Unsupervised Vision-Language Parsing: Seamlessly Bridging Visual Scene Graphs with Language Structures via Dependency Relationships

Chao Lou[1,2*], Wenjuan Han[1†], Yuhuan Lin[3], Zilong Zheng[1*]
[1] Beijing Institute for General Artificial Intelligence (BIGAI), Beijing, China
[2] ShanghaiTech University, Shanghai, China
[3] Tsinghua Unversity, Beijing, China
louchao@shanghaitech.edu.cn, hanwenjuan@bigai.ai
lin-yh20@mails.tsinghua.edu.cn, zlzheng@bigai.ai
https://github.com/bigai-research/VLGAE

Abstract

Understanding realistic visual scene images together with language descriptions is a fundamental task towards generic visual understanding. Previous works have shown compelling comprehensive results by building hierarchical structures for visual scenes (e.g., scene graphs) and natural languages (e.g., dependency trees), individually. However, how to construct a joint vision-language (VL) structure has barely been investigated. More challenging but worthwhile, we introduce a new task that targets on inducing such a joint VL structure in an unsupervised manner. Our goal is to bridge the visual scene graphs and linguistic dependency trees seamlessly. Due to the lack of VL structural data, we start by building a new dataset `VLParse`*. Rather than using labor-intensive labeling from scratch, we propose an automatic alignment procedure to produce coarse structures followed by human refinement to produce high-quality ones. Moreover, we benchmark our dataset by proposing a contrastive learning (CL)-based framework VLGAE, short for Vision-Language Graph Autoencoder. Our model obtains superior performance on two derived tasks,* i.e.*, language grammar induction and VL phrase grounding. Ablations show the effectiveness of both visual cues and dependency relationships on fine-grained VL structure construction.*

1. Introduction

Visual scene understanding has long been considered a primal goal for computer vision. Going beyond the success of high-accurate individual object detection in complicated environments, various attempts have been made for higher-order visual understanding, such as predicting an *explainable*, *structured*, and *semantically-aligned* representation from scene images [18, 22, 41]. Such representations not only provide fine-grained visual cues for low-level recognition tasks, but have further demonstrated their applications on numerous high-level visual reasoning tasks, *e.g.*, visual question answering (VQA) [34, 50], image captioning [3, 44], and scene synthesis [18, 21].

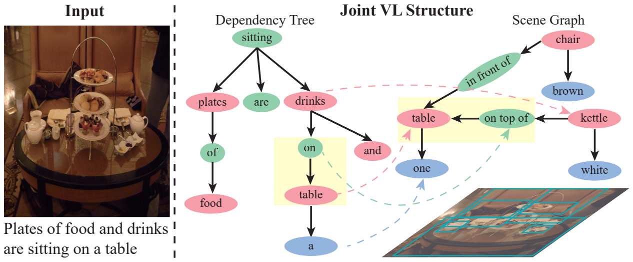

Figure 1. Task illustration of `VLParse`. Different node types are identified by their background colors and the yellow areas indicate first-order relationships (§3.1).

Scene graph (SG), one of the most popular visual structures, serves as an abstraction of objects and their complex relationships within scene images [22, 26]. Conventional scene graph generation models recognize and predict objects, attributes, relationships, and their corresponding semantic labels purely from natural images in a *fully-supervised* manner [33, 43]. Despite the promising performance achieved on large-scale SG benchmarks, these methods suffer from limitations on existing datasets and task setting [13]. First, a comprehensive scene graph requires different semantic levels of visual understanding [27], whilst most current datasets only capture a small portion of acces-

[*] Equal contribution. Author orders are coin clipped. This work was conducted when Chao Lou and Yuhuan Lin were research interns at BIGAI.

[†] Corresponding author.

sible semantics for classification [13], which will cause the prediction model bias towards those most-frequent labels. Second, building such datasets requires exhaustive labeling of bounding boxes, relations, and corresponding semantics, which are time-consuming and inefficient. Third, it is typically hard to induce a semantically-consistent graphical structure solely from visual inputs, which typically requires an extra visual relation recognition module with heavy manually-labeled supervision.

Different from dense and noisy visual information, natural language directly provides symbolic and structured information (*e.g.*, grammar) to support the comprehension process. Researches on language structure induction can date back to early computational linguistic theories [4, 5, 6]. Empowered by advances in deep learning techniques, a variety of neural structured prediction algorithms were proposed to analyze more complicated structure information and apply them to natural language tasks [9, 10, 23]. Dependency tree (DT) parsing, as one essential branch of language structured prediction, aims to generate a parse tree that is composed of vertices representing each word's semantic and syntactic meanings, and directed edges representing the dependency relationships among them. Of note, such tree structure shares a similar idea as in SG. However, the ground truth structure (commonly referred to as "gold structure") requires professional linguists' labeling. To mitigate the data issue, pioneer works have also demonstrated the success of DT learning in an unsupervised schema [19, 23].

In this work, we leverage the best of both modalities and introduce a new task – unsupervised vision-language (VL) parsing (short for `VLParse`) – aiming to devise a joint VL structure that bridges visual scene graphs with linguistic dependency trees seamlessly. By "seamless", we mean that each node in the VL structure shall present the well-aligned information of some node in SG and DT, so are their relationships, as shown in Figure 1. To the best of our knowledge, this is the first work that formally defines the joint representation of VL structure with dependency relationships. Respecting the semantic consistency and independent characteristics, the joint VL structure considers both the shared multimodal instances and the independent instances for each modality. In such a heterogeneous graph, semantically consistent instances across two graphs (DT and SG) are aligned in different levels, which maximizes the retention of the representation from two modalities. Some previous attempts have shown the benefits of exploring multi-modality information for structured understanding. For example, Shi et al. [31] first proposes a visually grounded syntax parser to induce the language structure. [46, 48] further exploit visual semantics to improve the structure for language. These structures, however, are still for language syntactic parsing rather than for joint vision-language understanding. One closest work to us is VL-Grammar [17], which builds separate image structures and language structures via compound PCFG [23]. However, the annotations (*i.e.*, segmentation parts) are provided in advance.

