXiv:1410.5401*, 2014.

[87] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, 2015.

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

[89] A. Agrawal, D. Batra, and D. Parikh, “Analyzing the behavior of visual question answering models,” *arXiv preprint arXiv:1606.07356*, 2016.

[90] 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,” *arXiv preprint arXiv:1612.00837*, 2016.

[91] A. Das, H. Agrawal, C. L. Zitnick, D. Parikh, and D. Batra, “Human attention in visual question answering: Do humans and deep networks look at the same regions?” in *Conference on Empirical Methods in Natural Language Processing*, 2016.

[92] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, “Bidirectional attention flow for machine comprehension,” *ICLR*, 2017.

[93] L. He, K. Lee, M. Lewis, and L. Zettlemoyer, “Deep semantic role labeling: What works and whats next,” in *ACL*, 2017.

[94] K. Lee, L. He, M. Lewis, and L. Zettlemoyer, “End-to-end neural coreference resolution,” *EMNLP*, 2017.

[95] M. E. Peters, W. Ammar, C. Bhagavatula, and R. Power, “Semi-supervised sequence tagging with bidirectional language models,” *ACL*, 2017.

[96] A. P. Parikh, O. Täckström, D. Das, and J. Uszkoreit, “A decomposable attention model for natural language inference,” in *EMNLP*, 2016.

[97] G. Stanovsky, J. Michael, L. S. Zettlemoyer, and I. Dagan, “Supervised open information extraction,” in *NAACL-HLT*, 2018.

[98] H. Rashkin, M. Sap, E. Allaway, N. A. Smith, and Y. Choi, “Event2mind: Commonsense inference on events, intents, and reactions,” in *ACL*, 2018.

[99] D. Massiceti, N. Siddharth, P. K. Dokania, and P. H. Torr, “Flipdial: A generative model for two-way visual dialogue,” in *CVPR*, 2018.

[100] L. Wang, Y. Li, J. Huang, and S. Lazebnik, “Learning two-branch neural networks for image-text matching tasks,” *TPAMI*, 2019.

[101] X. Wang, Y. Ye, and A. Gupta, “Zero-shot recognition via semantic embeddings and knowledge graphs,” in *CVPR*, 2018.

[102] L. Anne Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell, “Localizing moments in video with natural language,” in *ICCV*, 2017.

[103] B. A. Plummer, A. Mallya, C. M. Cervantes, J. Hockenmaier, and S. Lazebnik, "Phrase localization and visual relationship detection with comprehensive image-language cues," in *ICCV*, 2017.

[104] B. A. Plummer, P. Kordas, M. H. Kiapour, S. Zheng, R. Piramuthu, and S. Lazebnik, "Conditional image-text embedding networks," in *ECCV*, 2018.

[105] M. I. Vasileva, B. A. Plummer, K. Dusad, S. Rajpal, R. Kumar, and D. Forsyth, "Learning type-aware embeddings for fashion compatibility," in *ECCV*, 2018.

[106] T. Gupta, K. Shih, S. Singh, and D. Hoiem, "Aligned image-word representations improve inductive transfer across vision-language tasks," in *ICCV*, 2017.

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

[108] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *CVPR*, 2015.

[109] R. Luo and G. Shakhnarovich, "Comprehension-guided referring expressions," in *CVPR*, 2017.

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

[111] G. A. Miller, "Wordnet: a lexical database for english," *ACM*, 1995.

[112] C. E. Osgood, G. J. Suci, and P. H. Tannenbaum, *The measurement of meaning*. University of Illinois press, 1957.

[113] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, 1964.

[114] B. H. Ross and G. L. Murphy, "Food for thought: Cross-classification and category organization in a complex real-world domain," *Cognitive Psychology*, 1999.

[115] L. J. Rips, E. J. Shoben, and E. E. Smith, "Semantic distance and the verification of semantic relations," *Journal of verbal learning and verbal behavior*, 1973.

[116] J. A. Bullinaria and J. P. Levy, "Extracting semantic representations from word co-occurrence statistics: a computational study." *Behavior research methods*, 2007.

[117] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013.

[118] T. Mikolov, W. tau Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *HLT-NAACL*, 2013.

[119] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013.

[120] S. Kottur, R. Vedantam, J. M. F. Moura, and D. Parikh, "Visual word2vec (vis-w2v): Learning visually grounded word embeddings using abstract scenes," *CVPR*, 2016.

