
THE UNIVERSITY *of* EDINBURGH

# Unsupervised Structure Induction and Multimodal Grounding

*Yanpeng Zhao*


Doctor of Philosophy
Institute for Language, Cognition and Computation
School of Informatics
University of Edinburgh
2023

# Abstract

Structured representations build upon symbolic abstraction (e.g., words in natural language and visual concepts in natural images), offer a principled way of encoding our perceptions about the physical world, and enable the human-like generalization of machine learning systems. The predominant paradigm for learning structured representations of the observed data has been supervised learning, but it is limited in several respects. First, supervised learning is challenging given the scarcity of labeled data. Second, conventional approaches to structured prediction have been relying on a single modality (e.g., either images or text), ignoring the learning cues that may have been specified in and can be readily obtained from other modalities of data. In this thesis, we investigate unsupervised approaches to structure induction in a multimodal setting.

Unsupervised learning is inherently difficult in general, let alone inducing complex and discrete structures from data without direct supervision. By considering the multimodal setting, we leverage the alignments between different data modalities (e.g., text, audio, and images) to facilitate the learning of structure-induction models, e.g., knowing that the individual words in "a white pigeon" always appear with the same visual object, a language parser is likely to treat them as a whole (i.e., phrase). The multimodal learning setting is practically viable because multimodal alignments are generally abundant. For example, they can be found in online posts such as news and tweets that usually contain images and associated text, and in (YouTube) videos, where audio, scripts, and scenes are synchronized and grounded in each other.

We develop structure-induction models, which are capable of exploiting bimodal image-text alignments, for two modalities: (1) for natural language, we consider unsupervised syntactic parsing with phrase-structure grammars and regularize the parser by using visual image groundings; and (2) for visual images, we induce scene graph representations by mapping arguments and predicates in the text to their visual counterparts (i.e., visual objects and relations among them) in an unsupervised manner. While useful, crossmodal alignments are not always abundantly available on the web, e.g., the alignments between non-speech audio and text. We tackle the challenge by sharing the visual modality between image-text alignment and image-audio alignment; images function as a pivot and connect audio and text. The contributions of this thesis span from model development to data collection. We demonstrated the feasibility of applying multimodal learning techniques to unsupervised structure induction and multimodal alignment collection. Our work opens up new avenues for multimodal and unsupervised structured representation learning.

# Lay Summary

In this thesis, we focus on learning to represent the observed data, e.g., images, audio, and text. This is a fundamental step in many machine learning systems. Intuitively, to understand what we humans see, hear, and read, machine learning systems need to transform our observations into representations that are apprehensible to them. For example, to understand an image that consists of hundreds of thousands of pixels, machines usually transform it into a low-dimensional vector. But, cramming the whole image into a vector entangles visual concepts, i.e., it is unclear which dimensions correspond to which objects/attributes/relations. This poses a challenge for performing complex queries regarding an image, e.g., "*what is the color of the ball held by the boy?*" Conceptually, to answer the question, we would need a scene graph that represents objects, and their attributes and relations, so we can reason over the graph step by step. Since the reasoning steps are specified in the language form, we would also need to transform the query into another form that represents the reasoning chain.

To tackle the challenge, we set out to study the problem of structure induction. In particular, we are interested in learning structure-induction models in an unsupervised manner and from multimodal data. Unsupervised learning means that we use unlabeled data, e.g., images are not annotated with graph structures; instead, we will induce them from images. Multimodal learning means that: rather than learning from a single modality (e.g., either images or text), we exploit additional learning cues provided by other data modalities. Multimodal learning has been through aligned multimodal data, e.g., captioning data wherein each caption describes the associated image.

In these settings, we develop structure-induction models for text and images. For text, we leverage visual groundings for the induction of phrase structures that describe the process of merging adjacent phrases into larger phrases. The reason that visual groundings are helpful is based on the observation that: knowing adjacent words/phrases refer to the same visual object, a parser should probably treat them as a whole. For images, we induce their scene graph representations, i.e., labeling objects and predicting relations between objects. The object and relation labels are expressed as words in captions; our parser is trained to map them to objects and object pairs, respectively. We further tackle the issue of scarce multimodal alignments between environmental sound and language descriptions; we propose a model that connects audio and text via image pivots. In this thesis, we demonstrated the feasibility of applying multimodal learning techniques to unsupervised structure induction. Our work opens up new avenues for multimodal and unsupervised structured representation learning.

# Acknowledgements

I would like to first thank Ivan, my supervisor, for guiding me through my Ph.D. journey. Ivan has been steadfastly patient with and supportive of me, and has shaped my research perspectives that I will take with me all the way forward.

I also wanted to thank Shay Cohen and Desmond Elliott for examining my thesis, and special thanks to Shay for helpful discussions at the beginning of my third year. I am grateful to Mirella Lapata, Frank Keller, and Hakan Bilen for providing feedback on my annual reviews. Throughout this journey, I had the good fortune of expanding my research via internships. I would like to thank Jack Hessel, Youngjae Kim, and Yejin Choi for hosting me at Allen Institute for AI, and Xinyan Xiao and Liang Huang for offering me an exceptional internship opportunity at Baidu Research. I appreciate the freedom of exploring novel research directions, and many thanks to you all for connecting me with the excellent researchers on the team and beyond.

Over the years in Edinburgh, I have also met wonderful people and had good times. Thanks to the excellent members of Ivan's team: Michael, Diego, Caio, Tom, Serhii, Lena, Nicola, Xinchi, Chunchuan, Bailin, Arthur, Matthias, and Verna. I will definitely miss our discussions on research and beyond. In particular, I wanted to offer my gratitude to Diego and Serhii for collaborating with me on research. I am also thankful to my lovely office mates: Javad, Rui, Jiangming, Tom, Arthur, Karim, Nikos, Ruchika, Amir, and Mohammad. I enjoyed the cheerful vibe we created in the office, and special thanks to Arthur, Nikos, and Amir for fun road trips during the pandemic. I would further like to extend my thanks to all the colleagues I have interacted with, including Biao, Bowen, Kai, Shangmin, Muyang, Tianyi, Weihong, Hao, Yang, Yumo, Zhijiang, Guillermo, Marc, Parag, and Rohit. Thank you all for the good time we spent together on bouldering, cycling, hiking, skiing, dining, etc.

Finally, I would like to thank my family and friends for their unconditional support. Thank you for sticking with me through thick and thin. None of this would have been possible without you around.

# Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Yanpeng Zhao*)

*To my grandpa.*

# Table of Contents

# Chapter 1

# Introduction

## 1.1 Motivation

An important challenge in machine learning is to model diverse modalities of data such as images, audio, and text. In most scenarios, machine learning methods for modeling the observed data involve deriving alternative representations of the data, e.g., encoding images/text into continuous vectors or structured forms. We refer to this process as representation learning. There are many reasons for performing representation learning such as reducing the dimension of the observed data and finding better representations that are more predictive of some targets. Apart from these general reasons, we would like to highlight the reasons related to cognitive modeling and engineering.

In cognitive modeling, a primary goal of investigating representation learning is to advance our understanding of human intelligence. For example, there has been a wealth of research providing insights into the way humans acquire language and perceive the physical world. Specifically, hierarchical tree structures of sentences are used to model the way humans process language, and part-whole hierarchy is proposed to explain how humans perceive visual scenes. From the perspective of engineering, a crucial reason for studying representation learning is to find ways of communicating with intelligent systems, thus research on representation learning has focused on building efficient and interpretable interfaces. For example, search engines like Google need to represent heterogeneous contents on the Internet in such a way that it can be queried efficiently, voice assistants such as Alexa and Siri need to transform human language into a representation that is suitable for machines to comprehend, and ideally, general-purpose representations shared across domains, tasks, and even data modalities are preferred, in order to make applications flexible.

In this thesis, we study the problem of representation learning. We discuss the limitations of traditional learning settings and motivate new learning settings.

- **Supervised learning → unsupervised learning.** The conventional paradigm for learning representations of the observed data has been supervised learning and thus relies on labeled data, e.g., a supervised model for sentence structure prediction has to be trained on sentences annotated with desired target structures. Since human-annotated resources are usually specific to a domain (e.g., all annotated sentences may come from the same news domain), the representations derived via supervised learning tend to be difficult to generalize across domains, which is contrary to the goal of learning general-purpose representations. Moreover, due to the prohibitive cost of human annotations, labeled data is usually scarce, rendering supervised representation learning more challenging.

  Instead, recent deep learning approaches to representation learning eschew labeled data in favor of unsupervised learning. In particular, training large models on plentiful unlabeled data has been a popular means of learning general-purpose representations, which have demonstrated astonishing performance in various downstream tasks such as question answering (Devlin et al., 2019; Dong et al., 2019), image classification (Dosovitskiy et al., 2021; Radford et al., 2021), audio classification (Baevski et al., 2022), protein structure prediction (Jumper et al., 2021), and game playing (Kramár et al., 2022).

- **Continuous representation → structured representation.** While deep learning has propelled the phenomenal advance in artificial intelligence, deep learning models still fall short of human intelligence in some prominent aspects and, in particular, compositional generalization. Compositional generalization reflects humans' capability of recombining known components to construct novel inferences, predictions, and behaviors. A promising strategy for improving the compositional generalizability of neural models is through structured representations, e.g., hierarchical tree structures that are built from reusable production rules, and general graph structures that are composed of atomic entities and relations. Structured representations build upon symbolic abstraction, offer an implementation of the concept of compositional generalization, and importantly are amenable to learning from data (Ganchev et al., 2010; Kim et al., 2019b) and building into neural models (Tai et al., 2015; Kipf and Welling, 2017).

It is not entirely surprising that neural models struggle to generalize compositionally. To efficiently train large models on big data so as to maximize the benefits of model scaling and data scaling (Kaplan et al., 2020), neural models have been designed to prioritize *end-to-end learning* and minimize structural biases in computations and representations (e.g., ResNet (He et al., 2016) and Transformer (Vaswani et al., 2017)). While they have demonstrated great success, they fail spectacularly in tasks that require compositional generalization (Lake and Baroni, 2018; Keysers et al., 2020), presumably because they lack structured modeling. Thus, we advocate structured representation learning, which we believe will benefit challenging tasks that rely on reasoning (Barrett et al., 2018; Hudson and Manning, 2019a), involve learning from a few observations (Lake et al., 2015, 2017), and require domain generalization (Kim and Linzen, 2020).

- **Single modality → multimodality.** To eschew supervised learning and tackle the compositional generalization issue, we investigate unsupervised methods for learning structured representations. Structured representations are an important component of structure-aware neural models, e.g., graph neural networks follow a given structure to perform computations (Kipf and Welling, 2017). However, inducing complex and discrete structures from data without direct supervision is inherently difficult. To induce desired structures, we usually make *a priori* representational and computational assumptions, e.g., to induce tree structures, we define production rules and constrain the way they combine (representation). But, in order for learning to be tractable and for models to generalize, these assumptions are generally overly flexible, e.g., we may assume each rule in a tree is independent of the others (computation). Consequently, it is rather difficult to induce meaningful structures that best represent the data.

  A possible solution to this issue is to use extra regularization to further disambiguate models' predictions. Among available regularization choices, we are particularly interested in using multimodal alignment to regularize structure-induction models. Multimodal alignment has been in the form of aligned multimodal data such as images and text in captioning data. An explanation of the usefulness of multimodal alignment is based on the observation that: the same concept can be grounded in different modalities and aligned with each other via paired multimodal data, e.g., "a white pigeon" appears in both a caption "*a white pigeon sits in the grass*" and the associated images it describes. Regularities of this kind help with,

in a way, identifying meaningful structural patterns such as phrases in language. Specifically, while "white" and "pigeon" are two separate words in captions, they refer to (or appear with) the same visual region, thus knowing this fact supposedly encourages a language parser to treat them as a whole. Since phrases are basic units of phrase structures, visual groundings potentially help with phrase-structure induction. Conversely, the vector representations of visual "pigeon" and "grass" may be difficult to distinguish because they are spatially close to each other in an image, but knowing that they are two separate concepts in language, an image parser probably learns to make their visual representations more separable, i.e., textual groundings cause more symbolic object representations, an important component of structured image representations like scene graphs.

However, conventional approaches to structure induction have been relying on a single modality, so they can not use complementary learning cues provided by other modalities. Instead, we advocate learning structured representations from aligned multimodal data.

In these new settings, we develop novel representation learning models and tackle data scarcity issues we will confront in learning the models.

- **Unsupervised structure induction with multimodal alignment.** We investigate unsupervised structure induction for images and text in the multimodal setting because (1) there has been a wealth of research on inducing structures of images/text from individual modalities, thus we can build upon the existing work and focus on developing models that are capable of learning from multiple modalities; and (2) the image-text setting allows for using prevailing image-text pre-training techniques, and importantly, it requires little annotation effort since image-text alignment is abundantly available on the web and is relatively easy to collect (e.g., via online posts that usually contain images associated with language descriptions).

  For text, we focus on unsupervised syntactic structure induction and propose an end-to-end fully-differentiable visually grounded learning framework. The framework allows for learning a parser via joint unsupervised learning on raw text and contrastive image-text pre-training on image-text pairs. For images, we formulate the task of unsupervised scene graph induction and propose a visually-grounded masked language model. Our model can be trained on image-text pairs via a masked language modeling objective.

- **Unsupervised curation of multimodal alignment.** Our multimodal models for unsupervised structure induction rely on image-text pre-training and generally require large-scale parallel image-text data, but unlike visual and textual modalities, many other modality pairs lack sufficient co-occurrence data. This poses a grand challenge for extending our models to modality pairs that have scarce aligned data.

  We identify this problem in the pair of textual and auditory modalities and, specifically, text and non-speech audio (i.e., environmental sound). It is supposedly caused by *reporting bias:* we are less likely to describe what we hear than what we see. To tackle the problem, we propose a pivoting model; it connects audio and text via image pivots, without using any parallel audio-text data.

## 1.2 Overview

In this thesis, we focus on the problem of structure induction and investigate unsupervised approaches that exploit multimodal alignment (modeling). We also address scenarios wherein multimodal alignment is scarce and provide a remedy (multimodal alignment collection).

Structured formalisms are needed in many scenarios, from representing direct perceptual experiences such as images, audio, and text to modeling more sophisticated cognitive activities such as reasoning, decision-making, and problem-solving. In this thesis, we investigate unsupervised structure induction for images and text, on which there has been a wealth of research, but very little considers the multimodal setting.

For text, we focus on the classical syntactic structures (Chomsky, 1956, 1957; Hopcroft et al., 2006). The related structure induction problem is known as unsupervised grammar induction, and has a long history in computational linguistics (Carroll and Charniak, 1992; Pereira and Schabes, 1992; Brill, 1993; Stolcke and Omohundro, 1994; Klein and Manning, 2004). Though there have been various paradigms for grammar induction such as grammar induction via language modeling (Shen et al., 2018; Kim et al., 2019b) and grammar induction as syntax probing (Kim et al., 2020; Wu et al., 2020), all of them have relied solely on text, while we are more interested in a grounded learning setting, where parsers are learned from downstream tasks such as sentiment classification, textual entailment, and natural language inference (Yogatama et al., 2017; Choi et al., 2018; Maillard et al., 2019). By analogy to this task-dependent grammar induction, we formulate a multimodal learning task for grammar induction.

For images, we study scene graph representations (Johnson et al., 2015). Scene

graphs have been shown to be helpful in a variety of vision tasks, including image retrieval, captioning, and generation (Johnson et al., 2015, 2018; Yang et al., 2019) and visual question answering (Shi et al., 2019b; Hudson and Manning, 2019a,b). This has further stimulated the development of scene graph generation methods (Xu et al., 2017; Yang et al., 2018; Zareian et al., 2020a; Tang et al., 2020). The prevailing learning paradigm for scene graph generation has been supervised learning and thus relies on labeled data, but labeled data is prohibitively costly and the widely-used annotated resources suffer from the issue of skewed label distribution (Zellers et al., 2018; Yao et al., 2021). Those that do not require labeled data are, however, pipeline models (Ye and Kovashka, 2021); they rely on external language parsers and pre-trained object detectors to preprocess the inputs, and thus inherently suffer from the errors accumulated from preprocessing steps. These challenges necessitate the need for end-to-end unsupervised induction of scene graphs.

Our first study focuses on inducing syntactic structures of text with visual supervision, i.e., images that are aligned with the text. Drawing inspiration from the efforts of inducing text structures from supervised tasks (Havrylov et al., 2019; Maillard et al., 2019), we adopt a multimodal learning task and attempt to learn parsers from it. Specifically, we choose the contrastive image-text pre-training task (Radford et al., 2021). Contrastive image-text learning connects images and text through learning a joint image-text vector space. It requires little annotation effort (Jia et al., 2021) while providing a way of combining multimodal learning with grammar induction. We choose probabilistic context-free grammars as our parsing model and achieve an end-to-end fully-differentiable learning framework. Our model can be seen as a neural-symbolic (i.e., hybrid) model in the sense that it combines symbolic language modeling via grammar and continuous image-text modeling via contrastive learning. Apart from optimizing image-text alignment (i.e., task-dependent learning), our parser allows for optimizing a language modeling objective (i.e., self-supervised learning). Intuitively, the prior knowledge about image-text alignment is injected into the parser during joint training; it functions as a regularizer and potentially leads to a better parser.

In the previous study, we empirically found that visual groundings help with inducing structures of text. Conversely, *will textual groundings help with inducing structures of images?* To answer this question, we investigate unsupervised induction of scene graph representations of images with textual supervision, i.e., language descriptions that are aligned with the images. A scene graph consists of nodes and edges, where each node is an abstract description of a visual object (e.g., “girl” and “flowers”), and

each edge connects two objects via a relation expressed in a word (e.g., "hold" in the tuple (girl, hold, flowers)). Our ultimate goal is to perform object segmentation (representation), object labeling (abstraction), and visual relation prediction (composition) within a unified framework. But in this work, we assume that visual object representations have been given (e.g., via a pre-trained object detector) and focus only on object classification and relation prediction. We design a unified neural module to tackle the two subtasks. The customized module can be integrated into image-conditioned masked language models, so we can further learn it from abundantly available image-text pairs via multimodal masked language modeling (Lu et al., 2019). Our model can be seen as a unified connectionist model (*cf.* hybrid models) since it follows the philosophy of unified approaches within the framework of connectionism (Greff et al., 2020). To quantify model performance, we propose automatic evaluation metrics and create an artificial image-captioning dataset that focuses on spatial relational reasoning.

The findings from our previous work suggest that multimodal alignment helps with unsupervised text and image structure induction. Unfortunately, not all modality pairs have abundantly available co-occurrence data. This hinders the wider application of multimodal learning techniques to unsupervised structure induction. To tackle the challenge, we further investigate unsupervised curation of large-scale multimodal alignment and, specifically, the alignment between text and non-speech audio (i.e., environmental sound). Complementary to speech, environmental sound provides rich and diverse perspectives on the physical world, but it lacks large-scale and closely-related natural language descriptions. Inspired by pivot-based models for unsupervised machine translation (Wu and Wang, 2007), we propose to connect audio and text via images. The pivoting idea mirrors the connection between our mental imagery experience and language experience: hearing a sound, humans can visually *imagine* possibly associated events and literally *describe* them. Pivoting is practically viable because there are abundantly available image-text and video-audio co-occurrences on the web, from which we can train image-text and image-audio alignment models via contrastive bimodal pre-training (Radford et al., 2021). By sharing the visual modality between the two alignment models, we link audio and text implicitly in the vector space and mine novel audio-text pairs that never occurred together.

The main contributions of this thesis are:

- An end-to-end fully-differentiable framework for inducing phrase-structure grammars of language with natural image supervision.

- A multimodal masked language model for inducing scene graph representations of images with natural language supervision.
- A pivoting model for unsupervised multimodal alignment induction. We demonstrate its effectiveness in inducing audio-text alignment via image pivots.

## 1.3 Outline

We organize the rest of this thesis into six chapters. After reviewing background materials for unsupervised structure induction and multimodal learning (Chapter 2), we elaborate on our efforts in (1) unsupervised image and text structure induction in the multimodal setting (Chapters 3–4); and (2) unsupervised curation of audio-text alignment (Chapter 5). We conclude the thesis with Chapter 6.

**Chapter 2** provides an overview of background materials related to unsupervised structure induction and multimodal learning. We first review typical formalisms of text structures and image structures and, in particular, expand on (1) phrase structures of text (Chomsky, 1957; Allerton, 2016); and (2) scene graphs (Johnson et al., 2015) and scene grammars of images (Zhu and Mumford, 2006; Siskind et al., 2007). We further describe machine learning techniques for unsupervised structure induction and discuss multimodal learning for image and text understanding.

**Chapter 3** presents a fully-differentiable neural-symbolic model for inducing phrase-structure grammars of text with visual image supervision. We explicitly model hidden syntactic structures of text via a latent variable model (Kim et al., 2019b). Motivated by the observation that language is largely grounded in visual perceptions, we incorporate this prior knowledge (i.e., image-text alignment) into grammar induction via contrastive image-text learning. We learn our model by jointly optimizing a language modeling objective and an image-text alignment loss. Once trained, our parser can be directly used to parse text, without requiring the aligned images. We compare our visually grounded parser with the parser learned from text alone. Our experimental results suggest that visual groundings help with grammar induction. This chapter is based on the work published in Zhao and Titov (2020) and Zhao and Titov (2023a).

**Chapter 4** presents a unified connectionist model for inducing scene graph representations of images with natural language supervision. We formulate the structure

induction task as labeling objects and assigning relations to object pairs, i.e., predicting a label (i.e., word) given each object (pair). We design a unified computational module, which is capable of both object labeling and relation prediction. The module is integrated into an image-conditioned masked language model, thus we can further learn it via multimodal masked language modeling (Lu et al., 2019). Once trained, by virtue of the architecture design, our model can be directly used to make predictions given images, without requiring the aligned text. To quantify model performance, we create an artificial image-captioning dataset and propose automatic evaluation metrics. Our model demonstrates reasonable performance when using symbolic object representations, but the experimental results also suggest difficulties in inducing scene graphs of images via image-text pre-training. This chapter is based on a technical report available via Zhao and Titov (2023b).

**Chapter 5** presents a pivoting model for inducing multimodal alignment and, specifically, audio-text alignment. Multimodal alignment is referred to as prior knowledge that the same concept can be found in different modalities of data. It forms the basis for many multimodal learning tasks, including the structure induction problems we have studied. However, not all kinds of multimodal alignment are as abundantly available as image-text alignment, e.g., the alignment between text and non-speech audio (i.e., environmental sound). Inspired by pivot-based machine translation (Wu and Wang, 2007), we propose to induce audio-text alignment by using the images as the pivot. Our idea is to take advantage of the abundance of image-text alignment (e.g., via online news and tweets) and image-audio alignment (e.g., via YouTube videos) and connect audio and text via images. We realize the idea via a tri-modal contrastive pre-training framework. We conduct audio-text pre-training on the mined audio-text data; fine-tuning the learned audio encoder results in state-of-the-art results on a wide range of audio understanding tasks. This chapter is adapted from Zhao et al. (2022).

**Chapter 6** concludes the thesis. Along with a summary of our studies, we acknowledge the limitations of our work and discuss potential improvements and extensions, which we leave for future work.

# Chapter 2

# Background

In this chapter, we review background materials related to unsupervised structure induction and multimodal learning. We first present an overview of formalisms for structured representations of two data modalities: text and images. For text, we will elaborate on the syntax of language; for images, we will introduce scene graphs and scene grammars. Then we review unsupervised approaches to learning two families of structure induction models: neural-symbolic (hybrid) models and structure-aware (connectionist) models. Since we posit unsupervised structure induction in a multimodal setting and thus rely on multimodal data such as images and associated captions, we will also cover multimodal learning for image and text understanding.

## 2.1 Structured Text Representations

We are interested in structured modeling (i.e., syntax) of text on the sentence level (*cf.* the morphology/lexicon/phrase/document level). We first elaborate on phrase-structure grammar, a formalism for syntactic analysis (Section 2.1.1), which will be the theme of Chapter 3. Then we present a brief overview of related structured formalisms for sentence representation (Section 2.1.2).

### 2.1.1 Phrase-Structure Grammars

Phrase-structure grammars are a generative model of language. They specify the syntax of a language via a finite set of rewrite rules, e.g., $A \rightarrow B$ reads as "the string $A$ is replaced by the string $B$." Given a phrase-structure grammar, every sentence it admits is associated with a set of derivations, and each derivation is composed of a group of

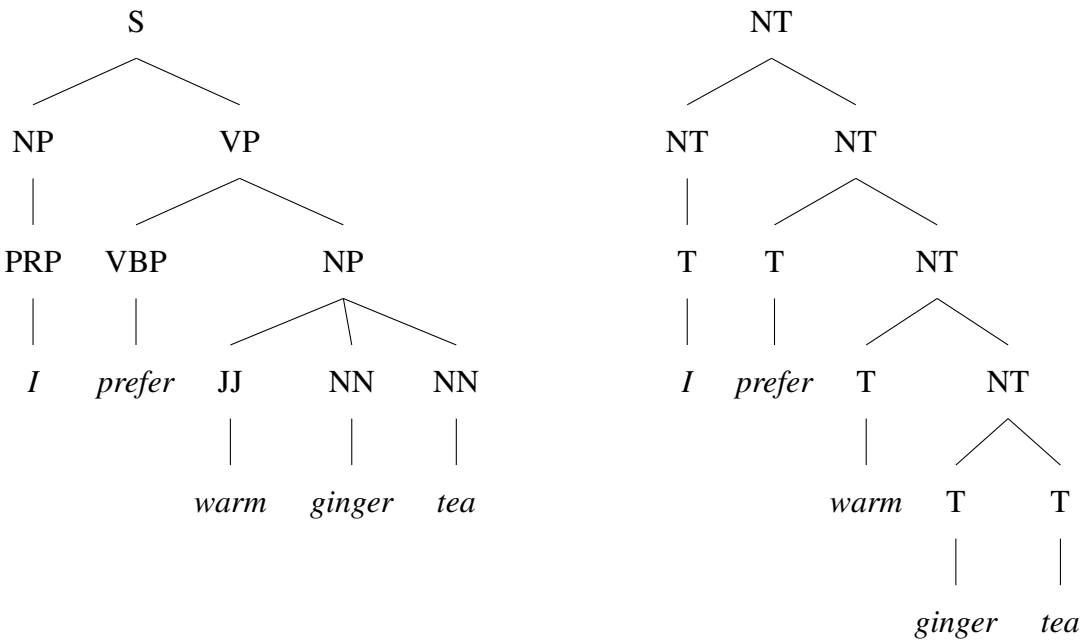



Figure 2.1: Constituent analyses for "*I prefer warm ginger tea*". Left: a parse tree in the Penn Treebank style. Right: binarized unlabeled parse tree.

rules from the grammar (see Figure 2.1). When traversed in a top-down manner, a derivation describes the process of recursively breaking down a sentence/phrase into smaller constituent phrases until the minimum constituent units (e.g., words or morphemes) are reached. These phrases are classified into one of phrasal categories such as noun phrase (NP) and verb phrase (VP) if they consist of two or more words,[1] and into one of lexical categories (i.e., part-of-speech tags such as Verb (V) and Noun (N)) if they are words (Chomsky, 1956). As we will see, these syntactic (phrasal/lexical) categories are used as grammar symbols in grammar rules. Since a derivation relies on the abstraction of constituent phrases and describes the structural relation between each constituent and the larger constituent that contains it, it is also called *constituency tree*, and phrase-structure grammars are also known as *constituency grammars*.

Depending on the restrictions placed on the form of phrase structure rules, Chomsky (1956) describes four types of grammars (i.e., regular grammar, context-free grammar, context-sensitive grammar, and recursively enumerable grammar), which form the Chomsky hierarchy. Among them, we are interested in context-free grammars. A Context-Free Grammar (CFG) is formally defined as a 4-tuple $\mathcal{G} = (\mathcal{N}, \Sigma, \mathcal{R}, S)$:

[1] Traditionally, a phrase is composed of multiple words, but, in some cases, single words can also be phrases, e.g., "I" in Figure 2.1, which is first classified into the lexical category PRP, short for "personal pronoun", and is further classified into the phrasal category NP.

- $\mathcal{N}$ is a finite set of grammar/nonterminal symbols (nonterminals for short);
- $\Sigma$ is a finite set of terminal symbols (terminals such as words for short);
- $\mathcal{R}$ is a set of grammar rules of the form:

$$A \to \alpha \quad \text{with} \quad A \in \mathcal{N}, \quad \alpha \in (\mathcal{N} \cup \Sigma)^*; \tag{2.1}$$

- $S \in \mathcal{N}$ is the start symbol dominating the whole sentence.

The context-freeness is evident from the fact that $A$ can be rewritten by $\alpha$ regardless of its context. To facilitate representation and natural language parsing, i.e., inferring the hidden tree structure that yields a sentence, context-free grammars are usually written in the Chomsky norm form (CNF), which restricts the form of grammar rules to be either $A \to BC$ or $A \to w$, where $A, B, C \in \mathcal{N}$ and $w \in \Sigma$ (Chomsky, 1959; Hopcroft et al., 2006). To distinguish phrasal categories from lexical categories, we adopt a formalism that defines a CFG in CNF as a 5-tuple $\mathcal{G} = (\mathcal{N}, \mathcal{P}, \Sigma, \mathcal{R}, S)$. It is distinguished from the previous definition of CFG in the following aspects:

- $\mathcal{N}$ is a finite set of phrasal categories and $S \notin \mathcal{N}$ (nonterminals for short);
- $\mathcal{P}$ is a finite set of part-of-speech tags and $\emptyset = \mathcal{N} \cap \mathcal{P}$ (preterminals for short);
- $\mathcal{R}$ is a set of grammar rules of the form:

$$\text{(1)} \quad S \to A \qquad \text{with} \quad A \in \mathcal{N}, \tag{2.2}$$

$$\text{(3)} \quad A \to w \qquad \text{with} \quad A \in \mathcal{P}, \quad w \in \Sigma, \tag{2.3}$$

$$\text{(2)} \quad A \to BC \qquad \text{with} \quad A \in \mathcal{N}, \quad B, C \in \mathcal{N} \cup \mathcal{P}. \tag{2.4}$$

Phrase-structure grammars are initially proposed as a simple yet revealing alternative to the finite-state Markov process, which lacks the capability of describing English (Chomsky, 1956). The class of CFG has been investigated in the context of language acquisition, where CFG serves as an approximation of the grammars of human languages (Shieber, 1985). Studies on the learnability of CFG have formed opposite (i.e., for and against) attitudes toward "*the argument from the poverty of the stimulus,*" which argues that the experience of language use is insufficient for language acquisition, so human infants are born with at least some innate linguistic-specific knowledge (Chomsky, 1965). While early analysis by Gold (1967) shows strong support for the argument, both empirical assessments of the argument (Pullum and Scholz,

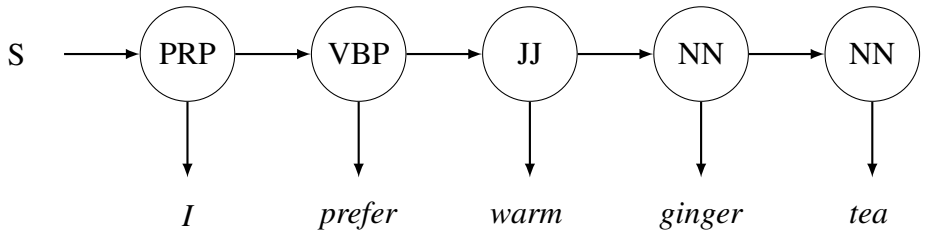



Figure 2.2: Linear grammatical structure for "*I prefer warm ginger tea*". S is the start symbol as in context-free grammars. The symbols in circles are part-of-speech tags.

2002) and the success of computational approaches to unsupervised CFG induction indicate the opposite, i.e., the fact that CFG can be automatically induced from text data alone undermines the argument (Zuidema, 2002; Klein and Manning, 2002). In addition to the cognitive implications of unsupervised CFG induction, most computational approaches are intended to examine the generative capacity of CFG models, i.e., to which extent the underlying tree structures of sentences can be recovered by CFG models (Klein and Manning, 2004; Jiang et al., 2016; Kim et al., 2019b). To deal with ambiguity (i.e., different derivations yield the same sentence), these approaches typically adopt Probabilistic CFG (PCFG; Booth and Thompson (1973)). PCFG extends CFG (the 5-tuple $\mathcal{G}$) by attaching each grammar rule with a probability such that:

$$\sum_{\alpha} p(A \rightarrow \alpha) = 1\,, \text{ where } A \in \mathcal{N} \cup \mathcal{P} \cup \{S\}\,. \tag{2.5}$$

In principle, PCFG can be learned (i.e., parameter estimation) via the expectation-maximization algorithm (Dempster et al., 1977), and the most probable tree of a given sentence can be inferred via, for example, the CKY algorithm (Cocke, 1969; Kasami, 1966; Younger, 1967).

### 2.1.2 Related Structured Formalisms

Syntactic structure analysis focuses on the syntax of language, i.e., the underlying process that governs the arrangement of words in a sentence. Apart from phrase-structure grammars, which describe hierarchical grammatical structures and give rise to deep syntactic analyses, there are other formalisms for syntactic analysis such as regular grammars and dependency grammars.

Regular grammars have been a popular formalism for shallow syntactic analysis. According to the Chomsky hierarchy, regular grammars are subsumed by CFG (Chomsky, 1956). A regular grammar consists of rules of the form $A \rightarrow wB$, where $A, B$ are

nonterminals and $w$ is a terminal; it describes linear grammatical structures of sentences (see Figure 2.2). The related inference task, which is known as part-of-speech (POS) tagging, aims to assign each word to a lexical category (i.e., a part of speech such as Verb (V) and Noun (N)) and gives rise to a bijective mapping between words and parts of speech. POS tagging has been the basis for more involved syntactic analysis such as constituency parsing.

Dependency grammars are another widely-used formalism for deep syntactic analysis. Differently from phrase-structure grammars, which emphasize the modeling of constituents, dependency grammars explicitly represent grammatical relations between words in a sentence (see the first parse in Figure 2.3). Such binary relations are represented as *directed* edges from one lexicon (i.e., headwords or heads) to the other (i.e., dependent words or dependants) (Jurafsky and Martin, 2000). The edge between a head and each of its direct dependants may be assigned to a label, which is chosen from a finite set of grammatical relations. Starting from the *root* node/word, when traversing all the directed edges recursively, all the words of the sentence will be visited and the traverse path will form a dependency tree (see the second parse in Figure 2.3). Each node of the tree is a headword and dominates a group of dependent/child words; together they form a phrase. For example, "tea" is a head with dependants "warm" and "ginger", and they form the phrase "warm ginger tea". But, compared to constituency grammars, dependency grammars have a weaker concept of phrases because there are no phrasal nodes (i.e., syntactic categories) in dependency structures. While word order is predictive of dependency structures, dependency structures on their own are determined by relations between words and do not explicitly represent phrases that respect word order. Consequently, dependency grammars lend themselves to modeling languages that have relatively free word order, including Catalan, Czech, and Finnish.

However, sentences that are grammatically correct are not necessarily semantically meaningful, e.g., "*colorless green ideas sleep furiously*" by Chomsky (1956). Thus, apart from analyzing the syntax of language, it is also important to understand semantics via, for example, semantic structure analysis, which focuses on the meaning representations of utterances. Semantic structure analysis usually resorts to probabilistic models to represent the meaning of an utterance in terms of meaning units (e.g., lexicons and phrases) and relate meaning units via semantic relations. There have been several formalisms developed for structured meaning representations, which fall roughly into two categories: (1) broad-coverage meaning representations and (2) executable meaning representations.

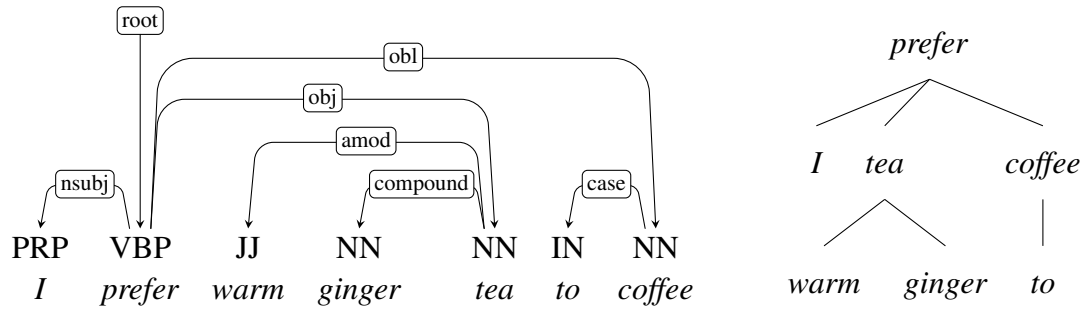



Figure 2.3: Dependency analyses for "*I prefer warm ginger tea to coffee*". The dependency relations in the first parse are optional. Starting from the "root" node, traversing all the words following the directed edges leads to the right tree structure.

Broad-coverage meaning representations aim at a unified representation across domains and tasks. In the area of shallow semantic structure analysis, predicate-argument structures have been the main focus (Gildea and Jurafsky, 2000). For deep semantic structure analysis, there are several popular formalisms, including semantic dependency structures (Oepen et al., 2014), abstract meaning representations (Banarescu et al., 2013), and combinatory categorial grammars (Steedman, 2000).[2] Notably, semantic dependency structures and combinatory categorial grammars are motivated by and built on syntactic dependency grammars and constituency grammars, respectively. Differently from broad-coverage meaning representations, executable meaning representations are usually domain-specific and application-oriented; they can be seen as machine-readable languages, including logic forms (Kate et al., 2005) and SQL (Yu et al., 2018). We will use logical form representations to automatically synthesize image captions in Chapter 4.

## 2.2 Structured Image Representations

We describe two formalisms of structured representations of images: (1) scene graphs with a focus on visual relation modeling (Section 2.2.1); and (2) scene grammars with an emphasis on hierarchy modeling (Section 2.2.2). We will further expand on scene graphs in Chapter 4.

[2]Combinatory categorial grammar derivations on their own do not necessarily carry semantics (Hockenmaier and Steedman, 2007), but they can be augmented with semantic interpretations via lambda calculus (Zettlemoyer and Collins, 2005).

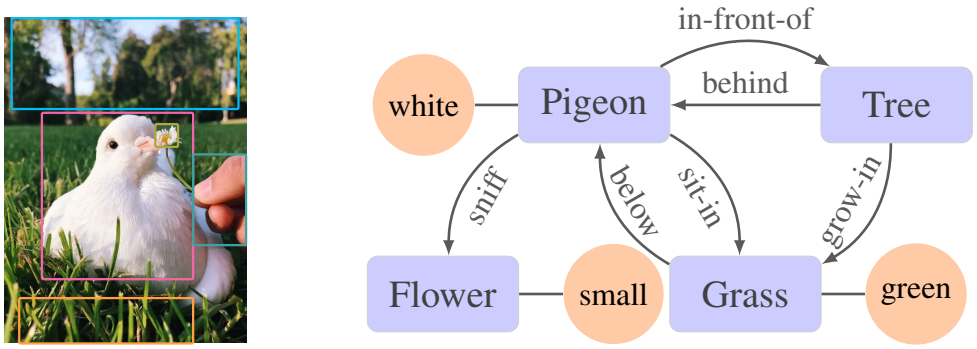



Figure 2.4: A scene graph (right) representation of the natural image (left). The scene graph is used for an illustration and does not include the "hand" object. Each object is localized in the image as a bounding box (not shown in the scene graph).

### 2.2.1 Scene Graphs

A scene graph is a directed graph consisting of objects as nodes and relationships as edges (see Figure 2.4). Formally, it can be defined as a 3-tuple $\mathcal{G} = (\mathcal{O}, \mathcal{P}, \mathcal{R})$, where

- $\mathcal{O}$ is a finite set of *localized* objects in a given image. Each object is labeled with a category (e.g., "pigeon", "flower", and "grass") and is associated with a bounding box (i.e., being localized). There might be multiple objects belonging to the same category, e.g., "man.01" and "man.02". They are distinguished from each other by the associated bounding boxes;

- $\mathcal{P}$ is a finite set of predicates. Each predicate represents an edge label (or relation type) such as "sniff", "below", and "grow-in". The special predicate "null" $\in \mathcal{P}$ indicates no relations;

- $\mathcal{R}$ is a set of relations in the form of $(o, p, o')$, where $o, o' \in \mathcal{O}$ and $p \in \mathcal{P}$. For example, (pigeon, sniff, flower) and (pigeon, sit-in, grass).

The scene graph representations of images build upon object-level abstractions and abstract away lower-level visual concepts (i.e., parts of objects), so they emphasize relations among objects. As a popular manually-annotated resource, Visual Genome (Krishna et al., 2017) also annotates object attributes. In this case, we can accordingly extend the above 3-tuple formalism to a 4-tuple $\mathcal{G} = (\mathcal{O}, \mathcal{A}, \mathcal{P}, \mathcal{R})$, where

- $\mathcal{A}$ is a finite set of attributes. Attributes can be color (e.g., green), state (e.g., smiling), and material (e.g., wood). Each attribute is associated with a set of admissible assignments, e.g., the "material" attribute can be assigned to wood, metal, glass, plastic, etc.;

- $\mathcal{R}$ is a set of relations in the form of (1) $(o, p, o')$ for describing relations between objects, where $o, o' \in O$ and $p \in \mathcal{P}$; and (2) $(o, a)$ for describing object attributes, where $o \in O$, and $a \in \mathcal{A}$ and is instantiated by a concrete attribute value, e.g., (lamp, off), (flower, small), and (pigeon, white).

### 2.2.2 Scene Grammars

While scene graphs offer concise representations of images, they lack the capacity to encode the hierarchies of visual scenes. Intuitively, a hierarchy describes the process of grouping primitives (i.e., visual words similar to tokens in text) into parts, parts into objects, and objects into scenes. This hierarchical construction process mirrors the compositional aspect of humans' perception of visual images (Biederman, 1987; Essen et al., 1992; Ullman, 1995; Sheinberg and Logothetis, 2001), and presumably hierarchical structured representations in terms of reusable parts and objects are preferred for better image understanding such as object detection and segmentation (Tu et al., 2003; Jin and Geman, 2006; Tighe and Lazebnik, 2010; Xu et al., 2022), scene classification (Socher et al., 2011), and image captioning (Yao et al., 2019).

Parsing images into hierarchical tree structures has been considered a challenging problem since the 1970s (Ohta et al., 1978). By analogy to natural language parsing, early approaches to modeling hierarchical structures of images have resorted to grammar-based models such as CFG (Chomsky, 1956). Given a CFG of two-dimensional (2D) images, each image admitted by the CFG is associated with a 3D derivation, which consists of grammar rules from the CFG. The 3D derivation essentially describes the process of recursively breaking down an image region into subregions until the minimum region units (e.g., part/pixel) are reached. While it is a straightforward extension of text CFG to 2D images, it is not practically applicable. The challenge lies in modeling visual concepts (e.g., parts and objects) that have arbitrary shapes. To find image regions that tightly encapsulate valid visual concepts (i.e., semantic segments via semantic segmentation), an inference algorithm will have to consider all possible ways of dividing an image region into subregions of arbitrary shapes, which is, however, intractable. A potential solution is to limit admissible decomposition by, for example, restricting image regions to be rectangular (see Figure 2.5). Though this restriction does not respect irregular shapes that visual concepts may have, it leads to the tractable spatial random tree model (SRT; Pollak et al. (2003a,b)). SRTs are nothing different from PCFG except that they rely on two sets

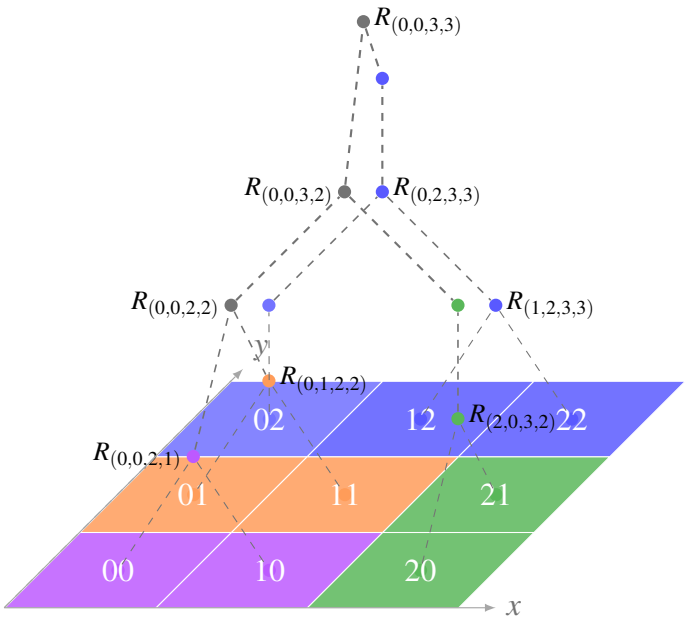



Figure 2.5: An example parse of a $3 \times 3$ grid. Each cell can be a pixel, or it can be an image token that corresponds to an image region if the image has been tokenized via the vector-quantization technique (van den Oord et al., 2017).

of binary rules to account for two different ways of (e.g., horizontal and vertical) decomposition, respectively. Formally, denoting an SRT by $\mathcal{G} = (\mathcal{N}, \mathcal{P}, \Sigma, \mathcal{R}, S)$, it is distinguished from PCFG in the following ways:

- $\Sigma$ is a finite set of terminals such as integer pixel values in image data;
- $\mathcal{R}$ consists of the following types of rules:

$$(1) \quad S \to A \qquad \text{with} \quad A \in \mathcal{N}, \tag{2.6}$$

$$(2) \quad A \to w \qquad \text{with} \quad A \in \mathcal{P}, \quad w \in \Sigma, \tag{2.7}$$

$$(3) \quad A \xrightarrow{d} BC \qquad \text{with} \quad A \in \mathcal{N}, \quad B, C \in \mathcal{N} \cup \mathcal{P}, \quad d \in \{h, v\}, \tag{2.8}$$

  where $d$ is a variable denoting the way of decomposition; $h$ and $v$ indicate horizontal and vertical decomposition, respectively.