`VLParse` aims to conduct thoughtful cross-modality understanding and bridge the gap between multiple subtasks: structure induction for the image and language separately and unsupervised visual grounding. As a complex task, it is comprised of several instances, such as objects, attributes, and different levels of relationships. The interactions among different instances and subtasks can provide rich information and play a complementary or restrictive role during identification and understanding.

To address this challenging task, we propose a novel contrastive learning (CL)-based architecture, Vision-Language Graph Autoencoder (VLGAE), aiming at constructing a multimodal structure and aligning VL information simultaneously. The VLGAE is comprised of feature extraction, structure construction, and cross-modality matching modules. The feature extraction module extracts features from both modalities and builds representations for all instances in DT and SG. The structure construction module follows the encoder-decoder paradigm, where the encoder obtains a compressed global VL representation from image-caption pair using attention mechanisms; the decoder incorporates the inside algorithm to construct the VL structure recursively as well as compute the posteriors of spans. The VL structure induction is optimized by Maximum Likelihood Estimation (MLE) with a negative likelihood loss. For cross-modality matching, we compute the vision-language matching score between visual image regions and language contexts. We further enhance the matching score with posterior values achieved from the structure construction module. This score is used to promote the cross-modality fine-grained correspondence with the supervisory signal of the image-caption pairs via a CL strategy; see Figure 3 and Section 5 for details.

In summary, our contributions are five-fold: (i) We design a joint VL structure that bridges visual scene graph and linguistic dependency tree; (ii) We introduce a new task `VLParse` for better cross-modality visual scene understanding (§4); (iii) We present a two-step VL dataset creation paradigm without labor-intensive labelling and deliver a new dataset (§3); (iv) We benchmark our dataset with a novel CL-based framework VLGAE (§5); (v) Empirical results demonstrate significant improvements on single modality structure induction and cross-modality alignment with the proposed framework.

2. Related Work

Weakly-supervised Visual Grounding Visual grounding (VG) aims to locate the most relevant object or region in an image referred by natural language expressions, such as phrases [39], sentences [1, 34] or dialogues [50]. Weakly-supervised visual phrase grounding, which infers region-phrase correspondences using only image-sentence pairs, has drawn researchers' attention. There are multiple approaches to weakly-supervised visual phrase grounding. Gupta et al. [14] leverage contrastive learning to train model based on image-sentence pairs data. Wang et al. [38] build visually-aware language representations for phrases that could be better aligned with the visual representations. Wang et al. [36] develop a method to distill knowledge from Faster R-CNN for weakly supervised phrase grounding.

Language sentences contain rich semantics and syntactics information. Thus, some researches focus on how to extract and leverage useful information in a sentence to facilitate visual grounding. For example, Xiao et al. [42] use the linguistic structure of natural language descriptions for visual phrase grounding. Yu et al. [45] learn to parse captions automatically into three modular components related to subject appearance, location, and relationship to other objects, which get rich different types of information from sentences. In this work, we propose to induce structures from realistic image-caption pairs without any structure annotations, nor phrase-region correspondence annotations. Note that different from Wang et al. [37] who predict the corresponding regions for a given set of noun phrases, noun phrases in VL grammar induction are unknown and all spans in the VL structure are corresponding regions in the image.

Language Dependency Parsing Dependency parsing, a fundamental challenge in natural language processing (NLP), aims to find syntactic dependency relations between words in sentences. Due to the challenge of achieving gold structures for all available language corpus, unsupervised dependency parsing, whose goal is to obtain a dependency parser without using annotated sentences, has attracted more attention over recent years. The pioneer work Dependency Model with Valence (DMV) [25] proposes to model dependency parsing as a generative process of dependency grammars. Empowered by deep learning techniques, NDMV [19] employ neural networks to capture the similarities between part-of-speech (POS) tags, and learn the grammar based on DMV. However, generative models often are limited by independence assumption, so more researchers have paid attention to autoencoder-based approaches [2], *e.g.*, Discriminative NDMV (D-NDMV) [15].

Visual-Aided Grammar Induction Visual-aided word representation learning and sentence representation learning achieve positive results. Shi et al. [31] first propose the visually grounded grammar induction task and present a visually-grounded neural syntax learner (VG-NSL). They use an easy-first bottom-up parser [12] and use REINFORCE [40] as gradient estimator for image-caption matching. Zhao and Titov [49] propose an end-to-end training algorithm for Compound PCFG [24], a powerful grammar inducer. Jin and Schuler [20] formulate a different visual grounding task. They use an autoencoder as a visual model and fuse language and vision features on the hidden states. Different from visually-grounded grammar induction, we not only care about the language structure accuracy but also the fine-grained alignment accuracy.

3. The `VLParse` Dataset

In this section, we start by formalizing the joint VL structure to represent the shared semantics for vision and language. Then we introduce how the dataset, `VLParse`, is formed in a semi-automatic manner.

3.1. Joint Vision-Language Structure

The vision-language (VL) structure is composed of a visual structure SG, a linguistic structure DT and a hierarchical alignment between SG and DT.

Scene graph (SG) We define SG on an image $\mathbf{I}$ as a structured representation composed of three types of nodes: $\mathcal{T} = \{\texttt{OBJECT}, \texttt{ATTRIBUTE}, \texttt{RELATIONSHIP}\}$, denoting the image's objects features, conceptual attribute features, and relationship features between two objects. Each `OBJECT` node is associated with an `ATTRIBUTE` node; between each pair of `OBJECT` nodes, there exists a `RELATIONSHIP` node. Let $\mathcal{R}$ be the set of all relationship types (including "none" relationship), we can denote the set of all variables in SG as $\{v_i^{cls}, v_i^{bbox}, v_i^{type}, v_{i\to j}; i \neq j\}$, where v_i^{cls} is the class label of the i-th bounding box, $v_i^{bbox} \in \mathbb{R}^4$ denotes the bounding box offsets, $v_i^{type} \in \mathcal{T}$ is the node type, and $v_{i\to j} \in \mathcal{R}$ is the relationship from node v_i to v_j.