[121] M. Hasegawa, T. Kobayashi, and Y. Hayashi, "Incorporating visual features into word embeddings: A bimodal autoencoder-based approach," in *IWCS*, 2017.

[122] A. Radford, "Improving language understanding by generative pre-training," 2018.

[123] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Citeseer, Tech. Rep., 2009.

[124] 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.

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

[126] C. L. Zitnick and D. Parikh, "Bringing semantics into focus using visual abstraction," in *CVPR*, 2013.

[127] A. Krebs, A. Lenci, and D. Paperno, "Semeval-2018 task 10: Capturing discriminative attributes," in *International Workshop on Semantic Evaluation*, 2018.

[128] 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.

[129] Y. Jiang, V. Natarajan, X. Chen, M. Rohrbach, D. Batra, and D. Parikh, "Pythia," https://github.com/facebookresearch/pythia, 2018.

[130] L. Yu, Z. Lin, X. Shen, J. Yang, X. Lu, M. Bansal, and T. L. Berg, "Mattnet: Modular attention network for referring expression comprehension," in *CVPR*, 2018.

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

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

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

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

[135] S. Datta, K. Sikka, A. Roy, K. Ahuja, D. Parikh, and A. Divakaran, "Align2ground: Weakly supervised phrase grounding guided by image-caption alignment," *ICCV*, 2019.

[136] H. Akbari, S. Karaman, S. Bhargava, B. Chen, C. Vondrick, and S.-F. Chang, "Multi-level multimodal common semantic space for image-phrase grounding," *CVPR*, 2018.

[137] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele, "Grounding of textual phrases in images by reconstruction," in *ECCV*, 2016.

[138] O. Maron and T. Lozano-Pérez, "A framework for multiple-instance learning," in *NeurIPS*, 1998.

[139] M. Ilse, J. M. Tomczak, and M. Welling, "Attention-based deep multiple instance learning," in *ICML*, 2018.

[140] H. Fang, S. Gupta, F. N. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, C. L. Zitnick, and G. Zweig, "From captions to visual concepts and back," *CVPR*, 2014.

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

[142] R. A. Yeh, M. N. Do, and A. G. Schwing, "Unsupervised textual grounding: Linking words to image concepts," in *CVPR*, 2018.

[143] J. Wang and L. Specia, "Phrase localization without paired training examples," *ICCV*, 2019.

[144] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013.

[145] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," in *ICLR*, 2017.

[146] H. Kim and A. Mnih, "Disentangling by factorising," in *ICML*, 2018.

[147] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm, "Mutual information neural estimation," in *ICML*, 2018.

[148] B. Poole, S. Ozair, A. Van Den Oord, A. Alemi, and G. Tucker, "On variational bounds of mutual information," in *ICML*, 2019.

[149] M. Tschannen, J. Djolonga, P. K. Rubenstein, S. Gelly, and M. Lucic, "On mutual information maximization for representation learning," in *ICLR*, 2020.

[150] D. McAllester and K. Stratos, "Formal limitations on the measurement of mutual information," *arXiv preprint arXiv:1811.04251*, 2018.

[151] J. Song and S. Ermon, "Understanding the limitations of variational mutual information estimators," in *ICLR*, 2020.

[152] O. J. Hénaff, A. Razavi, C. Doersch, S. Eslami, and A. v. d. Oord, “Data-efficient image recognition with contrastive predictive coding,” *arXiv preprint arXiv:1905.09272*, 2019.

[153] M. Gutmann and A. Hyvärinen, “Noise-contrastive estimation: A new estimation principle for unnormalized statistical models,” in *AISTATS*, 2010.

[154] A. Mnih and K. Kavukcuoglu, “Learning word embeddings efficiently with noise-contrastive estimation,” in *NeurIPS*, 2013.

[155] Y. Tian, D. Krishnan, and P. Isola, “Contrastive multiview coding,” *arXiv preprint arXiv:1906.05849*, 2019.

[156] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, “Unsupervised feature learning via non-parametric instance discrimination,” in *CVPR*, 2018.

[157] I. Misra and L. van der Maaten, “Self-supervised learning of pretext-invariant representations,” *arXiv preprint arXiv:1912.01991*, 2019.

[158] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, “Visualbert: A simple and performant baseline for vision and language,” *arXiv preprint arXiv:1908.03557*, 2019.

[159] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, “Vl-bert: Pre-training of generic visual-linguistic representations,” 2020.

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

[161] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid, “Videobert: A joint model for video and language representation learning,” in *ICCV*, 2019.