The probabilities of the binary rules satisfy:

$$\sum_{d} \sum_{B,C} p(A \xrightarrow{d} BC) = 1 . \tag{2.9}$$

While computationally tractable in theory, it is computationally impractical to apply SRTs to natural image parsing because of overly high time and space complexities, e.g., the time complexity of SRTs is as high as $O(l^5)$ ($l = \max(H, W)$ and the

height $H$ and width $W$ of natural images are usually larger than 100). Thus, to make SRTs practically usable, additional assumptions on the input form have to be made to reduce the input scale, e.g., previous SRT-based models parse images from given semantic segmentation rather than from raw pixels. While this substantially reduces the search space because inferring semantic segments from pixels have been done prior to parsing, extra spatial constraints are needed to ensure that the composition of two subregions/segments is admissible (Siskind et al., 2007; Socher et al., 2011). Moreover, the reliance on an external object segmentation model/algorithm renders these models error-prone and may hinder them from learning meaningful (sub-)parts. To tackle these challenges, Friesen and Domingos (2018) proposed submodular field grammars (SFGs), which combine CFG with submodular Markov random fields (MRFs; Gould et al. (2009); Blake et al. (2011)). SFGs enjoy the best of both worlds: hierarchical composition afforded by CFG and flexible region-shapes modeled via tractable submodular MRFs (Kolmogorov and Zabin, 2004). However, they only discussed the inference with SFGs, without presenting learning algorithms for SFGs.

And-Or graphs are another grammar-based model for representing hierarchical structures of images. There are two types of nodes in And-Or graphs: an And-node represents a decomposition of a pattern into its parts and an Or-node switches between alternative ways of composing a pattern (Zhu and Mumford, 2006). Similarly to phrase-structure grammars, And-Or graphs can be augmented with a probabilistic interpretation by assigning a probability to each rule. The resulting probabilistic models integrate PCFG with graphical models (e.g., MRFs), where PCFG accounts for hierarchy modeling, and graphical models are responsible for context modeling (i.e., pre-specified relations between nodes that determine the validity of combining two nodes). Thus, And-Or grammars represent probabilistic *context-sensitive* grammars of images (Zhu and Mumford, 2006; Tu et al., 2013), but it is also possible to make them context-free (Tu, 2016). And-Or grammars rely on a visual vocabulary consisting of visual concepts at all composition levels. These visual concepts can be recursively combined to give rise to larger and larger patterns for parts and objects. The bottom-up composition starts with image primitives (*cf.* tokens in text) such as 1D lines (Zhao and Zhu, 2011), shape elements (Wang et al., 2013), quantized image patches (Si and Zhu, 2013), which immediately generate pixels (e.g., an image patch/segment by itself is composed of pixels). To ensure valid concept combinations, certain composition constraints are attached to visual concepts (e.g., "table-leg" expects a "table-top" rather than a "laptop"). Learning of And-Or grammars includes

structure learning and parameter learning and is typically accomplished with iterative learning approaches (Zhao and Zhu, 2011; Tu et al., 2013). Relatedly, sum-product networks (SPNs) similar in spirit to And-Or graphs have also been used to model hierarchical structures of images, but we will not cover SPNs here; instead, we refer interested readers to Poon and Domingos (2011) and (Tu, 2016).

## 2.3 Unsupervised Structure Induction

We discuss two families of structure induction models: (1) neural-symbolic (hybrid) models where the underlying structures are treated as discrete latent variables; and (2) structure-aware (connectionist) models where the hidden structures are represented via specialized computational models. We begin with an exposition of latent variable modeling (e.g., variational inference) for neural-symbolic models (Section 2.3.1), and conclude with some examples of specialized computational models that rely on customized model architectures to model discrete structures (Section 2.3.2).

### 2.3.1 Neural-Symbolic Models

Neural-symbolic models combine symbolic modeling with connectionism. In unsupervised structure induction, symbolic modeling refers to explicit modeling of the underlying discrete structures of the observed data, and connectionism indicates that models are implemented and learned by using deep learning techniques, e.g., neural networks for parameterization and gradient descent methods for optimization.

Latent variable models have been a popular tool to operationalize symbolic modeling for unsupervised structure induction. A latent variable model defines a joint distribution $p(\boldsymbol{x}, \boldsymbol{z}; \theta)$, where $\boldsymbol{x}$ indicates the observed variable (i.e., observations), $\boldsymbol{z}$ represents the unobserved variable (e.g., latent structures), and $\theta$ parameterizes the joint distribution. The joint distribution can be factorized as $p(\boldsymbol{x}, \boldsymbol{z}; \theta) = p(\boldsymbol{x}|\boldsymbol{z}; \theta)p(\boldsymbol{z}; \theta)$, which defines a data generating process. Specifically, the latent variable $\boldsymbol{z}$ is first sampled from the prior distribution $p(\boldsymbol{z}; \theta)$, then conditioning on the sampled $\boldsymbol{z}$, the observed variable $\boldsymbol{x}$ is generated via the generative model $p(\boldsymbol{x}|\boldsymbol{z}; \theta)$. In the context of structure induction, the latent variable $\boldsymbol{z}$ explicitly specifies structured representations of an observation $\boldsymbol{x}$. For example, $\boldsymbol{z}$ can be a syntactic tree yielding a sentence, and alternatively, $\boldsymbol{z}$ can be a scene graph explaining an image.

To use latent variable approaches to model the observed data, we need to compute

the probability of each observation $\boldsymbol{x}$ under a model $\theta$ by marginalizing out the latent variables $\boldsymbol{z}$ (the sum needs to be replaced with an integral when $\boldsymbol{z}$ is continuous):

$$p(\boldsymbol{x};\theta) = \sum_{\boldsymbol{z}} p(\boldsymbol{x},\boldsymbol{z};\theta) = \sum_{\boldsymbol{z}} p(\boldsymbol{x}|\boldsymbol{z};\theta)p(\boldsymbol{z};\theta)\,, \tag{2.10}$$

where $p(\boldsymbol{x};\theta)$ represents unconditional modeling; it is usually referred to as marginal or model distribution. Since the goal of unsupervised structure induction is to uncover the hidden structures $\boldsymbol{z}$, i.e., alternative representations of a given observation $\boldsymbol{x}$, we are also interested in inferring the conditional distribution of the latent variable $\boldsymbol{z}$ given the observed variable $\boldsymbol{x}$, i.e., *posterior inference*:

$$p(\boldsymbol{z}|\boldsymbol{x};\theta) = \frac{p(\boldsymbol{x},\boldsymbol{z};\theta)}{p(\boldsymbol{x};\theta)}\,, \tag{2.11}$$

which involves estimating the marginal distribution $p(\boldsymbol{x};\theta)$.

As with probabilistic modeling, given a set of observations $\mathcal{X} = \{\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{N}\}$, latent variable models are usually learned through maximizing the log-likelihood $\mathcal{L}(\theta)$:

$$\mathcal{L}(\theta) = \sum_{i} \log p(\boldsymbol{x}^{(i)};\theta) = \sum_{i} \log \sum_{\boldsymbol{z}^{(i)}} p(\boldsymbol{x}^{(i)},\boldsymbol{z}^{(i)};\theta)\,, \tag{2.12}$$

In structure-induction tasks, after finding the model $\theta^*$ that best fits the observed data, we would like to use it to infer the most probable structure $\boldsymbol{z}^*$ of a given $\boldsymbol{x}$:

$$\boldsymbol{z}^* = \arg\max_{\boldsymbol{z}} p(\boldsymbol{x},\boldsymbol{z};\theta^*)\,, \qquad \text{(Inference)}$$

$$\theta^* = \arg\max_{\theta} \mathcal{L}(\theta)\,. \qquad \text{(Learning)}$$

The challenge with latent variable models lies in estimating the marginal likelihood $p(\boldsymbol{x};\theta)$, which is needed in learning and posterior inference but requires enumerating all possible configurations of the latent variable $\boldsymbol{z}$. Depending on whether the posterior inference $p(\boldsymbol{z}|\boldsymbol{x};\theta)$ is tractable, there are different approaches to optimizing the log-likelihood function $\mathcal{L}(\theta)$. We will elaborate on them in the remainder of this section.

#### 2.3.1.1 Exact Inference

The expectation-maximization (EM) algorithm is a popular tool for maximizing the marginal likelihood when the exact posterior $p(\boldsymbol{z}|\boldsymbol{x};\theta)$ can be calculated tractably (Dempster et al., 1977). Starting from a random guess of the parameters $\theta^{(0)}$, the EM algorithm finds an estimate of the parameters by iteratively alternating between the following E-step and M-step:

- Expectation step (E-step) calculates posteriors $p(\boldsymbol{z}|\boldsymbol{x}^{(i)};\theta^{(t)})$ with the current $\theta^{(t)}$ for $\boldsymbol{x}^{(i)} \in \mathcal{X}$ ($\mathcal{X} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(N)}\}$ is a set of examples) and constructs $\mathcal{Q}(\theta|\theta^{(t)})$ as the expected log-likelihood of the complete data under $p(\boldsymbol{z}|\boldsymbol{x}^{(i)};\theta^{(t)})$:

$$\mathcal{Q}(\theta|\theta^{(t)}) = \sum_i \mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x}^{(i)};\theta^{(t)})} \log p(\boldsymbol{x}^{(i)}, \boldsymbol{z}; \theta)\,. \tag{2.13}$$

- Maximization step (M-step) maximizes $\mathcal{Q}(\theta|\theta^{(t)})$ with $\theta^{(t)}$ fixed:

$$\theta^{(t+1)} = \arg\max_{\theta} \mathcal{Q}(\theta|\theta^{(t)})\,. \tag{2.14}$$

The EM algorithm can be derived in terms of optimizing a lower bound on the model log-likelihood (Neal and Hinton, 1998; Minka, 1998). To show this, we first define arbitrary probability distributions over the latent variables as $q_i(\boldsymbol{z})$ ($1 \leq i \leq N$) and derive the lower bound as:

$$\begin{aligned}
\mathcal{L}(\theta) &= \sum_i \log \sum_{\boldsymbol{z}^{(i)}} p(\boldsymbol{x}^{(i)}, \boldsymbol{z}^{(i)}; \theta) && \text{(Model log-likelihood)} \\
&= \sum_i \log \sum_{\boldsymbol{z}^{(i)}} q_i(\boldsymbol{z}^{(i)}) \frac{p(\boldsymbol{x}^{(i)}, \boldsymbol{z}^{(i)}; \theta)}{q_i(\boldsymbol{z}^{(i)})} && \text{(Rewriting in } q_i(\boldsymbol{z})\text{)} \\
&\geq \sum_i \sum_{\boldsymbol{z}^{(i)}} q_i(\boldsymbol{z}^{(i)}) \log \frac{p(\boldsymbol{x}^{(i)}, \boldsymbol{z}^{(i)}; \theta)}{q_i(\boldsymbol{z}^{(i)})} && \text{(Applying the Jensen's inequality)} \\
&= \sum_i \sum_{\boldsymbol{z}^{(i)}} q_i(\boldsymbol{z}^{(i)}) \log p(\boldsymbol{x}^{(i)}, \boldsymbol{z}^{(i)}; \theta) - q_i(\boldsymbol{z}^{(i)}) \log q_i(\boldsymbol{z}^{(i)}) && \text{(Distributing log)} \\
&= \sum_i \mathcal{B}(\theta; \boldsymbol{x}^{(i)})\,. && \text{(Evidence lower bound } \mathcal{B}\text{)}
\end{aligned}$$

An important observation from the above derivation is that $\mathcal{B}(\theta;\boldsymbol{x})$ is true for any $q(\boldsymbol{z})$. Among those valid probability distributions $q(\boldsymbol{z})$, the E-step finds a $q(\boldsymbol{z})$ that leads to a tight bound $\mathcal{B}(\theta;\boldsymbol{x})$ at the current $\theta^{(t)}$. One way to show this is to treat $\mathcal{B}(\theta;\boldsymbol{x})$ as a function of $q(\boldsymbol{z})$, and solve the maximization of $\mathcal{B}(\theta;\boldsymbol{x})$ by introducing a Lagrange multiplier that enforces the constraint $\sum_{\boldsymbol{z}} q(\boldsymbol{z}) = 1$. But a more intuitive explanation is given as follows by Minka (1998):

$$\mathcal{B}(\theta;\boldsymbol{x}) = \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \log p(\boldsymbol{x}, \boldsymbol{z}; \theta) - q(\boldsymbol{z}) \log q(\boldsymbol{z}) \tag{2.15}$$

$$= \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \log p(\boldsymbol{z}|\boldsymbol{x}; \theta) p(\boldsymbol{x}; \theta) - q(\boldsymbol{z}) \log q(\boldsymbol{z}) \tag{2.16}$$

$$= -\sum_{\boldsymbol{z}} q(\boldsymbol{z}) \log \frac{q(\boldsymbol{z})}{p(\boldsymbol{z}|\boldsymbol{x}; \theta)} + \log p(\boldsymbol{x}; \theta) \tag{2.17}$$

$$= -D_{\mathrm{KL}}(q(\boldsymbol{z}) || p(\boldsymbol{z}|\boldsymbol{x}; \theta)) + \mathcal{L}(\theta; \boldsymbol{x})\,, \tag{2.18}$$

where $D_{\mathrm{KL}}(q(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{x};\theta))$ calculates the Kullback-Leibler (KL) distance between a proposed posterior distribution $q(\boldsymbol{z})$ and the true posterior distribution $p(\boldsymbol{z}|\boldsymbol{x};\theta)$. It equivalently measures the difference between the lower bound $\mathcal{B}(\theta;\boldsymbol{x})$ and the model log-likelihood $\mathcal{L}(\theta;\boldsymbol{x})$. Since $\mathcal{L}(\theta;\boldsymbol{x})$ is constant at a given $\theta$ and $D_{\mathrm{KL}}(\cdot)$ is non-negative, the maximum of the lower bound can be obtained when $D_{\mathrm{KL}}(q(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{x};\theta))$ is zero, i.e., by setting $q(\boldsymbol{z}) = p(\boldsymbol{z}|\boldsymbol{x};\theta)$ ($\theta = \theta^{(t)}$). Therefore, the E-step yields a lower bound equal to $\mathcal{L}(\theta^{(t)})$.

Given the calculated posteriors from the E-step, the M-step maximizes the lower bound with respect to $\theta$:

$$\mathcal{B}(\theta;\boldsymbol{x}) = \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \log p(\boldsymbol{x},\boldsymbol{z};\theta) - q(\boldsymbol{z}) \log q(\boldsymbol{z}) \tag{2.19}$$

$$= \mathbb{E}_{q(\boldsymbol{z})}[\log p(\boldsymbol{x},\boldsymbol{z};\theta)] - H(q(\boldsymbol{z}))\,, \tag{2.20}$$

where $H(q(\boldsymbol{z}))$ indicates the entropy of $q(\boldsymbol{z})$. Since $H(q(\boldsymbol{z}))$ is independent of $\theta$, maximizing the lower bound $\mathcal{B}(\theta;\boldsymbol{x})$ is equivalent to maximizing the first expectation term, and the optimal parameters are obtained by optimizing $\mathcal{B}(\theta;\boldsymbol{x})$ over all $\boldsymbol{x}$:

$$\theta^{(t+1)} = \arg\max_{\theta} \sum_{i} \mathcal{B}(\theta;\boldsymbol{x}^{(i)}) = \arg\max_{\theta} \sum_{i} \mathbb{E}_{q(\boldsymbol{z})}[\log p(\boldsymbol{x}^{(i)},\boldsymbol{z};\theta)]\,. \tag{2.21}$$

Substituting $q(\boldsymbol{z})$ in the above derivation by $p(\boldsymbol{z}|\boldsymbol{x};\theta^{(t)})$ yields the expected complete-data likelihood under $p(\boldsymbol{z}|\boldsymbol{x};\theta^{(t)})$, i.e., the objective in the M-step. For models such as PCFG, the optimal parameters in the M-step can be estimated analytically, but for many other discrete latent variable models, especially those that are implemented with neural networks, there are no closed-form solutions. Instead, numerical optimization techniques (e.g., gradient ascent) are used to find an approximate estimate:

$$\theta^{(t+1)} = \theta^{(t)} + \eta \cdot \nabla_{\theta} Q(\theta;\theta^{(t)})\,, \tag{2.22}$$

where $\eta > 0$ is the learning rate and $\nabla_{\theta} Q(\theta;\theta^{(t)})$ is obtained by differentiating it with respect to $\theta$. While $Q(\theta^{(t+1)};\theta^{(t)}) > Q(\theta^{(t)};\theta^{(t)})$ holds, gradient-based optimization performs local maximization and does not necessarily give rise to the optimal estimate of $\theta^{(t+1)}$. This leads to a new variant of the EM algorithm that is referred to as the *generalized expectation-maximization* algorithm (Dempster et al., 1977; Neal and Hinton, 1998). In fact, assume $\mathcal{L}(\theta)$ is differentiable with respect to $\theta$, depending on the properties of $\boldsymbol{z}$, in some cases we can directly optimize $\mathcal{L}(\theta)$ via gradient-based methods. For example, when $\boldsymbol{z}$ is factorizable or $|\mathcal{Z}|$ ($\boldsymbol{z} \in \mathcal{Z}$) is small, and the marginalization

over $\boldsymbol{z}$ in calculating $\mathcal{L}(\theta)$ can be tractably computed (e.g., through the dynamic programming as with PCFG). It can be shown that the gradient formulation given by $\nabla_\theta \mathcal{L}(\theta)$ is the same as that given by $\nabla_\theta Q(\theta;\theta^{(t)})$.

#### 2.3.1.2 Variational Inference

We have discussed the optimization of $\mathcal{L}(\theta)$ in cases where the posterior inference $p(\boldsymbol{z}|\boldsymbol{x};\theta)$ is tractable, or equivalently, the marginal likelihood $p(\boldsymbol{x};\theta)$ is tractable to estimate, but in many cases the marginalization over $\boldsymbol{z}$ is intractable, so is the posterior. Variational inference tackles this challenge by using a tractable posterior to approximate the exact posterior. Defining the set of tractable probability distributions as $\mathcal{Q}$ (aka. *variational family*), a tractable posterior $q(\boldsymbol{z};\xi) \in \mathcal{Q}$ is parameterized by $\xi$ and is constructed to approximate the true posterior for each individual example, i.e., the set of parameters $\xi$ are example-dependent. The learning problem in variational inference is to minimize the distance between the variational posterior and the true posterior. As with variational inference, this distance is measured via the KL divergence between the two distributions:

$$q^*(\boldsymbol{z};\xi) = \arg\min_{q(\boldsymbol{z};\xi)\in\mathcal{Q}} D_{\mathrm{KL}}(q(\boldsymbol{z};\xi)||p(\boldsymbol{z}|\boldsymbol{x};\theta)) \,. \tag{2.23}$$

The KL distance has been used in the formulation of the evidence lower bound (ELBO) on the marginal likelihood, i.e., $\mathcal{B}(\theta;\boldsymbol{x})$ in Equation 2.18. By substituting the arbitrary probability distribution $q(\boldsymbol{z})$ in $\mathcal{B}(\theta;\boldsymbol{x})$ with $q(\boldsymbol{z};\xi)$, we can rewrite the KL distance in terms of ELBO as:

$$D_{\mathrm{KL}}(q(\boldsymbol{z};\xi)||p(\boldsymbol{z}|\boldsymbol{x};\theta)) = -\mathcal{B}(\theta,\xi;\boldsymbol{x}) + \mathcal{L}(\theta;\boldsymbol{x}) \,. \tag{2.24}$$

Since $\mathcal{L}(\theta;\boldsymbol{x})$ does not depend on $q(\boldsymbol{z};\xi)$ in Equation 2.24, minimizing the KL divergence between $q(\boldsymbol{z};\xi)$ and $p(\boldsymbol{z}|\boldsymbol{x};\theta)$ amounts to maximizing the evidence lower bound (Jordan et al., 1999; Wainwright and Jordan, 2008). To tractably compute ELBO, which requires estimating expectations with respect to $q$ (see Equation 2.19), many variational-inference models have resorted to the *mean-field variational family*. In this family, each latent variable is assumed to be independent of the others, and $q$ is decomposed as $q(\boldsymbol{z};\xi) = \prod_i q(\boldsymbol{z}_i;\xi)$. This is known as *mean field approximation*, which has its roots in mean field theory in physics (Parisi and Shankar, 1988) and gives rise to the *mean-field variational inference* (Opper and Saad, 2001). A natural extension to this mean field is to introduce dependencies between the latent variables, leading to

**Algorithm 1:** Variational Expectation Maximization with Coordinate Ascent

**Input :** $\mathcal{X} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(N)}\}$.

**Return:** $\theta$ and $\xi^{(i)}$ with $i \in [1, N]$.

$\theta$, $\xi^{(i)} \leftarrow$ random initializations;

**while** $\mathcal{B}(\theta, \xi; \mathcal{X})$ *has not converged* **do**

- Variational E-step: fix current $\theta$ and maximize ELBO with respect to $\xi^{(i)}$ for each $\boldsymbol{x}^{(i)} (i \in [1, N])$ by using coordinate ascent variation inference:

$$\xi^{(i)} = \arg\max_{\xi^{(i)}} \mathcal{B}(\theta, \xi^{(i)}; \boldsymbol{x}^{(i)}) = \arg\min_{\xi^{(i)}} D_{\mathrm{KL}}(q(\boldsymbol{z}; \xi^{(i)}) || p(\boldsymbol{z} | \boldsymbol{x}^{(i)}; \theta))$$

- Variational M-step: fix current $\xi$ and maximize ELBO with respect to $\theta$:

$$\theta = \arg\max_{\theta} \sum_i \mathcal{B}(\theta, \xi^{(i)}; \boldsymbol{x}^{(i)}) = \arg\max_{\theta} \sum_i \mathbb{E}_{q(\boldsymbol{z}; \xi^{(i)})}[\log p(\boldsymbol{x}^{(i)}, \boldsymbol{z}; \theta)]$$

$\mathcal{B}(\theta, \xi; \mathcal{X}) = \sum_i \mathcal{B}(\theta, \xi^{(i)}; \boldsymbol{x}^{(i)})$

**end**

the structured mean field (Saul et al., 1996; Barber and Wiegerinck, 1998; Wiegerinck, 2000; Xing et al., 2002).

In mean-field variational inference, coordinate ascent has been a popular method for optimizing variational parameters (see Equation 2.23). Concretely, we iteratively optimize with respect to the variational distributions while keeping the others fixed until ELBO converges. With proper assumptions on the functional forms of the model distribution and the variational distribution (e.g., both are in the exponential family), a closed-form update for each coordinate can be derived (Hoffman et al., 2013; Blei et al., 2017). In general, the ELBO objective function is non-convex, so the coordinate-ascent variational-inference algorithm (CAVI) only guarantees to converge to a local minimum, and similarly to the EM algorithm, it can be sensitive to model initialization.

To maximize ELBO with respect to both the variational parameters $\xi$ and the model parameters $\theta$, we can construct the CAVI algorithm in such a way that it resembles the EM algorithm. Specifically, in each iteration, we first maximize ELBO with respect to $\xi^{(i)}$ for all examples $i \in [1, N]$ while keeping $\theta$ fixed, then we maximize ELBO with respect to $\theta$ while keeping all $\xi^{(i)}$ fixed (see Algorithm 1). This iterative learning paradigm is known as *variational expectation maximization* (Neal and Hinton, 1998). Differently from the EM algorithm, which always obtains a tight bound by choosing the true posterior as the variational posterior in the E-step, variational inference is more

flexible: it aims to find parameters $\xi$ that yield as tight a bound as possible, then it uses the approximating posterior rather than the exact posterior to calculate the expected complete-data log-likelihood, which is further maximized in the M-step.

Again, as with most neural-based models, it is impossible to derive analytical solutions in the variational M-step. In this case, gradient-based optimization is used to update $\theta$ locally. Regarding the variational E-step, while there is usually a closed-form update for each variational parameter, coordinate ascent is inefficient for large data sets because each variational E-step requires optimizing a separate set of variational parameters for every training example. A possible solution is to perform the variational E-step (and M-step) over a mini-batch of training examples. This has led to the development of *stochastic variational inference* (SVI; Hoffman et al. (2013)). Apart from mini-batch training, SVI updates both $\xi$ in the variational E-step and $\theta$ in the variational M-step using gradient-based methods (i.e., natural gradients (Amari, 1998)). However, due to the reliance on stochastic optimization that requires subsampling the training data (Robbins and Monro, 1951), SVI suffers from noisy gradient estimates. While increasing the batch size could mitigate the issue, it leads to higher computational costs. To tackle the variance issue, many other strategies have been explored such as non-uniformly sampling training examples (Gopalan et al., 2012; Zhao and Zhang, 2015; Csiba and Richtárik, 2018) and using a control variate (Paisley et al., 2012; Ranganath et al., 2014).

#### 2.3.1.3 Amortized Variational Inference

While stochastic variational inference scales well as the dataset size increases, per-example posterior optimization in the E-step is inherently expensive, especially when there are no closed-form solutions and numerical optimization has to be used. Moreover, since each individual example has its own variational parameters, variational inference is unable to reuse previous inferences, so when new examples come, it will have to estimate all the parameters all over again. A possible strategy for resolving these issues is amortized inference (Stuhlmüller et al., 2013; Gershman and Goodman, 2014). Amortized inference refers to the philosophy of reusing past experience (Michie, 1968) to construct inferences for newer examples; it has been implemented for variational inference in deep learning models (Ritchie et al., 2016). The key idea of *amortized variational inference* is to employ a parameterized function $f(\cdot;\phi)$ to predict variational parameters for each example: $\xi = f(\boldsymbol{x};\phi)$, where $f(\cdot)$ is implemented as a neural network and $\phi$ is shared across examples (Ritchie et al., 2016).

Intuitively, $f(\cdot;\phi)$ outputs similar variational parameters for similar examples, thus the reuse of previous inferences becomes possible. During learning, in the variational E-step, amortized variational inference maximizes ELBO with respect to the shared variational parameters $\phi$, so the cost of variational posterior inference is amortized across examples through the encoding function $f(\cdot;\phi)$:

$$\phi^* = \arg\max_{\phi} \sum_i \mathcal{B}(\theta, f(\boldsymbol{x}^{(i)};\phi);\boldsymbol{x}^{(i)}) . \tag{2.25}$$

#### 2.3.1.4 Variational Autoencoder

The fact that amortized variational inference is amenable to gradient-based optimization connects it with deep learning models, which follow the philosophy of end-to-end learning with gradient-based optimization. Combining variational inference with neural models has led to the variational autoencoder framework (VAE; Kingma and Welling (2014)). A variational autoencoder consists of a neural encoder modeling the variational inference process and a neural decoder describing the data generating process. This will become more clear after we substitute the arbitrary posterior $q(\boldsymbol{z})$ in the original ELBO with the amortized variational posterior $q(\boldsymbol{z}|\boldsymbol{x};\phi) \triangleq q(\boldsymbol{z};f(\boldsymbol{x};\phi))$ (see Equation 2.15):

$$\mathcal{B}(\theta,\phi;\boldsymbol{x}) = \sum_{\boldsymbol{z}} q(\boldsymbol{z}|\boldsymbol{x};\phi)\log p(\boldsymbol{x},\boldsymbol{z};\theta) - q(\boldsymbol{z}|\boldsymbol{x};\phi)\log q(\boldsymbol{z}|\boldsymbol{x};\phi) \tag{2.26}$$

$$= \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x};\phi)}[\log p(\boldsymbol{x}|\boldsymbol{z};\theta)] - D_{\mathrm{KL}}(q(\boldsymbol{z}|\boldsymbol{x};\phi)||p(\boldsymbol{z};\theta)) . \tag{2.27}$$

In the above derivation, $q(\boldsymbol{z}|\boldsymbol{x};\phi)$ corresponds to a probabilistic encoder that infers a distribution over the latent $\boldsymbol{z}$ from each given example $\boldsymbol{x}$, and $p(\boldsymbol{x}|\boldsymbol{z};\theta)$ corresponds to a probabilistic decoder that produces a distribution over the example space for each given $\boldsymbol{z}$. To establish connections with the standard autoencoders, we can interpret the expectation term as the expected negative reconstruction loss under a variational posterior and the KL term as a regularizer that pushes the variational posterior toward the true posterior (see Equation 2.27).

#### 2.3.1.5 Gradient Estimation

As with the standard VAEs, the inference model and the generative model are jointly optimized by maximizing ELBO with respect to $\phi$ and $\theta$ via gradient-based methods (Kingma and Welling, 2014). We first present an estimate of the gradients of

ELBO with respect to $\theta$:

$$\nabla_\theta \mathcal{B}(\theta,\phi;\boldsymbol{x}) = \nabla_\theta \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x};\phi)}[\log p(\boldsymbol{x},\boldsymbol{z};\theta)] \tag{2.28}$$

$$= \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x};\phi)}[\nabla_\theta \log p(\boldsymbol{x},\boldsymbol{z};\theta)]\,, \tag{2.29}$$

where we have used Leibniz's rule to interchange the gradient operator and the expectation operator. Though the variational posterior is constructed to be tractable, the sum over $\boldsymbol{z}$ can still be intractable. In this case, the Monte Carlo method is used to calculate approximate gradients:

$$\nabla_\theta \mathcal{B}(\theta,\phi;\boldsymbol{x}) \approx \frac{1}{K}\sum_{i=1}^{K} \nabla_\theta \log p(\boldsymbol{x},\boldsymbol{z}^{(i)};\theta) \text{ with } \boldsymbol{z}^{(i)} \sim q(\boldsymbol{z}|\boldsymbol{x};\phi)\,. \tag{2.30}$$

Estimating the gradients of ELBO with respect to $\phi$ is nontrivial because the resulting gradient expression may not be written as an expectation with respect to a probability density; accordingly, the Monte Carlo estimator can no longer be used. To tackle this challenge, there have been several techniques developed, among which we elaborate on the score function estimator (Glynn, 1987), the pathwise gradient estimator (Kingma and Welling, 2014; Rezende et al., 2014), and the gumbel-softmax estimator (Jang et al., 2017; Maddison et al., 2017).

**Score Function Estimator.** The key idea of the *score function estimator* (Glynn, 1987) is to use the identity $\nabla q = q\nabla \log q$, which holds for any valid distribution $q$:

$$\nabla_\phi \mathcal{B}(\theta,\phi;\boldsymbol{x}) = \nabla_\phi \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x};\phi)}\left[\log \frac{p(\boldsymbol{x},\boldsymbol{z};\theta)}{q(\boldsymbol{z}|\boldsymbol{x};\phi)}\right] \tag{2.31}$$

$$= \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x};\phi)}\left[\log \frac{p(\boldsymbol{x},\boldsymbol{z};\theta)}{q(\boldsymbol{z}|\boldsymbol{x};\phi)} \nabla_\phi \log q(\boldsymbol{z}|\boldsymbol{x};\phi)\right]. \tag{2.32}$$

The resulting gradient formulation is also known as REINFORCE in the context of reinforcement learning, and $\log \frac{p(\boldsymbol{x},\boldsymbol{z};\theta)}{q(\boldsymbol{z}|\boldsymbol{x};\phi)}$ is referred to as the learning reward (Williams, 1992). Again, the Monte Carlo method is used to calculate approximate gradients when the sum over $\boldsymbol{z}$ is intractable:

$$\nabla_\phi \mathcal{B}(\theta,\phi;\boldsymbol{x}) \approx \frac{1}{K}\sum_{i=1}^{K} \log \frac{p(\boldsymbol{x},\boldsymbol{z};\theta)}{q(\boldsymbol{z}|\boldsymbol{x};\phi)} \nabla_\phi \log q(\boldsymbol{z}|\boldsymbol{x};\phi) \text{ with } \boldsymbol{z}^{(i)} \sim q(\boldsymbol{z}|\boldsymbol{x};\phi)\,. \tag{2.33}$$

The score function estimator is applicable to both discrete and continuous variational distributions. Though it is unbiased, it is prone to high variance, primarily because the multiplier $\log \frac{p(\boldsymbol{x},\boldsymbol{z};\theta)}{q(\boldsymbol{z}|\boldsymbol{x};\phi)}$ of the gradient $\nabla_\phi \log q(\boldsymbol{z}|\boldsymbol{x};\phi)$ is unbounded.

**Pathwise Gradient Estimator.** A low-variance estimator is the *pathwise gradient estimator*; it is also known as the reparameterization trick in the VAE literature (Kingma and Welling, 2014; Rezende et al., 2014). The key idea of the pathwise gradient estimator is to formalize the variational posterior in terms of a base probability density so that, instead of directly sampling from the variational posterior, we can sample from the base distribution and deterministically transform the sample:

$$\boldsymbol{z} = g(\varepsilon;\xi) \text{ with } \varepsilon \sim p(\varepsilon)\,, \tag{2.34}$$

where $p(\varepsilon)$ is a base distribution simpler than the variational posterior, and $g(\varepsilon,\xi)$ is a deterministic transform function differentiable with respect to $\xi$.

Below we present an example with the multivariate normal distribution as the base distribution since we will use it in Chapter 3:

$$\begin{aligned}
p(\varepsilon) &= \mathcal{N}(\varepsilon;\mathbf{0},\mathbf{I})\,, && \text{(Base distribution)}\\
q(\boldsymbol{z}|\boldsymbol{x};\phi) &= \mathcal{N}(\boldsymbol{z};\mu,\sigma^2) \text{ with } \xi = (\mu,\sigma^2) = f(\boldsymbol{x};\phi)\,, && \text{(Variational posterior)}\\
\boldsymbol{z} \sim q(\boldsymbol{z}|\boldsymbol{x};\phi) &\Leftrightarrow \boldsymbol{z} = g(\varepsilon;\xi) = \mu + \sigma\cdot\varepsilon \text{ with } \varepsilon \sim \mathcal{N}(\varepsilon;\mathbf{0},\mathbf{I})\,. && \text{(Reparameterization)}
\end{aligned}$$

The reparameterization trick separates the variational parameters $\phi$ from the discrete sampling path. We can instead estimate the expectation in the lower bound under the base distribution that does not involve any learnable parameters:

$$\mathcal{B}(\theta,\phi;\boldsymbol{x}) = \mathbb{E}_{p(\varepsilon)}\left[\log \frac{p(\boldsymbol{x},g(\varepsilon;\xi);\theta)}{q(g(\varepsilon;\xi)|\boldsymbol{x};\phi)}\right]. \tag{2.35}$$

To estimate the gradients, we use the Monte Carlo method to obtain a low-variance estimate of the lower bound $\tilde{\mathcal{B}}(\theta,\phi;\boldsymbol{x})$ and differentiate the empirical lower bound with respect to $\theta$ and $\phi$: $\nabla_{\theta,\phi}\tilde{\mathcal{B}}(\theta,\phi;\boldsymbol{x})$.

**Gumbel-Softmax Estimator.** The *Gumbel-Softmax estimator* extends the reparameterization trick that works with continuous distributions to discrete distributions, e.g, the categorical distribution. Denoting $\boldsymbol{z}$ as a categorical variable with $C$ categories, $\boldsymbol{z}$ is represented as a one-hot vector and takes on each category with a probability $\pi_i$ ($\sum_{i=1}^{C}\pi_i = 1$). The Gumbel-Softmax estimator (Jang et al., 2017; Maddison et al., 2017) achieves differentiable sampling by using the Gumbel-Max trick, i.e., applying the reparameterization trick to discrete variables (Maddison et al., 2014). Gumbel-Max uses the Gumbel distribution as the base distribution. Given a vector of independent

random Gumbel noises $\varepsilon(|\varepsilon| = C)$,[3] Gumbel-Max transforms it by using Perturb-and-MAP (Papandreou and Yuille, 2011):

$$\boldsymbol{z} = \text{one\_hot}\left(\arg\max_i \left(\log \pi_i + \varepsilon_i\right)\right) \text{ with } \varepsilon \sim \text{Gumbel}(\varepsilon; \mathbf{0}, \mathbf{I})\,. \tag{2.36}$$

But the arg max operator precludes gradient backpropagation; instead, the Gumbel-Softmax estimator uses a Gumbel-Softmax sample as a differentiable proxy of the discrete one-hot sample $\boldsymbol{z}$:

$$\tilde{\boldsymbol{z}}_i = \exp((\log \pi_i + \varepsilon_i)/\tau)/Z \text{ with } Z = \sum_i \exp((\log \pi_i + \varepsilon_i)/\tau)\,, \tag{2.37}$$

where $\tau$ is a positive temperature parameter controlling the closeness between the differential approximate and the corresponding discrete sample. But, at test time, the maximum a posteriori (MAP) estimate is usually required, and we will have to transform the continuous approximate back into a discrete sample. To bridge the gap between training and inference, the Gumbel-Softmax estimator is further augmented with the Straight-Through estimator (ST; Bengio et al. (2013)). The key idea of the ST estimator is to use the discrete (i.e., arg max) version of continuous relaxation $\tilde{\boldsymbol{z}}$ in the forward pass and backpropagate through $\tilde{\boldsymbol{z}}$ in the backward pass.

#### 2.3.1.6 Semi-supervised Learning

We have so far focused on learning structure-induction models in a fully-unsupervised manner. In cases where there are downstream tasks that benefit from structured representations, we could learn task-specific structure-induction models. Formally, suppose that, in the supervised setting, each observation $\boldsymbol{x}$ is associated with a target $\boldsymbol{y}$ (e.g., $\boldsymbol{y}$ can be a sentiment label if $\boldsymbol{x}$ is a sentence), a traditional supervised model would predict $\boldsymbol{y}$ directly from $\boldsymbol{x}$: $p(\boldsymbol{y}|\boldsymbol{x};\lambda)$. Rather than conditioning on $\boldsymbol{x}$ alone, we would like to predict the target $\boldsymbol{y}$ conditioning on both $\boldsymbol{x}$ and its structured representations. Since the structured representations are unobserved, we treat them as latent variables and indicate them by $\boldsymbol{z}$. Using the latent variable $\boldsymbol{z}$, we further reformulate the traditional conditional model as:

$$p(\boldsymbol{y}|\boldsymbol{x};\lambda) = \sum_{\boldsymbol{z}} p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{z};\lambda)p(\boldsymbol{z}|\boldsymbol{x};\lambda)\,. \tag{2.38}$$

[3] In practice, Gumbel variables are sampled via inverse transform sampling: $\varepsilon = -\log(-\log(\boldsymbol{u}))$ with $\boldsymbol{u} \sim \text{Uniform}(\mathbf{0}, \mathbf{I})$.

Compared with the traditional conditional model, the latent-variable formulation relies on an additional posterior distribution $p(\boldsymbol{z}|\boldsymbol{x};\lambda)$ to model latent structures. Explicitly modeling hidden structures has been shown to be useful for downstream supervised tasks (Chiang, 2005; Yogatama et al., 2017; Deng et al., 2018), especially those that require symbolic reasoning (Mao et al., 2019; Havrylov et al., 2019). But in our case, we are interested in uncovering the structures underlying observations $\boldsymbol{x}$, i.e., learning the inference model $p(\boldsymbol{z}|\boldsymbol{x};\lambda)$ from downstream tasks. Given that structured representations help with downstream tasks, we hypothesize that, in return, supervision from downstream tasks will help with learning structure-induction models. To favor unsupervised learning, we assume that $\boldsymbol{y}$ is easy to obtain, e.g., $\boldsymbol{y}$ can be images that are aligned with the given text $\boldsymbol{x}$ (see Chapter 3).

Learning the latent-variable conditional model can be formulated as maximizing the conditional log-likelihood of $\boldsymbol{y}$ given $\boldsymbol{x}$:

$$\mathcal{L}(\lambda) = \sum_i \log p(\boldsymbol{y}^{(i)}|\boldsymbol{x}^{(i)};\lambda) = \sum_i \log \sum_{\boldsymbol{z}} p(\boldsymbol{y}^{(i)}|\boldsymbol{x}^{(i)},\boldsymbol{z};\lambda) p(\boldsymbol{z}|\boldsymbol{x}^{(i)};\lambda)\,, \tag{2.39}$$

which involves learning the posterior distribution $p(\boldsymbol{z}|\boldsymbol{x};\lambda)$ over parameters $\lambda$. Since $p(\boldsymbol{z}|\boldsymbol{x};\lambda)$ can be derived from $p(\boldsymbol{x},\boldsymbol{z};\lambda)$, we instead replace it with a separate model $p(\boldsymbol{z}|\boldsymbol{x},\theta)$ and learn another unsupervised model $p(\boldsymbol{x};\theta) = \sum_z p(\boldsymbol{x},\boldsymbol{z};\theta)$. In particular, we are interested in a joint learning paradigm that combines supervised learning and unsupervised learning:

$$\mathcal{L}(\theta,\lambda) = \sum_i \log p(\boldsymbol{x}^{(i)};\theta) + \sum_i \log p(\boldsymbol{y}^{(i)}|\boldsymbol{x}^{(i)};\theta,\lambda)\,. \tag{2.40}$$

We refer to the joint learning paradigm as semi-supervised learning. To optimize the full model, we resort to the learning techniques discussed in Sections 2.3.1.1–2.3.1.5.

### 2.3.2 Structure-Aware Models

Differently from neural-symbolic models, which explicitly model latent structures via discrete latent variables, structure-aware models encode discrete structures via computation graphs, which can be specified by given structures (e.g., trees) or dynamically constructed via an algorithm (e.g., easy-first parsing (Goldberg and Elhadad, 2010)). The challenges with structure-aware models lie in representation and inference.

The representation challenge means that we have to design appropriate model architectures to encode desired structural patterns (e.g., phrase boundaries). Since these structural patterns are specific to a structure-induction task (e.g., phrase boundaries

can be used to induce syntactic structures), the model architectures are in general task-specific, but there are generic architectures that are capable of encoding discrete structures, including the self-attention mechanism for complete graphs (Vaswani et al., 2017) and graph neural networks for arbitrary graph-like structures (Kipf and Welling, 2017). Apart from encoding generic structures like graphs, in many cases, we are interested in encoding more restricted structures in neural models. A possible way to do so is to place proper restrictions on computations. This has led to more specialized model architectures, e.g., the recurrent computations of recurrent neural networks encode linear-chain structures (Elman, 1990) and the recursive computations of TreeLSTM encode tree structures (Tai et al., 2015). These structure-aware models generally rely on given structures, which can be implicitly defined (e.g., complete graphs), explicitly specified (e.g., parse trees), and dynamically constructed (Goldberg and Elhadad, 2010). Another way of encoding structures is to perform structure-aware computations without requiring specific structures. It is usually operationalized via customized attention mechanisms that are capable of capturing general hierarchies in images and text, i.e., these specialized computational mechanisms allow for semantically similar regions/spans to merge implicitly and recursively (Geng et al., 2023).

The inference challenge means that we need to tailor an inference algorithm to translate computations and representations of a structure-aware model into target structures (e.g., finding a set of phrase boundaries that best describe the syntactic structure of a sentence). An inference algorithm generally consists of two components: enumerating and ranking all possible structures. These structures may have been specified by the model (e.g., graph structures underlying graph neural networks), or they have to be interpreted via an algorithm (e.g., the CKY algorithm (Cocke, 1969) on its own produces tree-like structures). To score structures for disambiguation, an inference algorithm may directly use the model's computations (e.g., likelihoods of relations for a given object pair in scene graph induction) or it has to define a measure by using the intermediate representations produced by the model (e.g., measuring the distance between two words as the cosine similarity between their vector representations).

Since we are interested in unsupervised approaches to structure induction, we will focus on model architectures that are not only capable of encoding structural patterns but also allow for unsupervised learning. Specifically, we primarily use Transformer models (Vaswani et al., 2017) to demonstrate the idea of structure-aware structure-induction models since they have been widely used for self-supervised (i.e., unsupervised) learning in the language and vision community (Devlin et al., 2019; Radford

et al., 2018; Caron et al., 2021; He et al., 2022; Zellers et al., 2022).

At the core of Transformer is the self-attention mechanism. It produces attention scores that measure dependencies between the subparts of an input (e.g., words in a sentence and objects in an image); the attention scores are further used to compute contextualized subpart representations. Though the attention mechanism implicitly defines a complete graph of subparts, it encodes no preferences for a particular structural formalism. But, with a carefully designed inference algorithm, desired structured representations can be extracted from internal representations of Transformer models such as attention scores and contextualized subpart representations.

As a first example, in the area of Transformer-based language modeling, it has been shown that syntactic structures can be induced from pre-trained language models (Kim et al., 2020; Wu et al., 2020). Take dependency tree induction, two popular paradigms for inducing dependency structures of sentences have been explored. They are similar in that they both rely on a dynamic programming approach to produce tree-like structures but are distinguished from each other in their ways of scoring trees. Scoring dependency trees relies on a metric to quantify the dependencies between words. In doing so, one paradigm uses the fact that self-attention scores can be seen as a measure of the dependencies between words (Htut et al., 2019), while the other resorts to the similarities between contextualized word representations (Wu et al., 2020).

Transformer-based models are an example of generic model architectures for structure induction. In the context of unsupervised parsing, more sophisticated model architectures have also been explored. For example, ordered neurons LSTM (ON-LSTM) implements a specialized gating mechanism to model hierarchical phrase structures. Essentially, the gating mechanism computes the likelihood that a split point forms a phrase boundary. At inference time, a top-down greedy algorithm is used to recover tree structures from intermediate phrase-boundary likelihoods (Shen et al., 2019).