Dependency tree (DT) Conventional DT is a hierarchy with directed dependency relationships. Given the textual description denoted as a sequence of N words $\mathbf{w} = \{w_1, w_2, ..., w_N\}$, each dependency within DT can be denoted as triplet $(w_i, w_j, w_{i\to j})$, representing a parent node w_i, a child node w_j and the direct dependency relationship from w_i to w_j, respectively. Similar to SG, for each node's representation, we additionally append the node's type label $w_i^{type} \in \mathcal{T}$. Thus, all variables within in DT becomes $\{w_i, w_i^{type}, w_{i\to j}; i \neq j\}$.

Alignment The alignment between DT and SG can be seen as a realization of visual grounding for instances on different levels of the linguistic structure. We hereby define three levels of alignment (see Figure 1 for illustration):

- *Zero-order Alignment*. It defines connections between each node w_i in DT with a node v_i in SG.
- *First-order Alignment*. A first-order relationship can be defined as a triplet $(w_i, w_j, w_{i\to j})$, including two nodes and a directed dependency. Then the first-order

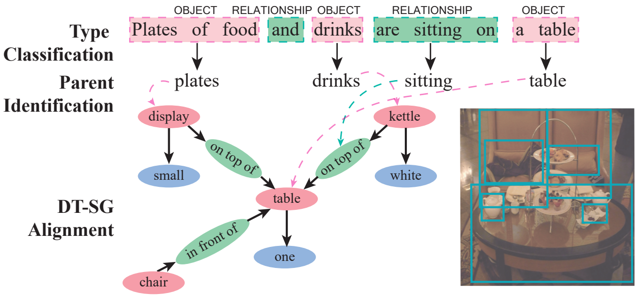

Figure 2. An illustration of the process of the automatic rule-based alignment. After the process of DT rewriting and DT-SG alignment, every instance in DT can be aligned to a SG instance. Through SG, instances in DT can match to image regions.

alignment aims to align the triplet in DT with a similar triplet $(v_i, v_j, v_{i\to j})$ in SG.

- *Second-order Alignment.* A second-order relationship builds upon the first-order relationship and is represented as dependencies among three nodes, *e.g.*, w_i, w_j, and w_k in DT. Similar as the first-order alignment, the second-order alignment aligns such relationships between DT and SG with similar semantics.

3.2. Automatic Rule-based Alignment

In practice, the alignment between SG and DT is labor-intense and expensive to obtain, whilst unlabeled data is low in cost and large in scale. Thus we design a set of rules to automatically ground the language instances of DT to the vision instances of SG. This automatic alignment provides beneficial information to reduce the labeling burden on workers. Specifically, we introduce a two-step alignment process, *i.e.*, rule-based DT rewriting (DT Rewriting) followed by the alignment between DT and SG (DT-SG Alignment).

DT Rewriting We start by introducing the rewriting procedure that extends and deforms DT in order to mitigate the difference between DT and SG. The rewriting considers two modules:

Type Classification We append the type label x_i^{type} for conventional DT. More specifically, we label words from DT with three node types as follows:

- OBJECT: The OBJECT node in DT is referred as to a word/phrase that can be grounded to a specific image area. A noun phrase including all words involved except for the attribute is designed as an OBJECT node.

- ATTRIBUTE: The ATTRIBUTE node is mostly an adjective used to decorate its linked OBJECT node. In our designed rules, we set words with dependency type *acomp* (adjectival complement) as ATTRIBUTE nodes.

- RELATIONSHIP: Two OBJECT nodes are linked to a RELATIONSHIP node with a directed dependency. OBJECT nodes are connected to each other through a RELATIONSHIP node. For example in Figure 2, "sitting" as a RELATIONSHIP node between two OBJECT nodes "drinks" and "table".

Parent Identification Since it is hard to identify a grounding area in image for an ATTRIBUTE node or a function word (such as the coordinating conjunction and determiner, *etc*.), we instead define words of these types share the corresponding OBJECT node's, says parent nodes' grounding area. A parent node is a *noun* representing the core semantics of a noun phrase and ATTRIBUTE nodes, as dependent, modify it. This parent-dependent relationship is encoded in dependency types and dependency directions of DT [7].

Through the rules we design, every word in DT is assigned with a node type and a parent node. We design 7 rules for OBJECT-ATTRIBUTE, 12 rules for RELATIONSHIP-OBJECT, 1 rule for OBJECT-OBJECT, 10 for OBJECT-RELATIONSHIP and 22 rules for function word processing.[1]

DT-SG Alignment Based on the rewritten DT, we perform DT-SG alignment to map the rewritten DT to SG. In details, we calculate the similarity score between the SG node and the word's parent, and choose top k results as the alignment result. The words labeled attribute leverage parent to retrieve OBJECT node it attributes in SG. Then we retrieve the ATTRIBUTE node in the subtree rooted by the OBJECT node by calculating the similarity score between the word and ATTRIBUTE node name. An illustration of the process of alignment from words to SG nodes is shown in Figure 2.

3.3. Crowd-Sourcing Human Refinement

To obtain a high-quality dataset, a human-refinement stage is adapted, providing an automatically annotated VL structure and asking the annotators to output the refined one. We utilize Amazon Mechanical Turk (AMT) to hire remotely native speakers to perform a crowd-sourcing survey.