[162] H. Tan and M. Bansal, “Lxmert: Learning cross-modality encoder representations from transformers,” in *EMNLP*, 2019.

[163] L. Zhou, H. Palangi, L. Zhang, H. Hu, J. J. Corso, and J. Gao, “Unified vision-language pre-training for image captioning and vqa,” in *AAAI*, 2020.

[164] G. Li, N. Duan, Y. Fang, D. Jiang, and M. Zhou, “Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training,” *arXiv preprint arXiv:1908.06066*, 2019.

[165] Y.-C. Chen, L. Li, L. Yu, A. E. Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu, “Uniter: Learning universal image-text representations,” *arXiv preprint arXiv:1909.11740*, 2019.

[166] C. Alberti, J. Ling, M. Collins, and D. Reitter, “Fusion of detected objects in text for visual question answering,” in *EMNLP*, 2019.

[167] C. Sun, F. Baradel, K. Murphy, and C. Schmid, "Contrastive bidirectional transformer for temporal representation learning," *arXiv preprint arXiv:1906.05743*, 2019.

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

[169] K. Chen, J. Gao, and R. Nevatia, "Knowledge aided consistency for weakly supervised phrase grounding," in *CVPR*, 2018.

[170] J. Choe, S. J. Oh, S. Lee, S. Chun, Z. Akata, and H. Shim, "Evaluating weakly supervised object localization methods right," *ArXiv*, 2020.

[171] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[172] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM transactions on graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 309–314.

[173] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics-TOG*, vol. 28, no. 3, p. 24, 2009.

[174] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *CoRR*, vol. abs/1312.6114, 2013.

[175] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014.

[176] Y. Bengio, G. Mesnil, Y. Dauphin, and S. Rifai, "Better mixing via deep representations," in *ICML*, 2013.

[177] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *CoRR*, vol. abs/1701.07875, 2017.

[178] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *CoRR*, vol. abs/1511.06434, 2015.

[179] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra, "Draw: A recurrent neural network for image generation," in *ICML*, 2015.

[180] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee, "Learning what and where to draw," in *NIPS*, 2016.

[181] S. E. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *ICML*, 2016.

[182] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. N. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," *CoRR*, vol. abs/1612.03242, 2016.

[183] E. Mansimov, E. Parisotto, J. Ba, and R. Salakhutdinov, "Generating images from captions with attention," in *ICLR*, 2016.

[184] X. Yan, J. Yang, K. Sohn, and H. Lee, "Attribute2image: Conditional image generation from visual attributes," in *ECCV*, 2016.

[185] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *ICML*, 2017.

[186] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *CoRR*, vol. abs/1710.10196, 2017.

[187] S. Hong, D. Yang, J. Choi, and H. Lee, "Inferring semantic layout for hierarchical text-to-image synthesis," *CoRR*, vol. abs/1801.05091, 2018.

[188] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *CoRR*, vol. abs/1611.07004, 2016.

[189] Q. Chen and V. Koltun, "Photographic image synthesis with cascaded refinement networks," *CoRR*, vol. abs/1707.09405, 2017.

[190] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," in *NIPS*, 2016.

[191] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee, "Learning to generate long-term future via hierarchical prediction," *arXiv preprint arXiv:1704.05831*, 2017.

[192] T. Marwah, G. Mittal, and V. N. Balasubramanian, "Attentive semantic video generation using captions," *CoRR*, vol. abs/1708.05980, 2017.

[193] Y. Li, M. R. Min, D. Shen, D. Carlson, and L. Carin, "Video generation from text," *arXiv preprint arXiv:1710.00421*, 2017.

[194] H. Kwak and B.-T. Zhang, "Generating images part by part with composite generative adversarial networks," *CoRR*, vol. abs/1607.05387, 2016.

[195] C. L. Zitnick and D. Parikh, "Bringing semantics into focus using visual abstraction," *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3009–3016, 2013.

[196] C. L. Zitnick, D. Parikh, and L. Vanderwende, "Learning the visual interpretation of sentences," *2013 IEEE International Conference on Computer Vision*, pp. 1681–1688, 2013.

[197] C. L. Zitnick, R. Vedantam, and D. Parikh, “Adopting abstract images for semantic scene understanding,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, pp. 627–638, 2016.

[198] S. Reed, A. van den Oord, N. Kalchbrenner, V. Bapst, M. Botvinick, and N. de Freitas, “Generating interpretable images with controllable structure,” in *OpenReview.net*, 2017.