In the area of vision-language learning, the generic self-attention mechanism has also been shown to be capable of capturing structural visual patterns. For example, in pre-training masked vision-language models (Lu et al., 2019; Su et al., 2020), the intermediate cross-modal attention has been shown to be predictive of visual relations (Li et al., 2020a). In Chapter 3, we will present a specialized computational model that is similar in spirit to masked language models (Devlin et al., 2019) but conditions on additional visual objects and is created with the goal of predicting relations for every pair of visual objects in a given image.

## 2.4 Multimodal Learning

Multimodal learning generally refers to machine learning problems that involve modeling connections among multiple modalities of data. We posit unsupervised structure induction in the multimodal setting and, specifically, the image-text learning setting, which relies on visual-textual groundings in the form of aligned image-text pairs. We first discuss the role of visual groundings in language understanding (Section 2.4.1), then show the potential of language abstraction for image understanding (Section 2.4.2), and finally review unsupervised curation of image-text alignments (Section 2.4.3).

### 2.4.1 Visual Groundings of Natural Language

Enabling machines to understand human language has long been a grand challenge. There have been different computational tools developed for representing, analyzing, and understanding natural language, from syntactic models to semantic models. Notably, recent breakthroughs in natural language understanding (NLU) have been made by distributional semantic models. Underlying these models is the distributional hypothesis, which states that words appearing in similar contexts are semantically similar (Harris, 1954), forms the basis of learning contextualized word representations from raw text corpora (Mikolov et al., 2013), and leads to the recent success of large language models (LLMs) for NLU (Devlin et al., 2019; Brown et al., 2020).

However, criticisms of these models have also arisen. For example, Bender and Koller (2020) argue that LLMs trained only on form (e.g., utterances) do not necessarily learn meanings (of words). One of the major reasons is that LLMs derive the representations of words (i.e., symbols) directly from their textual contexts, without considering the physical world (e.g., via visual and auditory perceptions) that grounds language (see *symbol grounding problem*; Harnad (1990)), which is contrary to the strong evidence that humans rely on visual information to learn language (O'Grady, 2005; Vigliocco et al., 2014). Moreover, linguistic data alone does not faithfully reflect the truth about the physical world, e.g., the relative frequency of an event, when described in text, does not necessarily match its relative likelihood in the world (Gordon and Van Durme, 2013), an issue that is usually referred to as the *reporting bias problem* (Van Durme, 2010). Consequently, learning from linguistic data alone will not plausibly lead to a system capable of understanding the world and human language.

To endow machine learning models with the ability of general language understanding, it is essential to learn them in an environment that demonstrates how lan-

guage is used (Wittgenstein, 1953), and practically, to learn from additional visual perceptions (Bruni et al., 2014; Bisk et al., 2020). To this end, many have been advocating visually grounded language learning, e.g., machine translation with image pivots (Elliott et al., 2016) and next-word prediction with image contexts (Ororbia et al., 2019). Apart from learning continuous sentence representations from visual groundings (Kiela et al., 2018; Bordes et al., 2019), visual groundings have been shown to be helpful for core NLP tasks such as unsupervised word segmentation (Kawakami et al., 2019) and syntax induction (Shi et al., 2019a). Further, with the large-scale image-text data (Sharma et al., 2018), multimodal pre-training progresses towards bridging the gap between vision and language (Lu et al., 2019; Zhou et al., 2020; Su et al., 2020).

### 2.4.2 Language Abstraction of Visual Concepts

Image understanding problems (e.g., object detection, image classification, and visual question answering) generally require machine learning models to capture visual concepts at a certain level, e.g., parts and objects. In the deep learning era, most neural models for these tasks are designed to follow the philosophy of end-to-end training, i.e., they transform pixel images into continuous feature vectors and decode a target directly from the feature vectors. While there is no explicit modeling of high-level abstraction of image contents, explainable visual concepts (e.g., objects) have been shown to emerge in unsupervised learning of object detectors (Le et al., 2012) and supervised scene classification (Zhou et al., 2015). These observations, to some extent, demonstrate the ability of neural models to understand complex scenes and account for their state-of-the-art performance, but meanwhile, the learned visual features have been shown to be correlated to some visual regularities such as textures and spatial closeness, e.g., visual objects “dog” and “couch” in the image “the dog is sitting on the couch” tend to have similar vector semantic representations because of the overlap between the corresponding object regions (Li et al., 2020b).

In contrast, words that refer to different concepts are likely to be separable in a learned word embedding space, partly because of the discrete nature of human language, e.g., words “dog” and “couch” are easy to distinguish from each other in the vector space. This observation suggests that language can be used as a tool to abstract visual concepts, giving rise to more symbolic visual representations. The argument is further bolstered by findings from cognitive science, which indicate that language experience causes more categorical visual perception (Lupyan et al., 2020).

Indeed, Li et al. (2020b) have shown that pre-trained word embeddings help learn more separable visual object representations. Abstracting visual concepts into textual concepts (e.g., translating images into scene graphs) has also improved image captioning and visual question answering (Wu et al., 2016). In low-level image understanding tasks such as object detection, a detector trained by additionally optimizing image-text alignment shows better performance than without using image-text alignment regularization (Kamath et al., 2021). Similarly, language supervision (e.g., semantic labels of pixels) via contrastive image-text learning enables few- and zero-shot semantic segmentation (Li et al., 2022a), and surprisingly contrastive image-text learning alone (i.e., without semantic mask annotations) results in meaningful semantic segmentation (Xu et al., 2022). However, all these models have been primarily focused on visual entities and learning their abstractions from language supervision. In addition to visual entities, visual relations (e.g., "behind" and "hold"), as another kind of visual concept, are essential for arranging visual entities and composing them into reasonable images, thus another interesting yet challenging problem would be inducing visual relations between objects from language supervision.

### 2.4.3 Curation of Image-Text Alignment

A prerequisite for training multimodal models is multimodal data and, specifically, the multimodal alignment that encodes informative connections among multiple modalities of data. Take aligned image-text pairs, images provide not only intuitive visual explanations of text but also extra contexts of text (i.e., what is unsaid in aligned text). Conversely, text offers high-level abstractions of complex visual concepts (e.g., the way "boy" and "ball" interact in an image can be abstracted as "play" in text). The methods for curating multimodal alignment vary depending on practical use scenarios. We will primarily focus on image-text alignment (i.e., via parallel image-text data) and briefly review the curation of representative image-text datasets.

The progress in computer vision has been a major drive for curating image-text data. Starting with image classification, the past decade has witnessed an increased interest in understanding interactions between vision and language in challenging yet practical multimodal tasks, including, *inter alia*, image captioning (Lin et al., 2014), visual question answering (Antol et al., 2015), visual (commonsense) reasoning (Zellers et al., 2019), and text-conditioned image generation (Ramesh et al., 2021). Depending on whether human labor is involved or not, the methods for curating image-text data

fall into two categories: manual curation and automatic curation.

Human-annotated resources have been used in vision-language tasks of varying difficulty, e.g., from visual perceptual tasks to visual cognitive tasks and higher-order visual reasoning tasks. For visual perceptual tasks, ImageNet (Deng et al., 2009), a hierarchical image database organized according to the hierarchy of WordNet (Fellbaum, 1998), has led to the breakthroughs of neural approaches to image classification (Krizhevsky et al., 2012) and is undoubtedly instrumental in the advances of deep learning. Progressing from perceptual tasks to cognitive tasks, MSCOCO (Lin et al., 2014; Chen et al., 2015), which is annotated with human-written captions, stimulates an interest in image captioning. Further, it is annotated with additional question-answer pairs for visual question answering (Antol et al., 2015), and has also become part of Visual Genome, an effort towards connecting structured visual concepts with language (Krishna et al., 2017). Advancing towards higher-order reasoning, NLVR (Suhr et al., 2019) and VCR (Zellers et al., 2019) provide testbeds for visually-grounded language reasoning and visual commonsense reasoning, respectively.

The fact that human annotations are prohibitively expensive to obtain on a large scale motivates automatic curation of image-text data. Synthetic image-text data is probably the easiest to create because it can be generated by following pre-defined procedures. Synthetic data is needed for two reasons: (1) to avoid difficult detection and labeling of natural image contents, e.g., AbstractScene assigns a set of semantically similar synthesized images to each human-written caption (Zitnick and Parikh, 2013); and (2) to function as a diagnostic dataset, e.g., CLEVR uses abstract images and executable functional language to facilitate the analysis of visual question-answering models (Johnson et al., 2017). Still, aligned natural image-text pairs are needed for training models for practical tasks, e.g., text-to-image generation. YFCC100M stands for the early large-scale automatically-curated dataset and contains around 100 million photos, but the associated tags are sparse and fail to fulfill the need for semantically diverse language descriptions (Thomee et al., 2016). Conceptual Captions demonstrates an automatic curation of large-scale high-quality image-text data from the Web (Sharma et al., 2018). Recent vision-language models such as CLIP (Radford et al., 2021) and DALLE (Ramesh et al., 2021) are learned from hundreds of millions of image-text pairs, which are also collected from the web. But aligned data is not always abundantly available for every modality pair (e.g., non-speech audio and text for auditory and textual modalities), posing an obstacle to working with data-scarce modality pairs. In Chapter 5, we will investigate this problem and present our solutions.

# Chapter 3

# Visually Grounded Grammar Induction

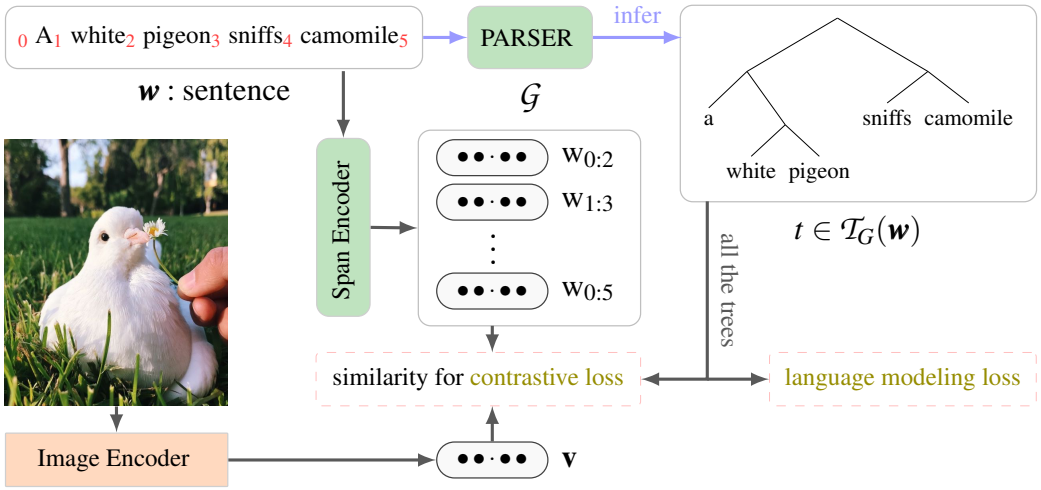



Figure 3.1: Visually-grounded Compound Probabilistic CFG (VC-PCFG). VC-PCFG is trained on image-text pairs by jointly optimizing a contrastive loss (negative pairs are not shown) and a language modeling loss. At test time, it infers syntactic structures directly from text, without access to the aligned images.

In this chapter, we investigate the task of inducing phrase-structure representations of text. While there have been different models developed for tackling the task, we are particularly interested in Context-Free Grammar (CFG). Given a CFG, every sentence admitted by it is associated with a derivation, i.e., phrase structure, which is composed of grammar rules from the CFG, and describes the process of recursively merging small constituent phrases into a larger constituent phrase. The task of finding latent phrase structures of language is known as grammar induction and has been formulated as an unsupervised learning task. As with grammar-based models, the goal of grammar induction is to learn a CFG from raw text, without using any annotated phrase

structures. As a fundamental problem in computational linguistics, grammar induction has been extensively studied prior to the deep learning era. Some of the early work on this problem is conducted with the purpose of examining the *Poverty of the Stimulus* argument (Chomsky, 1956) in the context of language acquisition (Gold, 1967; Pullum and Scholz, 2002; Zuidema, 2002), while many others are interested in evaluating the capability of CFG to represent English (Lari and Young, 1990; Carroll and Charniak, 1992; Clark, 2001; Klein and Manning, 2002). With the re-rising of neural networks, recent deep learning models have been applied to grammar induction and greatly advanced the area (Shen et al., 2018, 2019; Kim et al., 2019b,c). But these neural approaches to grammar induction follow conventional learning settings that were established decades ago. They have been generally limited to, for example, relying on text, without considering learning signals from other modalities.

In contrast, the crucial aspect of natural language learning is that it is grounded in perceptual experiences (Barsalou, 1999; Fincher-Kiefer, 2001; Bruni et al., 2014). As neural models have been approaching human performance on natural language processing benchmarks, grounded language learning has been argued to be an important next step towards better natural language understanding (Bisk et al., 2020; Bender and Koller, 2020). Existing work on grounded language learning has been primarily relying on visual groundings, largely because image-text pairs are abundantly available. It has been shown that visual groundings not only help with learning continuous sentence representations (Kiela et al., 2018; Bordes et al., 2019), but also benefit unsupervised word segmentation (Kawakami et al., 2019), machine translation (Elliott et al., 2016; Li et al., 2022b), and causal language modeling (Ororbia et al., 2019). In this chapter, we consider a more challenging problem: can visual groundings help us induce *syntactic structure*? We refer to this problem as *visually grounded grammar induction*.

The challenge with visually grounded grammar induction is: how to incorporate visual groundings into the learning of structure-induction models. Inspired by the work on task-dependent tree induction, where a latent-tree model induces tree-structured representations of text for downstream tasks such as sentiment classification and natural language inference (Yogatama et al., 2017; Choi et al., 2018; Maillard et al., 2019), we learn a latent-tree model from image-text pairs for a multimodal task. By analogy to the supervised natural language tasks in task-specific tree induction, where input text is grounded in the corresponding label (e.g., sentiment), the image in an image-text pair can be seen as the label of the aligned text. Predicting the image grounding of text can be formulated as finding an image that is most similar to the given text.

To learn such a similarity model, we adopt contrastive image-text learning, which has been shown to be effective in learning image and text representations from large-scale image-text data (Radford et al., 2021; Jia et al., 2021). Importantly, image and text representations are decoupled in contrastive image-text learning. This will be useful, at test time, for inferring syntactic structures without the reliance on aligned images.

Based on the framework of contrastive image-text pre-training, Shi et al. (2019a) has proposed a visually grounded neural syntax learner (VG-NSL) to tackle the task of visually grounded grammar induction. Specifically, they learn a parser from image-captioning data. The parser is optimized via REINFORCE, where the reward is computed by scoring the alignment of images and constituents. Though their model has demonstrated reasonable performance in latent tree induction, matching-based rewards can, as we will discuss further in the paper, make the parser focus only on more local and short constituents (e.g., 79.6% recall on NPs) and perform poorly on longer ones (e.g., 26.2% recall on VPs) (see Shi et al. (2019a)). While for the former it outperforms the text-only grammar induction methods, for the latter it substantially underachieves. This may not be surprising, as it is not guaranteed that every constituent of a sentence has its visual representation in the aligned image; the reward signals can be noisy and insufficient to capture all aspects of phrase-structure syntax. Consequently, Shi et al. (2019a) have to rely on language-specific inductive bias to obtain more informative reward signals. Another issue with VG-NSL is that the parser does not admit tractable estimation of the partition function and the necessary posterior probabilities – rewards have to be computed via point estimation, with learning potentially suffering from the high variance of gradient estimation.

To alleviate the first issue, we propose to complement the image-text alignment-based loss with a loss defined on unlabeled text (i.e., its log-likelihood). As re-confirmed with neural models in Shen et al. (2019) and Kim et al. (2019b), text itself can drive induction of rich syntactic knowledge, so additionally optimizing the parser on raw text can be beneficial and complementary to visually grounded learning. To resolve the second issue, we resort to an extension of Probabilistic CFG (PCFG) parsing model, compound PCFG (Kim et al., 2019b). It admits tractable estimation of the posteriors, needed in the alignment loss, with dynamical programming and leads to a fully-differentiable end-to-end visually grounded learning. More importantly, the PCFG parser lets us complement the alignment loss with a language modeling objective, leading to a framework of jointly maximizing the log-likelihood of text and optimizing image-text alignment.

Our key contributions can be summarized as follows: (1) we propose a fully-differentiable end-to-end visually grounded learning framework for grammar induction; (2) we additionally optimize a language modeling objective to complement visually grounded learning; and (3) we conduct experiments on MSCOCO (Lin et al., 2014) and observe that our model has a higher recall than VG-NSL for five out of the six most frequent constituent labels, e.g., it surpasses VG-NSL by 55% recall on VPs and by 48% recall on prepositional phrases (PPs). Comparing to a model trained purely via visually grounded learning, extending the loss with a language modeling objective improves the overall F1 from 50.5% to 59.4%.

## 3.1 Related Work

Grammar Induction has a long history in computational linguistics. Following observations that direct optimization of log-likelihood with the Expectation Maximization algorithm (Lari and Young, 1990) is not effective at producing effective grammars, a number of approaches have been developed, embodying various inductive biases or assumptions about the language structure and its relation to surface realizations (Klein and Manning, 2002; Smith and Eisner, 2005; Cohen and Smith, 2009; Spitkovsky et al., 2010; Zhao et al., 2018). The recent advances in the area have been brought by flexible neural models (Shen et al., 2018, 2019; Kim et al., 2019b,c; Drozdov et al., 2019). All these methods, with the exception of Shi et al. (2019a), rely solely on text.

Visually grounded learning is motivated by the observation that natural language is grounded in perceptual experiences (Steels, 1998; Barsalou, 1999; Fincher-Kiefer, 2001; Roy, 2002; Bisk et al., 2020). It has been shown effective in word representation learning (Bruni et al., 2014; Silberer and Lapata, 2014; Lazaridou et al., 2015) and sentence representation learning (Kiela et al., 2018; Bordes et al., 2019). All this work uses visual images as perceptual experience of language and exploits visual semantics derived from images to improve continuous vector representations of language. In contrast, we induce structured representations, discrete tree structures of language, by using visual groundings. We propose a model for the task within the contrastive learning framework. Learning involves estimating *concreteness* of spans, which generalizes word-level concreteness (Turney et al., 2011; Kiela et al., 2014).

In the vision and machine learning community, unsupervised induction of structured image representations (aka scene graphs or world models) has been receiving increasing attention (Eslami et al., 2016; Burgess et al., 2019; Kipf et al., 2020), how-

ever, they typically rely solely on visual signals. An interesting extension of our work would be to consider joint induction of structured representations of images and text while guiding learning by an alignment loss.

## 3.2 Background and Motivation

Our model relies on compound PCFG (Kim et al., 2019b) and generalizes the visually grounded grammar learning framework of Shi et al. (2019a). We will describe the relevant aspects of both frameworks in Sections 3.2.1-3.2.2, and then discuss their limitations (Section 3.2.3).

### 3.2.1 Compound PCFG

Compound PCFG extends Context-Free Grammar (CFG) and, to establish notation, we start by briefly introducing them. A CFG is defined as a 5-tuple $\mathcal{G} = (S, \mathcal{N}, \mathcal{P}, \Sigma, \mathcal{R})$ where $S$ is the start symbol, $\mathcal{N}$ is a finite set of nonterminals, $\mathcal{P}$ is a finite set of preterminals, $\Sigma$ is a finite set of terminals,[1] and $\mathcal{R}$ is a set of production rules in the Chomsky normal form:

$$S \rightarrow A \qquad A \in \mathcal{N}, \tag{3.1}$$

$$A \rightarrow BC \qquad A \in \mathcal{N}, \text{ and } B, C \in \mathcal{N} \cup \mathcal{P}, \tag{3.2}$$

$$T \rightarrow w \qquad T \in \mathcal{P}, w \in \Sigma. \tag{3.3}$$

PCFG extends CFG by associating each production rule $r \in \mathcal{R}$ with a non-negative scalar $\pi_r$ such that $\sum_{r:A\rightarrow\gamma} \pi_r = 1$, i.e., the probabilities of production rules with the same left-hand-side nonterminal sum to 1. The strong context-free assumption hinders PCFG and prevents them from being effective in the grammar induction context. Compound PCFG (C-PCFG) mitigates this issue by assuming that rule probabilities follow a compound probability distribution (Robbins, 1951):

$$\pi_r = g_r(\mathbf{z}; \theta) \quad \text{with} \quad \mathbf{z} \sim p(\mathbf{z}), \tag{3.4}$$

[1] Strictly, Context-Free Grammar does not distinguish nonterminals $\mathcal{N}$ (constituent labels) from preterminals $\mathcal{P}$ (part-of-speech tags). They are both treated as nonterminals. $\mathcal{N}, \mathcal{P}, \Sigma$ must satisfy $\mathcal{N} \cap \mathcal{P} = \emptyset$ and $(\mathcal{N} \cup \mathcal{P}) \cap \Sigma = \emptyset$.

where $p(\mathbf{z})$ is a prior distribution of the latent $\mathbf{z}$, and $g_r(\cdot;\theta)$ is parameterized by $\theta$ and yields a rule probability $\pi_r$. Depending on the rule type, $g_r(\cdot;\theta)$ has one of these forms:

$$\pi_{S\to A} = \frac{\exp(\mathbf{u}_A^T f_s([\mathbf{w}_S;\mathbf{z}]))}{\sum_{A'\in\mathcal{N}}\exp(\mathbf{u}_{A'}^T f_s([\mathbf{w}_S;\mathbf{z}]))}, \tag{3.5}$$

$$\pi_{A\to BC} = \frac{\exp(\mathbf{u}_{BC}^T[\mathbf{w}_A;\mathbf{z}])}{\sum_{B',C'\in\mathcal{N}\cup\mathcal{P}}\exp(\mathbf{u}_{B'C'}^T[\mathbf{w}_A;\mathbf{z}])}, \tag{3.6}$$

$$\pi_{T\to w} = \frac{\exp(\mathbf{u}_w^T f_t([\mathbf{w}_T;\mathbf{z}]))}{\sum_{w'\in\Sigma}\exp(\mathbf{u}_{w'}^T f_t([\mathbf{w}_T;\mathbf{z}]))}, \tag{3.7}$$

where $\mathbf{u}$ is a parameter vector, $\mathbf{w}_N$ is a symbol embedding and $N \in \{S\}\cup\mathcal{N}\cup\mathcal{P}$. $[\cdot;\cdot]$ indicates vector concatenation, and $f_s(\cdot)$ and $f_t(\cdot)$ encode the input into a vector (parameters are dropped for simplicity).

A C-PCFG defines a mixture of PCFGs (i.e., we can sample a set of PCFG parameters by sampling a vector $\mathbf{z}$). It satisfies the context-free assumption conditioned on $\mathbf{z}$ and thus admits exact inference for each given $\mathbf{z}$. Learning with C-PCFG involves maximizing the log-likelihood of every observed sentence $\boldsymbol{w} = w_1 w_2 \ldots w_n$:

$$\log p_\theta(\boldsymbol{w}) = \log \int_{\mathbf{z}} \sum_{t\in\mathcal{T}_\mathcal{G}(\boldsymbol{w})} p_\theta(t|\mathbf{z})p(\mathbf{z})\,d\mathbf{z}, \tag{3.8}$$

where $\mathcal{T}_\mathcal{G}(\boldsymbol{w})$ consists of all parses of $\boldsymbol{w}$ under a PCFG $\mathcal{G}$. Though for each given $\mathbf{z}$ the inner sum over parses can be efficiently computed using the inside algorithm (Baker, 1979), the integral over $\mathbf{z}$ renders optimization intractable. Instead, C-PCFG relies on variational inference and maximizes the evidence lower bound (ELBO):

$$\log p_\theta(\boldsymbol{w}) \geq \mathrm{ELBO}(\boldsymbol{w};\phi,\theta) = \mathbb{E}_{q_\phi(\mathbf{z}|\boldsymbol{w})}[\log p_\theta(\boldsymbol{w}|\mathbf{z})] - \mathrm{KL}[q_\phi(\mathbf{z}|\boldsymbol{w})||p(\mathbf{z})], \tag{3.9}$$

where $q_\phi(\mathbf{z}|\boldsymbol{w})$ is a variational posterior, a neural network parameterized with $\phi$. The expected log-likelihood term is estimated via the reparameterization trick (Kingma et al., 2014); the KL term can be computed analytically when $p(\mathbf{z})$ and $q_\phi(\mathbf{z}|\boldsymbol{w})$ are normally distributed.

### 3.2.2 Visually Grounded Neural Syntax Learner

The visually grounded neural syntax learner (VG-NSL) comprises a parsing model and an image-text matching model. The parsing model is an easy-first parser (Goldberg and Elhadad, 2010). It builds a parse greedily in a bottom-up manner while at the same time producing a semantic representation for each constituent in the parse (i.e., its

"embedding"). The parser is optimized through REINFORCE (Williams, 1992). The reward encourages merging two adjacent constituents if the merge results in a constituent that is *concrete*, i.e., if its semantic representation is predictive of the corresponding image, as measured with a matching function. We omit details of the parser and how the semantic representations of constituents are computed, as they are not relevant to our approach, and refer the reader to Shi et al. (2019a).

However, as we will extend their image-text matching model, we explain this component of their approach more formally. In their work, this loss is used to learn the textual and visual representations. For every constituent $c^{(i)}$ of a sentence $\boldsymbol{w}^{(i)}$, they define the following triplet hinge loss:

$$\begin{aligned} h(\mathbf{c}^{(i)}, \mathbf{v}^{(i)}) = & \mathbb{E}_{\mathbf{c}'} \left[ m(\mathbf{c}', \mathbf{v}^{(i)}) - m(\mathbf{c}^{(i)}, \mathbf{v}^{(i)}) + \varepsilon \right]_+ + \\ & \mathbb{E}_{\mathbf{v}'} \left[ m(\mathbf{c}^{(i)}, \mathbf{v}') - m(\mathbf{c}^{(i)}, \mathbf{v}^{(i)}) + \varepsilon \right]_+ , \end{aligned} \tag{3.10}$$

where $[\cdot]_+ = \max(0, \cdot)$, $\varepsilon$ is a positive margin, $m(\mathbf{c}, \mathbf{v}) \triangleq \cos(\mathbf{c}, \mathbf{v})$ is the matching function measuring similarity between the constituent representation $\mathbf{c}$ and the image representation $\mathbf{v}$. The expectation is taken with respect to "negative examples": $\mathbf{c}'$ and $\mathbf{v}'$. In practice, for efficiency reasons, a single representation of an image $\mathbf{v}'$ and a single representation of a constituent (span) $\mathbf{c}'$ from another example in the same batch are used as the negative examples. Intuitively, an aligned image-constituent pair $(\mathbf{v}^{(i)}, \mathbf{c}^{(i)})$ should score higher than an unaligned one ($(\mathbf{v}^{(i)}, \mathbf{c}')$ or $(\mathbf{v}', \mathbf{c}^{(i)})$).

The total loss for an image-sentence pair $(\mathbf{v}^{(i)}, \boldsymbol{w}^{(i)})$ is obtained by summing losses for all constituents in a tree $t^{(i)}$, sampled from the parsing model (we write $c^{(i)} \in \boldsymbol{w}^{(i)}$):

$$\hat{s}(\mathbf{v}^{(i)}, \boldsymbol{w}^{(i)}) = \sum_{c^{(i)} \in t^{(i)}} h(\mathbf{c}^{(i)}, \mathbf{v}^{(i)}) . \tag{3.11}$$

In their work, training alternates between optimizing the parser using rewards (relying on image and text representations) and optimizing the image-text matching model to refine image and text representations (relying on the fixed parsing model). Once trained, the parser can be directly applied to text, i.e., images are not used at test time.

### 3.2.3 Limitations of The VG-NSL Framework

While VG-NSL has shown reasonable performance in latent tree induction, there are several practical issues inhibiting this visually grounded learning framework. First, contrastive learning implicitly assumes that every constituent of a sentence has its visual representation in the aligned image. However, it is not guaranteed in practice and

would result in noisy reward signals. Besides, the loss in Equation 3.10 (and a similar component in the reward, see Shi et al. (2019a)) focuses on constituents corresponding to short spans. Long spans, independently of their syntactic structure, tend to be sufficiently discriminative to distinguish the aligned image $\mathbf{v}^{(i)}$ from an unaligned one. This implies that there is not much learning signal for such constituents. The tendency to focus on short spans and those more easily derivable from an image is evident from the results (Shi et al., 2019a). For example, their parser is accurate for noun phrases (recall 79.6%), which are often short for captions, but performs poorly on verb phrases (recall 26.2%), which have longer spans, and are more complex compositionally and also harder to predict from images (see our analysis in Section 3.4.3.2). While there may be ways to mitigate some of these issues, we believe that any image-text matching loss alone is unlikely to provide sufficient learning signal to accurately capture all aspects of syntax. Instead of resorting to language-specific inductive biases as done by Shi et al. (2019a) (i.e., head-initial bias (Baker, 2008) of English), we propose to complement the image-text matching loss with the objective derived from the unaligned text (i.e., log-likelihood), jointly training a parser to both explain the raw language data and the alignment with images.

Moreover, their learning is likely to suffer from large variance in gradient estimation as their parser does not admit tractable estimation of the partition function, and thus they have to rely on sampling decisions. This will be even more of a problem if we would attempt to use it in the joint learning setup. Similar parsing models do not yield linguistically-plausible structures when used in the conventional (i.e., non-grounded) grammar-induction set-ups (Williams et al., 2018; Havrylov et al., 2019).

In the next section, we will use compound PCFG (Kim et al., 2019b) and describe an improved visually grounded learning framework that can tackle these issues neatly.

## 3.3 Visually Grounded Compound PCFG

We use compound PCFG and develop Visually-grounded Compound PCFG (dubbed VC~PCFG) within the contrastive learning framework. Instead of sampling a tree and computing a point estimate of the image-text matching loss, we can compute the expected image-text matching loss under a tree distribution and use end-to-end contrastive learning (Section 3.3.1). Since it is inefficient to compute constituent representations relying on the chart, we will introduce an additional textual representation model to encode constituents (Section 3.3.2). Moreover, VC~PCFG lets us addition-

ally optimize a language modeling objective, complementing the visually grounded contrastive learning (Section 3.3.3).

### 3.3.1 End-to-End Contrastive Learning

In the visually grounded grammar induction framework, the parsing model is optimized through learning signals derived from the alignment of images and constituents, as scored by the image-text matching model. Denoting a set of image representations by $\mathcal{V} = \{\mathbf{v}^{(i)}\}$ and the corresponding set of sentences by $\mathcal{W} = \{\boldsymbol{w}^{(i)}\}$, the image-text matching model is optimized via contrastive learning:

$$\mathcal{L}(\mathcal{V}, \mathcal{W}; \phi, \theta) = \sum_i s(\mathbf{v}^{(i)}, \boldsymbol{w}^{(i)}) \,. \tag{3.12}$$

We define $s(\mathbf{v}^{(i)}, \boldsymbol{w}^{(i)})$ as the loss of aligning $\mathbf{v}^{(i)}$ and $\boldsymbol{w}^{(i)}$. In VG-NSL, it is estimated via point estimation (see Equation 3.11). While in VC-PCFG, given an aligned image-sentence pair $\langle \mathbf{v}, \boldsymbol{w} \rangle$, we compute the expected image-sentence matching loss under a tree distribution $p_\theta(t|\boldsymbol{w})$, leading to an end-to-end contrastive learning:

$$s(\mathbf{v}, \boldsymbol{w}) = \mathbb{E}_{p_\theta(t|\boldsymbol{w})} \sum_{c \in t} h(\mathbf{c}, \mathbf{v}) \,, \tag{3.13}$$

where $h(\mathbf{c}, \mathbf{v})$ is the hinge loss of aligning the unlabeled constituent $c$ and the image $\mathbf{v}$ (defined in Equation 3.10). Minimizing the hinge loss encourages an aligned image-constituent pair to rank higher than any unaligned one. Expanding the right-hand side of Equation 3.13 leads to:

$$\begin{aligned} s(\mathbf{v}, \boldsymbol{w}) &= \sum_{t \in \mathcal{T}_\mathcal{G}(\boldsymbol{w})} p_\theta(t|\boldsymbol{w}) \sum_{c \in t} h(\mathbf{c}, \mathbf{v}) \\ &= \sum_{c \in \boldsymbol{w}} \underbrace{\sum_{t \in \mathcal{T}_\mathcal{G}(\boldsymbol{w})} \mathbb{1}_{[c \in t]} p_\theta(t|\boldsymbol{w})}_{p(c|\boldsymbol{w})\text{: marginal of the span } c} h(\mathbf{c}, \mathbf{v}) \\ &= \sum_{c \in \boldsymbol{w}} p(c|\boldsymbol{w}) h(\mathbf{c}, \mathbf{v}) \,, \end{aligned} \tag{3.14}$$

where $p(c|\boldsymbol{w})$ is the conditional probability (i.e., marginal) of the span $c$ given $\boldsymbol{w}$. It can be efficiently computed by using the inside algorithm and automatic differentiation (Eisner, 2016; Rush, 2020).

### 3.3.2 Span Representation

Estimation of the expected image-text matching scores relies on span representations. Ideally, a span representation should encode semantics of a span with its computation guided by its syntactic structure (Socher et al., 2013). The reliance on the predicted tree structure will result in propagating learning signals derived from the alignment of images and sentences back to the parser. To realize this desideratum, we could follow the inside algorithm and recursively compose span representations (Le and Zuidema, 2015; Stern et al., 2017; Drozdov et al., 2019), which is, however, both time- and memory-inefficient in practice.

Instead, we produce span representations largely independently of the parser, as we will explain below. The only way the parser model influences this representation is through the predicted constituent label: we use its distribution to compute the representation.[2] Specifically, as a trade-off for better training efficiency, we adopt a single-layer bidirectional LSTM (BiLSTM) to encode spans. A mean-pooling layer is applied over the hidden states $\mathbf{h}$ of the BiLSTM and followed by a label-specific affine transformation $f_k(\cdot)$ to produce a label-specific span representation $\mathbf{c}_k$. Take a span $c_{i,j} = w_i \dots w_j$ $(0 < i < j \leq n)$:

$$\mathbf{c}_k = f_k\left(\frac{1}{j-i+1}\sum_{l=i}^{j}\mathbf{h}_l\right). \tag{3.15}$$

The BiLSTM encoding model operates at the span level and naturally encodes semantics of a span. Unlike using a single sentence-level BiLSTM encoder, it guarantees that no information from words outside of the span leaks into its representations. More importantly, it can run in $O(n)$ for a sentence of length $n$ with a parallel implementation. While the produced representation does not reflect the structural decisions made by the parser, it can be sensitive to word order and may be affected by its syntactic structure (Blevins et al., 2018).

In order to compute the representation of unlabeled constituent $\mathbf{c}$, we average the distribution $\mathbf{c}_k$ under the distribution of labels defined by the parser:

$$\mathbf{c} = \sum_{k=1}^{K} p(k|c,\boldsymbol{w})\mathbf{c}_k\,, \tag{3.16}$$

where $p(k|c,\boldsymbol{w})$ is the probability that the span $\mathbf{c}$ has label $k$, conditioned on having this constituent span in the tree.

[2] Intuitively, the key learning signal for the parser in our model comes through the marginals in Equation 3.14, not through the span representation.

To further reduce computation we estimate the matching loss only using the $\frac{n(n-1)}{4}$ shortest spans for a sentence of length $n$. Thus the image-text alignment loss will focus on small constituents. This is the case anyway (see discussion in Section 3.2.3), so we expect that this simplification would not hurt model performance significantly.

### 3.3.3 Joint Objective

Rather than simply optimizing the contrastive learning objective, we jointly maximize the log-likelihood of text data. As with C-PCFG, we optimize the ELBO:

$$\mathcal{L}(\mathcal{W};\phi,\theta) = -\sum_{\boldsymbol{w}\in\mathcal{W}} \text{ELBO}(\boldsymbol{w};\phi,\theta)\,. \tag{3.17}$$

This learning objective complements contrastive learning. As contrastive learning optimizes a parser by solely matching images and constituents, the parser would only focus on simple and local constituents (e.g., short NPs). Moreover, in practice, since not every constituent can be grounded in an image, contrastive learning would suffer from misleading or ambiguous learning signals.

To summarize, the overall loss function is

$$\mathcal{J}(\phi,\theta) = \mathcal{L}(\mathcal{W};\phi,\theta) + \alpha\cdot\mathcal{L}(\mathcal{V},\mathcal{W};\phi,\theta)\,, \tag{3.18}$$

where $\alpha$ is a hyper-parameter balancing the relative importance of contrastive learning.

### 3.3.4 Parsing

The parser can be directly used to parse raw text after training, without requiring access to visual groundings. Parsing seeks for the most probable parse $t^*$ of $\boldsymbol{w}$:

$$t^* = \arg\max \int_{\mathbf{z}} p_\theta(t|\boldsymbol{w},\mathbf{z}) p_\theta(\mathbf{z}|\boldsymbol{w})\, d\mathbf{z}\,. \tag{3.19}$$

Still, though the maximum a posterior (MAP) inference over $p_\theta(t|\boldsymbol{w})$ can be solved by the CKY algorithm (Cocke, 1969; Kasami, 1966; Younger, 1967), inference becomes intractable when introducing into $\mathbf{z}$. The MAP inference is instead approximated by

$$t^* \approx \arg\max \int_{\mathbf{z}} p_\theta(t|\boldsymbol{w},\mathbf{z})\delta(\mathbf{z}-\boldsymbol{\mu}_\phi(\boldsymbol{w}))\, d\mathbf{z}\,, \tag{3.20}$$

where $\delta(\cdot)$ is the Dirac delta function and $\boldsymbol{\mu}_\phi(\boldsymbol{w})$ is the mean vector of the variational posterior $q_\phi(\mathbf{z}|\boldsymbol{w})$. As $\delta(\cdot)$ has zero mass everywhere but at the mode $\boldsymbol{\mu}_\phi(\boldsymbol{w})$, it is equivalently solving $\arg\max_t p_\theta(t|\boldsymbol{w},\boldsymbol{\mu}_\phi(\boldsymbol{w}))$.

## 3.4 In-Domain Evaluation of VC-PCFG

### 3.4.1 Datasets and Evaluation

**Datasets**: We use MSCOCO (Lin et al., 2014). It consists of 82,783 training images, 1,000 validation images, and 1,000 test images. Each image is associated with 5 caption sentences. We encode images into 2048-dimensional vectors using the pre-trained ResNet-101 (He et al., 2016). At test time, only captions are used. We follow Shi et al. (2019a) and parse test captions with Benepar (Kitaev and Klein, 2018). We use the same data preprocessing as in Shen et al. (2019) and Kim et al. (2019b), where punctuation is removed from all data, and the top 10,000 frequent words in training sentences are kept as the vocabulary.

**Evaluation**: We mainly compare VC-PCFG with VG-NSL (Shi et al., 2019a). To verify the effectiveness of the use of visual groundings, we also compare our model with a C-PCFG trained only on the training captions. All models are run four times with different random seeds and for at most 15 epochs with early stopping (i.e., the image-caption loss/perplexity on the validation captions does not decrease). We report both averaged corpus-level F1 and averaged sentence-level F1 numbers as well as the unbiased standard deviations.

### 3.4.2 Settings and Hyperparameters

We adopt parameter settings suggested by the authors for the baseline models. For VG-NSL we run the authors' code. We re-implement C-PCFG using automatic differentiation (Eisner, 2016) to speed up training. Our VC-PCFG comprises a parsing model and an image-text matching model. The parsing model has the same parameters as the baseline C-PCFG; the image-text matching model has the same parameters as the baseline VG-NSL. Concretely, the parsing model has 30 nonterminals and 60 preterminals. Each of them is represented by a 256-dimensional vector. The inference model $q_\phi(\mathbf{z}|\boldsymbol{w})$ uses a single-layer BiLSTM. It has a 512-dimensional hidden state and relies on 512-dimensional word embeddings. We apply a max-pooling layer over the hidden states of the BiLSTM and then obtain 64-dimensional mean vectors $\boldsymbol{\mu}_\phi(\boldsymbol{w})$ and log-variances $\log \boldsymbol{\sigma}_\phi(\boldsymbol{w})$ by using an affine layer. The image-text matching model projects visual features into 512-dimensional feature vectors and encodes spans as 512-dimensional vectors. Our span representation model is another single-layer BiLSTM, with the same hyperparameters as in the inference model. $\alpha$ for visually grounded

| Model | NP | VP | PP | SBAR | ADJP | ADVP | C-F1 | S-F1 |
|---|---|---|---|---|---|---|---|---|
| Left Branching | 33.2 | 0.0 | 0.0 | 4.9 | 0.0 | 0.0 | 15.1 | 15.7 |
| Right Branching | 23.8 | 91.5 | 63.0 | **96.0** | 18.3 | 76.7 | 42.4 | 42.8 |
| Random Trees | $32.8_{\pm 0.5}$ | $18.4_{\pm 0.4}$ | $24.4_{\pm 0.3}$ | $17.7_{\pm 1.7}$ | $26.8_{\pm 2.6}$ | $20.9_{\pm 1.5}$ | $24.2_{\pm 0.3}$ | $24.6_{\pm 0.2}$ |
| C-PCFG | $43.0_{\pm 8.6}$ | $85.0_{\pm 2.6}$ | $78.4_{\pm 5.6}$ | $90.6_{\pm 2.1}$ | $36.6_{\pm 21}$ | **87.4**$_{\pm 1.0}$ | $53.6_{\pm 4.7}$ | $53.7_{\pm 4.6}$ |
| VG-NSL† | $79.6_{\pm 0.4}$ | $26.2_{\pm 0.4}$ | $42.0_{\pm 0.6}$ | | $22.0_{\pm 0.4}$ | | $50.4_{\pm 0.3}$ | |
| VG-NSL+HI† | $74.6_{\pm 0.5}$ | $32.5_{\pm 1.5}$ | $66.5_{\pm 1.2}$ | | $21.7_{\pm 1.1}$ | | $53.3_{\pm 0.2}$ | |
| VG-NSL* | **64.3**$_{\pm 1.1}$ | $28.1_{\pm 0.5}$ | $32.2_{\pm 1.1}$ | $16.9_{\pm 3.2}$ | $13.2_{\pm 1.5}$ | $5.6_{\pm 0.3}$ | $41.5_{\pm 0.5}$ | $41.8_{\pm 0.5}$ |
| VG-NSL+HI* | $61.0_{\pm 0.2}$ | $33.5_{\pm 1.6}$ | $62.7_{\pm 0.6}$ | $42.0_{\pm 5.1}$ | $13.9_{\pm 0.6}$ | $65.9_{\pm 2.5}$ | $48.8_{\pm 0.4}$ | $49.4_{\pm 0.5}$ |
| VC-PCFG (ours) | $54.9_{\pm 14}$ | $83.2_{\pm 3.9}$ | **80.9**$_{\pm 7.9}$ | $89.0_{\pm 2.0}$ | **38.8**$_{\pm 25}$ | $86.3_{\pm 4.1}$ | **59.3**$_{\pm 8.2}$ | **59.4**$_{\pm 8.3}$ |
| w/o LM | $35.6_{\pm 3.7}$ | **93.4**$_{\pm 2.1}$ | $70.1_{\pm 2.0}$ | $95.9_{\pm 3.9}$ | $20.6_{\pm 0.8}$ | $78.0_{\pm 2.2}$ | $49.7_{\pm 2.6}$ | $50.5_{\pm 2.5}$ |

Table 3.1: Recalls on six frequent constituent labels (i.e., NP, VP, PP, SBAR, ADJP, and ADVP) in the MSCOCO test captions and corpus-level F1 (C-F1) and sentence-level F1 (S-F1) results. The best mean number in each column is in bold. † indicates results reported by Shi et al. (2019a). $*$ denotes results obtained by running their code. Notice that the results from Shi et al. (2019a) are not comparable to ours because they keep punctuation and include trivial sentence-level spans in evaluation.

learning is set to 0.001. We implement[3] VC-PCFG relying on Torch-Struct (Rush, 2020), and optimize it using Adam (Kingma and Ba, 2015) with the learning rate set to 0.01, $\beta_1 = 0.75$, and $\beta_2 = 0.999$. All parameters are initialized with the Xavier uniform initializer (Glorot and Bengio, 2010).

## 3.4.3 Results and Analysis

### 3.4.3.1 Main Results

Our model outperforms all baselines according to both corpus-level F1 and sentence-level F1 (see Table 3.1). Notably, it surpasses VG-NSL+HI by 10% F1.[4] The right branching model is a strong baseline on image captions, as observed previously on the WSJ corpus, including in recent work (Shen et al., 2018; Kim et al., 2019b). Comparing with C-PCFG, which is trained solely on captions, VC-PCFG achieves a much

[3] https://github.com/zhaoyanpeng/vpcfg.

[4] We run the code of Shi et al. (2019a) and train VG-NSL and VG-NSL+HI on the training captions with punctuation removed. This is considered a more challenging setting as punctuation signals the boundaries of constituents and makes it easy for parsers to derive constituents. At test, as a common practice (Shen et al., 2018, 2019; Kim et al., 2019b), we discard punctuation and ignore trivial single-word and sentence-level spans. We notice that including sentence-level spans can improve the F1 of VG-NSL to around 48%.