Human Refinement We create a survey in AMT that allows workers to assess and refine data generated from the automatic rule-based alignment stage. We provide workers with comprehensive instructions and a set of well-defined examples to judge the quality of alignments, and modify those unsatisfied ones. During the task, we will show workers an interface with paired images and captions grouped by the image. We ask workers to check DT, SG, and the cross-modality alignment. Then the workers correct the inappropriate areas when necessary. The final results are combined using a majority vote.

[1] The dependencies used here is based on Stanford typed dependencies [7], a framework for annotation of grammar (parts of speech and syntactic dependencies https://catalog.ldc.upenn.edu/LDC99T42) [29]. Following [49], a learned parser [47] on this annotated data is used to label the dependency existence and dependency type.

Quality Control We adopt a set of measurements for quality control during the calibration process. Before submitting the task, the survey will first check the modifying parts by the worker to ensure that the modifying parts meet the base requirement: a dependency in DT is aligned to a RELATIONSHIP node in SG. If we find this kind of misalignment during annotation, we will prompt a message asking the workers to recheck their annotation. We publish datasets to workers one by one and request at least two workers to process the same sample to check whether there are disagreements. To ensure high-quality labeling, we restrict participated workers who have finished 500 human intelligence tasks (HITs) with high accuracy in the labeling history.

We do the post-processing double-check after the human refinement. We collect the flag disagreements for multiple decisions from several workers. All samples that have disagreement are double-checked manually by a third-party worker. We also flag annotations from workers whose work seems inadequate and filter out their results from the final collections.

3.4. Dataset Analysis

For the training dataset, we inherit MSCOCO training dataset [28][2]. We annotate VL structures based on the intersection of MSCOCO *dev+test* datasets and Visual Genome [26]. We collect an annotated dataset with 850 images and 4,250 captions (each image is associated with 5 captions). Then we split the 850 images into *dev* and *test* datasets by 1:1. The remains in *dev+test* are merged into the training dataset. Table 1 shows the data summary.

	Train	**Dev**	**Test**
# Images	83933	425	425
# Sentences	419665	2125	2125
# Avg. Instances in DT	-	20	21
# Avg. Instances in SG	-	135	134

Table 1. Data analysis of VLParse. # Avg.: The average number. Instances include zero-order instances, first-order relationships, and second-order relationships.

3.5. Human Performance

Five different workers are asked to label parse trees of 100 sentences from the test set. A different set of five workers on AMT were asked to align the visual terms and the language terms on the same sentences and their corresponding images. Then the averaged human performance is calculated as 96.15%.

Based on these observations, our designed dataset presents language representation and cross-modality understanding clearly and keeps vision-language alignment concrete. It demonstrates the reliability of our new dataset and benchmark through manual review.

4. Unsupervised Vision-Language Parsing

In this section, we introduce the task of unsupervised vision-language (VL) parsing, short for VLParse. We formalize the task of VL parsing followed by evaluation metrics.

4.1. Task Formulation

Given an input image $\mathbf{I}$ and the associating sentence with a sequence of N words $\mathbf{w} = \{w_1, w_2, ..., w_N\}$, the task is to predict the joint parse tree $\mathbf{pt}$ in a unsupervised manner. Specifically, the goal is to induce VL structures from only image-caption pairs without annotations of DT, SG nor phrase-region correspondence annotations for training. Of note, we do use a pre-trained object detector to obtain 50 bounding boxes as candidates, while the labels of the bounding boxes are not given. For a fully unsupervised setting, the process of obtaining the bounding boxes can be replaced by an object proposal method (*e.g.*, [35]). Compared with the weakly-supervised scene graph grounding task as in [32], the scene graphs in VLParse are unknown. Each OBJECT node in language DT will be mapped to a box region $o_i \in \mathbb{R}^4$ given M candidate object proposals $\mathcal{O} = \{o_i\}_{i=1}^M$ of the corresponding image. So are the relationships.

4.2. Evaluation Metrics

Due to lacking annotations of VL structure, we indirectly assess our model by two derived tasks from each modality's perspective, *i.e.*, language dependency parsing and phrase grounding.

Directed / Undirected Dependency Accuracy (DDA/UDA) DDA and UDA are two widely used evaluation metrics of dependency parsing. DDA denotes the proportion of tokens assigned with the correct parent node. UDA denotes the proportion of correctly predicted undirected dependency relation.

Zero-Order Alignment Accuracy (Zero-AA) Zero-AA assesses the alignment results on the zero-order level. A word is considered successfully grounded if two conditions are satisfied. First, the predicted bounding box of a language vertex has at least 0.5 IoU (Intersection over Union) with the box of ground-truth SG vertex if the ground-truth is a OBJECT node or ATTRIBUTE node, or the connected two boxes both have at least 0.5 IoU scores if the ground-truth is a RELATIONSHIP node. Second, although OBJECT node and ATTRIBUTE node share the same region, we ask models to distinguish them.

First/Second-Order Alignment Accuracy (First/Second-AA) We are also interested in whether the first- and

[2] We use the training data split following Zhao and Titov [48]. It contains 82,783 training images, 1,000 validation images, and 1,000 test images.

second-order relationships remain after alignment to another modality. That is, whether two zero-order instances (subject and predicate) in the first-order relationship remain adjacent in the aligned SG. For the second-order relationship, we consider whether three zero-order instances (a subject, a predicate, and an object) remain adjacent. For the second-order relationships, there are multiple approach to connect the three words obj-pred-sub (*e.g.*, obj→pred→sub and obj←pred→sub). We consider them all correct because distinguishing their adjacency to an unsupervised parser is more important to identify the semantics.

5. Vision-Language Graph Auto-Encoder

In this section, we introduce a novel CL based architecture, VLGAE, to benchmark the `VLParse` task. The architecture is composed of feature extraction, structure construction and cross-modality matching modules; Figure 3 depicts the overall computational framework. Below we will discuss details of each module and then the learning and inference algorithms.