[199] Y. Liu, Z. Qin, Z. Luo, and H. Wang, “Auto-painter: Cartoon image generation from sketch by using conditional generative adversarial networks,” *CoRR*, vol. abs/1705.01908, 2017.

[200] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov et al., “Devise: A deep visual-semantic embedding model,” in *NIPS*, 2013.

[201] L. Wang, Y. Li, and S. Lazebnik, “Learning deep structure-preserving image-text embeddings,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5005–5013.

[202] Y.-H. H. Tsai, L.-K. Huang, and R. Salakhutdinov, “Learning robust visual-semantic embeddings,” *arXiv preprint arXiv:1703.05908*, 2017.

[203] T. Gupta, K. Shih, S. Singh, D. Hoiem, K. J. Shih, A. Mallya, W. Di, V. Jagadeesh, R. Piramuthu, K. Shih et al., “Aligned image-word representations improve inductive transfer across vision-language tasks,” *arXiv preprint arXiv:1704.00260*, 2017.

[204] Y.-W. Chao, Y. Liu, X. Liu, H. Zeng, and J. Deng, “Learning to detect human-object interactions,” *arXiv preprint arXiv:1702.05448*, 2017.

[205] G. Gkioxari, R. Girshick, P. Dollár, and K. He, “Detecting and recognizing human-object interactions,” *arXiv preprint arXiv:1704.07333*, 2017.

[206] C. M. Bishop, “Mixture density networks,” 1994.

[207] C. Gao, Y. Zou, and J.-B. Huang, “ican: Instance-centric attention network for human-object interaction detection,” in *BMVC*, 2018.

[208] B. Yao and L. Fei-Fei, “Modeling mutual context of object and human pose in human-object interaction activities,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 17–24.

[209] B. Yao and L. Fei-Fei, “Grouplet: A structured image representation for recognizing human and object interactions,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 9–16.

[210] C. Desai, D. Ramanan, and C. Fowlkes, “Discriminative models for static human-object interactions,” in *Computer vision and pattern recognition workshops (CVPRW), 2010 IEEE computer society conference on*. IEEE, 2010, pp. 9–16.

[211] C. Desai and D. Ramanan, "Detecting actions, poses, and objects with relational phraselets," in *European Conference on Computer Vision.* Springer, 2012, pp. 158–172.

[212] V. Delaitre, J. Sivic, and I. Laptev, "Learning person-object interactions for action recognition in still images," in *Advances in neural information processing systems*, 2011, pp. 1503–1511.

[213] S. Maji, L. Bourdev, and J. Malik, "Action recognition from a distributed representation of pose and appearance," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on.* IEEE, 2011, pp. 3177–3184.

[214] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[215] H. Idrees, A. R. Zamir, Y.-G. Jiang, A. Gorban, I. Laptev, R. Sukthankar, and M. Shah, "The thumos challenge on action recognition for videos in the wild," *Computer Vision and Image Understanding*, vol. 155, pp. 1–23, 2017.

[216] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei, "Human action recognition by learning bases of action attributes and parts," in *Computer Vision (ICCV), 2011 IEEE International Conference on.* IEEE, 2011, pp. 1331–1338.

[217] Y.-W. Chao, Z. Wang, Y. He, J. Wang, and J. Deng, "Hico: A benchmark for recognizing human-object interactions in images," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1017–1025.

[218] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision.* Springer, 2014, pp. 740–755.

[219] 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.

[220] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, "Visual relationship detection with language priors," in *European Conference on Computer Vision.* Springer, 2016, pp. 852–869.

[221] S. Gupta and J. Malik, "Visual semantic role labeling," *arXiv preprint arXiv:1505.04474*, 2015.

[222] M. Yatskar, L. Zettlemoyer, and A. Farhadi, "Situation recognition: Visual semantic role labeling for image understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5534–5542.

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

[224] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[225] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, “Realtime multi-person 2d pose estimation using part affinity fields,” in *CVPR*, 2017.

[226] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, “Convolutional pose machines,” in *CVPR*, 2016.

[227] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, “Hand keypoint detection in single images using multiview bootstrapping,” in *CVPR*, 2017.

[228] S. Qi, W. Wang, B. Jia, J. Shen, and S.-C. Zhu, “Learning human-object interactions by graph parsing neural networks,” 2018.

[229] J. Johnson, B. Hariharan, L. Van Der Maaten, J. Hoffman, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick, “Inferring and executing programs for visual reasoning,” in *ICCV*, 2017.