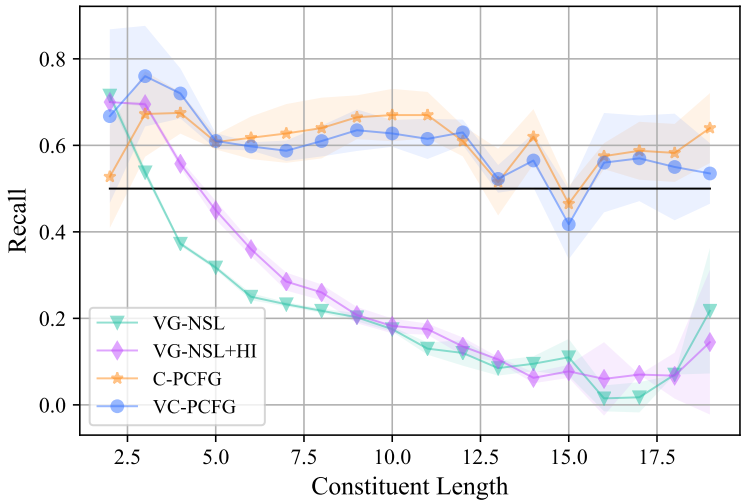



Figure 3.2: Recall broken down by constituent length.

higher mean F1 (+5.7% F1), demonstrating the informativeness of visual groundings. However, VC-PCFG suffers from a larger variance, presumably because the joint objective is harder to optimize. Visually grounded contrastive learning (w/o LM) has a mean F1 50.5%. It is further improved to 59.4% when additionally optimizing the language modeling objective.

Moreover, we show recall on six frequent constituent labels (NP, VP, PP, SBAR, ADJP, and ADVP) in the test captions. Unsurprisingly, VG-NSL is best on NPs because the matching-based reward signals optimize it to focus only on short and concrete NPs (recall 64.3%). It performs poorly on other constituent labels such as VPs (recall 28.1%). In contrast, VC-PCFG exhibits a relatively even performance across constituent labels, e.g., it is most accurate on PPs and SBARs and works fairly well on VPs (recall 83.2%). Meanwhile, it improves over C-PCFG for NPs, which are usually short and "concrete", once again confirming the usefulness of visual groundings. Visually grounded contrastive learning (w/o LM) tends to behave like the right branching baseline but performs slightly better on NPs (+2.8% recall). Additionally optimizing the language modeling objective brings a huge improvement for NPs (+19.3% recall).

#### 3.4.3.2 Analysis

We analyze model performance for constituents of different lengths (Figure 3.2). As expected, VG-NSL becomes weaker as constituent length increases, and the drop is very dramatic. C-PCFG and its grounded version VC-PCFG consistently outperform VG-NSL on constituents longer than four tokens and display a more even performance

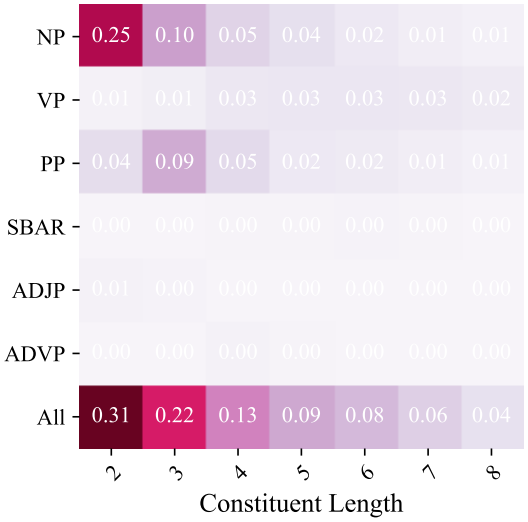



Figure 3.3: Label distribution over constituent length. "All" denotes frequencies of constituent lengths. Zero frequencies are due to the limited numerical precision.

across constituent lengths. Meanwhile, VC-PCFG beats C-PCFG on constituents of length below 5, confirming that visual groundings are beneficial for short spans. We further plot the distribution over constituent length for different phrase types (Figure 3.3) and find that around 75% constituents in our dataset are shorter than six tokens, and 60% of them are NPs. Thus, it is not surprising that the improvement on NPs, brought by visually grounded learning, has a large impact on the overall performance.

Next, we analyze induced tree structures. We compare model predictions against gold trees, left-branching trees, and right-branching trees. As there is little performance difference between corpus-level F1 and sentence-level F1, we focus on sentence-level F1 in this analysis. We report self F1 (Williams et al., 2018) to show model consistency across runs. The self F1 is computed by averaging over six model pairs from four different runs. All results are presented in Table 3.2. Overall, all models have self F1 above 70%, indicating a relatively high consistency. We observe that using the head-initial bias pushes VG-NSL closer to the right-branching baseline, while visually grounded learning leads to improvements over C-PCFG, forcing VC-PCFG to deviate from the default right-branching behaviour.

Moreover, we test VG-NSL+HI and VC-PCFG on 50 manually annotated captions released by Shi et al. (2019a). VC-PCFG achieves a mean F1 62.7%, surpassing VG-NSL+HI by 12.1% F1. In Figure 3.4 we visualize a parse tree predicted by the best run of VC-PCFG. We can see that VC-PCFG identifies most NPs but makes mistakes in PP attachment and consequently fails to identify the VP.

| Model | Gold | Left | Right | Self |
|---|---|---|---|---|
| VG-NSL | 41.8 | 28.3 | 20.6 | 84.3 |
| VG-NSL+HI | 49.4 | 24.5 | 29.2 | 88.6 |
| C-PCFG | 53.7 | 1.3 | 53.6 | 77.3 |
| VC-PCFG | 59.4 | 4.4 | 48.5 | 71.1 |

Table 3.2: Average sentence-level F1 results against gold trees (Gold), left-branching trees (Left), right-branching trees (Right), and self F1 (Self) (Williams et al., 2018).

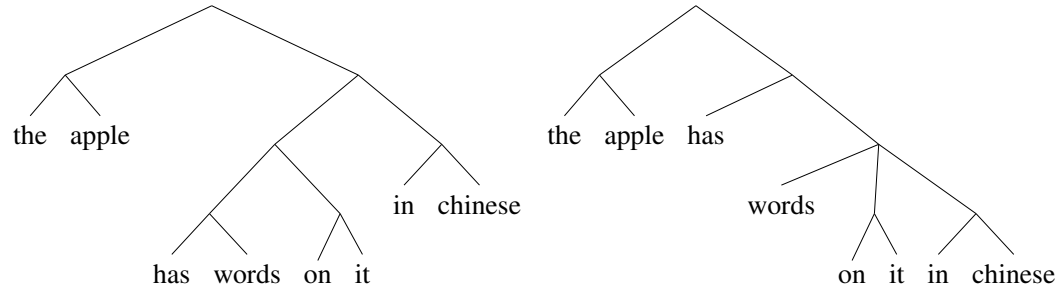



Figure 3.4: **Left:** A parse output by the best run of VC-PCFG. **Right:** The gold tree.

## 3.5 Cross-Domain Transfer of VC-PCFG

VC-PCFG has demonstrated impressive performance and, specifically, it improves over non-grounded C-PCFG on MSCOCO, but it has only been evaluated on in-domain text, so it is unclear if the improvements transfer across domains or, equivalently, if the models acquire a general grammar or only a grammar suitable for a certain domain.

To bridge the gap, we further study the transferability of VC-PCFG. First, to enable VC-PCFG to transfer across domains, we extend it using pre-trained word embeddings and obtain transferrable VC-PCFG (dubbed TVC-PCFG). This modification allows for directly applying VC-PCFG to a target domain, without requiring training on any data from the target domain.

### 3.5.1 Transfer Learning

#### 3.5.1.1 Background

To motivate transfer learning of VC-PCFG, we first reiterate the learning objective of VC-PCFG. Suppose a captioning data set $\mathcal{D} = \{(\mathrm{v}^{(i)}, \boldsymbol{w}^{(i)}) | 1 \leq i \leq N\}$ consists of $N$

pairs of image v and caption $\boldsymbol{w}$, the loss function of VC-PCFG is formally defined as:

$$\mathcal{L} = \sum_i -\log p(\boldsymbol{w}^{(i)}) + \alpha \cdot s(\mathrm{v}^{(i)}, \boldsymbol{w}^{(i)}), \tag{3.21}$$

where $\boldsymbol{w}$ is a caption; $\mathbf{v}$ is the vector representation of an image v and is precomputed by using a pre-trained image encoder such as ResNet (He et al., 2016). The hyper-parameter $\alpha$ controls the relative importance of the two loss terms. The first term $-\log p_\theta(\boldsymbol{w})$ computes the negative log-likelihood of the caption $\boldsymbol{w}$ under a PCFG $\mathcal{G}$; it can be seen as a language modeling (LM) loss. The second term $s(\mathbf{v}, \boldsymbol{w})$ defines a hinge loss. Intuitively, $s(\cdot)$ is optimized to score higher for an aligned pair $(\mathbf{v}^{(i)}, \boldsymbol{w}^{(i)})$ than for any un-aligned pair by a positive margin (see technical details in Section 3.3).

Essentially, the loss function defined in Equation 3.21 corresponds to multi-objective learning and can be applied to text alone or to image-text pairs. Specifically, for sentences that are paired with aligned images, both the LM loss and the hinge loss are minimized; for sentences without aligned images, only the LM loss is minimized. By treating images as the labels of the aligned text, this type of multi-objective learning can be seen as semi-supervised learning.

#### 3.5.1.2 Transferable VC-PCFG

We consider a zero-shot transfer learning setting: we directly apply and transfer a pre-trained VC-PCFG to the target domain. This setting is viable because VC-PCFG can be learned solely on text and does not rely on images to parse text at inference time.

To enable VC-PCFG to transfer across domains, we extend it by using pre-trained word embeddings and sharing them between the source domain and the target domain. Following our definition of PCFG, a VC-PCFG consists of three types of grammar rules: start rules (e.g., $S \rightarrow A$), binary rules (e.g., $A \rightarrow BC$), and preterminal rules (e.g., $T \rightarrow w$). The start rules and the binary rules are domain agnostic, but the preterminal rules, which generate a word conditioning on a preterminal, are domain-dependent because they rely on the domain-specific vocabulary. Thus, the key to transferring VC-PCFG from the source domain to the target domain is to share preterminal rules or, equivalently, a vocabulary between the source and target domains.

Still, sharing the same set of grammar rules between the two domains does not guarantee that a learned model transfers to unseen preterminal rules. This is because the target-domain vocabulary is not necessarily subsumed by the source-domain vocabulary. To make it more clear, we first note that VC-PCFG generates rule probabilities

| Split | CF | CG | CK | CL | CM | CN | CP | CR | All (Brown) |
|---|---|---|---|---|---|---|---|---|---|
| train | 2191 | 2324 | 2708 | 2745 | 615 | 3267 | 2801 | 648 | 17299 |
| dev | 507 | 461 | 570 | 518 | 115 | 599 | 543 | 164 | 3477 |
| test | 466 | 494 | 603 | 451 | 151 | 549 | 598 | 155 | 3467 |
| | File ID Splits | | | | | | | | |
| train | 1-22 | 1-25 | 1-19 | 1-18 | 1-4 | 1-21 | 1-20 | 1-6 | |
| dev | 23-27 | 26-31 | 20-23 | 19-21 | 5-5 | 22-25 | 21-25 | 7-7 | |
| test | 28-32 | 32-36 | 24-29 | 22-24 | 6-6 | 26-29 | 26-29 | 8-9 | |

Table 3.3: Nine subdomains of the Brown corpus of Penn Treebank (Marcus et al., 1999). CF: popular lore. CG: belles lettres, biography, memoires, etc. CK: general fiction. CL: mystery and detective fiction. CM: science fiction. CN: adventure and western fiction. CP: romance and love story. CR: humor.

conditioning on grammar symbols. Take preterminal rules of the form $T \to w$:

$$p(T \to w) \propto g(\mathbf{u}_T, \mathbf{e}_w, \mathbf{z}; \theta), \tag{3.22}$$

where $g_\theta$ is a neural network, $\mathbf{u}$ and $\mathbf{e}$ indicate preterminal-symbol embeddings and word embeddings, respectively, and $\mathbf{z}$ is a sentence-dependent latent vector. Since we train VC-PCFG only on the source domain, for preterminal rules that contain words outside of the source domain, their rule probabilities and, specifically, the word embeddings that are used to compute the rule probabilities, will never be learned. To resolve this issue, we propose to use pre-trained word embeddings, namely GloVe (Pennington et al., 2014), and refer to the resulting model as TVC-PCFG. Pre-trained word embeddings have encoded similarities between words, i.e., similar words are generally close to each other in the learned vector space. We keep pre-trained word embeddings frozen during training. Thus, at test time, for words (preterminal rules) that are unseen during training, our TVC-PCFG can exploit similarities in the embedding space to estimate rule probabilities.

## 3.5.2 Experiments

### 3.5.2.1 Data Sets and Evaluation

We use MSCOCO captioning data set (Lin et al., 2014; Chen et al., 2015) as the source domain and conduct *proximate-domain* transfer and *remote-domain* transfer experiments. For proximate-domain transfer, we consider Flickr30k (Flickr; Young

| Split | Answers | Email | Newsgroup | Reviews | Weblog | All (Enweb) |
|---|---|---|---|---|---|---|
| train | 2353 | 3362 | 1648 | 2565 | 1451 | 11379 |
| dev | 565 | 767 | 368 | 622 | 245 | 2567 |
| test | 569 | 759 | 371 | 626 | 334 | 2659 |

Table 3.4: Five subdomains of the English Web Treebank (Bies et al., 2012).

et al. (2014)). For remote-domain transfer, we consider Wall Street Journal (WSJ) and Brown portions of the Penn Treebank (Marcus et al., 1999), and the English Web Treebank (Enweb; Bies et al. (2012)). Note that Brown and Enweb consist of 8 and 5 subdomains, respectively, so we will be actually performing remote-domain transfer on 14 text domains (see below for details).

**Flickr** is an image captioning dataset. While the images of Flickr and MSCOCO are all sourced from Flickr, they focus on different aspects,[5] so do their captions. Though the guidelines for collecting MSCOCO captions (Chen et al., 2015) are inspired by those of Flickr (Hodosh et al., 2013; Young et al., 2014), due to the differences in the instructions, the statistics of the collected captions tend to be different, e.g., Flickr test captions are slightly longer than MSCOCO training captions (i.e., 12.4 vs 10.5 tokens on average). Nevertheless, Flickr is close to MSCOCO and thus we choose Flickr captions for *proximate-domain* transfer. Since Flickr does not contain gold phrase structures of captions, we follow the experimental settings of VC-PCFG and parse all the captions with Benepar (Kitaev and Klein, 2018).

**WSJ** is a news corpus and the central part of the Penn Treebank resource (Marcus et al., 1999). Sentences in WSJ have been manually annotated with phrase structures. We use WSJ for *remote-domain* transfer; sentences in newswire and image captions are very different as evident, for example, from the divergences in distributions of tokens, syntactic fragments, and sentence lengths (20.4 vs 10.5 tokens on average).

**Brown** is also part of the Penn Treebank resource and consists of manually parsed sentences from 8 domains, which cover various genres such as lore, biography, fiction, and humor (Marcus et al., 1999). We divide the sentences in each domain into three parts: around 70% of the sentences for training, 15% for development, and 15% for

[5]Flickr images focus on people and animals that perform some actions (Hodosh et al., 2013; Young et al., 2014; Plummer et al., 2015) while MSCOCO covers more diverse object categories (up to 80) and focuses on multiple-object images (Lin et al., 2014; Chen et al., 2015).

test. We further merge the training, development, and test subsets across domains and create a mixed-domain Brown (see Table 3.3). The average length of Brown test sentences is much longer than MSCOCO training captions, i.e., 17.1 vs 10.5 tokens. Since all these subdomains differ in terms of genre from image captions, we use them for *remote-domain* transfer.

**Enweb** is short for English Web Treebank and consists of sentences from 5 domains: weblogs, newsgroups, email, reviews, and question-answers (Bies et al., 2012). Each of these domains contains sentences that have been manually annotated with syntactic structures. We divide sentences in each domain in a similar way as we divide Brown sentences. We also create a mixed-domain Enweb (see Table 3.4). Enweb test sentences are slightly longer than MSCOCO training captions, i.e., 13.9 vs 10.5 tokens on average. Since they belong to genres different from image captions, we use them for *remote-domain* transfer.

#### 3.5.2.2 Model Configurations

**Transfer learning models.** We use the same implementations of the text-only parser C-PCFG and the visually-grounded version VC-PCFG as Zhao and Titov (2020) but replace their word embeddings with pre-trained GloVe embeddings (models are dubbed T(V)C-PCFG). We follow the setups in Zhao and Titov (2020) to learn and evaluate T(V)C-PCFG.[6] To measure model performance, we resort to unlabeled corpus-level F1 (C-F1) and sentence-level F1 (S-F1), which are equivalent to recall in unsupervised grammar induction.

**Domain-specific vocabulary.** For each corpus, we keep the top 10,000 frequent words in the corresponding training set as the vocabulary. In training and test, tokens outside of the given vocabulary are treated as a special "<unk>" token (short for "unknown"). We share the vocabulary of each mixed domain among its subdomains, e.g., the 8 subdomains of Brown shares the vocabulary of the mixed domain Brown, similarly for Enweb.

**Test-time vocabulary.** At test time, we use domain-specific vocabulary rather than the training-time vocabulary (i.e., the MSCOCO vocabulary). The reasons for doing so include: (1) domain-specific vocabulary is likely to cover more target-domain words

[6] https://github.com/zhaoyanpeng/cpcfg and https://github.com/zhaoyanpeng/xcfg.

| Model | NP | VP | PP | SBAR | ADJP | ADVP | C-F1 | S-F1 |
|---|---|---|---|---|---|---|---|---|
| Left Branching | 33.2 | 0.0 | 0.0 | 4.9 | 0.0 | 0.0 | 15.1 | 15.7 |
| Right Branching | 23.8 | 91.5 | 63.0 | **96.0** | 18.3 | 76.7 | 42.4 | 42.8 |
| Random Trees | $32.8_{\pm 0.5}$ | $18.4_{\pm 0.4}$ | $24.4_{\pm 0.3}$ | $17.7_{\pm 1.7}$ | $26.8_{\pm 2.6}$ | $20.9_{\pm 1.5}$ | $24.2_{\pm 0.3}$ | $24.6_{\pm 0.2}$ |
| C-PCFG † | $43.0_{\pm 8.6}$ | $\mathbf{85.0}_{\pm 2.6}$ | $78.4_{\pm 5.6}$ | $\mathbf{90.6}_{\pm 2.1}$ | $36.6_{\pm 21}$ | $\mathbf{87.4}_{\pm 1.0}$ | $53.6_{\pm 4.7}$ | $53.7_{\pm 4.6}$ |
| VC-PCFG † | $54.9_{\pm 14}$ | $83.2_{\pm 3.9}$ | $\mathbf{80.9}_{\pm 7.9}$ | $89.0_{\pm 2.0}$ | $38.8_{\pm 25}$ | $86.3_{\pm 4.1}$ | $59.3_{\pm 8.2}$ | $59.4_{\pm 8.3}$ |
| TC-PCFG * | $31.8_{\pm 13.5}$ | $60.0_{\pm 25.5}$ | $54.5_{\pm 14.0}$ | $73.0_{\pm 18.5}$ | $39.3_{\pm 23.0}$ | $59.5_{\pm 19.4}$ | $38.7_{\pm 2.6}$ | $38.8_{\pm 2.6}$ |
| TVC-PCFG * | $\mathbf{79.1}_{\pm 6.0}$ | $67.8_{\pm 13.7}$ | $71.4_{\pm 8.5}$ | $80.7_{\pm 9.2}$ | $\mathbf{59.1}_{\pm 17.9}$ | $84.9_{\pm 3.0}$ | $\mathbf{65.7}_{\pm 2.1}$ | $\mathbf{66.3}_{\pm 2.1}$ |

Table 3.5: Parsing performance on MSCOCO. † indicates the results from Zhao and Titov (2020) and * indicates models with pre-trained GloVe word embeddings.

| Model | NP | VP | PP | SBAR | ADJP | ADVP | C-F1 | S-F1 |
|---|---|---|---|---|---|---|---|---|
| Left Branching | 32.9 | 0.0 | 0.3 | 0.5 | 0.5 | 0.0 | 14.4 | 16.4 |
| Right Branching | 27.9 | **88.0** | 56.0 | **92.5** | 13.3 | 66.9 | 44.3 | 48.0 |
| Random Trees | $30.6_{\pm 0.2}$ | $17.4_{\pm 0.5}$ | $21.9_{\pm 0.6}$ | $15.8_{\pm 1.8}$ | $25.5_{\pm 2.5}$ | $19.6_{\pm 5.1}$ | $22.0_{\pm 0.3}$ | $24.2_{\pm 0.3}$ |
| C-PCFG † | $35.6_{\pm 23.4}$ | $64.6_{\pm 9.0}$ | $63.3_{\pm 25.0}$ | $55.1_{\pm 33.0}$ | $10.2_{\pm 4.4}$ | $58.6_{\pm 36.2}$ | $43.0_{\pm 16.6}$ | $45.8_{\pm 17.3}$ |
| VC-PCFG † | $33.7_{\pm 20.7}$ | $61.6_{\pm 6.2}$ | $46.8_{\pm 26.8}$ | $40.6_{\pm 37.8}$ | $12.7_{\pm 9.1}$ | $39.2_{\pm 39.0}$ | $38.0_{\pm 15.4}$ | $40.9_{\pm 15.4}$ |
| TC-PCFG * | $29.6_{\pm 15.5}$ | $58.3_{\pm 19.1}$ | $58.0_{\pm 12.4}$ | $66.7_{\pm 9.1}$ | $38.6_{\pm 27.2}$ | $55.8_{\pm 15.4}$ | $38.5_{\pm 2.1}$ | $40.5_{\pm 2.0}$ |
| TVC-PCFG * | $\mathbf{76.3}_{\pm 6.5}$ | $64.8_{\pm 11.1}$ | $\mathbf{72.7}_{\pm 5.6}$ | $69.1_{\pm 3.6}$ | $\mathbf{55.1}_{\pm 17.9}$ | $\mathbf{70.0}_{\pm 4.6}$ | $\mathbf{63.0}_{\pm 2.2}$ | $\mathbf{66.6}_{\pm 2.3}$ |

Table 3.6: Parsing performance on Flick. † indicates the results obtained by running (V)C-PCFG on Flick; * indicates the best models (w/ pre-trained GloVe word embeddings) trained on MSCOCO but evaluated on Flickr.

than the training-time vocabulary, and (2) this allows for fair comparison because the baseline C-PCFG also uses domain-specific vocabulary.

### 3.5.3 Main Results

**VC-PCFG benefits from pre-trained GloVe.** We run experiments on MSCOCO with pre-trained GloVe word embeddings (see Table 3.5). When trained on both images and text, TVC-PCFG improves over VC-PCFG (+6.9% S-F1). But when trained only on text, TC-PCFG lags far behind C-PCFG, i.e., using pre-trained GloVe leads to a reduction in performance.

We speculate that: this is because domain-specific lexical information is important for grammar induction models. GloVe has been pre-trained on diverse text and may not best reflect lexical information relevant to the domain of MSCOCO captions (e.g.,

| Model | NP | VP | PP | SBAR | ADJP | ADVP | C-F1 | S-F1 |
|---|---|---|---|---|---|---|---|---|
| Left Branching | 10.4 | 0.5 | 5.0 | 5.3 | 2.5 | 8.0 | 6.0 | 8.7 |
| Right Branching | 24.1 | **71.5** | 42.4 | **68.7** | 27.7 | 38.1 | 36.1 | 39.5 |
| Random Trees | $22.5_{\pm 0.3}$ | $12.3_{\pm 0.3}$ | $19.0_{\pm 0.5}$ | $9.3_{\pm 0.6}$ | $24.3_{\pm 1.7}$ | $26.9_{\pm 1.3}$ | $15.3_{\pm 0.1}$ | $18.1_{\pm 0.1}$ |
| C-PCFG † | $\mathbf{76.7}_{\pm 2.0}$ | $40.7_{\pm 5.5}$ | $\mathbf{71.3}_{\pm 2.1}$ | $53.8_{\pm 3.1}$ | $\mathbf{45.9}_{\pm 2.8}$ | $\mathbf{64.2}_{\pm 2.8}$ | $\mathbf{53.5}_{\pm 1.4}$ | $\mathbf{55.7}_{\pm 1.3}$ |
| L10C-PCFG † | $67.1_{\pm 3.8}$ | $31.0_{\pm 9.8}$ | $61.3_{\pm 2.2}$ | $45.9_{\pm 8.2}$ | $36.7_{\pm 2.3}$ | $41.3_{\pm 6.0}$ | $45.5_{\pm 2.4}$ | $48.2_{\pm 2.3}$ |
| TC-PCFG * | $30.9_{\pm 5.5}$ | $23.6_{\pm 7.3}$ | $36.4_{\pm 9.0}$ | $27.2_{\pm 5.3}$ | $24.7_{\pm 1.6}$ | $34.2_{\pm 4.7}$ | $24.4_{\pm 1.7}$ | $28.0_{\pm 1.9}$ |
| TVC-PCFG * | $48.6_{\pm 3.7}$ | $24.8_{\pm 4.1}$ | $39.4_{\pm 6.5}$ | $27.2_{\pm 1.1}$ | $30.2_{\pm 4.6}$ | $40.4_{\pm 2.0}$ | $32.0_{\pm 1.2}$ | $35.3_{\pm 1.3}$ |

Table 3.7: Parsing performance on WSJ. † indicates the models that are trained and evaluated on WSJ (Zhao and Titov, 2021). The prefix "L10" indicates that the models are trained on WSJ sentences shorter than 11 tokens but are tested on the full WSJ test set. * indicates the best models (w/ pre-trained GloVe word embeddings) trained on MSCOCO but evaluated on WSJ.

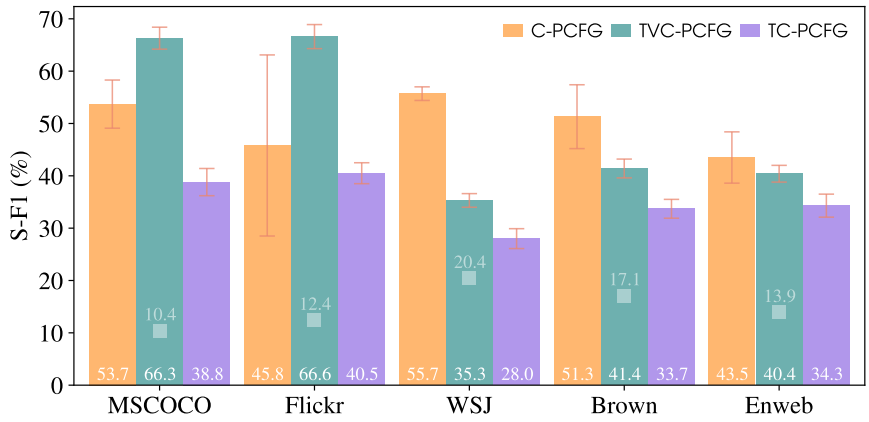



Figure 3.5: S-F1 numbers on different target domains. C-PCFG is trained only on the text data of each domain's training set. TVC-PCFG is our transfer learning model and TC-PCFG is the transfer learning model that is trained without using visual groundings. The squares indicate the average length of the test sentences of each domain.

wrong senses and parts of speech), so TC-PCFG underperforms C-PCFG. But visual groundings are specific to a domain and could regularize a parser to capture domain-specific lexical information (Zhao and Titov, 2020), so TVC-PCFG is less prone to the same issue as in TC-PCFG; instead, it might be making the best of both visual groundings and pre-trained GloVe, so it outperforms VC-PCFG.

| Model | NP | VP | PP | SBAR | ADJP | ADVP | C-F1 | S-F1 |
|---|---|---|---|---|---|---|---|---|
| Left Branching | 7.9 | 0.7 | 3.9 | 7.0 | 3.1 | 15.2 | 5.2 | 8.3 |
| Right Branching | 24.9 | **65.0** | 38.7 | **58.6** | 31.6 | 20.4 | 37.1 | 45.3 |
| Random Trees | $24.7_{\pm 0.2}$ | $15.0_{\pm 0.2}$ | $21.3_{\pm 0.6}$ | $11.7_{\pm 1.3}$ | $22.1_{\pm 0.9}$ | $28.9_{\pm 3.3}$ | $16.5_{\pm 0.2}$ | $21.2_{\pm 0.2}$ |
| C-PCFG † | **75.0**$_{\pm 3.1}$ | $31.9_{\pm 16.2}$ | **67.2**$_{\pm 8.5}$ | $54.6_{\pm 3.9}$ | **39.7**$_{\pm 7.8}$ | **59.4**$_{\pm 2.6}$ | **47.8**$_{\pm 4.4}$ | **51.3**$_{\pm 6.1}$ |
| L-C-PCFG † | $63.3_{\pm 1.8}$ | $25.5_{\pm 23.5}$ | $53.7_{\pm 6.7}$ | $36.2_{\pm 7.9}$ | $28.2_{\pm 8.9}$ | $40.2_{\pm 3.1}$ | $38.3_{\pm 6.2}$ | $42.8_{\pm 8.9}$ |
| TC-PCFG * | $34.7_{\pm 8.3}$ | $28.9_{\pm 5.9}$ | $38.8_{\pm 10.5}$ | $34.2_{\pm 3.8}$ | $26.3_{\pm 2.3}$ | $33.3_{\pm 2.6}$ | $27.5_{\pm 1.7}$ | $33.7_{\pm 1.8}$ |
| TVC-PCFG * | $58.5_{\pm 4.0}$ | $29.7_{\pm 2.7}$ | $44.8_{\pm 6.5}$ | $34.4_{\pm 1.3}$ | $32.9_{\pm 3.1}$ | $38.1_{\pm 1.2}$ | $35.9_{\pm 1.4}$ | $41.4_{\pm 1.8}$ |
| | Per-domain Performance of TVC-PCFG | | | | | | | |
| CF | $53.4_{\pm 4.9}$ | $25.4_{\pm 2.9}$ | $41.4_{\pm 7.0}$ | $32.1_{\pm 4.3}$ | $32.6_{\pm 6.3}$ | $31.0_{\pm 3.0}$ | $34.0_{\pm 1.3}$ | $37.5_{\pm 1.6}$ |
| CP | $64.1_{\pm 3.0}$ | $31.9_{\pm 3.8}$ | $49.0_{\pm 8.2}$ | $41.1_{\pm 1.9}$ | $34.5_{\pm 3.2}$ | $39.7_{\pm 7.3}$ | $37.7_{\pm 1.6}$ | $44.1_{\pm 2.1}$ |
| CN | $63.5_{\pm 3.2}$ | $33.8_{\pm 3.2}$ | $49.2_{\pm 5.7}$ | $39.0_{\pm 2.8}$ | $34.3_{\pm 5.6}$ | $45.8_{\pm 7.6}$ | $38.9_{\pm 1.2}$ | $42.9_{\pm 1.3}$ |
| CM | $61.4_{\pm 5.5}$ | $36.4_{\pm 2.9}$ | $49.6_{\pm 5.5}$ | $39.5_{\pm 4.5}$ | $44.8_{\pm 8.6}$ | $50.0_{\pm 10.2}$ | $40.0_{\pm 1.0}$ | $46.3_{\pm 1.3}$ |
| CG | $53.6_{\pm 4.6}$ | $25.8_{\pm 3.1}$ | $39.1_{\pm 6.8}$ | $31.0_{\pm 1.8}$ | $26.3_{\pm 3.3}$ | $28.6_{\pm 2.7}$ | $32.6_{\pm 1.8}$ | $37.1_{\pm 2.2}$ |
| CR | $52.0_{\pm 3.3}$ | $24.0_{\pm 2.9}$ | $39.0_{\pm 6.7}$ | $25.9_{\pm 3.2}$ | $26.8_{\pm 3.0}$ | $34.5_{\pm 7.4}$ | $31.8_{\pm 0.8}$ | $35.0_{\pm 1.4}$ |
| CK | $61.0_{\pm 4.8}$ | $29.7_{\pm 2.0}$ | $46.6_{\pm 6.3}$ | $33.1_{\pm 2.2}$ | $34.0_{\pm 3.3}$ | $35.5_{\pm 0.6}$ | $35.9_{\pm 1.5}$ | $42.6_{\pm 1.8}$ |
| CL | $62.2_{\pm 3.9}$ | $32.6_{\pm 1.6}$ | $46.4_{\pm 6.3}$ | $39.7_{\pm 3.4}$ | $35.3_{\pm 1.7}$ | $40.5_{\pm 4.2}$ | $36.7_{\pm 1.4}$ | $41.3_{\pm 2.0}$ |

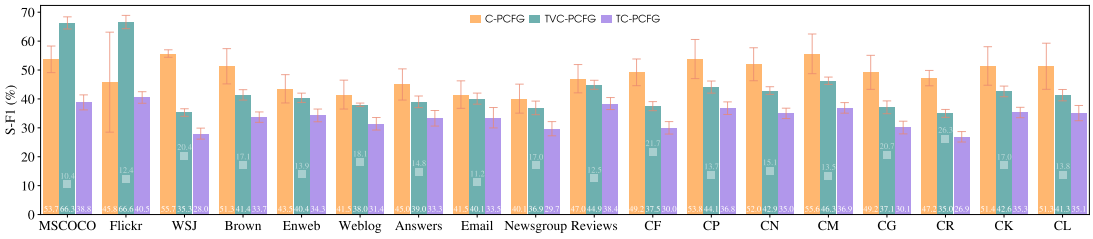
Table 3.8: Parsing performance on Brown. † indicates the results obtained by running C-PCFG on Brown; * indicates the best models (w/ pre-trained GloVe word embeddings) trained on MSCOCO but evaluated on Brown.

**TVC-PCFG succeeds in proximate-domain transfer.** We learn TVC-PCFG from MSCOCO and evaluate it on Flickr without further training (see Table 3.6). Our transfer learning model achieves the best corpus- and sentence-level F1 scores on MSCOCO. When evaluated on Flickr, it outperforms C-PCFG (+10.8% S-F1; see Figure 3.5), so the improvements brought by visual groundings can transfer to similar text domains.

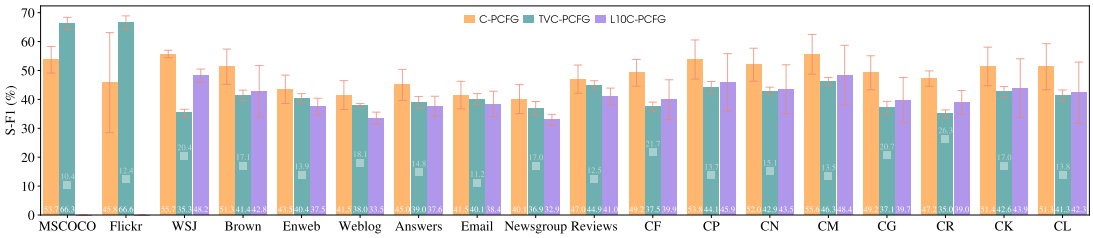
**TVC-PCFG fails in remote-domain transfer.** We further evaluate the pre-trained TVC-PCFG on remote-domain text, including WSJ, Brown, and Enweb (see Table 3.7-3.9). On the whole, the transfer learning model TVC-PCFG underperforms C-PCFG, which is trained individually on the training set of each target domain (see Figure 3.5). We observe the largest S-F1 gap between TVC-PCFG and C-PCFG on WSJ (-20.4%) and the smallest S-F1 gap on Enweb (-3.1%). This may be because of differences in language register. Both WSJ and Enweb are different from MSCOCO at a lexical level, but Enweb, consisting of web texts, contains informal language which is likely to be structurally similar to that of captions.

| Model | NP | VP | PP | SBAR | ADJP | ADVP | C-F1 | S-F1 |
|---|---|---|---|---|---|---|---|---|
| Left Branching | 9.9 | 0.9 | 3.4 | 10.1 | 3.9 | 11.2 | 5.8 | 10.9 |
| Right Branching | 27.1 | **66.3** | 41.6 | **59.3** | 30.9 | 29.8 | 38.3 | **45.9** |
| Random Trees | $25.1_{\pm 0.2}$ | $14.7_{\pm 0.3}$ | $21.6_{\pm 1.1}$ | $13.0_{\pm 1.0}$ | $22.1_{\pm 1.9}$ | $32.9_{\pm 2.0}$ | $16.8_{\pm 0.2}$ | $23.1_{\pm 0.3}$ |
| C-PCFG † | **62.8**$_{\pm 2.6}$ | $25.5_{\pm 10.4}$ | **53.5**$_{\pm 12.4}$ | $52.9_{\pm 2.4}$ | $32.6_{\pm 5.8}$ | **48.5**$_{\pm 9.1}$ | **39.7**$_{\pm 4.5}$ | $43.5_{\pm 4.9}$ |
| L10C-PCFG † | $56.4_{\pm 2.2}$ | $24.6_{\pm 9.6}$ | $33.0_{\pm 4.9}$ | $24.1_{\pm 4.3}$ | $24.2_{\pm 2.3}$ | $29.2_{\pm 2.8}$ | $31.5_{\pm 3.2}$ | $37.5_{\pm 2.9}$ |
| TC-PCFG * | $34.9_{\pm 6.8}$ | $28.0_{\pm 7.0}$ | $41.1_{\pm 10.2}$ | $34.2_{\pm 4.2}$ | $27.4_{\pm 1.7}$ | $38.3_{\pm 5.2}$ | $27.6_{\pm 2.0}$ | $34.3_{\pm 2.2}$ |
| TVC-PCFG * | $55.0_{\pm 3.9}$ | $28.6_{\pm 3.6}$ | $45.4_{\pm 6.1}$ | $34.9_{\pm 0.4}$ | **35.1**$_{\pm 2.5}$ | $41.4_{\pm 7.1}$ | $34.6_{\pm 1.5}$ | $40.4_{\pm 1.6}$ |
| Per-domain Performance of TVC-PCFG | | | | | | | | |
| Weblog | $51.2_{\pm 4.8}$ | $26.8_{\pm 3.4}$ | $40.4_{\pm 5.4}$ | $31.0_{\pm 3.7}$ | $30.8_{\pm 5.3}$ | $42.0_{\pm 4.5}$ | $32.7_{\pm 0.8}$ | $38.0_{\pm 0.6}$ |
| Answers | $58.9_{\pm 3.4}$ | $29.3_{\pm 2.9}$ | $50.7_{\pm 5.2}$ | $35.1_{\pm 1.9}$ | $37.7_{\pm 6.3}$ | $43.4_{\pm 6.9}$ | $34.8_{\pm 1.4}$ | $39.0_{\pm 2.0}$ |
| Email | $52.8_{\pm 3.9}$ | $26.2_{\pm 3.2}$ | $42.5_{\pm 6.7}$ | $32.6_{\pm 3.7}$ | $32.2_{\pm 6.9}$ | $35.9_{\pm 8.3}$ | $33.0_{\pm 1.4}$ | $40.1_{\pm 1.9}$ |
| Newsgroup | $50.9_{\pm 2.7}$ | $27.3_{\pm 3.5}$ | $41.2_{\pm 7.8}$ | $33.7_{\pm 1.4}$ | $29.2_{\pm 5.6}$ | $36.9_{\pm 4.6}$ | $33.7_{\pm 1.8}$ | $36.9_{\pm 2.4}$ |
| Reviews | $61.7_{\pm 5.1}$ | $31.5_{\pm 4.4}$ | $52.6_{\pm 6.9}$ | $42.5_{\pm 3.8}$ | $38.7_{\pm 6.5}$ | $42.6_{\pm 5.5}$ | $37.9_{\pm 1.3}$ | $44.9_{\pm 1.6}$ |

Table 3.9: Parsing performance on Enweb. † indicates the results obtained by running C-PCFG on Enweb; * indicates the best models (w/ pre-trained GloVe word embeddings) trained on MSCOCO but evaluated on Enweb.




(a) Comparison between C-PCFG, TVC-PCFG, and TC-PCFG on all target domains.




(b) Comparison between C-PCFG, TVC-PCFG, and L10C-PCFG on all target domains.

Figure 3.6: C-PCFG and L10C-PCFG are trained on sentences shorter than 41 tokens and 11 tokens, respectively, TVC-PCFG is our transfer learning model, and TC-PCFG is the transfer learning model that is trained without using visual groundings. The squares indicate the average length of the test sentences of each target domain.

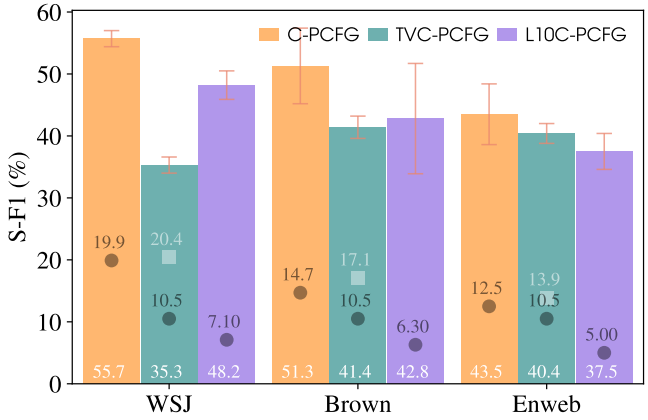



Figure 3.7: C-PCFG is trained on sentences shorter than 41 tokens, L10C-PCFG is trained on sentences shorter than 11 tokens, and TVC-PCFG is our transfer learning model. The circles represent the average length of the sentences for training each model; the squares indicate the average length of the test sentences of each domain.

Regarding model performance on subdomains, we observe similar trends as we see on mixed domains. Specifically, on the subdomains of both Brown and Enweb, C-PCFG performs best, and TVC-PCFG outperforms TC-PCFG (see Figure 3.6a).

**Remote-domain training is helpful.** Since the average lengths of WSJ, Brown, and Enweb training sentences are higher than that of MSCOCO training captions, to allow for fair comparison, for each target domain, we further train C-PCFG individually on the training sentences of the length below 10.5, the average length of MSCOCO training captions. We dub this model L10C-PCFG.

Surprisingly, though L10C-PCFG is individually trained only on the WSJ and Brown sentences that are shorter than 10.5 tokens, it surpasses TVC-PCFG by 12.9% and 1.4% S-F1, respectively (see Figure 3.7). On Enweb, while TVC-PCFG beats L10C-PCFG (+2.9% S-F1), it does not always outperform L10C-PCFG on every run, despite that the average length of the Enweb sentences used for training L10C-PCFG is only 5 (*cf.* 10.5 tokens).

Across the remote-domain test sets, we also observe that the longer the sentences are used for training C-PCFG, the better the performance is. For example, C-PCFG always surpasses L10C-PCFG, and interestingly, without considering the "dataset" variable, the improvement of L10C-PCFG over TVC-PCFG becomes larger as the average length of the sentences used for training L10C-PCFG increases: -2.9% < +1.4% < +12.9% S-F1 with 5.0 < 6.3 < 7.1 tokens for Enweb, Brown, and WSJ, respectively.

With regards to model performance on subdomains, again, we observe similar trends as we see on mixed domains (see Figure 3.6b). Specifically, on the subdomains of both Brown and Enweb, C-PCFG performs best, and TVC-PCFG underperforms L10C-PCFG on the subdomains of Brown but outperforms L10C-PCFG on the subdomains of Enweb.

## 3.6 Subsequent Work

Since our publication on VC-PCFG (Zhao and Titov, 2020), subsequent work has adapted our joint learning paradigm to tackle various challenging problems of unsupervised structure induction. Among them, a direct application of our learning paradigm is to induce syntactic structures from videos (Zhang et al., 2021), wherein, by analogy to image groundings, a variety of video features (e.g., object, action, scene, audio, speech, and OCR features) are extracted and used as the groundings of the aligned video captions. Apart from inducing phrase-structure grammars alone, Su et al. (2021) study visually grounded constituency and dependency grammar induction with a lexicalized PCFG parser (Zhu et al., 2020). Our learning paradigm has also been applied to joint image and text parsing. For example, Hong et al. (2021) treat visual artificial objects as sequences of constituent parts and extend our learning paradigm to jointly parse objects and text. Further, Wan et al. (2022) consider natural image parsing and jointly induce hierarchies of visual scenes and phrase structures of text. Notably, they use object-level groundings and replace compound PCFG with a more context-dependent parser (Drozdov et al., 2019). More recently, Lou et al. (2022) investigated graph-based representations of images and text; they jointly learn to parse images into scene graphs and text into dependency trees.

Compared with learning from language data alone, grounded language learning is believed to be important for better natural language understanding (Bisk et al., 2020; Bender and Koller, 2020). In efforts to promote grounded language learning, visual groundings have been empirically shown to be helpful for syntactic understanding, but it is still not entirely clear in which way and to which extent they help. Though there have been empirical studies attempting to answer these questions, e.g., both Kojima et al. (2020) and Zhao and Titov (2020) suggest that visual groundings help most with noun phrase induction, more research from both theoretical and empirical sides is needed, in order to investigate these problems thoroughly.

## 3.7 Summary

We have presented Visually-grounded Compound PCFG (VC-PCFG) that uses compound PCFG and generalizes the visually grounded grammar learning framework. VC-PCFG exploits visual groundings via contrastive learning, with learning signals derived from minimizing an image-text alignment loss. To tackle the issues of misleading and insufficient learning signals from purely agreement-based learning, we propose to complement the image-text alignment loss with a loss defined on unlabeled text. We resort to using compound PCFG which enables us to complement the alignment loss with a language modeling objective, resulting in a fully-differentiable end-to-end visually grounded learning. We empirically show that our VC-PCFG is superior to models that are trained only through visually grounded learning or only relying on text.

Further, we frame our proposed joint learning paradigm as a special case of multi-objective learning and connect it with semi-supervised learning. Specifically, we optimize the image-text alignment objective on text that is associated with images (*cf.* sentence labels) and the language modeling objective on pure text. We propose a simple approach that enables VC-PCFG to transfer to text domains beyond the training domain. Our approach relies on pre-trained word embeddings and does not require training on the target domain. We empirically find that VC-PCFG is able to transfer to similar image-caption domains such as Flickr but struggles to transfer to remote domains such as WSJ and Brown.