5.1. Modeling

Feature Extraction For visual features, we start by using an off-the-shelf object detector Faster R-CNN [30] to generate a set of object proposals (RoIs) $\mathcal{O} = \{o_i\}_{i=1}^{M}$ on an input image $\mathbf{I}$ and extracting corresponding features $\{v_i^o\}_{i=1}^{M} \in \mathbb{R}^D$ as `OBJECT` nodes' features, where D is the dimension of each RoI feature. For each `OBJECT` node v_i^o, an `ATTRIBUTE` node is tagged along, with its feature denoted as $v_i^a = \text{MLP}(v_i^o)$. For two arbitrary `OBJECT` nodes v_i^o and v_j^o, we denote the zero-order `RELATIONSHIP` node as $v_{i\to j,0}^{\text{img}}$. We also add an dummy node representing the full image and take the average of all `OBJECT` node features as its feature. For all nodes except for `OBJECT` nodes, we use randomly initialized neural networks to represent the features.

For textual features, each word w_i in sentence $\mathbf{w}$ is represented as the concatenation of a pretrained word embedding w_i and a randomly initialized POS tag embedding t_i. Similar to `RELATIONSHIP` nodes in SGs, the representation of dependency between two words, $w_{i\to j}$ is extracted by neural networks fed with (w_i, w_j). We use Biaffine scorers [8] for the first-order relationship:

$$\begin{aligned}
&w_i^{1st,parent}, w_i^{1st,child} = \text{MLP}^{1st,parent/child}(w_i)\\
&w_{i\to j}^{1st} = \text{Biaffine}(w_i^{1st,parent}, w_i^{1st,child})\\
&\text{Biaffine}(w_i, w_j) = w_i^{\text{T}}\mathbf{W}_1 w_j + (w_i + w_j)^{\text{T}}\mathbf{W}_2 + b,
\end{aligned}$$

where MLP denotes the multi-layer perceptron, $\mathbf{W}_1$, $\mathbf{W}_2$ and b are trainable parameters. The calculation of a second-order relationship's score follows a similar way.

Structure Construction Inspired by neural DT construction algorithms [15], we use an encoder-decoder framework that employs the dynamic programming algorithm (namely, inside algorithm) and calculate the posteriors of instances $p(\mathbf{pt}|\mathbf{w}, \mathbf{I})$ recursively retrieved during the structure construction.

Encoder The encoder is to produce a joint representation of an input image $\mathbf{I}$ and its corresponding caption $\mathbf{w}$. Specifically, we obtain contextual encoding $c \in \mathcal{C}$ by fusing the text features with the visual information via attention mechanisms, where $\mathcal{C}$ denotes the space for the attended language context. For each token in captions $\{w_i\}$ and SG representations $\mathcal{V} = \{v_i, v_{i\to j}\}$, we calculate attention scores between them and then obtain weighted summation over all terms, *i.e.*, $c_i = \sum \text{Attn}(w_i, v_i)w_i$. Finally, we use an average-pooling layer to summarize all information into a continuous context vector $\mathbf{s}$, which represents the global information of the vision-language context.

Decoder The decoder generates the tag sequence $\mathbf{t}$ and parse tree $\mathbf{pt}$ conditioned on the joint representation s w.r.t. the joint probability $p(\mathbf{t}, \mathbf{pt}|\mathbf{s})$. To consider the exponential scale of possible parse trees, we use dynamic programming to consider all possible dependencies over the sentence. Refer to Section 5.2 for the learning process.

Cross-modality Matching We employ cross-modality matching to align vision and language features in different levels.

Matching Score We define $sim(\cdot, \cdot)$ as the cross-modality matching function. Following Wang et al. [38], we first compute the similarity score between each $c \in \mathcal{C}$ and each $v \in \mathcal{V}$:

$$sim(v, c) = \langle v, c\rangle, \tag{1}$$

where $\langle\cdot, \cdot\rangle$ is an inter-product function. Heuristically, we can define the similarity score between instance c and the entire image $\mathbf{I}$ as

$$sim(\mathbf{I}, c) = \max_{v\in\mathcal{V}} sim(v, c), \tag{2}$$

Matching Score Enhanced by Posterior To leverage the contextual information, we use a posterior $p(c|\mathbf{s})$ computed from the decoder to reflect how likely c exists given the joint representation $\mathbf{s}$. Then we fuse the matching scores with the posteriors to provide an enhanced simliarity function, $sim^+(\mathbf{I}, c) = sim(\mathbf{I}, c) \times p(c|\mathbf{s})$.

5.2. Learning

Maximum Likelihood Estimation (MLE) With the compressed representation $\mathbf{s}_i$ for image-sentence pair $(\mathbf{I}_i, \mathbf{w}_i)$, VLGAE generates the tag sequence $\mathbf{t}_i$ and the parse tree $\mathbf{pt}$. The learning objective is to maximize the