The major difference between our grammar-induction settings and conventional settings is that we consider learning cues from multimodal data beyond text, which constitutes an effort to favor grounded language learning. We focus specifically on visual groundings of text and propose a joint learning paradigm that allows for leveraging visual groundings for learning PCFG induction models. We note that it is also possible to use other types of groundings rather than images. To a large extent, our work suggests that general image-text pre-training and, specifically, the visual groundings of text help with the challenging syntactic-structure induction task. Conversely, we would like to investigate if image-structure induction benefits from textual groundings via image-text pre-training. In Chapter 4, we will formulate an unsupervised textually-grounded image-structure induction task. Differently from the neural-symbolic model, i.e., VC-PCFG, which explicitly models hidden structures via latent variables, we will devise a neural architecture that implicitly derives the induction of image structures.

# Chapter 4

# Textually Grounded Scene Graph Induction

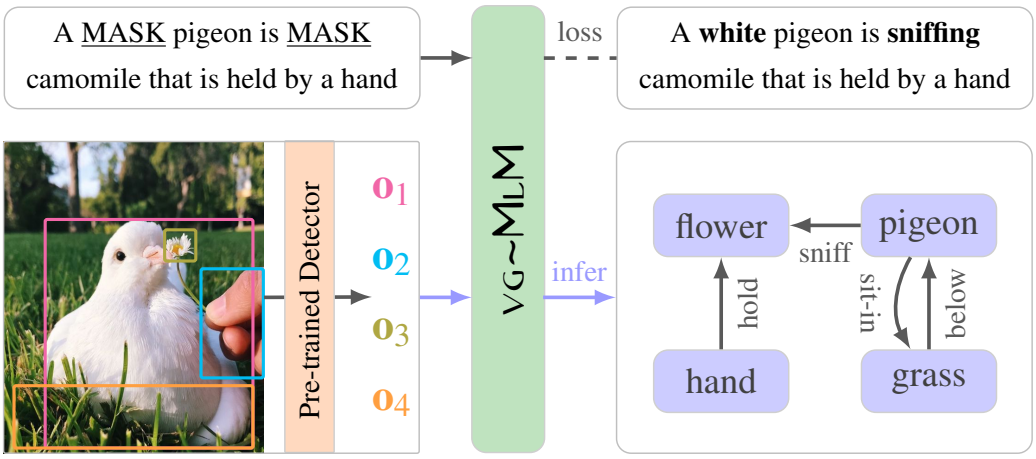



Figure 4.1: Visually Grounded Masked Language Model (VG~MLM) for unsupervised scene graph induction. VG~MLM is trained on image-text pairs and learns directly from natural language supervision. At test time, it infers scene graphs from images, without access to the aligned text.

In the previous chapter, we empirically found that visual groundings of text help with learning syntactic structures. Conversely, we would like to investigate if textual groundings of images help with learning structured image representations. In particular, we focus on inducing scene graph representations of images.

A scene graph represents relations between objects. Given an image, scene graph generation models detect objects (including localizing and classifying objects) and predict relations among them. The resulting scene graph representations abstract away low-level image features and represent image contents with high-level concepts that

are expressed in language (see Figure 4.1). There has been a substantial body of work showing that scene graph representations are useful in a variety of vision tasks, including image retrieval (Johnson et al., 2015; Schuster et al., 2015), image captioning (Elliott and Keller, 2013; Yang et al., 2019; Li and Jiang, 2019; Gu et al., 2019a), image synthesis (Johnson et al., 2018; Dhamo et al., 2020), and visual question answering (Shi et al., 2019b; Hudson and Manning, 2019b,a).

Most approaches to scene graph generation adopt a supervised learning paradigm and thus require scene graph annotations. Apart from the high cost of obtaining manual annotations, the annotated scene graphs tend to be biased. Specifically, the distribution of annotated relations is heavily uneven and many relations appear only a handful number of times in the annotated data (Krishna et al., 2017; Yao et al., 2021). Learning from biased data further leads to biased models. While weakly-supervised approaches have been studied, most of them mitigate only the annotation cost issue by using image-level labels, without requiring gold bounding boxes (Zhang et al., 2017b; Peyre et al., 2017), so the label bias issue remains. A possible solution to the issue is to use distant supervision, e.g., mining pseudo labels from large image-captioning datasets via an external parser (Schuster et al., 2015), but pipeline models are potentially error-prone (Zhong et al., 2021; Ye and Kovashka, 2021).

In this work, we propose to learn scene graph induction models by using free-form captions as direct supervision, without relying on linguistic preprocessing, e.g., converting captions into structured forms such as ⟨subject, predicate, object⟩ triplets. We are partly inspired by the recent phenomenal progress in learning visual representations from natural language supervision, which is usually in the form of image-text pairs. Image-text data is abundantly available on the web (e.g., online posts and tweets usually contain images and associated text) and can be curated via automatic tools, without requiring intensive human labor (Sharma et al., 2018). In the context of image-text pre-training, though continuous visual representations have been the main focus (Lu et al., 2019; Li et al., 2020b; Radford et al., 2021), it has recently been shown that more symbolic representations (e.g., via semantic segmentation and object detection) can also be learned (Xu et al., 2022; Geng et al., 2023). Motivated by this line of work, we focus on learning structured image representation (i.e., scene graphs) via image-text pre-training.

The challenges of inducing scene graphs from image-text pairs are two-fold: inducing object representations and aligning each object (pair) to a textual concept that is expressed as a word. Object representations are usually produced by using a pre-

trained object detector, but it is also possible to obtain them via unsupervised semantic segmentation (Burgess et al., 2019; Locatello et al., 2020). Given object representations, the next step[1] is to classify individual objects and assign relations to pairs of objects, i.e., aligning objects and object pairs to words that best describe them. In annotated scene graphs, two main clusters of words have been distinguished: entities for referring to objects and predicates for referring to visual relations. Since we learn directly from captions, we do not use prior knowledge about word clusters in learning and thus work in a more challenging setting.

We propose VG~MLM, short for Visually Grounded Masked Language Model, for object and relation classification. VG~MLM has an encoder-decoder architecture. Given object embeddings, the encoder produces a contextualized representation for each pair of objects. The decoder implements masked language modeling and conditions on the outputs of the encoder. We design a special computational mechanism such that (1) at training time, VG~MLM predicts target words conditioning on both visual and textual contexts; and (2) at test time, VG~MLM is able to make predictions for each object (pair), without access to captions (see Figure 4.2).

To study our model, we create CLEVR-TV, an artificial image-captioning dataset built off CLEVR (Johnson et al., 2017). CLEVR-TV consists of descriptions of relations among abstract 3D shapes. By using abstract objects, we are able to focus on visual relation induction in isolation. We predict an object category for each object and assign a relation to each pair of objects. We propose automatic evaluation metrics to quantify model performance in terms of object and relation classification accuracy. By experimenting with different methods for visual object encoding, we find that symbolic object representation is important for VG~MLM to achieve reasonable performance. Our experiments also suggest that VG~MLM is hard to optimize, e.g., VG~MLM is sensitive to hyperparameters such as the learning rate.

## 4.1 Related Work

**Visual Relationship Detection.** Visual relations capture interactions among objects in an image and are important for representing and understanding detailed visual semantics. Early approaches to modeling visual relations have focused on spatial relations, which are generally overly generic, e.g., “above”, “near”, and “around” (Gal-

[1] In principle, inducing object representations and aligning them to words can be formulated as multitask learning and learned jointly.

leguillos et al., 2008; Gould et al., 2008; Kulkarni et al., 2011). More complex visual relations have been studied in the literature of human-object interactions (Desai et al., 2010; Chao et al., 2015; Ramanathan et al., 2015), referential expression comprehension (Mao et al., 2016; Yu et al., 2016; Hu et al., 2017), and visual phrase detection (Sadeghi and Farhadi, 2011). To automatically induce diverse relations between arbitrary objects, previous work has formulated a more general task called visual relationship detection (VRD; (Lu et al., 2016)). VRD aims to predict triplets of the form ⟨subject, predicate, object⟩. It involves localizing pairs of objects (i.e., subject and object) as bounding boxes, labeling objects, and assigning a relation (i.e., predicate) to each pair. Differently from VRD, which targets pairwise relation detection, scene graph generation (SGG; (Xu et al., 2017)) is introduced to induce scene graphs as a whole. In essence, SGG is simply a redefinition of VRD, but from a modeling perspective, it is proposed to emphasize the aspect of reasoning with surrounding contexts (Xu et al., 2017), while previous approaches to VRD make predictions for each object pair independently (Lu et al., 2016).

**(Weakely) Supervised Scene Graph Generation.** Supervised approaches to scene graph generation have been dominant (Lu et al., 2016; Zhang et al., 2017a; Yu et al., 2017; Li et al., 2017; Newell and Deng, 2017; Yang et al., 2018; Li et al., 2018; Woo et al., 2018), presumably because of the availability of human-annotated scene graphs such as the Scene Graph dataset (Johnson et al., 2015), the Visual Relationship Detection dataset (Lu et al., 2016), and the Visual Genome dataset (Krishna et al., 2017). Among them, Visual Genome has been widely used, but it has been found that the scene graphs in Visual Genome have noisy and sparse annotations (Xu et al., 2017) and exhibit strong structural regularities (Zellers et al., 2018), presenting a great obstacle to learning reliable and generalizable supervised models. To tackle these challenges, previous work has resorted to different strategies, such as data refinement (Xu et al., 2017), multitask learning (Li et al., 2017), regularized learning with linguistic knowledge (Lu et al., 2016; Yu et al., 2017), exploiting correlations between relations and object labels (Dai et al., 2017a; Zellers et al., 2018; Chen et al., 2019), learning from commonsense knowledge (Gu et al., 2019b; Zareian et al., 2020a), and debiasing based on counterfactual analysis (Tang et al., 2020), to name but a few.

Despite the impressive development of supervised learning approaches, they are inherently limited due to the reliance on expensive human annotations. A popular strategy for mitigating the need for labeled data is to use weakly-supervised learning.

Unlike supervised approaches, which assume localized scene graphs and thus require bounding box annotations, most weakly-supervised methods relax the assumption by assuming unlocalized scene graphs (i.e., image-level object and relation labels) and thus avoid costly manually-annotated bounding boxes. To obtain object proposals, some of the prior weakly-supervised approaches rely on pre-trained detectors (Peyre et al., 2017; Baldassarre et al., 2020; Zareian et al., 2020b; Shi et al., 2021), and others jointly learn an object proposal module and a relation detector (Zhang et al., 2017b).

**Learning Visual Representations from Natural Language Supervision.** In the area of visual representation learning, a recent breakthrough is to learn general-purpose visual representations from natural language supervision (Lu et al., 2019; Zhou et al., 2020; Su et al., 2020; Li et al., 2020b). Natural language supervision has been primarily in the form of parallel images and text, which are abundant on the web and thus have led to the development of large-scale image-text pre-training (Radford et al., 2021; Jia et al., 2021). Apart from learning continuous visual representations, natural language supervision has also been shown to be helpful for learning more symbolic representations via, for example, object detection (Sadeghi and Farhadi, 2011; Kamath et al., 2021) and segmentation (Li et al., 2022a; Xu et al., 2022). But little work has been carried out to learn more complex structured visual representations (e.g., scene graphs) from natural language supervision. Those that have used image-text pairs in scene graph generation usually require preprocessing text. For example, Yao et al. (2021) and Zhong et al. (2021) use a rule-based parser (Schuster et al., 2015) to extract ⟨subject, predicate, object⟩ triplets from image captions, and curate pseudo labels by aligning the noisy triplets with gold object annotations. Ye and Kovashka (2021) use the same parser to parse captions into graph-based semantic structures and use them as the source of supervision. While we also use image captions, we directly learn from them, without converting them into structured forms.

## 4.2 Problem Statement

**Scene Graph.** A scene graph is a directed graph where each node represents a visual object and each directed edge represents the relation between an object pair (see Figure 4.1). Formally, we define a scene graph as a 3-tuple $\mathcal{G} = (O, \mathcal{P}, \mathcal{R})$, where

- $O$ is a finite set of objects in a given image. Each object is labeled with a category such as “pigeon”, “grass”, and “flower”, and is associated with a bounding box.

There might be multiple objects labeled with the same category in the given image (e.g., "man.01" and "man.02"); they can be distinguished from each other by their bounding boxes;

- $\mathcal{P}$ is a finite set of predicates such as "sniff", "under", and "sit-in". The special predicate "null" $\in \mathcal{P}$ indicates no relation;

- $R$ is a finite set of triplets in the form of $(o, p, o')$, where $o, o' \in O$ and $p \in \mathcal{P}$. Each triplet indicates that one object $o$ is related to the other object $o'$ via the relation $p$. For example, (pigeon, sniff, flower).

**Problem Formulation.** Assuming a dataset $\mathcal{D} = \{(v^{(i)}, t^{(i)}) | 1 \leq i \leq N\}$ consisting of $N$ pairs of image $v$ and caption $t$, our goal is to learn a scene graph induction model from $D$. We consider a novel unsupervised learning setting and contrast it with previous learning settings in the following two important respects:

- We provide object representations. To obtain object representations, we assume that a pre-trained object detector is available. The assumption is practical because object detection has been widely studied (Girshick, 2015; Ren et al., 2015; He et al., 2017) and there are off-the-shelf performant detectors (Wu et al., 2019). Most weakly-supervised learning approaches also use a pre-trained detector, but some of them use extra object label distributions predicted by the detector (Peyre et al., 2017; Baldassarre et al., 2020), while we do not;

- We do not use image-level object labels and relation labels. Image-level captions are the only source of supervision. Unlike previous work, which uses unlocalized gold scene graphs or parses captions to create image-level pseudo labels, we learn directly from captions. At inference time, a model should make predictions (object and relation classification) conditioning on only images, without access to the aligned captions.

## 4.3 Scene Graph Induction Model

### 4.3.1 Visually Grounded Masked Language Model

Conceptually, a scene graph induction model predicts object categories for individual objects and assigns relations to object pairs, so it is desirable to have two separate sets

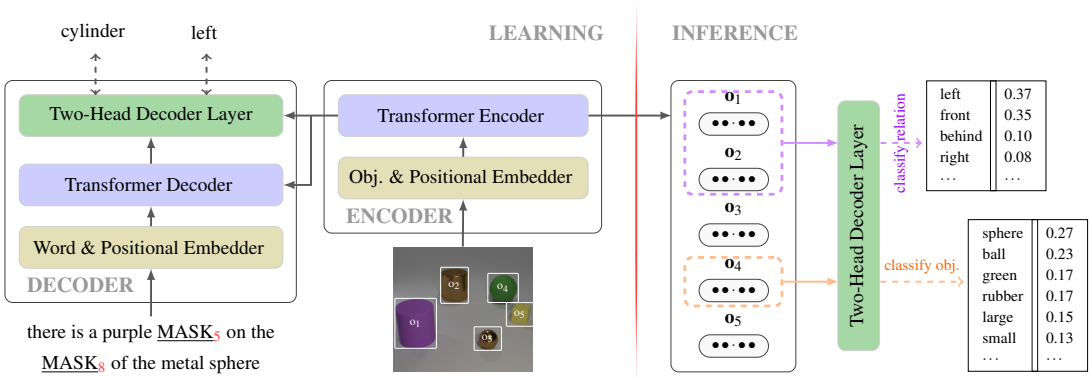



Figure 4.2: Visually Grounded Masked Language Model (VG~MLM). **Left:** VG~MLM has an encoder-decoder architecture. We customize a two-head Transformer decoder layer (see Figure 4.3), which stacks above the standard Transformer decoder. **Right:** At inference time, VG~MLM classifies individual objects and assigns relations to individual pairs, without access to the aligned text.

of labels for objects and relations, respectively. But, in our setting, we aim to learn directly from captions, so we assume that all the labels are contained in captions,[2] but we do not use the prior knowledge about these labels during learning. While this assumption poses a challenge for inference, it leads to the same setting as that used for image-text pre-training and allows for tapping into a large body of work in that area. Specifically, we draw inspiration from masked multimodal learning (Lu et al., 2019; Zhou et al., 2020). Observing that the object and relation classification can be formulated as predicting a target word given certain visual objects, we propose VG~MLM[3] for unsupervised scene graph induction.

Though both encoder-decoder and decoder-only architectures would suffice, we adopt a Transformer encoder-decoder architecture because separating the encoder from the decoder lets us inject different architectural biases into them. As we will expand on in Section 4.3.2, these architectural biases are the key to enabling classify-

[2]For example, the caption "the pigeon sniffs the flower" contains two object labels "pigeon" and "flower", and a relation label "sniffs". At inference time, we may post-process inferred labels and use their base forms, e.g., "sniffs" will be replaced by "sniff".

[3]Visual-image-conditioned *causal* language modeling is another option, but the left-side contexts of a token do not necessarily contain all the relevant information needed for predicting the token, especially for English that tends to have a subject-verb-object word order, while masked language modeling does not have this limitation. Take the caption "the pigeon sniffs the flower", when predicting "sniffs", it is desirable to know both the left-side entity "pigeon" and the right-side entity "flower" because this not only narrows down possible targets but also guides the model to attend to relevant visual objects, i.e., visual "pigeon" and "flower" segments. Nevertheless, it is possible to jointly perform masked language modeling and causal language modeling as in Zhou et al. (2020).

ing objects and relations conditioning on solely visual objects. We learn VG~MLM by optimizing a masked language modeling objective. Formally, given a training set $\mathcal{D} = \{(v^{(i)}, t^{(i)}) | 1 \leq i \leq N\}$, we maximize the following conditional log-likelihood:

$$\mathcal{L}(\mathcal{D}; \theta) = \sum_{i=1}^{N} \sum_{j \in \mathcal{M}^{(i)}} \log p(t_j^{(i)} | v^{(i)}, t_{\_\mathcal{M}}^{(i)}; \theta), \tag{4.1}$$

where $\mathcal{M}$ represents a set of random token indices with $\max(\mathcal{M}) \leq |t|$ (the length of the caption $t$), $t_j$ denotes $j$-th token of $t$, and $t_{\_\mathcal{M}}$ indicates the caption with $t_j$ (for every $j \in \mathcal{M}$) masked out, i.e., replaced by a special symbol "MASK". For example, if the caption $t$ is "the pigeon sniffs the flower" and $\mathcal{M} = \{2,4\}$, $t_{\_\mathcal{M}}$ will be "the MASK sniffs MASK flower". Intuitively, VG~MLM is trained to predict a target token $t_j$ conditioning on both visual contexts $v$, which are encoded by the encoder, and textual contexts $t_{\_\mathcal{M}}$, which are encoded by the decoder.

### 4.3.2 VG-MLM for Scene Graph Induction

One of our goals is to infer the most probable relation for a pair of objects, without access to captions. Suppose an $L$-layer Transformer encoder outputs $n_o$ contextualized object representations $\mathbf{o}_1^{L+1}, \mathbf{o}_2^{L+1}, \ldots, \mathbf{o}_{n_o}^{L+1} \in \mathbb{R}^{d_m}$. At inference time, we solve the following task:

$$\arg\max_r p(r | \mathbf{o}_i^{L+1}, \mathbf{o}_j^{L+1}; \theta). \tag{4.2}$$

But, during training, the only assumption we have made about VG~MLM is that it predicts a target word conditioning on both visual contexts (i.e., objects) and textual contexts. Let us assume a single random token with the index $k$ in a caption $t$ is masked out, the learning task with a single image-text pair is formalized as:

$$\arg\max_\theta \log p(t_k | t_{\_k}, \mathbf{o}_{1:n_o}^{L+1}; \theta), \tag{4.3}$$

where $t_{\_k}$ indicates the caption with the $k$-th token masked out, similarly to $t_{\_\mathcal{M}}$.

Here the problem is that the inference model is inconsistent with the model defined by Equation 4.3. To solve the problem, we tailor VG~MLM to make it capable of (1) inferring a distribution over relations rather than over the whole vocabulary, (2) making inferences conditioning on individual pairs of objects rather than on all the individual objects, and (3) inferring relation distributions conditioning on only visual objects rather than on both visual objects and textual contexts. Below we expand on our solutions to achieving these goals at inference time.

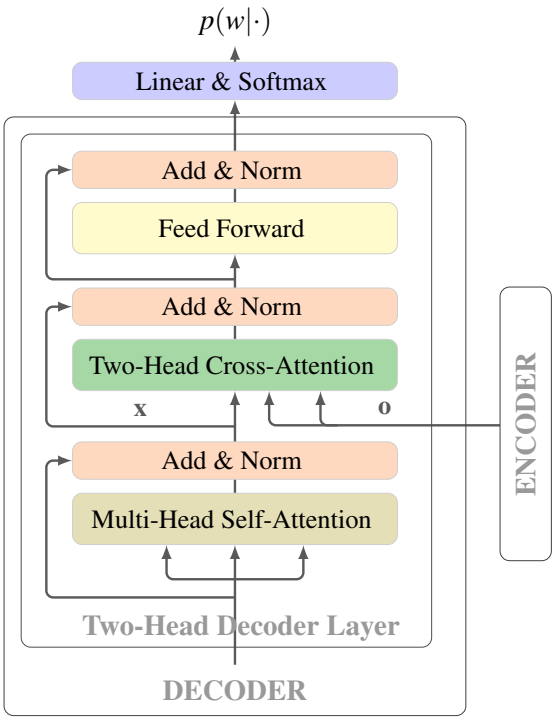



Figure 4.3: The two-head decoder layer. It is the same as the standard Transformer decoder layer except that we replace the standard cross-attention module with our customized two-head cross-attention module (see Figure 4.4).

**Inferring Relation Distributions.** We assume that the vocabulary subsumes all the object and relation labels. But, since we do not distinguish them during learning, VG~MLM always predicts distributions over the whole vocabulary. At inference time, to focus on a specific set of words, e.g., relation words in relation classification, we simply reset the logits that correspond to non-relation words to "$-\inf$", i.e., ignoring all the non-relation words. In doing so, we need to identify all relation words in the vocabulary. Practically, we detect all the predicates and treat them as relation words.

**Conditioning on Object Pairs.** To keep consistent with the inference setting, where VG~MLM predicts relations conditioning on individual pairs of objects, we first construct VG~MLM to condition on all the individual pairs of objects during training. A simple way to represent object pairs is to concatenate the vector representations of the two objects for each pair. But, usually, not all the pairs are equally predictive of a target word, so it is desirable to prioritize object pairs by assigning a predictiveness score to each pair. Moreover, since the semantic roles of two objects determine the relations

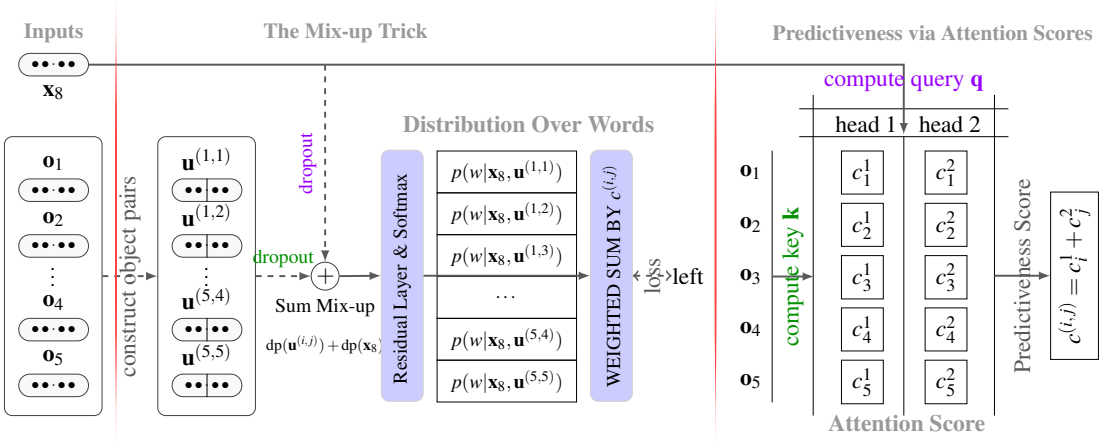



Figure 4.4: The two-head cross-attention module takes a contextualized word representation $\mathbf{x}_8$ and contextualized object representations $\mathbf{o}_{1:5}$ as the inputs. It uses the $\mathbf{x}$ to compute the query and all $\mathbf{o}$'s to compute the key and value (see Equation 4.4). Unlike the standard attention mechanism, which summarizes all the individual objects as a single vector, our module uses object pairs $\mathbf{u}^{(i,j)}$ (the concatenation of the representations of two objects $i$ and $j$ (see Equation 4.6)). Each pair $(i, j)$ is associated with a predictiveness score $c^{(i,j)}$ and independently performs a prediction $p(w|\mathbf{x}_8, \mathbf{u}^{(i,j)})$, we instead summarize the predictions by using the predictiveness scores (see Equation 4.8). The final part of the diagram illustrates how we compute predictiveness scores from the attention scores of the two attention heads (see Equation 4.6).

between them,[4] it is desirable to learn two sets of object representations to indicate the "subject" role and the "object" role, respectively.

We realize all the desiderata by using a two-head cross-attention mechanism. Following the standard attention mechanism, each attention head computes query $\mathbf{q}$, key $\mathbf{k}$, and value $\mathbf{v}$:

$$\mathbf{q}_k = \mathbf{W}^Q \mathbf{x}_k\,, \quad \mathbf{k}_i = \mathbf{W}^K \mathbf{o}_i^{L+1}\,, \quad \mathbf{v}_i = \mathbf{W}^V \mathbf{o}_i^{L+1}\,, \tag{4.4}$$

where $\mathbf{x}_k \in \mathbb{R}^{d_m}$ is output by the self-attention module of the two-head decoder layer and indicates the contextualized representation at the position $k$ of $t_{-k}$ (see Figure 4.3). $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{d_h \times d_m}$ (with $d_h = \frac{d_m}{2}$) are learnable parameters and head-specific. Given $\mathbf{q}$ and $\mathbf{k}$, the attention scores are computed as:

$$c_i = \frac{\mathbf{q}_k{}^T \cdot \mathbf{k}_i}{\sqrt{d_h}} \quad \text{with } 1 \le i \le n_o\,. \tag{4.5}$$

[4]For example, assuming "cube" is the subject, and "sphere" is the object, and the relation between them is "in-front-of", switching the roles of the two objects will change the relation into "behind".

We use the value representations $\{\mathbf{v}_i^s\}$ from one head as the "subject" representations of objects, and the attention scores $\{c_i^s\}$ as the confidence of assigning the "subject" role to the corresponding objects. From the other head, we obtain the "object" representations $\{\mathbf{v}_j^o\}$ and the associated attention scores $\{c_j^o\}$. Then the representation $\mathbf{u}^{(i,j)}$ of an object pair $(i,j)$ and the associated predictiveness score $c^{(i,j)}$ are given by:

$$\mathbf{u}^{(i,j)} = [\mathbf{v}_i^s : \mathbf{v}_j^o], \quad c^{(i,j)} = c_i^s + c_j^o, \tag{4.6}$$

where $i, j \in [1, n_o]$ and $[:]$ indicates vector concatenation.[5]

Our decoder architecture follows that of the standard Transformer decoder except that, in the final layer, we replace the standard cross-attention mechanism with our customized two-head cross-attention mechanism (see Figure 4.2 and 4.3). Intuitively, we are implicitly assuming that: given the textual contexts of $t_k$ and visual contexts $\mathbf{o}_{1:n_o}^{L+1}$, the model should be able to infer object pairs that are most predictive of $t_k$ and assign higher scores to them.

**Decoupling Object Pairs.** Given representations of object pairs $\{\mathbf{u}^{(i,j)}\}$ and the associated predictiveness scores $\{c^{(i,j)}\}$ $(1 \leq i, j \leq n_o)$, by analogy to the standard attention mechanism, we would summarize visual contexts by averaging $\{\mathbf{u}^{(i,j)}\}$ according to the normalized predictiveness scores: $\hat{c}^{(i,j)} = \exp\left(c^{(i,j)}\right) / \sum_{i,j} \exp\left(c^{(i,j)}\right)$, merge textual contexts $\mathbf{x}_k$ with the summarized visual contexts, and infer a distribution over the vocabulary. Formally,

$$p(w|\mathbf{x}_k, \mathbf{o}_{1:n_o}^{L+1}; \theta) = h\left(\mathbf{x}_k + \sum_{i,j} \hat{c}^{(i,j)} \cdot \mathbf{u}^{(i,j)}\right), \tag{4.7}$$

where $h(\cdot)$ is implemented as a residual layer followed by the softmax activation function. But this couples all the pairs of objects. A simple solution to this issue is to move the sum operator outside of $h(\cdot)$:

$$p(w|\mathbf{x}_k, \mathbf{o}_{1:n_o}^{L+1}; \theta) = \sum_{i,j} \hat{c}^{(i,j)} \cdot h\left(\mathbf{x}_k + \mathbf{u}^{(i,j)}\right). \tag{4.8}$$

Intuitively, for each pair $\mathbf{u}^{(i,j)}$, we merge it with the textual contexts $\mathbf{x}_k$ and infer a distribution over the vocabulary, then we average all the inferred distributions according to the normalized predictiveness scores $\{\hat{c}^{(i,j)}\}$.

[5] Note that $i = j$ implies that the pair is composed of an object and its replication, so it is equivalent to an individual object. This will be useful for object labeling, which is conditioned on individual objects.

**Conditioning on Only Visual Objects.** At inference time, Equation 4.8 enables inference conditioning on a given pair of objects:

$$p(w|\mathbf{x}_k, \mathbf{o}_i^{L+1}, \mathbf{o}_j^{L+1}; \theta) = \hat{c}^{(i,j)} \cdot h\left(\mathbf{x}_k + \mathbf{u}^{(i,j)}\right), \tag{4.9}$$

but still, it relies on the textual contexts $\mathbf{x}_k$ in two different ways: (1) for the outer scalar $\hat{c}^{(i,j)}$, which is computed by using $\mathbf{x}_k$, we can simply drop it; and (2) for the inner $\mathbf{x}_k$, since it is merged with the pair via addition, we can also drop it. This leads to an inference procedure that is not conditioned on textual contexts:

$$p(w|\mathbf{o}_i^{L+1}, \mathbf{o}_j^{L+1}; \theta) = h\left(\mathbf{u}^{(i,j)}\right). \tag{4.10}$$

But the textual contexts of $t_k$, which are encoded in $\mathbf{x}_k$, are predictive of $t_k$ in general, dropping $\mathbf{x}_k$ at inference time is likely to lead to a less accurate estimate of the word distribution for a given pair. A possible strategy for retaining the informative textual contexts encoded in $\mathbf{x}_k$ is to distill them into the representations of object pairs. Specifically, we randomly mix-up $\mathbf{x}_k$ and $\mathbf{u}^{(i,j)}$ during training:[6]

$$p(w|\mathbf{x}_k, \mathbf{o}_{1:n_o}^{L+1}; \theta) = \sum_{i,j} \hat{c}^{(i,j)} \cdot h\left(f_t^{dp}(\mathbf{x}_k)\mathbb{1}_{[\text{train}]} + f_v^{dp}(\mathbf{u}^{(i,j)})\right), \tag{4.11}$$

where $f_t^{dp}(\cdot)$ and $f_v^{dp}(\cdot)$ are dropout functions applied to textual contexts and visual contexts, respectively. $\mathbb{1}_{[\text{train}]}$ is an indicator function and evaluates to 1 only at training time. Intuitively, when part of $\mathbf{x}_k$ that is predictive of a target is masked out, to maintain accurate predictions, $\mathbf{u}$ has to fill in the missing part.

### 4.3.3 Encoding Objects

We have so far discussed the decoder of VG~MLM, and specifically, the tailored cross-attention mechanism, which relies on contextualized object representations output by the encoder of VG~MLM. In this section, we elaborate on the encoder (Section 4.3.3.1), and describe ways of representing object positions (Section 4.3.3.2) and encoding symbolic visual objects (Section 4.3.3.3).

#### 4.3.3.1 Contextualized Object Representations

Suppose $n_o$ objects are initially embedded as $d_o$-dimensional continuous vectors denoted by $\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_{n_o}$, respectively, and each object is associated with a positional

[6] We also tried the manifold mixup technique (Zhang et al., 2018; Verma et al., 2019) but did not observe improvements.

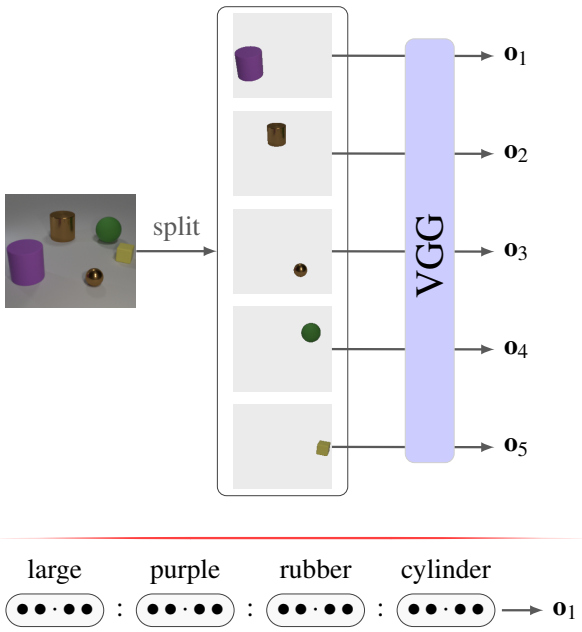



Figure 4.5: Object embedder. The above illustrates visual object representations via a pre-trained VGG model. Below is an example of symbolic object representations (":" indicates vector concatenation).

embedding $\mathbf{e} \in \mathbb{R}^{d_b}$. For each object, we concatenate $\mathbf{o}_i$ and $\mathbf{e}_i$, then we apply a linear map $f : \mathbb{R}^{d_o+d_e} \rightarrow \mathbb{R}^{d_m}$ and input the resultant object representations to an $L$-layer Transformer encoder. Following the multi-head attention mechanism, in the $l$-th Transformer layer, given an object representation $\mathbf{o}_i^l \in \mathbb{R}^{d_m}$ and for every $\mathbf{o}_j^l \in \mathbb{R}^{d_m}$ $(i, j \in [1, n_o])$, an attention head estimates the importance of $\mathrm{o}_j^l$ to $\mathrm{o}_i^l$ as:

$$s_{i,j} = \frac{\mathbf{q}_i^T \cdot \mathbf{k}_j}{\sqrt{d_h}} \quad \text{with} \quad \mathbf{q}_i = \mathbf{W}_l^Q \mathbf{o}_i^l, \mathbf{k}_j = \mathbf{W}_l^K \mathbf{o}_j^l, \tag{4.12}$$

from which the $i$-th context-aware object representation is computed as:

$$\mathbf{o}_i^{l+1} = \sum_{j=1}^{n_o} \hat{s}_{i,j} \cdot \mathbf{v}_j^l \quad \text{with} \quad \mathbf{v}_j^l = \mathbf{W}_l^V \mathbf{o}_j^l, \quad \hat{s}_{i,j} = \frac{\exp(s_{i,j})}{\sum_{j=1}^{n_o} \exp(s_{i,j})}. \tag{4.13}$$

In the above formulations, we have used head- and layer-specific learnable parameters $\mathbf{W}_l^Q, \mathbf{W}_l^K, \mathbf{W}_l^V \in \mathbb{R}^{d_h \times d_m}$ to transform object embeddings into query $\mathbf{q}$, key $\mathbf{k}$, and value $\mathbf{v}$. Suppose there are $n_h$ heads in a Transformer encoder layer and $d_h = d_m/n_h$, the Transformer encoder layer will output $\mathbf{o}_i^{l+1} = [\mathbf{o}_{i,1}^l; \mathbf{o}_{i,2}^l; \ldots; \mathbf{o}_{i,n_h}^l]$, which is the concatenation of the $i$-th object representations from the $n_h$ heads. A Transformer encoder may have multiple layers. In this case, the outputs $\mathbf{o}^{l+1}$ from the last layer $l$ will be input to the next layer.

#### 4.3.3.2 Positional Representations

We represent object positions as normalized bounding boxes, which are 4-dimensional vectors, e.g., $(x_1/W, y_1/H, x_2/W, y_2/H)$, where $(W, H)$ is the image size of the form (width, height), and $(x_1, y_1)$, $(x_2, y_2)$ are the upper left and bottom right coordinates of a bounding box, respectively. Each 4-dimensional vector is further transformed into $d_e$-dimensional positional embedding $\mathbf{e}$ via a learnable linear map: $f : \mathbb{R}^4 \rightarrow \mathbb{R}^{d_b}$.

#### 4.3.3.3 Object Representations

An important concept of scene graphs is symbolic object modeling, i.e., we abstract away detailed visual features of objects and represent them as symbolic units, i.e., object labels such as "dog" and "bird". Following the common practice, we would use a pre-trained detector to encode visual objects as continuous representations, while generally performant in terms of detection accuracy, since the detected bounding boxes usually do not contain exact objects, extracting object features from rectangular regions inevitably results in noisy object representations. Moreover, since an object usually has different appearances in different images, the extracted visual representations are generally specific to an image. Thus, object embeddings obtained from a detector defy the reusability concept of symbolic representations. To study our model, we instead consider two alternatives for object embedding. These alternatives are compatible with CLEVR images (Johnson et al., 2017), an artificial image set we use for model study.

**Symbolic Object Representations.** Prior to training with visual object representations, we also consider a simpler setting: learning VG~MLM with symbolic object representations. To this end, we represent objects in such a way that symbolic object representations are ensured. Our idea is to represent an object with four attribute values since each object can be characterized by four attributes. Specifically, we create an object template "⟨size⟩ ⟨color⟩ ⟨material⟩ ⟨shape⟩", and substituting the attribute variables with the attribute values of an object gives rise to a symbolic object representation, e.g., "small red rubber sphere". To encode objects into continuous representations, we learn a finite set of attribute value embeddings. These embeddings are shared across objects and images and thus meet the goal of reusability. We concatenate the embeddings of the four attribute values of an object to obtain its symbolic representation. By using symbolic object representations, we essentially obtain an upper bound on the performance of our model.

| Attribute | Value |
|---|---|
| Shape | cylinder, sphere, cube |
| Size | large, small |
| Material | metal, rubber |
| Color | gray, red, blue, cyan, green, brown, purple, yellow |
| Relationship | front, behind, right, left |

Table 4.1: Object attributes and relationships in CLEVR-TV.

**Visual Object Representations.** Visual object representations encoded by a pre-trained detector are inevitably noisy. One possible way to reduce the noise is to use an object segmenter. A segmenter produces object regions that roughly encapsulate exact objects and contain less noisy pixels than bounding boxes. For CLEVR images, we can actually obtain gold object segmentation; encoding each object segment via a pre-trained image encoder presumably gives rise to less noisy object embeddings. Practically, for each object segment, we first create a canvas that has the same size as the original image, then we copy the object to the canvas and ensure that it is in the same position as in the original image. Finally, we use a pre-trained image encoder to encode individual objects (see Figure 4.5).

## 4.4 CLEVR-TV: An Image-Captioning Dataset

Prior to applying VG~MLM to natural images, we would like to learn and test it on artificial data, which helps validate the effectiveness of our model design. In doing so, we propose CLEVR-TV, an artificial dataset for learning scene graph induction models from language supervision. CLEVR-TV consists of image-text pairs and builds upon CLEVR, a diagnostic dataset for evaluating visual reasoning capabilities of visual question-answering systems (Johnson et al., 2017). The text in CLEVR-TV describes relations between visual objects. The images in CLEVR-TV are composed of abstract 3D shapes. By focusing on abstract objects, we try to isolate relational reasoning from visual regularities, e.g., the co-occurrence of two objects "man" and "horse" is likely to entail the "riding" relation, while abstract objects minimize regularities of this kind.

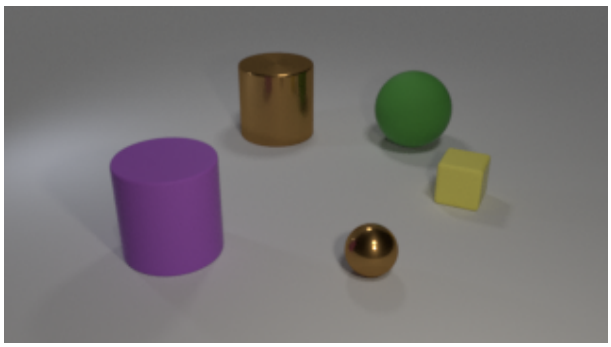


Figure 4.6: An example CLEVR-TV image. We only consider captions that describe the relations between two objects, e.g., "*there is a cylinder on the left of the green ball*".

### 4.4.1 Image Generation

Following CLEVR (Johnson et al., 2017), we render images from randomly sampled scene graphs by using Blender (Blender Online Community, 2016). A scene graph represents objects as nodes and relations as edges. Each object is annotated with shape, size, material, and color and is related to other objects via four spatial relations, i.e., "front", "behind", "left", and "right" (see Table 4.1). A scene graph contains all the information necessary for rendering an image.

### 4.4.2 Caption Generation

We are interested in automatically generating diverse relational descriptions. In doing so, we draw inspiration from CLEVR and generate captions from the corresponding functional programs, which can be executed on scene graphs. A functional program is composed of elementary building blocks such as *counting*, *querying*, and *comparing* functions. Suppose the following functional program that requires a *shape* variable $\langle \mathrm{S} \rangle$ as the input and produces a *shape* output $\langle \mathrm{A} \rangle$:

$$\langle \mathrm{A} \rangle := \mathrm{query\_shape}(\mathrm{left\_of}(\mathrm{filter\_shape}(\langle \mathrm{S} \rangle, \mathrm{scene}()))),$$

where scene() returns the scene graph representation of an image, and the elementary functions filter_shape(), left_of(), and query_shape() return a list of objects. There are multiple ways of instantiating the program such as (1) "*there is a* $\langle A \rangle$ *on the left of the* $\langle S \rangle$", and (2) "*there is a* $\langle S \rangle$*; the* $\langle A \rangle$ *is on the left of it*". These instantiations are called caption templates. To generate captions, we simply replace the variables $\langle \mathrm{S} \rangle$ and $\langle \mathrm{A} \rangle$ with valid assignments. For example, the two caption templates may lead to the following captions: "*there is a cylinder on the left of the cube*" and "*there is a cube; a*

*cylinder is on the left of it*".[7]

We generate captions by roughly following the same procedure as was used for generating CLEVR question-answer pairs. Specifically, given the gold scene graph of an image, we select a functional program and execute it on the scene graph to obtain groups of valid variable assignments (e.g., $\langle \mathrm{S} \rangle := \textit{cube}$ and $\langle \mathrm{A} \rangle := \textit{cylinder}$ in the aforementioned captions are a group of valid assignments). To generate captions, we randomly select a group of assignments and a caption template that corresponds to the program, then we substitute the variables in the caption template with the assignments. To increase caption diversity, CLEVR defines a set of synonyms for some attribute values, e.g., "metal" is associated with {metallic, metal, shiny}, and the assignment bound to a variable will be randomly replaced by one of its synonyms.

## 4.5 Experiments

### 4.5.1 Evaluation Metrics

The goal of scene graph induction models is to label individual objects and assign a relation to each object pair, thus, in evaluation, we are interested in precisions of object labeling and relation prediction.

#### 4.5.1.1 Object Labeling

At inference time, VG~MLM requires an object pair as the input (see Equation 4.10). To enable inference conditioning on individual objects, we create an object pair by concatenating the representations of an object and its copy.

We compute per-attribute precision. Conceptually, for a given attribute $A$, we only focus on the distribution (i.e., $\mathbf{s}^A$) of its admissible assignments (i.e., $\mathcal{H}(A)$) and find the most probable assignment from the admissible assignments ($\arg\max_k \mathbf{s}^A_k$). Since an attribute value can be described in different ways (e.g., "metal" can be described as "metal", "metallic", and "shiny"), we count it as a correct prediction as long as the prediction is one of the synonyms (i.e., $\hat{A}(\mathrm{o})$) of the value of the attribute $A$. For example, suppose the material of an object is "metal", a prediction "metallic" is considered correct because it is a synonym of "metal".

Formally, for each attribute $A \in \{\text{shape}, \text{size}, \text{material}, \text{color}\}$, we denote the set of values that can be assigned to $A$ by $\mathcal{H}(A)$ (e.g., $\mathcal{H}(A) = \{\text{large}, \text{tiny}, \text{big}, \text{small}\}$ with

[7] https://github.com/zhaoyanpeng/clevr-ed.

$A =$ "shape"). Given an object, suppose the logits (i.e., unnormalized log probabilities) of the inferred categorical distribution over the vocabulary $\mathcal{V}$ is $\mathbf{s} \in \mathbb{R}^{|\mathcal{V}|}$. For a given attribute $A$, we focus on only its valid assignments $\mathcal{H}(A)$, so we reset the logits that correspond to the words that are not in $\mathcal{H}(A)$ to "$-\inf$", and indicate the resultant logits as $\mathbf{s}^A$. Then the attribute-specific precision over $N$ objects is computed as:

$$p_A^{\text{same}} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{[\arg\max_k \mathbf{s}_k^A \in \hat{A}(\mathrm{o}_i)]}\,, \tag{4.14}$$

where $\hat{A}(\mathrm{o})$ accepts an input object o and returns the indices of all the synonyms of the value of the object's attribute $A$. For example, suppose $A =$ "material" and an object's *material* is "metal", $\hat{A}(\mathrm{o})$ will return the indices of "metal", "metallic", and "shiny", which are the synonyms of "metal".

We can further generalize the metric to any pair of objects $\mathrm{o}_i$ and $\mathrm{o}_j$:

$$p_A = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \mathbb{1}_{[\arg\max_k \mathbf{s}_k^A \in \hat{A}(\mathrm{o}_i) \cup \hat{A}(\mathrm{o}_j)]}\,. \tag{4.15}$$

#### 4.5.1.2 Relation Prediction

We compute the precision of relation prediction in a similar way as we compute the precision of attribute prediction. Note that the four spatial relations in CLEVR-TV are composed of two pairs of opposite relations: (left, right) and (front, behind). Given an object pair, only one relation is valid when considering two opposite relations (e.g., either "left" or "right"). But, if we consider the four relations together, there will be two valid relations, which come from the two pairs, respectively. For example, an object can be in *front* and to the *left* of another object at the same time, i.e., there are two gold relation labels for an object pair, but we only predict a single label, which is the most probable label. To resolve this issue, we compute precisions for the two pairs of relations, respectively.

A remaining problem is that relations are sensitive to the roles of the participating objects. When developing our model, we specifically assume the first object and the second object are assigned the "subject" role and the "object" role, respectively (see Equation 4.6), but since we are working with unsupervised learning, a learned model may switch the assumed role assignments and reverse the relation, i.e., the model may assign the "subject" role to the second object and the "object" role to the first object. Consequently, the relation between the two objects will also be reversed, e.g.,

$(o_1^s, \text{left\_of}, o_2^0) \Leftrightarrow (o_2^s, \text{right\_of}, o_1^o)$. Thus, we need to consider both cases in evaluation. Specifically, we first hypothesize default role assignments (defined by Equation 4.6), then we compute a precision $p_R^{\text{null}}$ provided that the hypothesis is true and a precision $p_R^{\text{reject}}$ given that the hypothesis is false:

$$p_R^{\text{null}} = \frac{1}{|\mathcal{H}(R)|} \sum_{r \in \mathcal{H}(R)} \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \mathbb{1}_{[\arg\max_k \mathbf{s}_k^R \in \hat{R}(r)]}\,, \tag{4.16}$$

$$p_R^{\text{reject}} = \frac{1}{|\mathcal{H}(R)|} \sum_{r \in \mathcal{H}(R)} \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \mathbb{1}_{[\arg\max_k \mathbf{s}_k^R \in \hat{R}(\bar{r})]}\,, \tag{4.17}$$

where $R \in \{\text{LR}, \text{FB}\}$ with $\mathcal{H}(\text{LR}) = \{\text{left}, \text{right}\}$ and $\mathcal{H}(\text{FB}) = \{\text{front}, \text{behind}\}$, $r \in \mathcal{H}(R)$ and $\bar{r}$ is opposite to $r$ (e.g., $r =$ "left" and $\bar{r} =$ "right"), and $\hat{R}(r)$ returns the indices of the synonyms of $r$. Finally, the precision for a pair of opposite relations is computed as:

$$P_R = \max\{P_R^{\text{null}}, P_R^{\text{reject}}\}\,. \tag{4.18}$$

Intuitively, for a given pair $(i, j)$ with the gold label $r$, the model should predict $r$ if the hypothesis is true; otherwise, it should predict the opposite relation $\bar{r}$ because switching role assignments reverses the relation. For each case, we estimate the model's prediction precision and use the higher precision as the quantification of model performance.

### 4.5.2 Datasets and Baselines

**Dataset.** The training set of CLEVR-TV consists of 32102 images and 96064 captions, and the test set of CLEVR-TV consists of 1000 images. The distribution of the four relations, i.e., "left", "right", "front", and "behind", is roughly uniform.

**Baseline.** Since our evaluation metrics are conditioned on attributes/relations, we consider a baseline model that relies on conditional sampling. Specifically, given a pair of objects, for each attribute, we randomly sample one of the values that can be assigned to the attribute; for each pair of opposite relations, we randomly sample a relation from the pair of opposite relations.

### 4.5.3 Settings and Hyperparameters

**Standard Transformer Encoder Layers.** The encoder and the decoder have 1 and 2 standard Transformer encoder layers, respectively. We set the number of attention

heads $n_h = 4$ and the input feature dimension $d_m = 512$. We use the GeLU activation function (Hendrycks and Gimpel, 2016) and disable dropout in the standard Transformer encoder layers.[8]

**Decoder Inputs.** Both word embeddings and learnable positional embeddings are 256-dimensional. We concatenate word embeddings and the corresponding positional embeddings as the inputs to the decoder. Following the convention (Devlin et al., 2019), we randomly mask out 15% of tokens.

**Encoder Inputs.** When using symbolic object embeddings, we only need gold bounding boxes and object names (e.g., "small yellow rubber cube"). We embed each attribute value (e.g., "yellow") as a 64-dimensional vector, so each object embedding will be 256-dimensional. To obtain visual object embeddings, we use gold object segmentation and encode each object into a 4096-dimensional vector by using a pre-trained VGG[9] model (Simonyan and Zisserman, 2015). We set the dimension of objects' positional embeddings $d_e = 256$.

**Two-head Crossmodal Layer.** We empirically set all dropout rates $f_s^{dp} = f_s^{dp} = f^{dp} = 0.25$.

**Learning.** We optimize VG~MLM with Adam, where the learning rate is $5 \times 10^{-5}$, the weight decay is $10^{-8}$, and $\beta_1 = 0.9$, $\beta_2 = 0.999$. We use the MulTiStepLR learning rate scheduler, where milestones are $[15, 36, 45, 50]$ and $\gamma = 0.5$. We train VG~MLM for 100 epochs with a batch size of 50 and evaluate the final checkpoint.

**Evaluation.** For each setting, we run 5 times with different random seeds and report the mean and standard deviation of precision values.

### 4.5.4 Experimental Design

The goal of our experiments is to validate the effectiveness of our model architecture. We set out to investigate different ways of embedding objects and representing object pairs, and the strength of language supervision because these are major factors that affect model performance.

[8] https://github.com/zhaoyanpeng/sgi.

[9] VGG-19_BN: https://pytorch.org/vision/stable/models/vgg.html.

**Object Representations.** We consider symbolic object embeddings and visual object embeddings (see Section 4.3.3.3). Symbolic embeddings are used to estimate an upper bound on model performance, while visual embeddings are more practical. We indicate models that use symbolic and visual object embeddings by affixes "w/ S" and "w/ V", respectively.

**Object Pair Representations.** We have proposed to learn two sets of object representations for the subject role and the object role, respectively (see Equation 4.6), but a single set of object representations would also suffice. Recall that we represent a pair as the concatenation of the representations of the two objects: $\mathbf{u}^{(i,j)} = [\mathbf{o}_i : \mathbf{o}_j]$. With a single set of object representations, we need to additionally assume that the object $o_i$ on the left-hand side of ":" and the object $o_j$ on the right-hand side of ":" are assigned the subject role and the object role, respectively, i.e., role assignments are tied to the concatenation operator ":" rather than object representations. For example, the object $o_i$ in $\mathbf{u}^{(i,j)} = [\mathbf{o}_i : \mathbf{o}_j]$ and $\mathbf{u}^{(j,i)} = [\mathbf{o}_j : \mathbf{o}_i]$ has the same representation but is assigned a "subject" role and a "object" role, respectively. In our experiments, we will use a single set of object representations by default because this simplifies our model.

**Strength of Language Supervision.** Language supervision is generally weaker compared to direct supervision in the form of the (subject, predicate, object) triplets. To study how it influences the learning of our model, we vary the strength of language supervision by using different ways of referring to objects.

- *Ambiguous Captions.* The automatically synthesized captions are ambiguous by design. Specifically, to mimic natural language, which is ambiguous to some extent, when synthesizing captions, we introduce ambiguities by randomly dropping some attributes of each object. Take the image in Figure 4.6, a synthesized caption could be "*there is a large cylinder on the left of the yellow cube*", where the "large cylinder" may refer to either "large brown metal cylinder" or "large purple rubber cylinder". Though similar to natural language, ambiguities of this kind make learning more difficult.

- *Unambiguous Captions.* We further consider five types of unambiguous captions and group them according to how specifically the objects are referred to. (1) FULL indicates that we use all four attributes to refer to objects when possible. For example, the above ambiguous caption can be disambiguated as "*there is a*

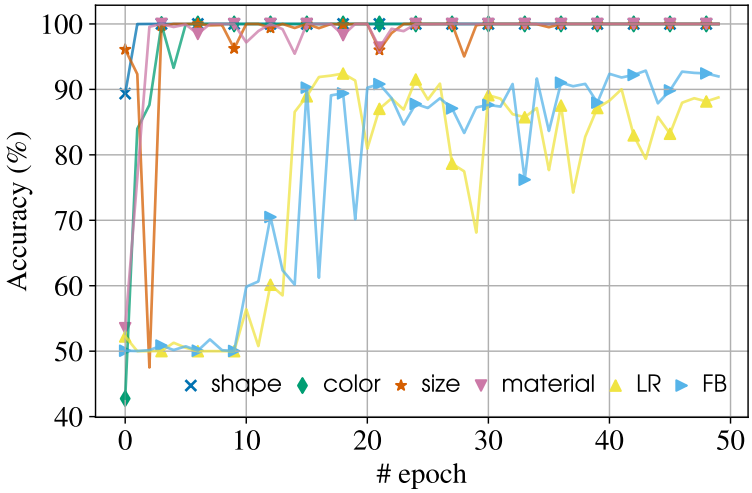



Figure 4.7: Per-attribute classification and relation prediction accuracy on the development set. Results are from the best run: unambiguous FULL†. LR and FB indicate relation prediction for the left-right pair and the front-behind pair, respectively.

*large purple rubber cylinder* *on the left of the* *small yellow rubber cube*", while in captions like "*the* *purple rubber cylinder* *on the left of the small yellow rubber cube is* *large*", we use three attributes to refer to "large purple rubber cylinder" because we would like models to infer "large" from the contexts; and (2) $\leq n$ (where $n \in \{1,2,3,4\}$) indicates that we use up to $n$ attributes to refer to objects. As $n$ decreases, there are fewer captions that are unambiguous.

### 4.5.5 Main Results

**Unambiguous captions are helpful.** Compared to "Ambiguous", unambiguous FULL achieves perfect or nearly perfect object classification performance and demonstrates a decent relation prediction precision for the front-behind pair (i.e., 78.5%). Thus, unambiguous language descriptions make learning easier.

**The more specific object descriptions, the better.** We vary the strength of language supervision by using different maximum numbers of attributes to refer to objects. For relation prediction, considering the variance of performance, using more attributes does not give rise to significant improvements, e.g., "$\leq 4$" and "$\leq 1$" have a similar mean precision (e.g., 53% for the front-behind pair), though "$\leq 1$" only uses around 60% of the training examples that "$\leq 4$" uses. But, for object classification, us-

| Model | Relation | | Pairs of Same Objects | | | | All Object Pairs | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $p_{\text{LR}}$ | $p_{\text{FB}}$ | $p_{\text{shape}}$ | $p_{\text{color}}$ | $p_{\text{size}}$ | $p_{\text{material}}$ | $p_{\text{shape}}$ | $p_{\text{color}}$ | $p_{\text{size}}$ | $p_{\text{material}}$ |
| Baseline | *50.00 | *50.00 | 34.50 | 11.62 | 49.63 | 49.69 | 52.50 | 21.58 | 71.01 | 71.66 |
| VG~MLM with Symbolic Object Embeddings (w/ S) | | | | | | | | | | |
| Ambiguous | $50.14_{\pm 0.1}$ | $51.06_{\pm 0.7}$ | $99.85_{\pm 0.3}$ | $100.00_{\pm 0.0}$ | $99.98_{\pm 0.0}$ | $99.96_{\pm 0.1}$ | $98.48_{\pm 2.4}$ | $97.69_{\pm 1.3}$ | $99.98_{\pm 0.0}$ | $99.97_{\pm 0.1}$ |
| FULL† | 88.75 | 90.66 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 |
| FULL | $50.40_{\pm 0.2}$ | $78.47_{\pm 5.4}$ | $100.00_{\pm 0.0}$ | $100.00_{\pm 0.0}$ | $100.00_{\pm 0.0}$ | $100.00_{\pm 0.0}$ | $99.64_{\pm 0.6}$ | $100.00_{\pm 0.0}$ | $99.95_{\pm 0.1}$ | $99.89_{\pm 0.2}$ |
| $\leq 4$ | $50.13_{\pm 0.1}$ | $53.32_{\pm 3.2}$ | $100.00_{\pm 0.0}$ | $100.00_{\pm 0.0}$ | $100.00_{\pm 0.0}$ | $100.00_{\pm 0.0}$ | $95.85_{\pm 3.2}$ | $95.22_{\pm 2.8}$ | $100.00_{\pm 0.0}$ | $100.00_{\pm 0.0}$ |
| $\leq 3$ | $50.07_{\pm 0.1}$ | $53.79_{\pm 5.6}$ | $100.00_{\pm 0.0}$ | $99.99_{\pm 0.0}$ | $99.98_{\pm 0.0}$ | $99.96_{\pm 0.1}$ | $93.28_{\pm 3.5}$ | $96.76_{\pm 1.8}$ | $99.88_{\pm 0.2}$ | $99.97_{\pm 0.1}$ |
| $\leq 2$ | $50.20_{\pm 0.1}$ | $51.66_{\pm 0.9}$ | $100.00_{\pm 0.0}$ | $97.60_{\pm 3.4}$ | $99.38_{\pm 1.2}$ | $99.74_{\pm 0.4}$ | $95.11_{\pm 3.9}$ | $94.62_{\pm 3.6}$ | $99.19_{\pm 1.5}$ | $99.78_{\pm 0.3}$ |
| $\leq 1$ | $50.56_{\pm 0.4}$ | $53.20_{\pm 1.1}$ | $44.39_{\pm 21.2}$ | $0.00_{\pm 0.0}$ | $0.00_{\pm 0.0}$ | $0.00_{\pm 0.0}$ | $62.46_{\pm 18.0}$ | $7.70_{\pm 6.6}$ | $41.55_{\pm 0.0}$ | $41.26_{\pm 0.0}$ |
| VG~MLM (w/ S) + Causal Language Modeling (w/ CLM) | | | | | | | | | | |
| FULL | $50.31_{\pm 0.1}$ | $71.58_{\pm 9.1}$ | $100.00_{\pm 0.0}$ | $100.00_{\pm 0.0}$ | $100.00_{\pm 0.0}$ | $100.00_{\pm 0.0}$ | $100.00_{\pm 0.0}$ | $99.99_{\pm 0.0}$ | $100.00_{\pm 0.0}$ | $100.00_{\pm 0.0}$ |
| VG~MLM with Visual Object Embeddings (w/ V) | | | | | | | | | | |
| FULL | $50.45_{\pm 0.5}$ | $55.76_{\pm 4.6}$ | $99.14_{\pm 0.1}$ | $94.76_{\pm 0.4}$ | $99.95_{\pm 0.0}$ | $96.78_{\pm 0.2}$ | $99.30_{\pm 0.2}$ | $81.84_{\pm 1.1}$ | $99.98_{\pm 0.0}$ | $98.81_{\pm 0.5}$ |

Table 4.2: * denotes theoretical performance. † indicates the best run. "w/ S" and "w/ V" indicate that VG~MLM uses symbolic and visual object embeddings, respectively. FULL indicates that all the four attributes are used to refer to objects when possible and "$\leq n$" ($1 \leq n \leq 4$) indicates that up to $n$ attributes are used to refer to objects (see Section 4.5.4).

ing more attributes leads to higher precision. This is unsurprising because using more attributes gives rise to stronger supervision.

**Relation prediction relies on object classification.** Figure 4.7 illustrates object classification and relation prediction accuracy after each training epoch. Interestingly, only after object classification performance stabilizes (e.g., from the 15-th epoch), does relation prediction performance tend to plateau, i.e., the model first learns to map objects to their labels in captions, then it learns to infer relations between objects.

**Symbolic object representations are helpful.** Compared to VG~MLM (w/ V), which uses visual object representations, VG~MLM (w/ S) performs better in general, indicating that symbolic object representations are the key to achieving good performance. Nevertheless, visual object representations (w/ V) result in nearly perfect object classification for all the attributes except "color" and "material", presumably because these two attributes have more valid assignments, i.e., 8 for "color" and 5 for "material", while other attributes have fewer than 5.

**Causal language modeling does not necessarily help.** We additionally optimize a causal language modeling (CLM) objective during training. Compared to VG~MLM (w/ S), though VG~MLM (w/ S+CLM) also achieves nearly perfect object classification, it does not improve relation prediction. As discussed before, masked language modeling alone should be adequate for learning mappings between words and their visual counterparts (see Footnote 3).

## 4.6 Summary

We have presented a novel setting for unsupervised scene graph induction, where a scene graph induction model is trained on image-text pairs and learns from only image-level captions. We propose an image-conditioned masked language model (VG~MLM) to tackle the task. VG~MLM adopts a Transformer encoder-decoder architecture. We tailor a multi-head attention module to connect the object encoder and caption decoder. The crossmodal module, by virtue of its architecture design, enables VG~MLM to infer scene graphs from images without relying on text. We create CLEVR-TV, which is an artificial image-captioning dataset, to learn and study VG~MLM, and propose automatic evaluation metrics to quantify the performance of VG~MLM. Though we

empirically find that VG~MLM can achieve reasonable performance when using symbolic object representations, we also observe that VG~MLM is unstable and sensitive to hyperparameters.

In the future, we plan to improve the proposed VG~MLM in the following respects:

- Exploring alternative model architectures. The architectural biases, which are implemented in VG~MLM, might be inappropriate and account for the difficulties of optimization;

- Jointly classifying objects and predicting relations. Currently we label objects and predict relations independently, but when working with natural images, knowing object labels arguably helps with relation prediction. For example, given an object pair: (man, book), the relation between them is more likely to be "read"/"hold" rather than "eat"/"ride". This type of commonsense knowledge has been exploited in previous work (Lu et al., 2016; Zareian et al., 2020a). We would expect VG~MLM to be able to derive it from abundant text, without relying on external knowledge bases. For example, we may substitute the decoder of VG~MLM with a pre-trained masked language model (Petroni et al., 2019).

We have so far discussed unsupervised text- and image-structure induction. We considered novel settings with the goal of learning structure-induction models from multimodal data. To tackle the challenges arising from the new settings, we designed end-to-end neural models that follow neural-symbolic and connectionist paradigms, i.e., neural-symbolic VC~PCFG and structure-aware VG~MLM.

The reasons that we have been focusing on visual and textual modalities are twofold: (1) unsupervised structure induction is a fundamental problem in both natural language and computer vision communities. There has been a considerable body of research on this problem, providing well-justified structured formalisms, promising models, and plausible benchmarks. When extending the single-modality setting to the multimodal setting, we can tap into established work and focus primarily on techniques for exploiting learning cues specified in multimodal data; and (2) aligned image-text data is abundantly available and easy to scale up. Apart from existing annotated large-scale image-text resources from the image-captioning area, image-text co-occurrences are relatively frequent on the web and, in principle, can be curated at a large scale via automatic approaches (Sharma et al., 2018). Thus, the proposed bimodal image-text learning setting introduces little to no annotation cost; accordingly, it aligns with our ultimate goal of achieving fully unsupervised learning.

Besides dominant visual and textual modalities, there are other important data modalities providing indispensable perspectives of the physical world, e.g., audio. But unlike images and text, which have abundant co-occurrences, many other modality pairs lack sufficient and easy-to-curate co-occurrence data. For example, non-speech audio (i.e., environmental sound) is rarely associated with informative natural language descriptions. This data scarcity issue potentially presents an obstacle to using multi-modal pre-training techniques, which usually require millions of training pairs (Radford et al., 2021; Jia et al., 2021; Akbari et al., 2021). In Chapter 5, we will look closer at this problem and develop unsupervised curation methods to solve it.

# Chapter 5

# Unsupervised Audio-Text Alignment Induction


Figure 5.1: VIP~ANT pivots audio and text via visual imagination.

In the previous chapters, we studied the problems of unsupervised structure induction that arise particularly from natural language and visual image understanding. We introduced novel learning settings so that we can use multimodal learning approaches to tackle these problems. An important reason that we focus on textual and visual modalities is that there are abundantly available image-text pairs for multimodal learning, thus requiring little to no annotation effort when applying multimodal learning techniques to unsupervised structure induction. However, differently from image-text co-occurrences, many other modality pairs lack sufficient and high-quality co-occurrence data. This data scarcity issue poses a grand challenge for using multimodal learning paradigms (e.g., contrastive image-text pre-training and multimodal masked language modeling), which tend to be rather data-hungry. In this chapter, we address the issue of scarce multimodal alignments and focus specifically on the alignment between

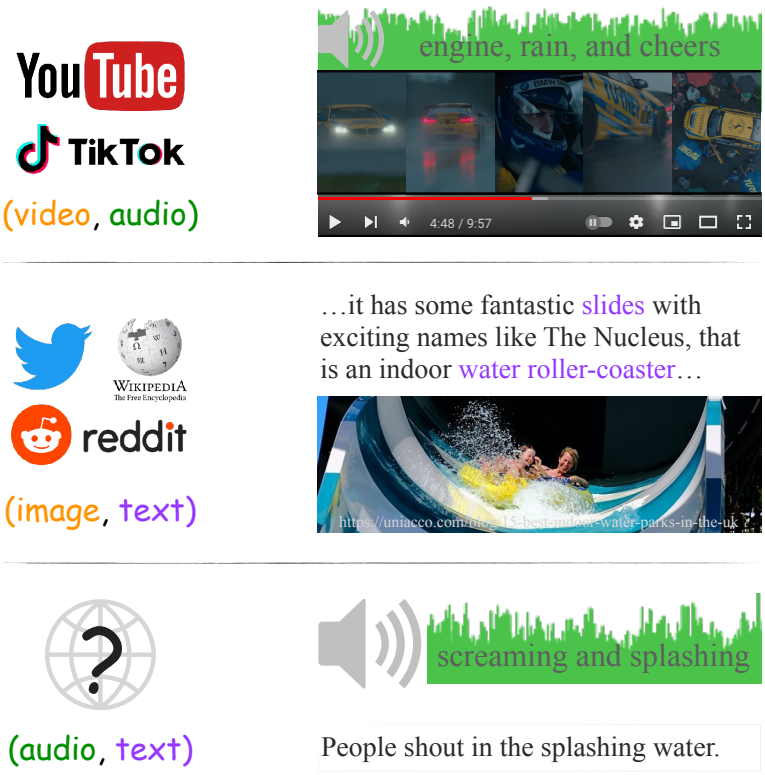



Figure 5.2: Video-audio and image-text co-occurrences are abundantly available on the web to support the learning of video-audio alignment and image-text alignment (e.g., via video-audio and image-text pre-training), but audio-text co-occurrences are not.

non-speech audio (i.e., environmental sound) and natural language descriptions.

Environmental sound provides rich perspectives on the physical world. For example, if we hear: *joyful laughing, a playful scream, and a splash;* we not only can visualize literal objects/actions that might have given rise to the audio scene, but also, we can reason about plausible higher-level facets, e.g., a child speeding down a water slide at a water park, splashing through the water (see Figure 5.1).

Machines capable of parsing, representing, and describing such environmental sound hold practical promise. For example, according to the National Association of the Deaf's captioning guide, accessible audio caption generation systems should go beyond speech recognition (i.e., identifying speakers and transcribing the literal content of their speech) and provide the textual description of all the sound effects, e.g., "a large group of people talking excitedly at a party", in order to provide the full information contained in that audio.[1]

[1] nad.org's captioning guide; Gernsbacher (2015) discusses the benefits of video captions beyond d/Deaf users.

The dominant paradigm for studying *machine hearing* (Lyon, 2010) has been relying on human-annotated audio-text data, where text is either free-form audio descriptions (e.g., "the sound of heavy rain") or tagsets (Salamon et al., 2014; Gemmeke et al., 2017; Kim et al., 2019a; Drossos et al., 2020). But existing supervised audio-text resources are limited. While some audio-text co-occurences can be sourced from audio-tag co-occurrences (Font et al., 2013) or from video captioning data (Rohrbach et al., 2015; Xu et al., 2016; Oncescu et al., 2021a), they are either not sufficiently related to environmental sound or limited in their scale and coverage.

In this paper, we study large-scale audio-text alignment without paired audio-text (AT) data. Inspired by pivot-based models for unsupervised machine translation (Wu and Wang, 2007; Utiyama and Isahara, 2007), we propose VIP~ANT, short for Visually Pivoted Audio and(N) Text. VIP~ANT uses images as a pivot modality to connect audio and text. It parallels our motivating example: hearing a sound, humans can visually *imagine* the associated situation and literally *describe* it. Pivoting is practically viable because there are abundantly available image-text (VT) and video-audio (VA) co-occurrences on the web, from which bimodal correspondence models can be trained (see Figure 5.2). By linking audio and text implicitly via the combination of the VT and VA models, we enable *zero-resource* connection between audio and text, i.e., VIP~ANT can reason about audio-text connections despite never having observed these modalities co-occur explicitly.

We evaluate on zero-shot audio-text retrieval and zero-shot audio classification. On the Clotho caption retrieval task (Drossos et al., 2020), without any parallel AT data, VIP~ANT surpasses the supervised state of the art by 2.2% R@1; on zero-shot audio classification tasks, it establishes new state of the arts, achieving 57.1% accuracy on ESC50 (Piczak, 2015) and 44.7% accuracy on US8K (Salamon et al., 2014). We also show that the zero-resource pivoting AT model VIP~ANT can be improved by:

- *Unsupervised curation.* Whereby noisy AT pairs are explicitly mined from the pivoting model and serve as additional training data (e.g., +5.7% on ESC50 and +9.3% on US8K);

- *Few-shot curation.* Whereby a small number of human-annotated audio caption pairs are made available at training time (e.g., a few hundred pairs increase the zero-shot audio classification accuracy by 8% on US8K).

However, for ESC-50, according to the empirical scaling relationship we find, it would require around $2^{21} \approx$ 2M aligned audio-text pairs for the zero-shot model to

| Model | AE Initialization | Objective | AT Supervision | VT Alignment | Zero-shot AT Retrieval |
|---|---|---|---|---|---|
| MMV (Alayrac et al., 2020) | Random | $\mathcal{L}_{bi\text{-}bi}$ | None | Trainable | ✗ |
| VATT (Akbari et al., 2021) | Random | $\mathcal{L}_{bi\text{-}bi}$ | None | Trainable | ✗ |
| AudioCLIP (Guzhov et al., 2022) | ImageNet | $\mathcal{L}_{tri}$ | 2M Audio Tags | Trainable | ✗ |
| Wav2CLIP (Wu et al., 2022) | Random | $\mathcal{L}_{bi\text{-}bi}$ | None | Frozen | ✗ |
| VIP~ANT (ours) | Image CLIP | $\mathcal{L}_{bi\text{-}bi}$ | None | Frozen | ✓ |
| VIP~ANT +AT (ours) | Image CLIP | $\mathcal{L}_{bi\text{-}bi}$ | Caption Curation | Frozen | ✓ |

Table 5.1: Survey of recent prior work studying for tri-modal (images, audio, and text) representation learning. AE is short for Audio Encoder. Some work experiments with more than one objective, we report the best or the one it advocates. Importantly, we report zero-shot audio-text retrieval between audio and full-sentence text descriptions, along with scaling laws associated with that setup.

match human parity on ESC50 under our setup, which is an order-of-magnitude more than the largest currently-available audio-text corpus of Kim et al. (2019a).

## 5.1 Related Work

**Supervised audio representation learning.** While automatic speech recognition has been a core focus of the audio processing community, environment sound classification has emerged as a new challenge and is drawing more attention (Salamon et al., 2014; Piczak, 2015; Gemmeke et al., 2017). Some prior work in learning sound event representations is supervised by category labels (Dai et al., 2017b; Boddapati et al., 2017; Kumar et al., 2018; Guzhov et al., 2021; Gong et al., 2021). Others use weaker forms of supervision for tagging Kumar and Raj (2017); Kong et al. (2018) and localization McFee et al. (2018); Kim and Pardo (2019).

**Learning audio representations from visual imagination.** There have been two main paradigms for using visual information to derive audio representations. In the two-stage setup, an image encoder is first pre-trained; these weights are used as the initialization of the supervised audio model (Guzhov et al., 2021; Gong et al., 2021). The other adopts contrastive learning: it exploits the image-audio alignment inherent in videos and learns audio and image/video representations jointly (Korbar et al., 2018; Wang et al., 2021; Nagrani et al., 2021). We use insights from both directions by (1)

using CLIP's image encoder, which has been pre-trained on image-text pairs (Radford et al., 2021), to initialize an audio encoder and (2) using contrastive pre-training on image-audio pairs. During training, we do not require any labeled images or audio.

**Tri-modal learning of audio-text alignment.** Our work extends recent work that generalizes the bi-modal contrastive learning to a tri-modal setting (Alayrac et al., 2020; Akbari et al., 2021). While they also connect audio and text implicitly by using images as a pivot, the quality of this audio-text alignment has rarely been studied. To our knowledge, we present the first comprehensive evaluation of the inferred audio-text alignment via zero-shot retrieval/classification.

The works closest to ours are AudioCLIP (Guzhov et al., 2022) and Wav2CLIP (Wu et al., 2022). AudioCLIP's pre-training setup is similar to ours, but requires human-annotated textual labels of audio, while ours does not. Wav2CLIP is concurrent with our work; while similar-in-spirit, our model not only performs significantly better, but also, we more closely explore methods for improving audio-text alignment, e.g., unsupervised curation.

**Pivot-based alignment models.** The pivoting idea for alignment learning can date back to Brown et al. (1991). Language pivots (Wu and Wang, 2007; Utiyama and Isahara, 2007) and image pivots (Specia et al., 2016; Hitschler et al., 2016; Nakayama and Nishida, 2017) have been explored in zero-resource machine translation. Pivot-based models have also been shown to be helpful in learning image-text alignment (Li et al., 2020b). Differently from the previous uni-modal (i.e., language) and bi-modal (i.e., vision and language) learning settings, we focus on a tri-modal setting and propose to use visual images as a pivot to bridge audio and text.

## 5.2 Model

We first formalize tri-modal learning by assuming available co-occurrence data for every pair of modalities (Section 5.2.1). Then we present bi-bi-modal pre-training as an alternative when there is no paired audio-text data, and implement VIP~ANT via bi-bi-modal pre-training (Section 5.2.2). Finally, we describe model variants for cases of varying AT supervision (Section 5.2.3).

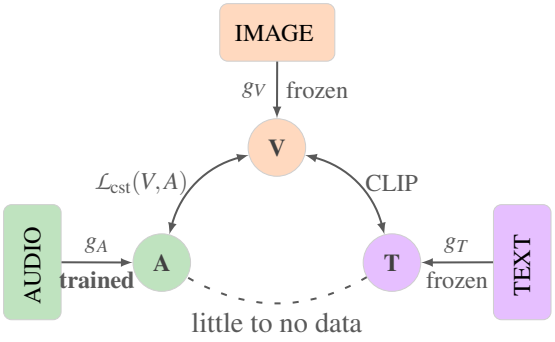



Figure 5.3: Learning paradigm of VIP~ANT.

### 5.2.1 Tri-modal Representation Learning

Tri-modal representation learning between images, audio, and text aims to derive representations from co-occurrence patterns among the three modalities (Alayrac et al., 2020; Akbari et al., 2021). We consider a simple tri-modal representation space, which relies on encoding functions $g_V : V \rightarrow \mathbf{V}$, $g_A : A \rightarrow \mathbf{A}$, and $g_T : T \rightarrow \mathbf{T}$ to map images $v$, audio $a$, and text $t$ ($v \in V, a \in A,$ and $t \in T$), respectively, to a shared vector space: $\boldsymbol{v}, \boldsymbol{a}, \boldsymbol{t} \in \mathbb{R}^d$ ($\boldsymbol{v} \in \mathbf{V}, \boldsymbol{a} \in \mathbf{A},$ and $\boldsymbol{t} \in \mathbf{T}$). Instead of pre-specifying the precise semantics of this continuous space, vector similarities across modalities are optimized to reconstruct co-occurrence patterns in training corpora, i.e., two vectors should have a higher dot product if they are more likely to co-occur. We use contrastive learning with the InfoNCE loss (Sohn, 2016; van den Oord et al., 2018):

$$\mathcal{L}_{\text{cst}}(A,B) = \sum_i \frac{\exp s(\boldsymbol{a}^{(i)}, \boldsymbol{b}^{(i)})}{\sum_{\boldsymbol{a}} \exp s(\boldsymbol{a}, \boldsymbol{b}^{(i)})} + \frac{\exp s(\boldsymbol{a}^{(i)}, \boldsymbol{b}^{(i)})}{\sum_{\boldsymbol{b}} \exp s(\boldsymbol{a}^{(i)}, \boldsymbol{b})}, \tag{5.1}$$

where $A, B$ are two sets of data points from two different modal domains, respectively; $\boldsymbol{a}^{(i)}, \boldsymbol{b}^{(i)}$ are vector representations of the co-occurring pair $(a^{(i)}, b^{(i)})$ which are encoded by $g_A(a^{(i)})$ and $g_B(b^{(i)})$, respectively; $s(\boldsymbol{a}, \boldsymbol{b})$ computes the similarity between $\boldsymbol{a}$ and $\boldsymbol{b}$, which we take to be scaled cosine similarity.

If we had access to co-occurrence data between all pairs of modalities, we could optimize the tri-modal loss:

$$\mathcal{L}_{\text{tri}}(V,A,T) = \mathcal{L}_{\text{cst}}(V,A) + \mathcal{L}_{\text{cst}}(A,T) + \mathcal{L}_{\text{cst}}(V,T). \tag{5.2}$$

### 5.2.2 Visually Pivoted Audio and Text

Differently from image-text and image-audio pairs, which are abundantly available on the web, audio-text data is scarce. Instead of Equation 5.2, in VIP~ANT, we consider a

"bi-bi-modal" loss, which does not require parallel AT data:

$$\mathcal{L}_{\text{bi-bi}}(V,A,T) = \mathcal{L}_{\text{cst}}(V,A) + \mathcal{L}_{\text{cst}}(V,T)\,. \tag{5.3}$$

The image encoder is shared between the VA alignment model (i.e., $\mathcal{L}_{\text{cst}}(V,A)$) and the VT alignment model (i.e., $\mathcal{L}_{\text{cst}}(V,T)$) and thus provides a zero-resource connection between audio and text in the tri-modal embedding space implicitly.

#### 5.2.2.1 Model Architecture

**Image and text encoders.** Instead of learning $g_V$ and $g_T$ from scratch, we build on a pre-trained CLIP model, which has been pre-trained on WebImageText (WIT), a dataset of 400 million image-text pairs gathered from the internet (Radford et al., 2021). CLIP has been shown highly performant on VT tasks, e.g., zero-shot image classification. We use the ViT-B/32 model in this work, which consists of a 12-layer vision Transformer (ViT) and a 12-layer language Transformer (Vaswani et al., 2017; Dosovitskiy et al., 2021). Given CLIP's strong VT alignment, we use its image encoder as $g_V$ and text encoder as $g_T$. During learning, $g_V$ and $g_T$ are kept frozen and thus the joint VT representation space is untouched (see Figure 5.3). We minimize only the first loss term of Equation 5.3:

$$\min_{\Theta_A} \mathcal{L}_{\text{cst}}(V,A)\,, \tag{5.4}$$

where $\Theta_A$ are the trainable parameters of the audio encoder $g_A$.

**Audio encoder.** Our audio encoder has the same vision Transformer architecture as CLIP's image encoder (ViT-B/32). In Section 5.3, we show that initializing the audio encoder with CLIP's visual weights significantly improves convergence speed and accuracy. The architectural modifications, which enable the use of visual CLIP's architecture for audio, include (see Figure 5.4 for an illustration):[2]

- We customize the convolution stride to allow for overlaps between neighbor patches of Spectrogram features of audio. ViT uses the same convolution stride as the kernel size, so it encodes non-overlapped image regions, while we set the convolution stride as half of the kernel size to compute cross-correlation between patches.
- In the input embedding layer, we average the kernel weights of the convolution layer along the input channel to account for 1-channel Mel-filter bank features of audio (*cf.* RGB channels of images).

[2] https://github.com/zhaoyanpeng/vipant.

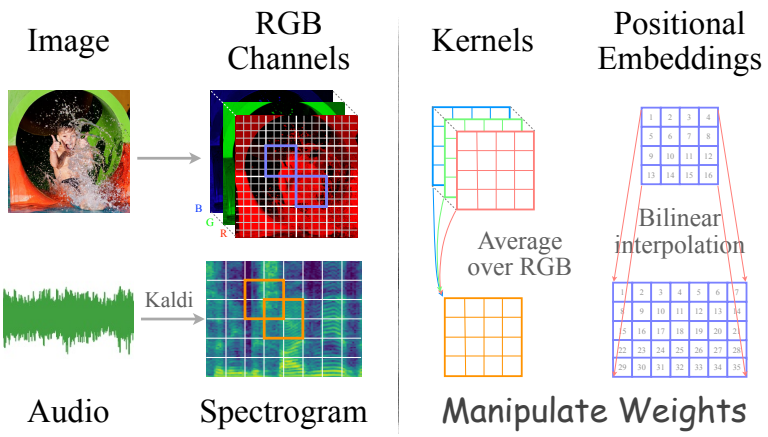



Figure 5.4: **Left:** three-channel images versus one-channel Spectrogram features of audio. We use ViT to encode images and audio. ViT uses a convolution layer to encode non-overlapped image patches into a sequence of image tokens, but for audio, we modify the convolution stride to allow for overlaps between neighbor patches. Note that the convolutional layers of both ViT and our model do not use padding. **Right:** adapting the convolution layer of ViT for audio encoding. For simplicity's sake, we omit the output channels of kernel weights and positional embeddings.

- We up-sample the 2-dimensional positional embeddings of image tokens to account for longer audio token sequences. Practically, this is done via bilinear interpolation (Gonzalez and Woods, 2018) (see the right part of Figure 5.4).

#### 5.2.2.2 Bi-bi-modal Pre-training Details

**Video-audio co-occurences.** To optimize Equation 5.4, we gather VA co-occurrences from AudioSet (AS; Gemmeke et al. (2017)),[3] which contains temporally aligned audio and video frames from 10-second clips gathered from around 2 million YouTube videos. To construct aligned image-audio pairs from AS, we adopt a sparse sampling approach (Lei et al., 2021): we first, extract four equal-spaced video frames from each clip. Then, during training, we randomly sample a frame from the four, and treat it as co-occurring with the corresponding audio clip. At test time, we always use the second video frame as the middle frame to construct image-audio pairs. We use the unbalanced training set, which consists of around 2 million video clips, to pre-train the audio encoder. Since AudioSet does not provide an official validation set, we validate the audio encoder and tune model hyperparameters on the balanced training set.

[3] https://github.com/zhaoyanpeng/audioset-dl.

| STAT. | AudioSet | ESC50 | US8K | AudioCaps | Clotho |
|---|---|---|---|---|---|
| # Train | 2041789 (unbalanced) | 2000 (5-fold) | 8732 (10-fold) | 44118 (×1 caption) | 3839 (dev-train) |
| # Dev | 22160 (balanced) | | | | 1045 (dev-val) |
| # Val | | | | 441 (×5 caption) | 1045 (dev-test) |
| # Test | 20371 (balanced) | | | 860 (×5 caption) | 1043 (withheld) |
| # Class | 527 | 50 | 10 | | 5 captions/audio |
| Duration | 10s | 5s | 0-4s | 10s | 15-30s |
| Task | Multi-label CLF | Multi-class CLF | Multi-class CLF | Captioning | Captioning |
| Source | YouTube | Freesound | Freesound | YouTube (AudioSet) | Freesound |

Table 5.2: Statistics of the data used in this paper. CLF is the abbreviation of "classification". In AudioSet (Gemmeke et al., 2017) audio clips come from distinct videos. Balanced split means that there are at least 59 samples for each of the 527 sound classes. We managed to download 18,036 out of 22,160 videos in the balanced training split, 16,416 out of 20,371 videos in the test/validation split, and 1,715,367 out of 2,041,789 videos in the unbalanced split.

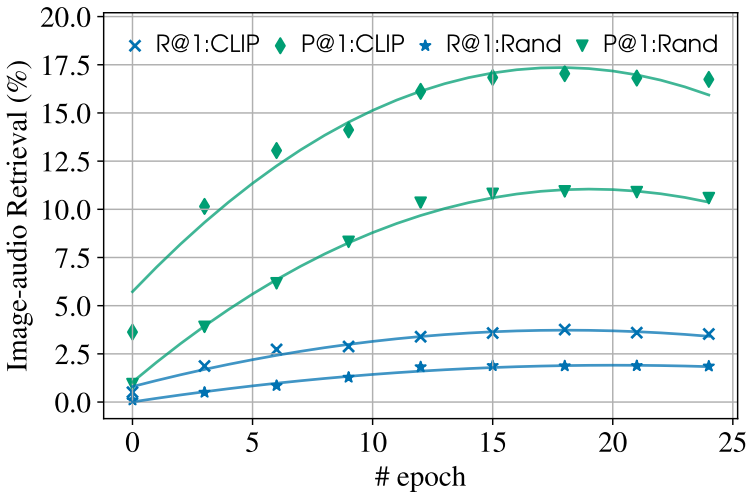



Figure 5.5: Image → Audio retrieval performance per image-audio pre-training epoch. We conduct this evaluation on the AS balanced training set. "CLIP" and "Rand" indicate that the audio encoder is initialized from CLIP's image encoder and has random initialization, respectively.

**Audio preprocessing.** We use Kaldi (Povey et al., 2011) to create Mel-filter bank features (FBANK) from the raw audio signals. Specifically, we use the Hanning window, 128 triangular Mel-frequency bins, and a 10-millisecond frameshift. We always use the first audio channel when an audio clip has more than one channel. We apply two normalizations: (1) before applying Kaldi, we subtract the mean from the raw

audio signals; and (2) we compute the mean and standard deviation of FBANK on the unbalanced AS training set, and then normalize the FBANK of each audio clip. For data augmentation, inspired by Gong et al. (2021), we use frequency masking and time masking: we randomly mask out one-fifth FBANK along the time dimension and one-fourth FBANK along the frequency dimension during training.

**Training dynamics.** The architecture of our audio encoder follows the vision Transformer of CLIP (ViT-B/32, see Radford et al. (2021) for more details). For the trade-off of efficiency and efficacy, we set the convolution stride to $16 \times 24$. This results in around 300 audio tokens for a kernel size of $32 \times 32$ and an input size of $1000 \times 128$ (all in the form of $time \times frequency$). We optimize the model with LARS (You et al., 2017), where the initial learning rates for model weights and model biases are set to 2e-1 and 4.8e-3, respectively. We pre-train our model on 4 NVIDIA Quadro RTX 8000 GPUs and for 25 epochs. We empirically set the batch size to 432 to fit the GPU memory. The full pre-training can be done within 24 hours.

**Evaluation.** We measure the image-audio pre-training performance by retrieval precision and recall:

$$p = \frac{\#(\text{relevant items among the retrieved})}{\#(\text{retrieved items})}, \tag{5.5}$$

$$r = \frac{\#(\text{relevant items among the retrieved})}{\#(\text{relevant items})}. \tag{5.6}$$

Audio is relevant if it has the same set of labels as the image query, and vice versa.[4] We average precisions and recalls over all samples in the balanced AS training set. Figure 5.5 illustrates the top-1 retrieval performance with images as the query (similar trends are observed when using audio as the query). Compared with random initialization, initializing the audio encoder from CLIP's image encoder leads to faster convergence and better VA alignment. As we will see, this performance on VA retrieval transfers to downstream AT tasks.

[4]We say audio has a set of labels because each audio clip in AudioSet is annotated with multiple labels, e.g., an audio clip can be labeled with both "music" and "bark". This definition of relevance will lead to a large number of relevant items, i.e., a large denominator in Equation 5.6, and thus the recall will be much smaller than the corresponding precision (see Figure 5.5).

| | | |
|---|---|---|
| Unsupervised (Zero-resource) | **AC** | **A**udio-focused **C**aptions originate from the training captions of AudioCaps and Clotho. We perform caption retrieval by using CLIP and the prompt "the sound of" (1,080,078 aligned pairs). |
| | example | *A balloon is rubbed quickly and slowly to make squeaking sounds.* |
| | **FC** | **F**ree **C**aptions are generated by priming GPT-J with MSCOCO captions. We perform caption retrieval by using CLIP and the prompt "a photo of" (1,224,621 aligned pairs). |
| | example | *The blue colored person is jumping on the white and yellow beach ball.* |
| | **VC** | **V**ision-focused **C**aptions originate from MSCOCO. We perform caption retrieval by using CLIP and the prompt "a photo of" (1,172,276 aligned pairs). |
| | example | *A sky view looking at a large parachute in the sky.* |
| | **RC** | **R**andom **C**aptions indicates that we break the gold AL alignment in AudioCaps by randomly sampling a caption for each audio clip. They are used as a lower bound on the quality of AL alignment (44,118 aligned pairs). |
| | example | *A whoosh sound is heard loudly as a car revs its engines.* |
| Supervised | **GL** | **G**old textual **L**abels are used to construct AL pairs (120,816 aligned pairs). |
| | example | *Gurgling* |
| | **GC** | **G**old **C**aptions from AudioCaps provide an upper bound on the quality of AL alignment (44,118 aligned pairs). |
| | example | *Children screaming in the background as the sound of water flowing by.* |

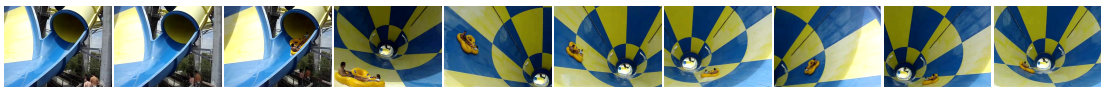


Table 5.3: Different ways of curating AT pairs. *Gurgling* is described as "the bubbling sound of water flowing through a narrow constriction, such as from a bottle with a narrow neck." The example comes from this YouTube video: 1O7-QuhweZE.

### 5.2.3 Unsupervised and Few-Shot Curation

To improve the AT alignment beyond pivoting, we consider curating audio-text pairs, and then performing an additional fine-tuning step by training the audio encoder with the AT loss, i.e., $\mathcal{L}_{cst}(A,T)$.[5] During AT fine-tuning, we keep the text encoder $g_T$ frozen and only fine-tune the audio encoder.