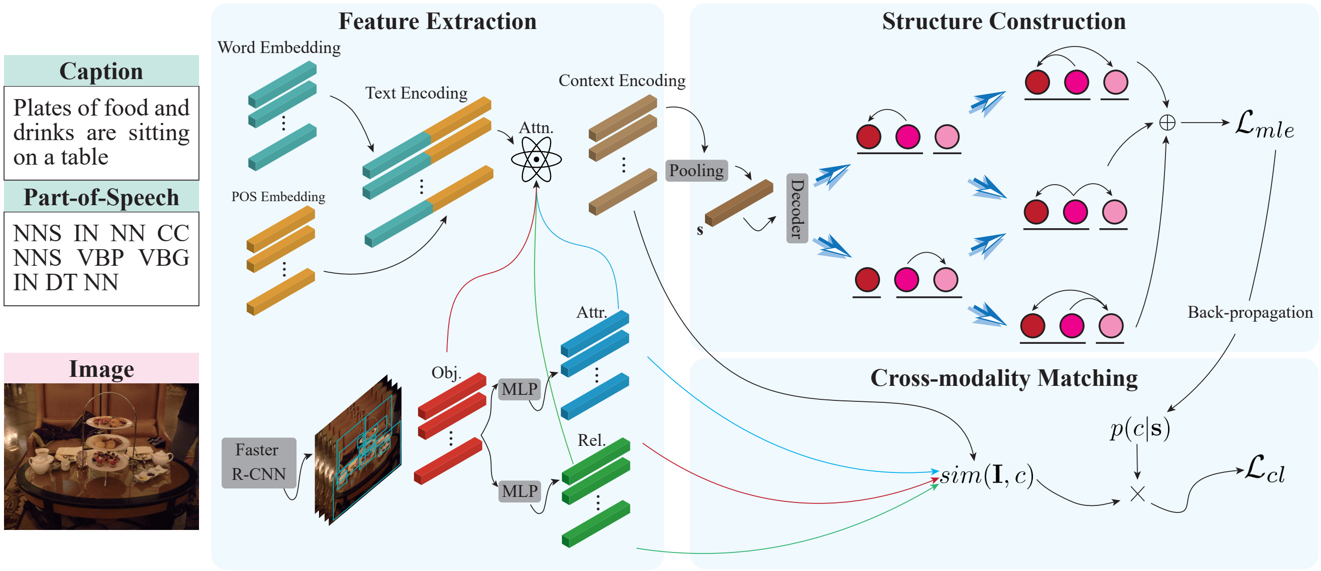

Figure 3. Diagram of VLGAE. It first extracts features from both modalities and builds representations for all instances in DT and SG. Then the encoder of the structure construction module encodes the language feature with visual cues and output a compressed representation $\mathbf{s}$. With this compressed global representation $\mathbf{s}$, the decoder incorporates the inside algorithm to construct the VL structure recursively as well as compute the posteriors. On top of the resulting posteriors generated from the structure construction module, an enhanced matching score between language context c and image region v is used to promote the cross-modality fine-grained correspondence.

conditional log-likelihood of K training sentences:

$$\begin{aligned}\mathcal{L}_{mle} &= -\frac{1}{K}\sum_{i=1}^{K}\log p_{\Theta}(\mathbf{t}_i|\mathbf{w}_i) \\ &= -\frac{1}{K}\sum_{i=1}^{K}\log\sum_{\mathbf{pt}\in PT(\mathbf{s}_i)} p_{\Theta}(\mathbf{t}_i, \mathbf{pt}|\mathbf{w}_i)\end{aligned} \tag{3}$$

where Θ parameterizes the encoder-decoder neural network and $PT(\mathbf{s}_i)$ denotes the set of all possible parse trees. Given some Θ, Eqn. (3) can be computed using the inside algorithm, an $\mathcal{O}(n^3)$ dynamic programming procedure. Therefore, we perform structure construction and parameter learning via an expectation-maximization (EM) process. Specifically, the E-step is to compute possible structures given current Θ and the M-step is to optimize Θ by gradient descent w.r.t. Eqn. (3). Of note, the posterior $p(c|\mathbf{s})$ used for matching score can be computed in the back-propagation process [11].

Contrastive Loss Due to the lack of fine-grained annotations in an unsupervised setting, the objective referring to the alignment is employed in a contrastive loss. The contrastive learning strategy is based on maximizing the matching score between paired fine-grained instances. For each c, the sentence's corresponding image is a positive example and all the other images in the current batch are negative examples. Of note, compared with coarse image-sentence pairs, our design of fine-grained vision-language alignments yield stronger negative pairs for contrastive training. Formally, given a vision-language pair $(\mathbf{w}, \mathbf{I})$ within a batch, the contrastive loss can be defined as,

$$\mathcal{L}_{cl}(\mathbf{w}, \mathbf{I}) = \mathbb{E}_{p(\mathbf{pt}|\mathbf{w})}\sum_{c\in\mathbf{pt}}\ell(\mathbf{I}, c), \tag{4}$$

$$\ell(\mathbf{I}, c) = -\log\frac{\exp[sim^{+}(\mathbf{I}, c)]}{\sum_{\hat{\mathbf{I}}\in batch}\exp[sim^{+}(\hat{\mathbf{I}}, c)]}, \tag{5}$$

where $\mathbf{pt}$ is a valid parse tree, $\hat{\mathbf{I}}$ are negative examples in a batch. $\ell(\mathbf{I}, c)$ shows an possibly aligned pair that ranks higher than other unaligned ones in a batch.

Finally, the total loss is defined as

$$\mathcal{L}_{tot} = (1-\lambda)\cdot\mathcal{L}_{mle} + \lambda\cdot\mathcal{L}_{cl}, \tag{6}$$

where λ is pre-defined to balance different scalars between two losses.

5.3. Inference

Given a trained model with trained parameters $\mathbf{\Theta}$, the model can predict the VL structure and further the parse tree of the sentence and its visual grounding on the SG. The parse tree can be parsed by searching for $\mathbf{pt}^*$ with the highest conditional probability among all valid parse trees $PT(\mathbf{s})$ using the dynamic programming [25]:

$$\mathbf{pt}^* = \underset{\mathbf{pt}\in PT(\mathbf{s})}{\arg\max}\ p(\mathbf{pt}|\mathbf{s};\mathbf{\Theta}) \tag{7}$$

For each $c \in \mathcal{C}$, we can predict its corresponding image region o_{m^c} using enhanced similarity score as in Eqn. (1):

	UDA	**DDA**
Language Only		
Left branch	53.61	30.75
Right branch	53.19	23.01
Random	32.44	19.29
DMV [25]	58.06	41.36
D-NDMV [15]	70.77	65.88
Vision-Language (VL)		
VLGAE	**71.43**	**67.57**

Table 2. Dependency structure induction results on the test split.

$$v^* = \arg\max_{v} \ sim^+(v, c) \tag{8}$$

It is worth noting that, when the ground truth dependency tree is known for sentence, we can directly retrieve corresponding scene graph w.r.t. Eqn. (8).