[5]Since our goal is to improve AT alignment, we primarily focus on AT fine-tuning; nonetheless, we compare AT fine-tuning to full VAT fine-tuning as in Equation 5.2 in Section 5.3.3.

**Unsupervised curation.** We consider explicitly mining AT pairs from VIP~ANT. Because this zero-resource method uses no human supervision, we refer to it as "unsupervised curation". Concretely, for each video segment in AudioSet, we extract a video frame, and input that frame to the original CLIP image encoder. Then, we encode a large set of candidate captions, and perform Image $\rightarrow$ Text retrieval over them by using the CLIP text encoder. The top candidate captions according to cosine similarity are then paired with the audio that corresponds to the original video clip.

We consider multiple caption sources to search over. As noted by Kim et al. (2019a), captions for images and captions for environmental audio are significantly different in focus. We consider two vision-focused caption sets: (1) MSCOCO (Lin et al., 2014) captions (VC); and (2) because MSCOCO captions are limited to 80 object categories, we generate free-captions from GPT-J (Wang and Komatsuzaki, 2021) conditioned on MSCOCO captions as a prompt (FC).[6] We additionally consider audio-focused captions from the training set of AudioCaps (Kim et al., 2019a) and Clotho Drossos et al. (2020) (AC).[7] As a baseline, we also consider a random caption alignment, which assigns a random caption from AC to each clip (instead of pivoting on images). The upper half of Table 5.3 summarizes different ways of curating AT pairs without additional supervision.

**Few-shot curation.** We also explore the effect of incorporating limited amounts of AT supervision, specifically, via captions from AudioCaps (GC) and textual labels of AudioCaps (GL) (see the bottom half of Table 5.3).

## 5.3 Audio-Text Experiments

We use two types of tasks to evaluate the quality of the audio-text alignments learned by our model: AT retrieval and zero-shot audio classification.

**AT retrieval.** We conduct audio-text retrieval on AudioCaps and Clotho for in-domain evaluation and out-of-domain evaluation, respectively:

[6] We create prompts by randomly sampling several captions from MSCOCO captions and separating them with a newline character "\n". Given such a prompt, GPT-J can generate text in a similar style to MSCOCO captions. We empirically find that GPT-J can generate diverse text using novel entities and entity combinations, presumably because we do not require that each generated caption corresponds to a specific natural image; accordingly, GPT-J is free to use entity synonyms or hallucinate.

[7] We do not use the *alignment* of these captions — just the captions themselves.

| | Model | 10-second Clotho (eval) | | | | 18-second Clotho (eval) | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Text→Audio | | Audio→Text | | Text→Audio | | Audio→Text | |
| | | R@1 | R@10 | R@1 | R@10 | R@1 | R@10 | R@1 | R@10 |
| Zero-resource | VA-Rand | 1.4 | 7.4 | 3.2 | 13.1 | 1.3 | 7.5 | 3.2 | 13.5 |
| | VIP~ANT | 1.9 | 10.1 | 6.1 | 23.7 | 1.9 | 9.5 | 7.0 | 25.6 |
| | +AT w/ AC | 5.9 | 26.3 | 8.2 | 30.3 | 6.7 | 29.1 | 7.1 | 30.7 |
| | +AT w/ FC | 5.7 | 26.6 | 6.6 | 28.0 | 6.5 | 27.7 | 7.8 | 29.7 |
| | +AT w/ VC | 5.2 | 25.2 | 7.0 | 25.9 | 5.5 | 25.6 | 7.6 | 28.2 |
| | +AT w/ RC | 3.5 | 16.3 | 5.7 | 23.6 | 3.5 | 16.9 | 5.5 | 24.9 |
| Zero-shot | +AT w/ GL | 6.0 | 27.1 | 6.1 | 25.4 | 6.7 | 29.0 | 6.8 | 27.0 |
| | +AT w/ GC | 10.2 | 39.0 | 10.3 | 37.2 | **11.1** | **40.5** | **11.8** | **41.0** |

Table 5.4: Interpolating positional embeddings to account for Clotho audios which are longer than 10 seconds.

- **AudioCaps** (Kim et al., 2019a) builds on AudioSet (Gemmeke et al., 2017) and provides captions for a subset of audio clips in AudioSet (sourced from YouTube). As we have pre-trained the audio encoder on AudioSet, we consider audio-text retrieval on AudioCaps as *in-domain* evaluation.

- **Clotho** (Drossos et al., 2020) consists of audio clips which have a duration of 15-30 seconds and come from Freesound (Font et al., 2013). It has a different sound source from AudioCaps and is used for *out-of-domain* evaluation.

We study the out-of-domain generalizability of our models by applying them to Clotho directly, without further fine-tuning on it. Since Clotho audio clips (15-30s) are longer than our pre-training audio clips (10s), to apply our pre-trained audio encoder to Clotho audio-caption retrieval, we up-sample the pre-trained positional embeddings to account for longer audio token sequences. Table 5.4 shows the retrieval performance of 10-second and 18-second Clotho audio. In general, longer audio gives rise to better audio-caption retrieval performance, so we will use up-sampling as the default setting.

**Zero-shot audio classification.** We consider the following three widely used datasets for audio classification.

- **ESC50** (Piczak, 2015) contains 2000 audio clips from 50 classes. Each audio clip

has a duration of 5 seconds and a single textual label. We follow the standard $k$-fold data splits.

- **US8K** (Salamon et al., 2014) contains 8732 audio clips from 10 classes. Each audio clip has a duration of less than 4 seconds and a single textual label. We follow the standard $k$-fold data splits.

- **AudioSet** (Gemmeke et al., 2017) is a benchmark dataset for multi-label classification. AudioSet provides balanced and unbalanced training sets. The balanced set consists of 22 thousand audio clips and the unbalanced set contains around 2 million audio clips. It also provides 20 thousand balanced audio clips for evaluation (more data statistics can be found in Table 5.2).

For each audio clip $\boldsymbol{a}$, we first compute the cosine similarity between it and every possible textual label in the tri-modal representation space. Then we predict the label $t$ with the highest similarity:

$$\arg\max_i \cos(\boldsymbol{t}^{(i)}, \boldsymbol{a}) . \tag{5.7}$$

### 5.3.1 Main Results

Our prediction results for AT retrieval are given in Table 5.5 and for zero-shot classification in Table 5.6.

**Initializing with visual CLIP weights helps.** Comparing VA-Rand to VIP~AnT, we see accuracy increases in all classification and retrieval setups. For example, on AudioCaps, VIP~AnT outperforms VA-Rand by 4.5% R@1 and 13.6% R@10. This confirms that the findings of Gong et al. (2021) carry over to unsupervised audio pre-training.

**Pivoting works well for Audio → Text.** VIP~AnT exhibits surprisingly strong performance on AT retrieval tasks and zero-shot classification. Take AT retrieval on Clotho, it outperforms the supervised baseline (Oncescu et al., 2021b) by 2.2% R@1 for text retrieval, without being trained or fine-tuned on Clotho, and without ever having seen an aligned AT pair.

| | Model | AudioCaps | | | | Clotho | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Text→Audio | | Audio→Text | | Text→Audio | | Audio→Text | |
| | | R@1 | R@10 | R@1 | R@10 | R@1 | R@10 | R@1 | R@10 |
| | Supervised SoTA | 18.0 | 62.0 | 21.0 | 62.7 | 4.0 | 25.4 | 4.8 | 25.8 |
| Zero-resource | VA-Rand | 1.3 | 7.3 | 5.6 | 24.5 | 1.3 | 7.5 | 3.2 | 13.5 |
| Zero-resource | VIP~ANT | 0.8 | 7.9 | 10.1 | 38.1 | 1.9 | 9.5 | 7.0 | 25.6 |
| Zero-resource | +AT w/ AC | 9.9 | 45.6 | 15.2 | 52.9 | 6.7 | 29.1 | 7.1 | 30.7 |
| Zero-resource | +AT w/ FC | 8.9 | 41.5 | 14.7 | 50.0 | 6.5 | 27.7 | 7.8 | 29.7 |
| Zero-resource | +AT w/ VC | 6.9 | 35.7 | 13.5 | 49.4 | 5.5 | 25.6 | 7.6 | 28.2 |
| Zero-resource | +AT w/ RC | 3.8 | 19.9 | 10.7 | 38.1 | 3.5 | 16.9 | 5.5 | 24.9 |
| Zero-shot | +AT w/ GL | 12.4 | 52.9 | 13.0 | 51.2 | 6.7 | 29.0 | 6.8 | 27.0 |
| Zero-shot | +AT w/ GC | **27.7** | **78.0** | **34.3** | **79.7** | **11.1** | **40.5** | **11.8** | **41.0** |
| | OracleAV-CLIP | 4.8 | 27.8 | 6.6 | 31.2 | | | | |

Table 5.5: Audio caption retrieval performance (%) on AudioCaps test set and Clotho evaluation set. "Supervised SoTA" comes from Oncescu et al. (2021b). OracleAV-CLIP: we replace audio with the corresponding image and evaluate the image-text retrieval performance of CLIP (Radford et al., 2021). VA-Rand is the same as VIP~ANT but the audio encoder is initialized randomly, instead of from CLIP visual weights. We further fine-tune VIP~ANT on AT data curated in different fashions, e.g., AC, FC, VC, and RC are mined explicitly from the zero-resource pivoting model (see Table 5.3 for details).

**Prompting (usually) helps.** Inspired by the zero-shot image classification setups of CLIP (Radford et al., 2021), we prefix textual labels with a prompt in zero-shot audio classification. We empirically find that the prompt "*the sound of*" works well. Using it greatly improves zero-shot multi-class classification accuracy (see Table 5.6). Take VIP~ANT, the prompt gives rise to an improvement of 7.2% on ESC50 and 6.9% on US8K, but hurts multi-label classification performance on AS.

**Random curation helps.** Even when the audio-text pairs used to train that objective are sampled entirely at random (+AT w/ RC), VIP~ANT improves, e.g., R@1 for Text → Audio retrieval increases from 0.8% to 3.8%. We conjecture that RC at least makes audio representations aware of and lean towards the text cluster of the joint VT rep-

| | Model | ESC50 | US8K | AS |
|---|---|---|---|---|
| | Supervised | $95.7_{\pm 1.4}$ | $86.0_{\pm 2.8}$ | 37.9 |
| Zero-resource | VA-Rand | 37.6(33.0) | 41.9(38.1) | 1.7( 2.0) |
| | VIP~ANT | 57.1(49.9) | 44.7(37.8) | 2.6( 2.8) |
| | +AT w/ AC | 62.8(55.7) | 54.0(47.0) | 11.6(12.3) |
| | +AT w/ FC | 62.5(58.0) | 52.7(50.0) | 11.2(12.2) |
| | +AT w/ VC | 61.9(58.0) | 52.7(50.3) | 8.9(10.7) |
| | +AT w/ RC | 51.6(36.1) | 42.3(28.5) | 4.1( 4.6) |
| | Wav2CLIP | 41.4 | 40.4 | |
| Zero-shot | +AT w/ GL | 67.2(64.5) | 62.6(61.0) | 15.4(**18.9**) |
| | +AT w/ GC | **69.5**(64.2) | **71.9**(67.1) | 13.3(13.6) |
| | AudioCLIP | 69.4 | 65.3 | |

Table 5.6: Zero-shot audio classification accuracies (%) on ESC50 and US8K and mAPs (%) on AudioSet (AS). "Supervised" = upper bound performance of VIP~ANT when fine-tuned with supervised audio labels. In the zero-shot/zero-resource settings, we use a prompt "*the sound of*" by default (results in parenthesis are without the prompt). "+AT" = fine-tuned VIP~ANT on AT pairs with different curations. AudioCLIP is pre-trainined using the 2 million textual labels of AudioSet; +AT w/ GL and +AT w/ GC are trained with only 44K labels/captions. Wav2CLIP is most directly comparable to our zero-resource pivoting model VIP~ANT with unsupervised curation.

resentation space.[8] While this result also holds for AS classification (+1.5% mAP), performance decreases for ESC50 (-5.5% accuracy) and US8K (-2.4% accuracy).

**Unsupervised curation is universally helpful.** VIP~ANT fine-tuned with unsupervised audio captions (+AT w/ AC) outperforms both pivoting (VIP~ANT) and random curation (+AT w/ RC) in all cases. Thus, explicitly mining unsupervised AT pairs can be a helpful zero-resource approach. Performance with automatically generated captions (FC) is similar to captions written by humans (AC).

[8]Concretely, VA pre-training pushes audio embeddings towards the image cluster (V) of the VT space of the pre-trained CLIP, but it does not guarantee that audio embeddings will be as close to the text cluster (T) of the VT space as to V. Random curation provides an estimate of the text-cluster's distributional properties, i.e., the audio embeddings are moved on top of the distribution of the text cluster of the VT space *explicitly*; surprisingly, this crude "semantic-free" alignment method improves the quality of audio-text alignment.

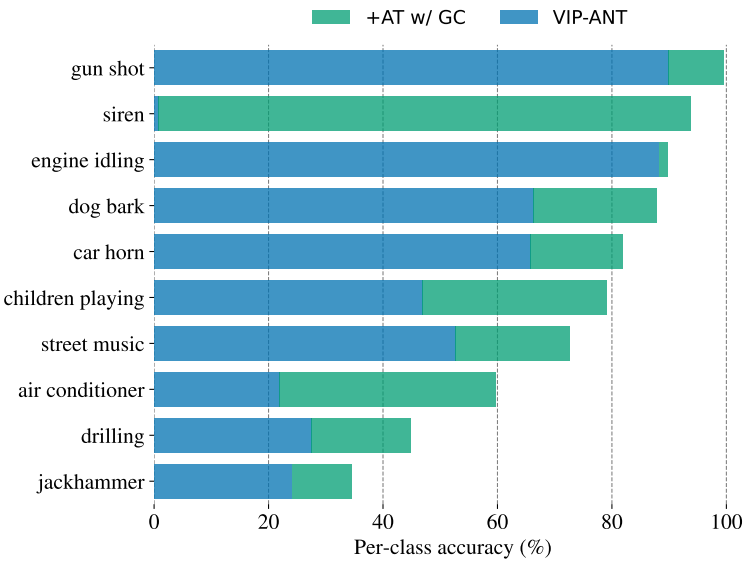



Figure 5.6: Per-class accuracy on US8K.

**Supervision is still the most helpful.** Fine-tuning VIP~ANT on GC pairs leads to the highest accuracies on ESC50 and US8K. However, we do not observe similar improvements on AS, presumably because multi-label classification is more challenging – it requires more direct language supervision, such as audio labels. This is further evident when we fine-tune VIP~ANT on GL and achieve the highest accuracy (18.9% mAP) on AS (see Table 5.6).

For retrieval, GL uses only audio labels as the text, and thus less dense language supervision than GC; accordingly, it performs slightly worse than GC, but still, it gives better AT alignment than all automatic methods. As captions become semantically further from the audio-caption domain, e.g., GC < AC < FC < VC, the AT alignment becomes weaker, and thus leading to worse retrieval performance. The fine-tuned audio encoder generalizes to the out-of-domain Clotho successfully, displaying a trend similar to AudioCaps.

**Supervision improves per-class accuracy in general.** We further plot zero-shot classification accuracy for each audio class (see Figure 5.6 for US8K and Figure 5.7 for ESC50). Clearly, language supervision improves per-class accuracy in general. The highest improvement is observed on "siren" because "siren" rarely appears in image descriptions while GC contains a lot of textual descriptions of "vehicle" audio.

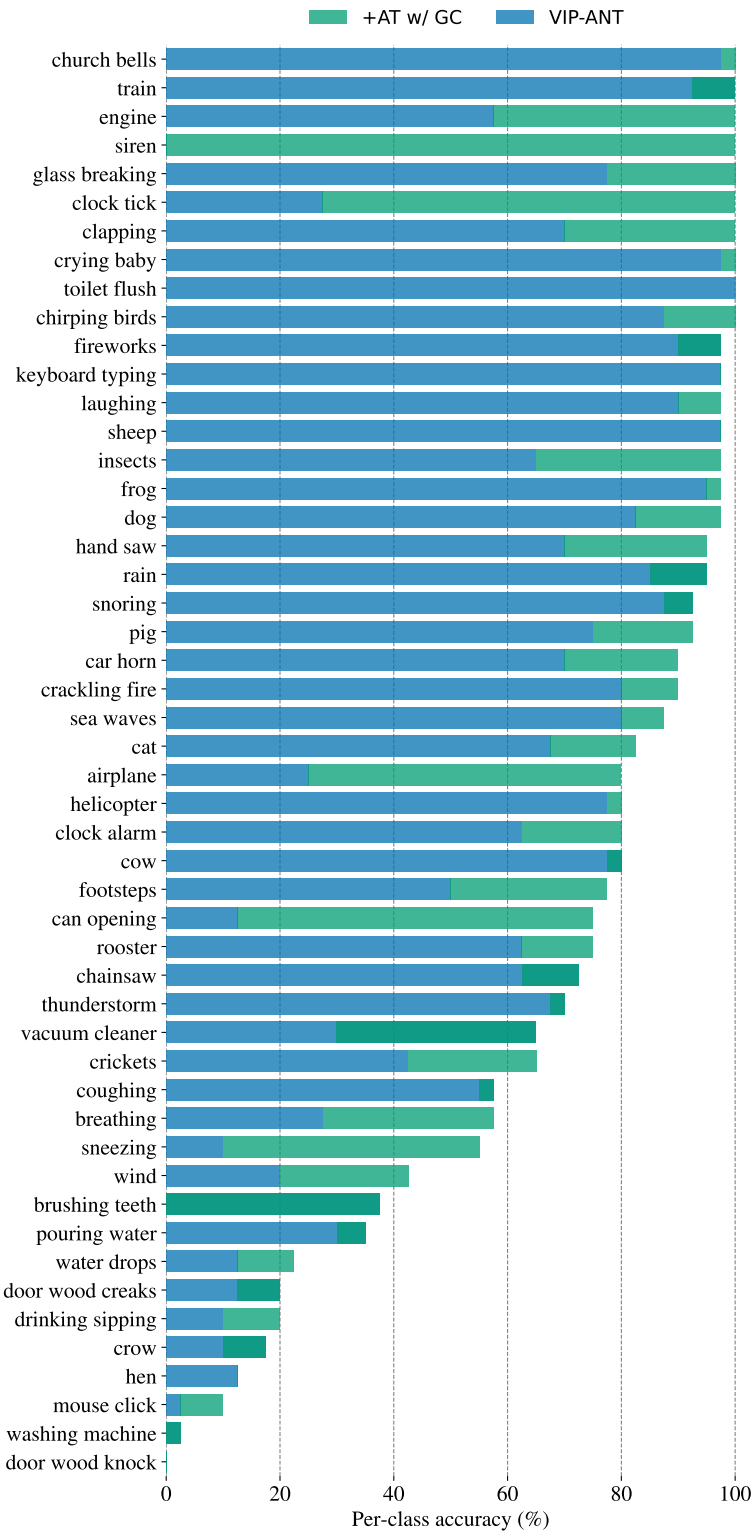



Figure 5.7: Per-class accuracy on US8K.

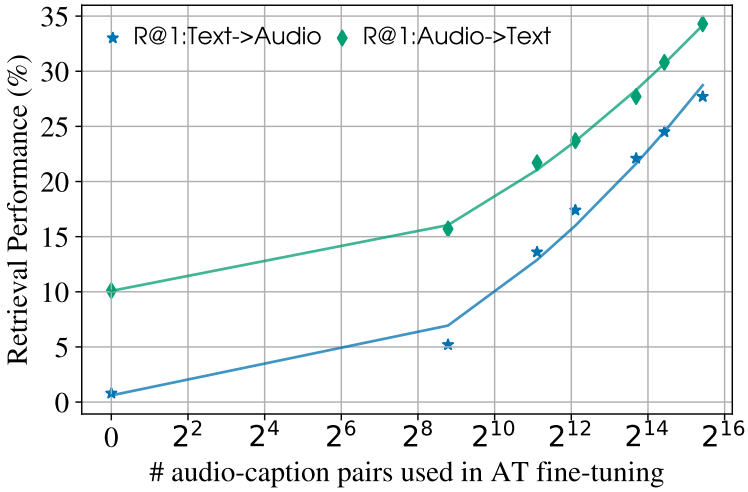
**Asymmetric audio-text retrieval performance.** For Text → Audio retrieval, our unsupervised pivoting model is not as good as on Audio → Text. This could be because

audio is intrinsically more difficult to retrieve with specificity than text in our corpus, e.g., because sound events co-occur (a baby may cry in the street with sirens in the background or in a room with dogs barking), there may be a broader range of captions that accurately describe them. However, it could also be the case that AT alignment is bounded by VT alignment because VA pre-training biases audio representations towards image representations. We check this hypothesis by conducting image-text retrieval on AudioCaps. AudioCaps provides aligned image-audio-text triplets, so we simply replace audio with the corresponding image. We find that the Text $\rightarrow$ Image retrieval performance of CLIP is much better than the Text $\rightarrow$ Audio retrieval performance of VIP~ANT (see OracleAV-CLIP in Table 5.5); it is also close to the Image $\rightarrow$ Text retrieval performance of CLIP, but VIP~ANT exhibits a large gap between the Text $\rightarrow$ Audio retrieval performance and the Audio $\rightarrow$ Text retrieval performance.

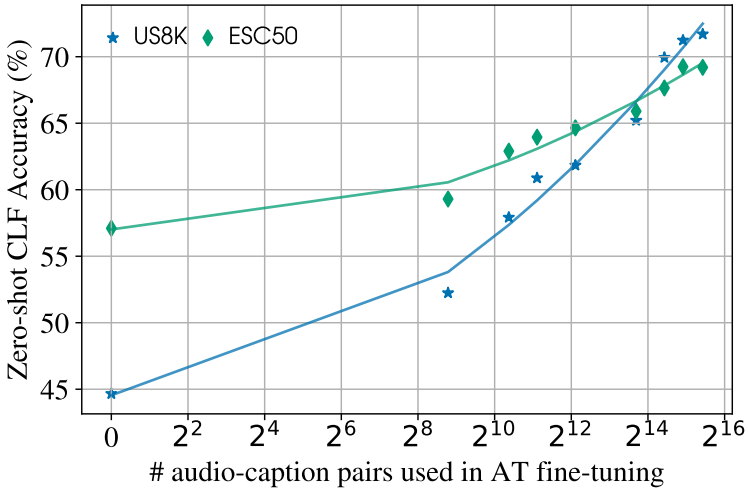
### 5.3.2 Level of Language Supervision

We have observed that AT fine-tuning on AT pairs mined without any additional supervision (e.g., AC, FC, and VC) can improve the AT alignment, but supervised alignments are still the most effective. But: how much supervised data is really needed? To understand the relationship between supervision and performance, we vary the number of gold AT pairs (i.e., training samples of AudioCaps) used for AT fine-tuning. Unsurprisingly, fine-tuning on more aligned AT pairs results in higher audio-text retrieval and zero-shot classification performance (see Figure 5.8). Surprisingly, using only 442 (around 1%) AT pairs of AudioCaps gives rise to as strong AT alignment as VT alignment (*cf.* OracleAV-CLIP in Table 5.5).

As we increase the number of supervised AT pairs used during fine-tuning, we observe a roughly linear relationship between the zero-shot performance and the log of the number of supervised pairs, similar to Kaplan et al. (2020)'s observations regarding Transformers. While it is not clear how reliable extrapolations from this roughly linear trend are, we roughly estimate the amount of annotated AT pairs that is required for the zero-shot performance to equal human parity for ESC50 of 81% Piczak (2015): our estimate is that $2^{21} \approx$ 2M supervised audio caption pairs would be needed. We are hopeful both (1) that larger curated audio-text datasets will become available; and (2) that future work can improve the data efficiency of the pre-training process.




(a) R@1 of AT retrieval on AudioCaps test set.

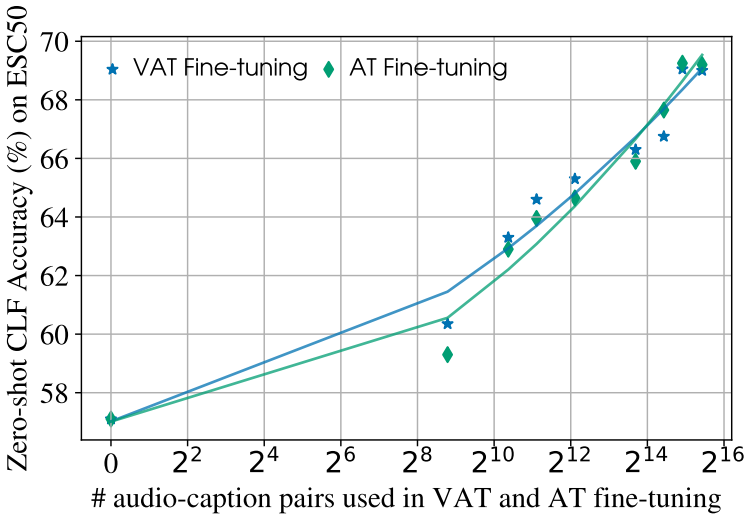



(b) Zero-shot classification (CLF) on ESC50 and US8K.

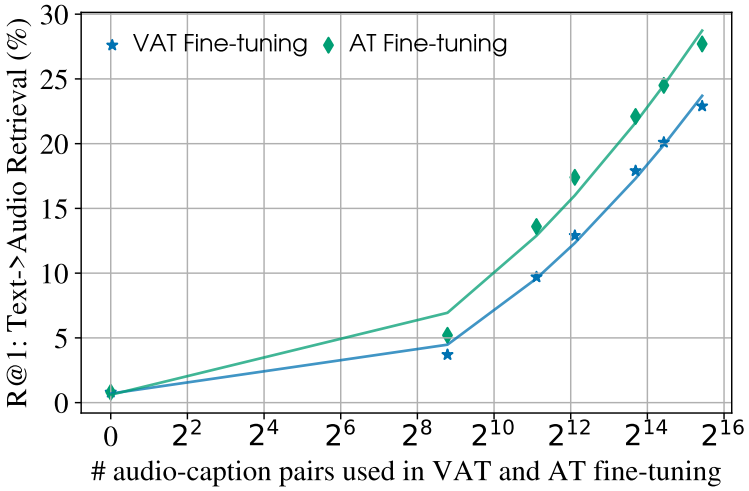
Figure 5.8: Audio retrieval and zero-shot classification performance versus the level of language supervision.

### 5.3.3 VAT versus AT Fine-tuning

Given caption-augmented AudioCaps audio (Kim et al., 2019a), we can improve the pre-trained audio encoder via contrastive vision-audio-text (VAT) fine-tuning and contrastive audio-text (AT) fine-tuning. Figure 5.9 shows a comparison between the two fine-tuning techniques on zero-shot ESC50 classification and AudioCaps audio retrieval. In general, AT fine-tuning results in better results on the two tasks.




(a) Zero-shot classification (CLF) accuracy.




(b) R@1 of audio retrieval with text as the query.

Figure 5.9: Comparing VAT and AT fine-tuning on zero-shot ESC50 classification and AudioCaps audio retrieval.

## 5.4 Supervised Audio Classification

### 5.4.1 Problem Formulation

To perform supervised audio classification, we add a classification head (a linear layer) on top of the pre-trained audio encoder. For *multi-class* classification, the classification head projects the vector representation of an audio clip onto the class space. We fine-

| AS Classification | | | | |
|---|---|---|---|---|
| Dataset | AST | AST$^{\star}$ | AST$^{\dagger}$ | VIP~ANT |
| Unbalanced | | | 43.4 | **44.7** |
| Balanced | 34.7 | 35.8 | 31.4 | **37.9** |
| **US8K and ESC50 Classification** | | | | |
| Dataset | AST-S | AST-P | CLIP | VIP~ANT |
| US8K | | | $82.5_{\pm 6.0}$ | $\mathbf{86.0}_{\pm 2.8}$ |
| ESC50 | $88.7_{\pm 0.7}$ | $95.6_{\pm 0.4}$ | $89.7_{\pm 1.5}$ | $\mathbf{95.7}_{\pm 1.4}$ |

Table 5.7: Multi-label classification mAPs (%) on AS and Supervised audio classification accuracies (%) on ESC50 and US8K. AST, AST-S, and AST-P indicate the results reported by Gong et al. (2021). We follow their suggestions and test their best model (AST$^{\star}$) on our test set. Note that the best model has been trained on the combination of balanced and unbalanced AS training sets. $^{\dagger}$ indicates that we follow the settings of AST and train it on our data. CLIP and VIP~ANT indicate that the audio encoder is initialized from CLIP and from VIP~ANT, respectively.

tune the model by minimizing the cross-entropy loss:

$$\sum_{i} \log p(y^{(i)}|\boldsymbol{a}^{(i)}), \tag{5.8}$$

where $y^{(i)}$ is the gold label of $\boldsymbol{a}^{(i)}$. For supervised *multi-label* classification, the classification head estimates the likelihood that an audio clip has some textual label. We thus minimize the per-label binary cross-entropy loss:

$$\sum_{i} \sum_{l} \log p(l = 1|\boldsymbol{a}^{(i)}), \tag{5.9}$$

where $l$ enumerates all possible audio labels.

### 5.4.2 Experimental Results

**ESC50 and US8K classification.** We initialize the audio encoder from random initialization, CLIP, and VIP~ANT, respectively. Among them, VIP~ANT performs best. It surpasses random initialization and CLIP on both datasets (see Table 5.7).[9] Notably,

[9]We find that VIP~ANT initialization leads to fast convergence, so it can bring better classification results than other initialization methods with the same number of training epochs.

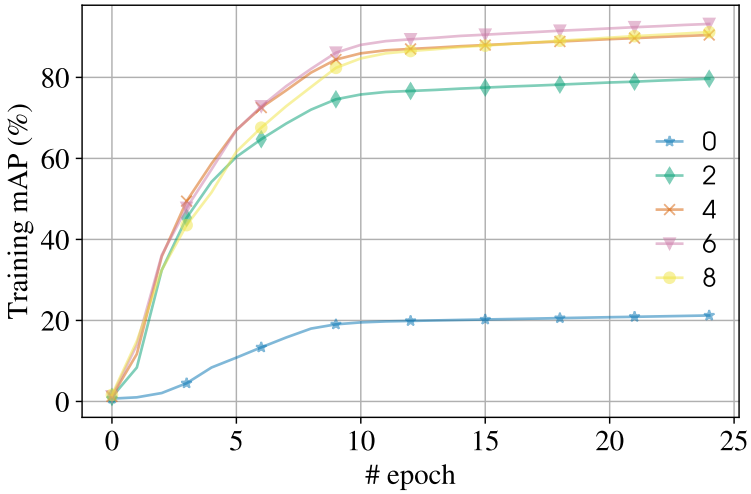



Figure 5.10: Fine-tuning the last $k = 0, 2, 4, 6, 8$ layers of the pre-trained audio encoder for supervised AS classification. mAP is measured on the AS balanced training set after each fine-tuning epoch.

it outperforms the strong baseline AST-P on ESC50 (+0.1%), though AST-P has used gold audio labels for supervised pre-training.

**AS classification.** We consider balanced and unbalanced training for AS classification and train an individual model on the balanced set and the unbalanced set, respectively. Since the audio encoder has been pre-trained on the unbalanced AudioSet training set, it can be directly used without further fine-tuning. Nevertheless, we fine-tune the last $k$ layers of the Transformer architecture of VIP~ANT and investigate whether task-specific fine-tuning helps (see Figure 5.10). When $k = 0$ the model is basically a linear probe; it inspects if contrastive image-audio pre-training learns separable audio representations. Since the unbalanced AudioSet training set has a very skewed audio distribution (e.g., it is dominated by music, speech, and vehicle), it is unsurprising to see that the linear probe has a low performance, i.e., around 20% mAP.

As we increase $k$, i.e., fine-tune more layers, the model exhibits a tendency to over-fit the training set. We use $k = 4$ as a trade-off between under-fitting and over-fitting. Our model achieves the best mAP of 37.9% for balanced training, which surpasses AST by 6.5% (see Table 5.7). While for unbalanced training, we find it crucial to fine-tune the whole model. Again, our model outperforms AST (+1.4% mAP).

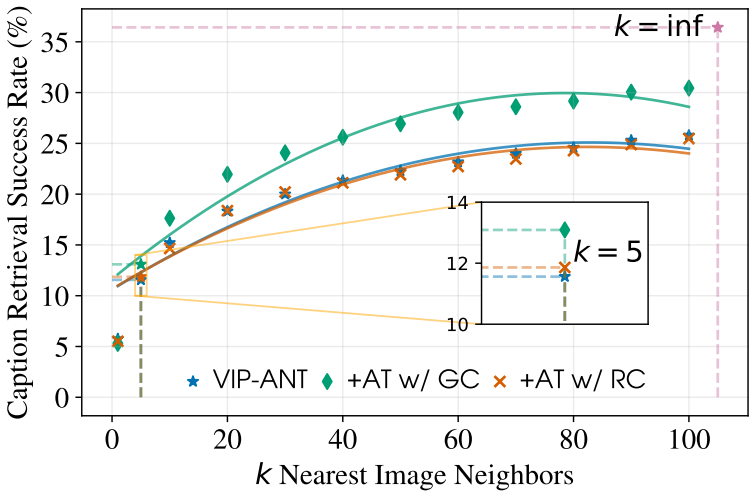
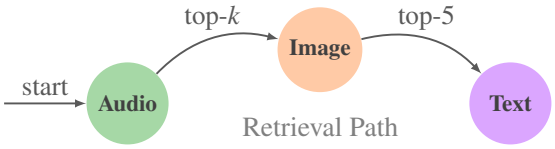



Figure 5.11: Tri-modal pivotability. +AT (w/ GC) and +AT (w/ RC) indicate that VIP~ANT is further fine-tuned on GC and RC, respectively.

## 5.5 Analyzing Tri-modal Representations

To better understand the geometry of tri-modal embeddings of our pivoting, unsupervised curation, and supervised curation, we study how AT fine-tuning influences the tri-modal representation space. Specifically, we analyze VIP~ANT (pivoting), VIP~ANT +AT (w/ RC) (unsupervised curation), and VIP~ANT +AT (w/ GC) (supervised curation) using *pivotability*.




We define pivotability as a metric that measures how likely images can pivot audio and text. We quantify it for each aligned VAT triplet via a two-step retrieval probe. Starting from a given audio clip, we retrieve $k$ nearest image neighbors; for each image neighbor, we retrieve the top 5 nearest captions. Since each audio clip has 5 gold captions, we compute pivotability as the ratio of the number of retrieved gold captions to 5. A gold caption may be retrieved more than once, but we always count it as 1, so pivotability is always between 0 and 1.

We conduct this experiment on the AudioCaps test set. For each $k$, i.e., how many images will be retrieved for a given audio clip, we average pivotability scores over all test triplets (see Figure 5.11).

| | |
|---|---|
| +AT w/ GC | "female speech, woman speaking", "narration, monologue", "vibration" |
| +AT w/ RC | "bee, wasp, etc.", "female speech, woman speaking", "insect", "narration, monologue", "vibration" |

Table 5.8: Compared against VIP~ANT, the two fine-tuned versions of VIP~ANT find novel audio categories in pivotable AT pairs.

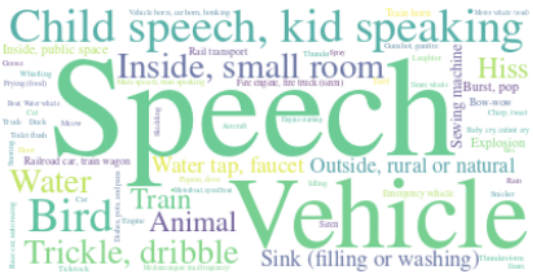



Figure 5.12: Categories of the audio that can be pivoted with text by images. The larger text indicates that the related audio is more likely to be pivoted with text.

**Which pairs are pivotable?** To study what kinds of audio are more likely to be pivoted with text by images, we set $k = 5$ (i.e., 5 images will be retrieved for each given audio clip). We consider an AT pair pivotable if at least 3 out of 5 gold captions of the audio clip are retrieved, i.e., pivotability is equal to or larger than 0.6. Figure 5.12 illustrates the categories of the audio clips in pivotable AT pairs. Unsurprisingly, speech and vehicle audio are more pivotable because the two categories are among the top three frequent categories in AS.[10] Since AT fine-tuning improves Audio → Image retrieval, we wonder if it could also help find novel categories of audio that can be pivoted with text. We find that this is indeed the case (see Table 5.8). For example, VIP~ANT +AT (w/ GC) finds more fine-grained speech categories because most AT pairs in AudioCaps are about speech. In contrast, VIP~ANT +AT (w/ RC) finds two additional novel insect categories, presumably because RC suffers from less data bias than GC.

[10]Music is the second most frequent category in AS. It is not shown in the figure because AudioCaps excludes all music audio.

## 5.6 Summary

We have presented VIP~ANT for unsupervised audio-text alignment induction. Based on the pivoting idea, our model learns image-text alignment and image-audio alignment explicitly and separately via bi-modal contrastive pre-training. The image modality is shared between the two and thus pivots audio and text in the tri-modal embedding space implicitly, without using any paired audio-text data. We empirically find that our model achieves strong performance on zero-shot audio-text tasks. We further strengthen the audio-text alignment by using varying kinds of audio-text supervision. Experimental results show that even unaligned audio-caption pairs can help.

# Chapter 6

# Conclusions

In this thesis, we have investigated unsupervised structured representation learning in the multimodal setting. Underlying our studies is the need for interpretable and controllable machine-learning systems with human-like compositional generalizability. To fulfill the need, we resorted to structured representations and addressed practical yet challenging learning settings, including unsupervised learning and multimodal learning. Unsupervised learning gets away with costly manual annotations in favor of natural supervision, while multimodal learning encourages the incorporation of world knowledge into machine learning systems. Together they lead to a more realistic learning setting. Revolving around the new setting, we identify and cope with several challenges. Our contributions span two dimensions: model development and data collection; the findings from each dimension can be summarized as follows:

- **Model.** Both neural-symbolic (hybrid) and structure-aware (connectionist) structure induction models can be augmented with and benefit from multimodal learning (i.e., image-text alignment). We integrate combine image-text learning with text-only grammar induction to enable a syntactic parser to learn from visual groundings (Chapter 3), and we customize a neural module for scene graph induction and integrate it into a visually-grounded masked language model (Chapter 4).

- **Data.** Multimodal alignment synthesis is enabled by structured representations, e.g., we use scene graph representations of images and functional program representations of captions to create artificial image-captioning data (Chapter 4). Moreover, multimodal alignment can be induced from naturally aligned modalities in an unsupervised manner, e.g., we collect audio-text alignment from image-text alignment and image-audio alignment without using paired audio and text (Chapter 5).

An alternative way to view our contributions is from the modeling perspective: each of Chapters 3–5 features a model that addresses special challenges faced by multimodal and unsupervised structured representation learning. We summarize these models and acknowledge their limitations (Section 6.1), and further discuss future directions (Section 6.2).

## 6.1 Summary of Models

**Syntactic Parser:** Visually Grounded Grammar Induction (Chapter 3). We have presented VC~PCFG, short for Visually-grounded Compound PCFG, for inducing constituency grammars with visual image supervision. VC~PCFG integrate PCFG for language modeling with contrastive image-text learning. The joint learning paradigm allows for deriving learning signals both from text alone and from image-text alignment. By choosing PCFG as the parsing model, we are able to optimize VC~PCFG, a neural-symbolic model, within an end-to-end fully-differentiable framework. We empirically find that VC~PCFG improves over PCFG that is learned from text alone or from only visual groundings.

**Image Parser:** Textually Grounded Scene Graph Induction (Chapter 4). We have presented, VG~MLM, short for Visually Grounded Masked Language Model, for inducing scene graphs from natural language supervision. VG~MLM has a Transformer encoder-decoder architecture; the encoder models visual contexts, and the decoder is responsible for masked language modeling. Since we formulate scene graph induction as a visual relation prediction task, we tailor a computational module to predict words conditioning on object pairs, which can be integrated into VG~MLM seamlessly. VG~MLM demonstrates decent performance on an artificial image-captioning dataset, as measured by our proposed unsupervised evaluation metric.

**Multimodal Alignment Curator:** Unsupervised Audio-Text Alignment Induction (Chapter 5). We have presented VIP~ANT, short for VIsually Pivoted Audio and(N) Text, for unsupervised curation of audio-text alignment. VIP~ANT adopts a contrastive pre-training framework, where bimodal image-text and image-audio models are trained from abundantly available image-text and image-audio co-occurrences, respectively. By sharing the visual modality between the two alignment models, we can link audio and text in a trimodal vector space. We find that the learned audio-text alignment is

capable of unsupervised zero-shot audio classification; it can be further improved by finetuning on automatically curated audio-text data.

## 6.2 Future Directions

**Grounded grammar induction beyond visual groundings.** VC~PCFG, which is proposed for grounded grammar induction, has been shown to be capable of exploiting visual groundings, but the underlying learning paradigm of jointly optimizing a language modeling objective and minimizing an image-text alignment loss is generalizable to many other types of groundings. For instance, our VC~PCFG has been extended to the video-grounded setting, where diverse groundings such as audio, speech, action, and object groundings have been used (Zhang et al., 2021). Using fine-grained visual groundings such as objects has also led to joint induction of image and text structures (Hong et al., 2021). Despite the promising progress, rigorous studies on the role of groundings in grammar induction are still limited, presumably because crossmodal analysis is hard. One possible way is to slightly simplify the grounded setting, e.g., rather than using crossmodal groundings, we can ground the target language in a foreign language via supervised machine translation data. We can further perform grammar induction for the foreign language, leading to a framework for bilingual grammar induction from sentence alignment.

**Scene graph induction within an end-to-end framework.** VG~MLM, which is proposed for scene graph induction from language supervision, relies on the assumption that objects have been detected by using a pre-trained detector, but the pipeline-style method is error-prone. A desirable way would be to induce objects in an unsupervised way and integrate object induction with VG~MLM. This direction becomes especially appealing and promising given the recent stunning progress in object-centric representation learning, e.g., MONet (Burgess et al., 2019) and SlotAttention (Locatello et al., 2020) have demonstrated that it is possible to induce objects from pixel images alone. Integrating an unsupervised object discovery module with VG~MLM will lead to a joint learning paradigm similar to that of VC~PCFG. Specifically, we can jointly minimize a reconstruction loss for object discovery and optimize the masked language modeling objective for object and relation classification. Moreover, recent work has shown that natural language supervision helps with unsupervised object segmentation (Xu et al., 2022); it is promising that the two learning tasks/objectives would benefit each other,

making the joint learning paradigm more efficient and intriguing.

**Efficient neural-symbolic (hybrid) models.** We have described hybrid models such as VC-PCFG for grounded grammar induction. As latent variable models, VC-PCFG admits tractable learning and inference via dynamic programming but still suffers from high computational (i.e., time and space) complexities even with automatic differentiation (Eisner, 2016) and parallel implementations (Rush, 2020). A future direction would be to devise more efficient dynamic programming algorithms for learning and inference, e.g., approximate inference with latent-variable PCFG (Cohen et al., 2013). Alternatively, we may trade off inexact learning for efficiency. As we have discussed in Chapter 2, though Monte Carlo sampling-based optimization methods lead to inexact learning, they are generally more efficient, without requiring enumerating all possible latent structures. For example, rather than computing the expected crossmodal alignment loss, we may use a random sample to estimate it when efficient sampling algorithms exist. With the Perturb-and-MAP and straight-through (or continuous relaxation) techniques (Papandreou and Yuille, 2011; Bengio et al., 2013), we can still learn discrete latent-variable models within an end-to-end differentiable framework.

**Towards more practical multimodal and structured learning.** We evaluated our structure induction models only on relatively easy data. Specifically, the image captions we used for unsupervised grammar induction represent a restricted subdomain of language data and lack syntactic diversity; and the artificial images we used in unsupervised scene graph induction contain clear object boundaries and involve only four spatial relations. Thus, though our experimental results suggest that: the proposed multimodal neural-symbolic and structure-aware models are, in a way, effective in inferring hidden structures and capable of exploiting crossmodal alignment, these experiments might be relatively restrictive.

In contrast, recent self-supervised language/vision models are trained on large-scale single-modality data and have fewer structural biases but, surprisingly, exhibit the capability of learning structured representations. For example, (1) in the language domain, Wu et al. (2020) show that syntactic trees can be extracted from pre-trained BERT models (Devlin et al., 2019) and obtain parsing performance competitive to models that have explicit structural biases (e.g., neural PCFG); and (2) in the vision domain, Caron et al. (2021) find that visual concepts emerge in self-supervised vision Transformers without using any symbolic representational biases. Moreover, when

finetuned/prompted for practical tasks, these pre-trained unimodal structure-free neural models demonstrate surprisingly strong performance and reasonable generalizability.

Nevertheless, as we have discussed in Chapter 2, multimodal learning is believed to be one of the most important directions in machine learning (Bisk et al., 2020; Bender and Koller, 2020), and more and more research has shown that multimodal learning helps with both image and language understanding (Radford et al., 2021; Ramesh et al., 2021; Rombach et al., 2022). In the future, we would like to extend our multimodal models to more practical settings so that we can thoroughly examine the role of multimodal learning in unsupervised structure induction. Moreover, though in some practical applications, modeling structures does not give rise to very substantial benefits when considering both performance improvements and learning efficiency, structured modeling is indeed needed in cases where interpretability, controllability, and generalizability are the top priorities, as we have discussed at the beginning of this thesis. We anticipate that incorporating structured modeling into current structure-free machine learning systems will improve them further, and meanwhile, we acknowledge the learning inflexibility it may cause; future work would be well-suited to finding a balance between structured modeling and flexible learning.