6. Experiments

6.1. Setup

The candidate bounding boxes are given in the following setting. For an input image, we use an external object detector, Faster R-CNN as MAF [38], to generate top-50 object proposals. For each proposal, we use RoI-Align [16] and global average pooling to compute the object feature [38]. Since we do not have the ground-truth structure of the captions, we follow [31] and [49] to use predictions as ground truth produced by an external parser. We report the average score of three runs with different random seeds.

6.2. Evaluation on Language Structure Induction

We compare VLGAE with prior language-only baselines on language structure induction using UDA and DDA metrics in Table 2. We can obverse a performance boosting after incorporating visual cues. In particular, VLGAE outperforms D-NDMV by 1.69% score on DDA and 0.66% score on UDA.

6.3. Evaluation on Visual Phrase Grounding

In addition to language structure induction, we evaluate our approach on the weakly-supervised visual phrase grounding task.[3] Experimental results in Table 3 show that VLGAE outperforms the previous mutlimodal baseline MAF [38] by 1.0%. Moreover, a significant improvement is observed, especially for high-order relations, indicating the effectiveness of our multi-order alignments. We also report performance if ground truth bounding boxes (and relationships) are used as a reference instead of proposals (and dense connections); see VLGAE† in Table 3.

	All	*Obj.*	*Attr.*	*Rel.*	*First*	*Second*
Random	12.2	15.9	9.4	0.0	0.0	0.0
MAF*	27.7	38.5	20.7	0.1	0.0	0.0
VLGAE	**28.7**	**36.1**	**21.0**	**10.2**	**3.4**	**0.2**
VLGAE†	42.3	67.2	41.8	15.9	-	-

Table 3. Visual grounding results on the test split. $*$ refers to re-implemented results. $\dagger$ refers to experiments using gold scene graphs. *All*: Zero-AA on all zero-order instances. *Obj.*: Zero-AA on objective nodes. *Attr.*:Zero-AA on attribute nodes. *Rel.*: Zero-AA on relationship nodes. *First*: First-AA. *Second*: Second-AA.

6.4. Ablation Analysis on Arc Length

We further investigate the recall rate for different lengths of arcs $len(w_{i \to j})$ in Figure 4. The experiments are on the Dev split. VLGAE enhanced by visual cues has been proven to boost DDA/UDA than its non-visual version (D-NDMV) in Table 2. Moreover, this boost is observed not only on short arcs but also longer arcs. This phenomenon is contrary to VC-PCFG [49], showing that dependency structures in VLGAE can be beneficial for all the arcs regardless of the arc length, compared with constituent structures.

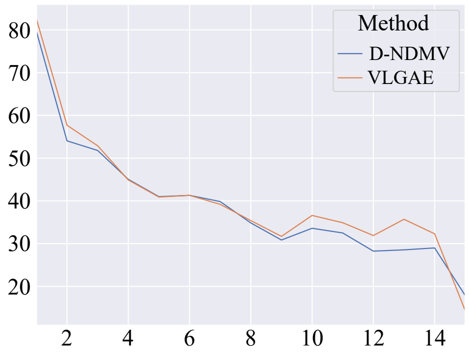

Figure 4. DDA of different arc length on the Dev dataset.

7. Conclusion

In this work, we introduce a new task `VLParse` that aims to construct a joint VL structure that leverages both visual scene graphs and language dependency trees in an unsupervised manner. Meanwhile, we deliver a semi-automatic strategy for creating a benchmark for the proposed task. Lastly, we devise a baseline framework VLGAE based on contrastive learning, aiming to construct such structure and build VL alignment simultaneously. Evaluations on structure induction and visually phrase grounding show that VLGAE enhanced by visual cues can boost performance than its non-visual version. Despite of the compelling boosted results, the performance on both tasks are far from satisfactory. Nevertheless, this work sheds light on explainable multimodal understanding and calls for future research in this direction.

[3] We apply the learning strategy of MAF on weakly-supervised visually grounding of a DT instead of given noun phrases in the training stage.

References

[1] Arjun Akula, Spandana Gella, Yaser Al-Onaizan, Song-Chun Zhu, and Siva Reddy. Words aren't enough, their order matters: On the robustness of grounding visual referring expressions. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6555–6565, Online, July 2020. Association for Computational Linguistics.

[2] Jiong Cai, Yong Jiang, and Kewei Tu. CRF autoencoder for unsupervised dependency parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1638–1643, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics.

[3] Shizhe Chen, Qin Jin, Peng Wang, and Qi Wu. Say as you wish: Fine-grained control of image caption generation with abstract scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9962–9971, 2020.

[4] Noam Chomsky. Three models for the description of language. *IRE Transactions on information theory*, 2(3):113–124, 1956.

[5] Noam Chomsky. On certain formal properties of grammars. *Information and control*, 2(2):137–167, 1959.

[6] Noam Chomsky. *Syntactic structures*. De Gruyter Mouton, 2009.

[7] Marie-Catherine De Marneffe and Christopher D Manning. Stanford typed dependencies manual. Technical report, Technical report, Stanford University, 2008.

[8] Timothy Dozat and Christopher D. Manning. Deep biaffine attention for neural dependency parsing. In *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2017.

[9] Andrew Drozdov, Subendhu Rongali, Yi-Pei Chen, Tim O'Gorman, Mohit Iyyer, and Andrew McCallum. Unsupervised parsing with s-diora: Single tree encoding for deep inside-outside recursive autoencoders. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[10] Andrew Drozdov, Pat Verga, Mohit Yadav, Mohit Iyyer, and Andrew McCallum. Unsupervised latent tree induction with deep inside-outside recursive autoencoders. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

[11] Jason Eisner. 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, Nov. 2016. Association for Computational Linguistics.

[12] Yoav Goldberg and Michael Elhadad. 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, June 2010. Association for Computational Linguistics.

[13] Jiuxiang Gu, Handong Zhao, Zhe Lin, Sheng Li, Jianfei Cai, and Mingyang Ling. Scene graph generation with external knowledge and image reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1969–1978, 2019.

[14] Tanmay Gupta, Arash Vahdat, Gal Chechik, Xiaodong Yang, Jan Kautz, and Derek Hoiem. Contrastive learning for weakly supervised phrase grounding. In *European Conference on Computer Vision (ECCV)*, pages 752–768. Springer, 2020.

[15] Wenjuan Han, Yong Jiang, and Kewei Tu. Enhancing unsupervised generative dependency parser with contextual information. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5315–5325, 2019.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[17] Yining Hong, Qing Li, Song-Chun Zhu, and Siyuan Huang. Vlgrammar: Grounded grammar induction of vision and language. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021.

[18] Chenfanfu Jiang, Siyuan Qi, Yixin Zhu, Siyuan Huang, Jenny Lin, Lap-Fai Yu, Demetri Terzopoulos, and Song-Chun Zhu. Configurable 3d scene synthesis and 2d image rendering with per-pixel ground truth using stochastic grammars. *International Journal of Computer Vision (IJCV)*, 126(9):920–941, 2018.

[19] Yong Jiang, Wenjuan Han, and Kewei Tu. Unsupervised neural dependency parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 763–771, 2016.

[20] Lifeng Jin and William Schuler. Grounded PCFG induction with images. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 396–408, Suzhou, China, Dec. 2020. Association for Computational Linguistics.

[21] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1219–1228, 2018.

[22] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3668–3678, 2015.

[23] Yoon Kim, Chris Dyer, and Alexander Rush. Compound probabilistic context-free grammars for grammar induction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2369–2385, Florence, Italy, July 2019. Association for Computational Linguistics.

[24] Yoon Kim, Chris Dyer, and Alexander Rush. 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, July 2019. Association for Computational Linguistics.

[25] Dan Klein and Christopher D Manning. Corpus-based induction of syntactic structure: Models of dependency and con-

stituency. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 478–485, 2004.

[26] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vision*, 123(1):32–73, may 2017.

[27] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1261–1270, 2017.

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *European Conference on Computer Vision (ECCV)*, pages 740–755, Cham, 2014. Springer International Publishing.

[29] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[30] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Workshop on Conference on Neural Information Processing Systems (NeurIPS)*, 2015.

[31] Haoyue Shi, Jiayuan Mao, Kevin Gimpel, and Karen Livescu. Visually grounded neural syntax acquisition. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1842–1861, 2019.

[32] Jing Shi, Yiwu Zhong, Ning Xu, Yin Li, and Chenliang Xu. A simple baseline for weakly-supervised scene graph generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16393–16402, 2021.

[33] Mohammed Suhail, Abhay Mittal, Behjat Siddiquie, Chris Broaddus, Jayan Eledath, Gerard Medioni, and Leonid Sigal. Energy-based learning for scene graph generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13936–13945, 2021.

[34] Kewei Tu, Meng Meng, Mun Wai Lee, Tae Eun Choe, and Song-Chun Zhu. Joint video and text parsing for understanding events and answering queries. *IEEE MultiMedia*, 21(2):42–70, 2014.

[35] J. R. Uijlings, K. E. Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International Journal of Computer Vision (IJCV)*, 2013.

[36] Liwei Wang, Jing Huang, Yin Li, Kun Xu, Zhengyuan Yang, and Dong Yu. Improving weakly supervised visual grounding by contrastive knowledge distillation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14090–14100, 2021.

[37] Qinxin Wang, Hao Tan, Sheng Shen, Michael Mahoney, and Zhewei Yao. MAF: Multimodal alignment framework for weakly-supervised phrase grounding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2030–2038, Online, Nov. 2020. Association for Computational Linguistics.

[38] Qinxin Wang, Hao Tan, Sheng Shen, Michael W Mahoney, and Zhewei Yao. Maf: Multimodal alignment framework for weakly-supervised phrase grounding. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[39] Shuo Wang, Yizhou Wang, and Song-Chun Zhu. Learning hierarchical space tiling for scene modeling, parsing and attribute tagging. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 37(12):2478–2491, 2015.

[40] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 2004.

[41] Tian-Fu Wu, Gui-Song Xia, and Song-Chun Zhu. Compositional boosting for computing hierarchical image structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2007.

[42] Fanyi Xiao, Leonid Sigal, and Yong Jae Lee. Weakly-supervised visual grounding of phrases with linguistic structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5945–5954, 2017.

[43] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *European Conference on Computer Vision (ECCV)*, pages 670–685, 2018.

[44] Benjamin Z Yao, Xiong Yang, Liang Lin, Mun Wai Lee, and Song-Chun Zhu. I2t: Image parsing to text description. *Proceedings of the IEEE*, 98(8):1485–1508, 2010.

[45] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. Mattnet: Modular attention network for referring expression comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1307–1315, 2018.

[46] Songyang Zhang, Linfeng Song, Lifeng Jin, Kun Xu, Dong Yu, and Jiebo Luo. Video-aided unsupervised grammar induction. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 1513–1524, 2021.

[47] Yu Zhang, Zhenghua Li, and Min Zhang. Efficient second-order TreeCRF for neural dependency parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3295–3305, Online, July 2020. Association for Computational Linguistics.

[48] Yanpeng Zhao and Ivan Titov. Visually grounded compound pcfgs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4369–4379, 2020.

[49] Yanpeng Zhao and Ivan Titov. Visually grounded compound PCFGs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4369–4379, Online, Nov. 2020. Association for Computational Linguistics.

[50] Zilong Zheng, Wenguan Wang, Siyuan Qi, and Song-Chun Zhu. Reasoning visual dialogs with structural and partial observations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6669–6678, 2019.