# Bibliography

Akbari, H., Yuan, L., Qian, R., Chuang, W.-H., Chang, S.-F., Cui, Y., and Gong, B. (2021). VATT: Transformers for multimodal self-supervised learning from raw video, audio and text. In *Thirty-Fifth Conference on Neural Information Processing Systems*.

Alayrac, J.-B., Recasens, A., Schneider, R., Arandjelović, R., Ramapuram, J., De Fauw, J., Smaira, L., Dieleman, S., and Zisserman, A. (2020). Self-supervised multimodal versatile networks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 25–37. Curran Associates, Inc.

Allerton, D. (2016). *Essentials of Grammatical Theory: A Consensus View of Syntax and Morphology*. Routledge Library Editions: Syntax. Taylor & Francis.

Amari, S.-i. (1998). Natural Gradient Works Efficiently in Learning. *Neural Computation*, 10(2):251–276.

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., and Parikh, D. (2015). Vqa: Visual question answering. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2425–2433.

Baevski, A., Hsu, W.-N., Xu, Q., Babu, A., Gu, J., and Auli, M. (2022). data2vec: A general framework for self-supervised learning in speech, vision and language. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S., editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 1298–1312. PMLR.

Baker, J. K. (1979). Trainable grammars for speech recognition. *The Journal of the Acoustical Society of America*, 65(S1):S132–S132.

Baker, M. C. (2008). *The atoms of language: The mind's hidden rules of grammar*. Basic books.

Baldassarre, F., Smith, K., Sullivan, J., and Azizpour, H. (2020). Explanation-based weakly-supervised learning of visual relations with graph networks. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M., editors, *Computer Vision – ECCV 2020*, pages 612–630, Cham. Springer International Publishing.

Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., and Schneider, N. (2013). Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Barber, D. and Wiegerinck, W. (1998). Tractable variational structures for approximating graphical models. In Kearns, M., Solla, S., and Cohn, D., editors, *Advances in Neural Information Processing Systems*, volume 11. MIT Press.

Barrett, D., Hill, F., Santoro, A., Morcos, A., and Lillicrap, T. (2018). Measuring abstract reasoning in neural networks. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 511–520. PMLR.

Barsalou, L. W. (1999). Perceptual symbol systems. *Behavioral and Brain Sciences*, 22(4):577–660.

Bender, E. M. and Koller, A. (2020). Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Bengio, Y., Léonard, N., and Courville, A. C. (2013). Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432.

Biederman, I. (1987). Recognition-by-components: a theory of human image understanding. *Psychol Rev*, 94(2):115–147.

Bies, A., Mott, J., Warner, C., and Kulick, S. (2012). English web treebank. *Linguistic Data Consortium, Philadelphia, PA*.

Bisk, Y., Holtzman, A., Thomason, J., Andreas, J., Bengio, Y., Chai, J., Lapata, M., Lazaridou, A., May, J., Nisnevich, A., Pinto, N., and Turian, J. (2020). Experience grounds language. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8718–8735, Online. Association for Computational Linguistics.

Blake, A., Kohli, P., and Rother, C. (2011). *Markov Random Fields for Vision and Image Processing*. The MIT Press.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

Blender Online Community (2016). Blender - a 3d modelling and rendering package. Technical report. https://www.blender.org.

Blevins, T., Levy, O., and Zettlemoyer, L. (2018). Deep RNNs encode soft hierarchical syntax. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 14–19, Melbourne, Australia. Association for Computational Linguistics.

Boddapati, V., Petef, A., Rasmusson, J., and Lundberg, L. (2017). Classifying environmental sounds using image recognition networks. *Procedia Computer Science*, 112:2048–2056. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 21st International Conference, KES-20176-8 September 2017, Marseille, France.

Booth, T. and Thompson, R. (1973). Applying probability measures to abstract languages. *IEEE Transactions on Computers*, C-22(5):442–450.

Bordes, P., Zablocki, E., Soulier, L., Piwowarski, B., and Gallinari, P. (2019). Incorporating visual semantics into sentence representations within a grounded space. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 696–707, Hong Kong, China. Association for Computational Linguistics.

Brill, E. (1993). Automatic grammar induction and parsing free text: A transformation-based approach. In *31st Annual Meeting of the Association for Computational*

*Linguistics*, pages 259–265, Columbus, Ohio, USA. Association for Computational Linguistics.

Brown, P. F., Lai, J. C., and Mercer, R. L. (1991). Aligning sentences in parallel corpora. In *29th Annual Meeting of the Association for Computational Linguistics*, pages 169–176, Berkeley, California, USA. Association for Computational Linguistics.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Bruni, E., Tran, N. K., and Baroni, M. (2014). Multimodal distributional semantics. *J. Artif. Int. Res.*, 49(1):1–47.

Burgess, C. P., Matthey, L., Watters, N., Kabra, R., Higgins, I., Botvinick, M. M., and Lerchner, A. (2019). Monet: Unsupervised scene decomposition and representation. *CoRR*, abs/1901.11390.

Caron, M., Touvron, H., Misra, I., Jegou, H., Mairal, J., Bojanowski, P., and Joulin, A. (2021). Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9630–9640.

Carroll, G. and Charniak, E. (1992). *Two experiments on learning probabilistic dependency grammars from corpora*. AAAI Press.

Chao, Y.-W., Wang, Z., He, Y., Wang, J., and Deng, J. (2015). Hico: A benchmark for recognizing human-object interactions in images. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1017–1025.

Chen, T., Yu, W., Chen, R., and Lin, L. (2019). Knowledge-embedded routing network for scene graph generation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6156–6164.

Chen, X., Fang, H., Lin, T., Vedantam, R., Gupta, S., Dollár, P., and Zitnick, C. L. (2015). Microsoft COCO captions: Data collection and evaluation server. *CoRR*, abs/1504.00325.

Chiang, D. (2005). A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270, Ann Arbor, Michigan. Association for Computational Linguistics.

Choi, J., Yoo, K. M., and Lee, S.-g. (2018). Learning to compose task-specific tree structures. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on Information Theory*, 2(3):113–124.

Chomsky, N. (1957). *Syntactic Structures*. De Gruyter Mouton, Berlin, Boston.

Chomsky, N. (1959). On certain formal properties of grammars. *Information and Control*, 2(2):137–167.

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. The MIT Press, 50 edition.

Clark, A. (2001). Unsupervised induction of stochastic context-free grammars using distributional clustering. In *Proceedings of the ACL 2001 Workshop on Computational Natural Language Learning (ConLL)*.

Cocke, J. (1969). *Programming Languages and Their Compilers: Preliminary Notes*. New York University, USA.

Cohen, S. and Smith, N. A. (2009). Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 74–82, Boulder, Colorado. Association for Computational Linguistics.

Cohen, S. B., Satta, G., and Collins, M. (2013). Approximate PCFG parsing using tensor decomposition. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 487–496, Atlanta, Georgia. Association for Computational Linguistics.

Csiba, D. and Richtárik, P. (2018). Importance sampling for minibatches. *Journal of Machine Learning Research*, 19(27):1–21.

Dai, B., Zhang, Y., and Lin, D. (2017a). Detecting visual relationships with deep relational networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3298–3308.

Dai, W., Dai, C., Qu, S., Li, J., and Das, S. (2017b). Very deep convolutional neural networks for raw waveforms. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 421–425.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

Deng, Y., Kim, Y., Chiu, J., Guo, D., and Rush, A. (2018). Latent alignment and variational attention. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Desai, C., Ramanan, D., and Fowlkes, C. (2010). Discriminative models for static human-object interactions. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pages 9–16.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dhamo, H., Farshad, A., Laina, I., Navab, N., Hager, G. D., Tombari, F., and Rupprecht, C. (2020). Semantic image manipulation using scene graphs. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5212–5221.

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M., and Hon, H.-W. (2019). Unified language model pre-training for natural language understanding and generation. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Drossos, K., Lipping, S., and Virtanen, T. (2020). Clotho: an audio captioning dataset. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 736–740.

Drozdov, A., Verga, P., Chen, Y.-P., Iyyer, M., and McCallum, A. (2019). Unsupervised labeled parsing with deep inside-outside recursive autoencoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1507–1512, Hong Kong, China. Association for Computational Linguistics.

Eisner, J. (2016). Inside-outside and forward-backward algorithms are just backprop (tutorial paper). In *Proceedings of the Workshop on Structured Prediction for NLP*, pages 1–17, Austin, TX. Association for Computational Linguistics.

Elliott, D., Frank, S., Sima'an, K., and Specia, L. (2016). Multi30K: Multilingual English-German image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74, Berlin, Germany. Association for Computational Linguistics.

Elliott, D. and Keller, F. (2013). Image description using visual dependency representations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1292–1302, Seattle, Washington, USA. Association for Computational Linguistics.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2):179–211.

Eslami, S. M. A., Heess, N., Weber, T., Tassa, Y., Szepesvari, D., kavukcuoglu, k., and Hinton, G. E. (2016). Attend, infer, repeat: Fast scene understanding with generative models. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 3225–3233. Curran Associates, Inc.

Essen, D. C. V., Anderson, C. H., and Felleman, D. J. (1992). Information processing in the primate visual system: An integrated systems perspective. *Science*, 255(5043):419–423.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. Bradford Books.

Fincher-Kiefer, R. (2001). Perceptual components of situation models. *Memory & Cognition*, 29(2):336–343.

Font, F., Roma, G., and Serra, X. (2013). Freesound technical demo. In *Proceedings of the 21st ACM International Conference on Multimedia*, MM '13, page 411–412, New York, NY, USA. Association for Computing Machinery.

Friesen, A. L. and Domingos, P. M. (2018). Submodular field grammars: Representation, inference, and application to image parsing. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Galleguillos, C., Rabinovich, A., and Belongie, S. (2008). Object categorization using co-occurrence, location and appearance. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

Ganchev, K., Graça, J., Gillenwater, J., and Taskar, B. (2010). Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11(67):2001–2049.

Gemmeke, J. F., Ellis, D. P. W., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M., and Ritter, M. (2017). Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780.

Geng, S., Yuan, J., Tian, Y., Chen, Y., and Zhang, Y. (2023). HiCLIP: Contrastive language-image pretraining with hierarchy-aware attention. In *The Eleventh International Conference on Learning Representations*.

Gernsbacher, M. A. (2015). Video captions benefit everyone. *Policy Insights from the Behavioral and Brain Sciences*, 2(1):195–202. PMID: 28066803.

Gershman, S. and Goodman, N. (2014). Amortized inference in probabilistic reasoning. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 36.

Gildea, D. and Jurafsky, D. (2000). Automatic labeling of semantic roles. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 512–520, Hong Kong. Association for Computational Linguistics.

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, page 1440–1448, USA. IEEE Computer Society.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In Teh, Y. W. and Titterington, M., editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.

Glynn, P. W. (1987). Likelilood ratio gradient estimation: An overview. In *Proceedings of the 19th Conference on Winter Simulation*, WSC '87, page 366–375, New York, NY, USA. Association for Computing Machinery.

Gold, E. M. (1967). Language identification in the limit. *Information and Control*, 10(5):447–474.

Goldberg, Y. and Elhadad, M. (2010). An efficient algorithm for easy-first non-directional dependency parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 742–750, Los Angeles, California. Association for Computational Linguistics.

Gong, Y., Chung, Y.-A., and Glass, J. (2021). AST: Audio Spectrogram Transformer. In *Proc. Interspeech 2021*, pages 571–575.

Gonzalez, R. and Woods, R. (2018). *Digital Image Processing*. Pearson.

Gopalan, P. K., Gerrish, S., Freedman, M., Blei, D., and Mimno, D. (2012). Scalable inference of overlapping communities. In Pereira, F., Burges, C., Bottou, L.,

and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Gordon, J. and Van Durme, B. (2013). Reporting bias and knowledge acquisition. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*, AKBC '13, page 25–30, New York, NY, USA. Association for Computing Machinery.

Gould, S., Fulton, R., and Koller, D. (2009). Decomposing a scene into geometric and semantically consistent regions. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1–8.

Gould, S., Rodgers, J., Cohen, D., Elidan, G., and Koller, D. (2008). Multi-class segmentation with relative location prior. *Int. J. Comput. Vision*, 80(3):300–316.

Greff, K., van Steenkiste, S., and Schmidhuber, J. (2020). On the binding problem in artificial neural networks. *CoRR*, abs/2012.05208.

Gu, J., Joty, S., Cai, J., Zhao, H., Yang, X., and Wang, G. (2019a). Unpaired image captioning via scene graph alignments. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10322–10331.

Gu, J., Zhao, H., Lin, Z., Li, S., Cai, J., and Ling, M. (2019b). Scene graph generation with external knowledge and image reconstruction. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1969–1978.

Guzhov, A., Raue, F., Hees, J., and Dengel, A. (2021). Esresnet: Environmental sound classification based on visual domain models. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4933–4940.

Guzhov, A., Raue, F., Hees, J., and Dengel, A. (2022). Audioclip: Extending clip to image, text and audio. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 976–980.

Harnad, S. (1990). The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1):335–346.

Harris, Z. S. (1954). Distributional structure. *WORD*, 10(2-3):146–162.

Havrylov, S., Kruszewski, G., and Joulin, A. (2019). Cooperative learning of disjoint syntax and semantics. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1118–1128, Minneapolis, Minnesota. Association for Computational Linguistics.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. (2022). Masked autoencoders are scalable vision learners. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15979–15988.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hendrycks, D. and Gimpel, K. (2016). Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415.

Hitschler, J., Schamoni, S., and Riezler, S. (2016). Multimodal pivots for image caption translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2399–2409, Berlin, Germany. Association for Computational Linguistics.

Hockenmaier, J. and Steedman, M. (2007). CCGbank: A corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396.

Hodosh, M., Young, P., and Hockenmaier, J. (2013). Framing image description as a ranking task: Data, models and evaluation metrics. *J. Artif. Int. Res.*, 47(1):853–899.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *Journal of Machine Learning Research*, 14(40):1303–1347.

Hong, Y., Li, Q., Zhu, S.-C., and Huang, S. (2021). Vlgrammar: Grounded grammar induction of vision and language. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1665–1674.

Hopcroft, J. E., Motwani, R., and Ullman, J. D. (2006). *Introduction to Automata Theory, Languages, and Computation (3rd Edition)*. Addison-Wesley Longman Publishing Co., Inc., USA.

Htut, P. M., Phang, J., Bordia, S., and Bowman, S. R. (2019). Do attention heads in BERT track syntactic dependencies? *CoRR*, abs/1911.12246.

Hu, R., Rohrbach, M., Andreas, J., Darrell, T., and Saenko, K. (2017). Modeling relationships in referential expressions with compositional modular networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4418–4427.

Hudson, D. and Manning, C. D. (2019a). Learning by abstraction: The neural state machine. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Hudson, D. A. and Manning, C. D. (2019b). Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6693–6702.

Jang, E., Gu, S., and Poole, B. (2017). Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*.

Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q., Sung, Y.-H., Li, Z., and Duerig, T. (2021). Scaling up visual and vision-language representation learning with noisy text supervision. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4904–4916. PMLR.

Jiang, Y., Han, W., and Tu, K. (2016). Unsupervised neural dependency parsing. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 763–771, Austin, Texas. Association for Computational Linguistics.

Jin, Y. and Geman, S. (2006). Context and hierarchy in a probabilistic image model. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2145–2152.

Johnson, J., Gupta, A., and Fei-Fei, L. (2018). Image generation from scene graphs. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1219–1228.

Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C. L., and Girshick, R. (2017). Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1988–1997.

Johnson, J., Krishna, R., Stark, M., Li, L.-J., Shamma, D. A., Bernstein, M. S., and Fei-Fei, L. (2015). Image retrieval using scene graphs. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3668–3678.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. (2021). Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589.

Jurafsky, D. and Martin, J. H. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, USA, 1st edition.

Kamath, A., Singh, M., LeCun, Y., Synnaeve, G., Misra, I., and Carion, N. (2021). Mdetr - modulated detection for end-to-end multi-modal understanding. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1760–1770, Los Alamitos, CA, USA. IEEE Computer Society.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. (2020). Scaling laws for neural language models. *CoRR*, abs/2001.08361.

Kasami, T. (1966). An efficient recognition and syntax-analysis algorithm for context-free languages. *Coordinated Science Laboratory Report no. R-257*.

Kate, R. J., Wong, Y. W., and Mooney, R. J. (2005). Learning to transform natural to formal languages. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 3*, AAAI'05, page 1062–1068. AAAI Press.

Kawakami, K., Dyer, C., and Blunsom, P. (2019). Learning to discover, ground and use words with segmental neural language models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6429–6441, Florence, Italy. Association for Computational Linguistics.

Keysers, D., Schärli, N., Scales, N., Buisman, H., Furrer, D., Kashubin, S., Momchev, N., Sinopalnikov, D., Stafiniak, L., Tihon, T., Tsarkov, D., Wang, X., van Zee, M., and Bousquet, O. (2020). Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations*.

Kiela, D., Conneau, A., Jabri, A., and Nickel, M. (2018). Learning visually grounded sentence representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 408–418, New Orleans, Louisiana. Association for Computational Linguistics.

Kiela, D., Hill, F., Korhonen, A., and Clark, S. (2014). Improving multi-modal representations using image dispersion: Why less is sometimes more. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 835–841, Baltimore, Maryland. Association for Computational Linguistics.

Kim, B. and Pardo, B. (2019). Sound event detection using point-labeled data. In *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 1–5.

Kim, C. D., Kim, B., Lee, H., and Kim, G. (2019a). AudioCaps: Generating captions for audios in the wild. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 119–132, Minneapolis, Minnesota. Association for Computational Linguistics.

Kim, N. and Linzen, T. (2020). COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empir-*

*ical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105, Online. Association for Computational Linguistics.

Kim, T., Choi, J., Edmiston, D., and goo Lee, S. (2020). Are pre-trained language models aware of phrases? simple but strong baselines for grammar induction. In *International Conference on Learning Representations*.

Kim, Y., Dyer, C., and Rush, A. (2019b). Compound probabilistic context-free grammars for grammar induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2369–2385, Florence, Italy. Association for Computational Linguistics.

Kim, Y., Rush, A., Yu, L., Kuncoro, A., Dyer, C., and Melis, G. (2019c). Unsupervised recurrent neural network grammars. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1105–1117, Minneapolis, Minnesota. Association for Computational Linguistics.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*.

Kingma, D. P., Mohamed, S., Jimenez Rezende, D., and Welling, M. (2014). Semi-supervised learning with deep generative models. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 3581–3589. Curran Associates, Inc.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *International Conference on Learning Representations*.

Kipf, T., van der Pol, E., and Welling, M. (2020). Contrastive learning of structured world models. In *International Conference on Learning Representations*.

Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Kitaev, N. and Klein, D. (2018). Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

Klein, D. and Manning, C. (2004). Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 478–485, Barcelona, Spain.

Klein, D. and Manning, C. D. (2002). A generative constituent-context model for improved grammar induction. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 128–135, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Kojima, N., Averbuch-Elor, H., Rush, A., and Artzi, Y. (2020). What is learned in visually grounded neural syntax acquisition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2615–2635, Online. Association for Computational Linguistics.

Kolmogorov, V. and Zabin, R. (2004). What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159.

Kong, Q., Xu, Y., Wang, W., and Plumbley, M. D. (2018). Audio set classification with attention model: A probabilistic perspective. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 316–320.

Korbar, B., Tran, D., and Torresani, L. (2018). Cooperative learning of audio and video models from self-supervised synchronization. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Kramár, J., Eccles, T., Gemp, I., Tacchetti, A., McKee, K. R., Malinowski, M., Graepel, T., and Bachrach, Y. (2022). Negotiation and honesty in artificial intelligence methods for the board game of diplomacy. *Nature Communications*, 13(1):7214.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., Bernstein, M. S., and Fei-Fei, L. (2017). Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., and Wein-

berger, K., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Kulkarni, G., Premraj, V., Dhar, S., Li, S., Choi, Y., Berg, A. C., and Berg, T. L. (2011). Baby talk: Understanding and generating simple image descriptions. In *CVPR 2011*, pages 1601–1608.

Kumar, A., Khadkevich, M., and Fügen, C. (2018). Knowledge transfer from weakly labeled audio using convolutional neural network for sound events and scenes. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 326–330.

Kumar, A. and Raj, B. (2017). Audio event and scene recognition: A unified approach using strongly and weakly labeled data. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 3475–3482.

Lake, B. and Baroni, M. (2018). Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2873–2882. PMLR.

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40:e253.

Lari, K. and Young, S. (1990). The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language*, 4(1):35 – 56.

Lazaridou, A., Dinu, G., and Baroni, M. (2015). Hubness and pollution: Delving into cross-space mapping for zero-shot learning. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 270–280, Beijing, China. Association for Computational Linguistics.

Le, P. and Zuidema, W. (2015). The forest convolutional network: Compositional distributional semantics with a neural chart and without binarization. In *Proceedings*

*of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1155–1164, Lisbon, Portugal. Association for Computational Linguistics.

Le, Q., Ranzato, M., Monga, R., Devin, M., Chen, K., Corrado, G., Dean, J., and Ng, A. (2012). Building high-level features using large scale unsupervised learning. In Langford, J. and Pineau, J., editors, *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, ICML '12, pages 81–88, New York, NY, USA. Omnipress.

Lei, J., Li, L., Zhou, L., Gan, Z., Berg, T. L., Bansal, M., and Liu, J. (2021). Less is more: Clipbert for video-and-language learning via sparse sampling. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7327–7337.

Li, B., Weinberger, K. Q., Belongie, S., Koltun, V., and Ranftl, R. (2022a). Language-driven semantic segmentation. In *International Conference on Learning Representations*.

Li, L. H., Yatskar, M., Yin, D., Hsieh, C.-J., and Chang, K.-W. (2020a). What does BERT with vision look at? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5265–5275, Online. Association for Computational Linguistics.

Li, X. and Jiang, S. (2019). Know more say less: Image captioning based on scene graphs. *IEEE Transactions on Multimedia*, 21(8):2117–2130.

Li, X., Yin, X., Li, C., Zhang, P., Hu, X., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., Choi, Y., and Gao, J. (2020b). Oscar: Object-semantics aligned pre-training for vision-language tasks. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M., editors, *Computer Vision – ECCV 2020*, pages 121–137, Cham. Springer International Publishing.

Li, Y., Ouyang, W., Zhou, B., Shi, J., Zhang, C., and Wang, X. (2018). Factorizable net: An efficient subgraph-based framework for scene graph generation. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y., editors, *Computer Vision – ECCV 2018*, pages 346–363, Cham. Springer International Publishing.

Li, Y., Ouyang, W., Zhou, B., Wang, K., and Wang, X. (2017). Scene graph generation from objects, phrases and region captions. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1270–1279.

Li, Y., Panda, R., Kim, Y., Chen, C.-F. R., Feris, R., Cox, D., and Vasconcelos, N. (2022b). Valhalla: Visual hallucination for machine translation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5206–5216.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T., editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham. Springer International Publishing.

Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., and Kipf, T. (2020). Object-centric learning with slot attention. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 11525–11538. Curran Associates, Inc.

Lou, C., Han, W., Lin, Y., and Zheng, Z. (2022). Unsupervised vision-language parsing: Seamlessly bridging visual scene graphs with language structures via dependency relationships. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15586–15595.

Lu, C., Krishna, R., Bernstein, M., and Fei-Fei, L. (2016). Visual relationship detection with language priors. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Computer Vision – ECCV 2016*, pages 852–869, Cham. Springer International Publishing.

Lu, J., Batra, D., Parikh, D., and Lee, S. (2019). Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Lupyan, G., Abdel Rahman, R., Boroditsky, L., and Clark, A. (2020). Effects of language on visual perception. *Trends in Cognitive Sciences*, 24(11):930–944.

Lyon, R. F. (2010). Machine hearing: An emerging field [exploratory dsp]. *IEEE Signal Processing Magazine*, 27(5):131–139.

Maddison, C. J., Mnih, A., and Teh, Y. W. (2017). The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*.

Maddison, C. J., Tarlow, D., and Minka, T. (2014). A$*$ sampling. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Maillard, J., Clark, S., and Yogatama, D. (2019). Jointly learning sentence embeddings and syntax with unsupervised tree-lstms. *Natural Language Engineering*, 25(4):433–449.

Mao, J., Gan, C., Kohli, P., Tenenbaum, J. B., and Wu, J. (2019). The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *International Conference on Learning Representations*.

Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A., and Murphy, K. (2016). Generation and comprehension of unambiguous object descriptions. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11–20.

Marcus, M. P., Santorini, B., Marcinkiewicz, M. A., and Taylor, A. (1999). Treebank-3. *Linguistic Data Consortium, Philadelphia*.

McFee, B., Salamon, J., and Bello, J. P. (2018). Adaptive pooling operators for weakly labeled sound event detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(11):2180–2193.

Michie, D. (1968). "Memo" functions and machine learning. *Nature*, 218(5136):19–22.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Minka, T. (1998). Expectation-maximization as lower bound maximization. Technical report. https://tminka.github.io/papers/em.html.

Nagrani, A., Yang, S., Arnab, A., Jansen, A., Schmid, C., and Sun, C. (2021). Attention bottlenecks for multimodal fusion. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*.

Nakayama, H. and Nishida, N. (2017). Zero-resource machine translation by multimodal encoder-decoder network with multimedia pivot. *Machine Translation*, 31(1/2):49–64.

Neal, R. M. and Hinton, G. E. (1998). *A View of the Em Algorithm that Justifies Incremental, Sparse, and other Variants*, pages 355–368. Springer Netherlands, Dordrecht.

Newell, A. and Deng, J. (2017). Pixels to graphs by associative embedding. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Oepen, S., Kuhlmann, M., Miyao, Y., Zeman, D., Flickinger, D., Hajič, J., Ivanova, A., and Zhang, Y. (2014). SemEval 2014 task 8: Broad-coverage semantic dependency parsing. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 63–72, Dublin, Ireland. Association for Computational Linguistics.

O'Grady, W. (2005). *How children learn language.* Cambridge approaches to linguistics. Cambridge University Press, New York, NY, US.

Ohta, Y., Kanade, T., and Sakai, T. (1978). An analysis system for scenes containing objects with substructures. In *Proceedings of 4th International Joint Conference on Pattern Recognition (IJCPR '78)*, pages 752 – 754.

Oncescu, A.-M., Henriques, J. F., Liu, Y., Zisserman, A., and Albanie, S. (2021a). Queryd: A video dataset with high-quality text and audio narrations. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2265–2269.

Oncescu, A.-M., Koepke, A. S., Henriques, J. F., Akata, Z., and Albanie, S. (2021b). Audio Retrieval with Natural Language Queries. In *Proc. Interspeech 2021*, pages 2411–2415.

Opper, M. and Saad, D. (2001). *Advanced Mean Field Methods: Theory and Practice*. The MIT Press.

Ororbia, A., Mali, A., Kelly, M., and Reitter, D. (2019). Like a baby: Visually situated neural language acquisition. In *Proceedings of the 57th Annual Meeting of*

*the Association for Computational Linguistics*, pages 5127–5136, Florence, Italy. Association for Computational Linguistics.

Paisley, J., Blei, D. M., and Jordan, M. I. (2012). Variational bayesian inference with stochastic search. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, page 1363–1370, Madison, WI, USA. Omnipress.

Papandreou, G. and Yuille, A. L. (2011). Perturb-and-map random fields: Using discrete optimization to learn and sample from energy models. In *2011 International Conference on Computer Vision*, pages 193–200.

Parisi, G. and Shankar, R. (1988). Statistical field theory. *Physics Today*, 41(12):110–110.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Pereira, F. and Schabes, Y. (1992). Inside-outside reestimation from partially bracketed corpora. In *30th Annual Meeting of the Association for Computational Linguistics*, pages 128–135, Newark, Delaware, USA. Association for Computational Linguistics.

Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., and Miller, A. (2019). Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Peyre, J., Laptev, I., Schmid, C., and Sivic, J. (2017). Weakly-supervised learning of visual relations. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5189–5198.

Piczak, K. J. (2015). ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pages 1015–1018. ACM Press.

Plummer, B. A., Wang, L., Cervantes, C. M., Caicedo, J. C., Hockenmaier, J., and Lazebnik, S. (2015). Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2641–2649.

Pollak, I., Siskind, J., Harper, M., and Bouman, C. (2003a). Parameter estimation for spatial random trees using the em algorithm. In *Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429)*, volume 1, pages I–257.

Pollak, L., Siskind, J., Harper, M., and Bouman, C. (2003b). Modeling and estimation of spatial random trees with application to image classification. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, volume 3, pages III–305.

Poon, H. and Domingos, P. (2011). Sum-product networks: A new deep architecture. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, UAI'11, page 337–346, Arlington, Virginia, USA. AUAI Press.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., and Vesely, K. (2011). The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society. IEEE Catalog No.: CFP11SRW-USB.

Pullum, G. K. and Scholz, B. C. (2002). Empirical assessment of stimulus poverty arguments. *The Linguistic Review*, 19(1-2):9–50.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training. Technical report. https://openai.com/research/language-unsupervised.

Ramanathan, V., Li, C., Deng, J., Han, W., Li, Z., Gu, K., Song, Y., Bengio, S., Rossenberg, C., and Fei-Fei, L. (2015). Learning semantic relationships for better action retrieval in images. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1100–1109.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (2021). Zero-shot text-to-image generation. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8821–8831. PMLR.

Ranganath, R., Gerrish, S., and Blei, D. (2014). Black Box Variational Inference. In Kaski, S. and Corander, J., editors, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pages 814–822, Reykjavik, Iceland. PMLR.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In Xing, E. P. and Jebara, T., editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, Bejing, China. PMLR.

Ritchie, D., Horsfall, P., and Goodman, N. D. (2016). Deep amortized inference for probabilistic programs. *CoRR*, abs/1610.05735.

Robbins, H. (1951). Asymptotically subminimax solutions of compound statistical decision problems. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 131–149, Berkeley, Calif. University of California Press.

Robbins, H. and Monro, S. (1951). A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400 – 407.

Rohrbach, A., Rohrbach, M., Tandon, N., and Schiele, B. (2015). A dataset for movie description. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3202–3212.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695.

Roy, D. K. (2002). Learning visually grounded words and syntax for a scene description task. *Computer Speech and Language*, 16(3):353 – 385. Spoken Language Generation.

Rush, A. (2020). Torch-struct: Deep structured prediction library. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 335–342, Online. Association for Computational Linguistics.

Sadeghi, M. A. and Farhadi, A. (2011). Recognition using visual phrases. In *CVPR 2011*, pages 1745–1752.

Salamon, J., Jacoby, C., and Bello, J. P. (2014). A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM International Conference on Multimedia*, MM '14, page 1041–1044, New York, NY, USA. Association for Computing Machinery.

Saul, L. K., Jaakkola, T., and Jordan, M. I. (1996). Mean field theory for sigmoid belief networks. *J. Artif. Int. Res.*, 4(1):61–76.

Schuster, S., Krishna, R., Chang, A., Fei-Fei, L., and Manning, C. D. (2015). Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the Fourth Workshop on Vision and Language*, pages 70–80, Lisbon, Portugal. Association for Computational Linguistics.

Sharma, P., Ding, N., Goodman, S., and Soricut, R. (2018). Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia. Association for Computational Linguistics.

Sheinberg, D. L. and Logothetis, N. K. (2001). Noticing familiar objects in real world scenes: the role of temporal cortical neurons in natural vision. *J. Neurosci.*, 21(4):1340–1350.

Shen, Y., Lin, Z., wei Huang, C., and Courville, A. (2018). Neural language modeling by jointly learning syntax and lexicon. In *International Conference on Learning Representations*.

Shen, Y., Tan, S., Sordoni, A., and Courville, A. (2019). Ordered neurons: Integrating tree structures into recurrent neural networks. In *International Conference on Learning Representations*.

Shi, H., Mao, J., Gimpel, K., and Livescu, K. (2019a). Visually grounded neural syntax acquisition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1842–1861, Florence, Italy. Association for Computational Linguistics.

Shi, J., Zhang, H., and Li, J. (2019b). Explainable and explicit visual reasoning over scene graphs. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8368–8376.

Shi, J., Zhong, Y., Xu, N., Li, Y., and Xu, C. (2021). A simple baseline for weakly-supervised scene graph generation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16373–16382.

Shieber, S. M. (1985). Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, 8(3):333–343.

Si, Z. and Zhu, S.-C. (2013). Learning and-or templates for object recognition and detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(9):2189–2205.

Silberer, C. and Lapata, M. (2014). Learning grounded meaning representations with autoencoders. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 721–732, Baltimore, Maryland. Association for Computational Linguistics.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*.

Siskind, J., Sherman, J., Pollak, I., Harper, M., and Bouman, C. A. (2007). Spatial random tree grammars for modeling hierarchal structure in images with regions of arbitrary shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1504–1519.

Smith, N. A. and Eisner, J. (2005). Guiding unsupervised grammar induction using contrastive estimation. In *In Proc. of IJCAI Workshop on Grammatical Inference Applications*, pages 73–82.

Socher, R., Bauer, J., Manning, C. D., and Ng, A. Y. (2013). Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 455–465, Sofia, Bulgaria. Association for Computational Linguistics.

Socher, R., Lin, C. C.-Y., Ng, A. Y., and Manning, C. D. (2011). Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 129–136, Madison, WI, USA. Omnipress.

Sohn, K. (2016). Improved deep metric learning with multi-class n-pair loss objective. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Specia, L., Frank, S., Sima'an, K., and Elliott, D. (2016). A shared task on multimodal machine translation and crosslingual image description. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 543–553, Berlin, Germany. Association for Computational Linguistics.

Spitkovsky, V. I., Alshawi, H., Jurafsky, D., and Manning, C. D. (2010). Viterbi training improves unsupervised dependency parsing. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 9–17, Uppsala, Sweden. Association for Computational Linguistics.

Steedman, M. (2000). *The Syntactic Process*. MIT Press, Cambridge, MA, USA.

Steels, L. (1998). The origins of syntax in visually grounded robotic agents. *Artificial Intelligence*, 103(1):133 – 156. Artificial Intelligence 40 years later.

Stern, M., Andreas, J., and Klein, D. (2017). A minimal span-based neural constituency parser. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 818–827, Vancouver, Canada. Association for Computational Linguistics.

Stolcke, A. and Omohundro, S. (1994). Inducing probabilistic grammars by bayesian model merging. In Carrasco, R. C. and Oncina, J., editors, *Grammatical Inference and Applications*, pages 106–118, Berlin, Heidelberg. Springer Berlin Heidelberg.

Stuhlmüller, A., Taylor, J., and Goodman, N. (2013). Learning stochastic inverses. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Su, R., Rijhwani, S., Zhu, H., He, J., Wang, X., Bisk, Y., and Neubig, G. (2021). Dependency induction through the lens of visual perception. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 17–26, Online. Association for Computational Linguistics.

Su, W., Zhu, X., Cao, Y., Li, B., Lu, L., Wei, F., and Dai, J. (2020). Vl-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations*.

Suhr, A., Zhou, S., Zhang, A., Zhang, I., Bai, H., and Artzi, Y. (2019). A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6418–6428, Florence, Italy. Association for Computational Linguistics.

Tai, K. S., Socher, R., and Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China. Association for Computational Linguistics.

Tang, K., Niu, Y., Huang, J., Shi, J., and Zhang, H. (2020). Unbiased scene graph generation from biased training. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3713–3722.

Thomee, B., Shamma, D. A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., and Li, L.-J. (2016). Yfcc100m: The new data in multimedia research. *Commun. ACM*, 59(2):64–73.

Tighe, J. and Lazebnik, S. (2010). Superparsing: Scalable nonparametric image parsing with superpixels. In Daniilidis, K., Maragos, P., and Paragios, N., editors, *Computer Vision – ECCV 2010*, pages 352–365, Berlin, Heidelberg. Springer Berlin Heidelberg.

Tu, K. (2016). Stochastic and-or grammars: A unified framework and logic perspective. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, page 2654–2660. AAAI Press.

Tu, K., Pavlovskaia, M., and Zhu, S.-C. (2013). Unsupervised structure learning of stochastic and-or grammars. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Tu, Z., Chen, X., Yuille, and Zhu (2003). Image parsing: unifying segmentation, detection, and recognition. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 18–25 vol.1.

Turney, P., Neuman, Y., Assaf, D., and Cohen, Y. (2011). Literal and metaphorical sense identification through concrete and abstract context. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 680–690, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Ullman, S. (1995). Sequence seeking and counter streams: a computational model for bidirectional information flow in the visual cortex. *Cereb. Cortex*, 5(1):1–11.

Utiyama, M. and Isahara, H. (2007). A comparison of pivot methods for phrase-based statistical machine translation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 484–491, Rochester, New York. Association for Computational Linguistics.

van den Oord, A., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

van den Oord, A., Vinyals, O., and kavukcuoglu, k. (2017). Neural discrete representation learning. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Van Durme, B. D. (2010). *Extracting implicit knowledge from text*. University of Rochester.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D., and Bengio, Y. (2019). Manifold mixup: Better representations by interpolating hidden states. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6438–6447. PMLR.

Vigliocco, G., Perniss, P., and Vinson, D. (2014). Language as a multimodal phenomenon: implications for language learning, processing and evolution. *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, 369(1651):20130292.

Wainwright, M. J. and Jordan, M. I. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305.

Wan, B., Han, W., Zheng, Z., and Tuytelaars, T. (2022). Unsupervised vision-language grammar induction with shared structure modeling. In *International Conference on Learning Representations*.

Wang, B. and Komatsuzaki, A. (2021). GPT-J-6B: a 6 billion parameter autoregressive language model. Technical report. https://github.com/kingoflolz/mesh-transformer-jax.

Wang, L., Luc, P., Recasens, A., Alayrac, J., and van den Oord, A. (2021). Multimodal self-supervised learning of general audio representations. *CoRR*, abs/2104.12807.

Wang, S., Wang, Y., and Zhu, S.-C. (2013). Hierarchical space tiling for scene modeling. In Lee, K. M., Matsushita, Y., Rehg, J. M., and Hu, Z., editors, *Computer Vision – ACCV 2012*, pages 796–810, Berlin, Heidelberg. Springer Berlin Heidelberg.

Wiegerinck, W. (2000). Variational approximations between mean field theory and the junction tree algorithm. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, UAI'00, page 626–633, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Williams, A., Drozdov, A., and Bowman, S. R. (2018). Do latent tree learning models identify meaningful structure in sentences? *Transactions of the Association for Computational Linguistics*, 6:253–267.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256.

Wittgenstein, L. (1953). *Philosophical Investigations*. Basil Blackwell, Oxford.

Woo, S., Kim, D., Cho, D., and Kweon, I. S. (2018). Linknet: Relational embedding for scene graph. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Wu, H. and Wang, H. (2007). Pivot language approach for phrase-based statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 856–863, Prague, Czech Republic. Association for Computational Linguistics.

Wu, H.-H., Seetharaman, P., Kumar, K., and Bello, J. P. (2022). Wav2clip: Learning robust audio representations from clip. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4563–4567.

Wu, Q., Shen, C., Liu, L., Dick, A., and Van Den Hengel, A. (2016). What value do explicit high level concepts have in vision to language problems? In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 203–212.

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., and Girshick, R. (2019). Detectron2. Technical report. https://github.com/facebookresearch/detectron2.

Wu, Z., Chen, Y., Kao, B., and Liu, Q. (2020). Perturbed masking: Parameter-free probing for analyzing and interpreting BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4166–4176, Online. Association for Computational Linguistics.

Xing, E. P., Jordan, M. I., and Russell, S. (2002). A generalized mean field algorithm for variational inference in exponential families. In *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*, UAI'03, page 583–591, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Xu, D., Zhu, Y., Choy, C. B., and Fei-Fei, L. (2017). Scene graph generation by iterative message passing. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3097–3106.

Xu, J., De Mello, S., Liu, S., Byeon, W., Breuel, T., Kautz, J., and Wang, X. (2022). Groupvit: Semantic segmentation emerges from text supervision. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18113–18123.

Xu, J., Mei, T., Yao, T., and Rui, Y. (2016). Msr-vtt: A large video description dataset for bridging video and language. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5288–5296.

Yang, J., Lu, J., Lee, S., Batra, D., and Parikh, D. (2018). Graph r-cnn for scene graph generation. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y., editors, *Computer Vision – ECCV 2018*, pages 690–706, Cham. Springer International Publishing.

Yang, X., Tang, K., Zhang, H., and Cai, J. (2019). Auto-encoding scene graphs for image captioning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10677–10686.

Yao, T., Pan, Y., Li, Y., and Mei, T. (2019). Hierarchy parsing for image captioning. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2621–2629.

Yao, Y., Zhang, A., Han, X., Li, M., Weber, C., Liu, Z., Wermter, S., and Sun, M. (2021). Visual distant supervision for scene graph generation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15796–15806.

Ye, K. and Kovashka, A. (2021). Linguistic structures as weak supervision for visual scene graph generation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8285–8295.

Yogatama, D., Blunsom, P., Dyer, C., Grefenstette, E., and Ling, W. (2017). Learning to compose words into sentences with reinforcement learning. In *International Conference on Learning Representations*.

You, Y., Gitman, I., and Ginsburg, B. (2017). Scaling SGD batch size to 32k for imagenet training. *CoRR*, abs/1708.03888.

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

Younger, D. H. (1967). Recognition and parsing of context-free languages in time n3. *Information and Control*, 10(2):189 – 208.

Yu, L., Poirson, P., Yang, S., Berg, A. C., and Berg, T. L. (2016). Modeling context in referring expressions. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Computer Vision – ECCV 2016*, pages 69–85, Cham. Springer International Publishing.

Yu, R., Li, A., Morariu, V. I., and Davis, L. S. (2017). Visual relationship detection with internal and external linguistic knowledge distillation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1068–1076.

Yu, T., Zhang, R., Yang, K., Yasunaga, M., Wang, D., Li, Z., Ma, J., Li, I., Yao, Q., Roman, S., Zhang, Z., and Radev, D. (2018). Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics.

Zareian, A., Karaman, S., and Chang, S.-F. (2020a). Bridging knowledge graphs to generate scene graphs. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII*, page 606–623, Berlin, Heidelberg. Springer-Verlag.

Zareian, A., Karaman, S., and Chang, S.-F. (2020b). Weakly supervised visual semantic parsing. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3733–3742.

Zellers, R., Bisk, Y., Farhadi, A., and Choi, Y. (2019). From recognition to cognition: Visual commonsense reasoning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6713–6724.

Zellers, R., Lu, J., Lu, X., Yu, Y., Zhao, Y., Salehi, M., Kusupati, A., Hessel, J., Farhadi, A., and Choi, Y. (2022). Merlot reserve: Neural script knowledge through vision and language and sound. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16354–16366.

Zellers, R., Yatskar, M., Thomson, S., and Choi, Y. (2018). Neural motifs: Scene graph parsing with global context. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5831–5840.

Zettlemoyer, L. S. and Collins, M. (2005). Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, UAI'05, page 658–666, Arlington, Virginia, USA. AUAI Press.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2018). mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

Zhang, H., Kyaw, Z., Chang, S.-F., and Chua, T.-S. (2017a). Visual translation embedding network for visual relation detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3107–3115.

Zhang, H., Kyaw, Z., Yu, J., and Chang, S.-F. (2017b). Ppr-fcn: Weakly supervised visual relation detection via parallel pairwise r-fcn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4243–4251.

Zhang, S., Song, L., Jin, L., Xu, K., Yu, D., and Luo, J. (2021). Video-aided unsupervised grammar induction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1513–1524, Online. Association for Computational Linguistics.

Zhao, P. and Zhang, T. (2015). Stochastic optimization with importance sampling for regularized loss minimization. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1–9, Lille, France. PMLR.

Zhao, Y., Hessel, J., Yu, Y., Lu, X., Zellers, R., and Choi, Y. (2022). Connecting the dots between audio and text without parallel data through visual knowledge transfer. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4492–4507, Seattle, United States. Association for Computational Linguistics.

Zhao, Y. and Titov, I. (2020). Visually grounded compound PCFGs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4369–4379, Online. Association for Computational Linguistics.

Zhao, Y. and Titov, I. (2021). An empirical study of compound PCFGs. In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 166–171, Kyiv, Ukraine. Association for Computational Linguistics.

Zhao, Y. and Titov, I. (2023a). On the transferability of visually grounded PCFGs. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Singapore. Association for Computational Linguistics.

Zhao, Y. and Titov, I. (2023b). Unsupervised scene graph induction from natural language supervision. Technical report. https://github.com/zhaoyanpeng/sgi.

Zhao, Y., Zhang, L., and Tu, K. (2018). Gaussian mixture latent vector grammars. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1181–1189, Melbourne, Australia. Association for Computational Linguistics.

Zhao, Y. and Zhu, S.-c. (2011). Image parsing with stochastic scene grammar. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc.

Zhong, Y., Shi, J., Yang, J., Xu, C., and Li, Y. (2021). Learning to generate scene graph from natural language supervision. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1803–1814.

Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., and Torralba, A. (2015). Object detectors emerge in deep scene cnns. In Bengio, Y. and LeCun, Y., editors, *International Conference on Learning Representations*.

Zhou, L., Palangi, H., Zhang, L., Hu, H., Corso, J., and Gao, J. (2020). Unified vision-language pre-training for image captioning and vqa. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):13041–13049.

Zhu, H., Bisk, Y., and Neubig, G. (2020). The return of lexical dependencies: Neural lexicalized PCFGs. *Transactions of the Association for Computational Linguistics*, 8:647–661.

Zhu, S.-C. and Mumford, D. (2006). A stochastic grammar of images. *Found. Trends. Comput. Graph. Vis.*, 2(4):259–362.

Zitnick, C. L. and Parikh, D. (2013). Bringing semantics into focus using visual abstraction. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3009–3016.

Zuidema, W. (2002). How the poverty of the stimulus solves the poverty of the stimulus. In Becker, S., Thrun, S., and Obermayer, K., editors, *Advances in Neural Information Processing Systems*, volume 15. MIT Press.