VLGrammar: Grounded Grammar Induction of Vision and Language

Yining Hong[1], Qing Li[1], Song-Chun Zhu[2,3,4], Siyuan Huang[1]

[1] University of California, Los Angeles

[2] Beijing Institute for General Artificial Intelligence, [3] Tsinghua University, [4] Peking University

Abstract

Cognitive grammar suggests that the acquisition of language grammar is grounded within visual structures. While grammar is an essential representation of natural language, it also exists ubiquitously in vision to represent the hierarchical part-whole structure. In this work, we study grounded grammar induction of vision and language in a joint learning framework. Specifically, we present VLGrammar, a method that uses compound probabilistic context-free grammars (compound PCFGs) to induce the language grammar and the image grammar simultaneously. We propose a novel contrastive learning framework to guide the joint learning of both modules. To provide a benchmark for the grounded grammar induction task, we collect a large-scale dataset, PARTIT, which contains human-written sentences that describe part-level semantics for 3D objects. Experiments on the PARTIT dataset show that VLGrammar outperforms all baselines in image grammar induction and language grammar induction. The learned VLGrammar naturally benefits related downstream tasks. Specifically, it improves the image unsupervised clustering accuracy by 30%, and performs well in image retrieval and text retrieval. Notably, the induced grammar shows superior generalizability by easily generalizing to unseen categories. Code and pre-trained models are released at https:// github.com/evelinehong/VLGrammar.

1. Introduction

Natural and man-made dynamical systems tend to have a nested multi-scale organization, which might be a general property of all physical and biological systems. According to [37], building complex stable systems requires the reuse of stable sub-systems that can be assembled to build larger systems. Therefore, exploring the low-dimensional structures in sensory data is critical for understanding the world and helping the design, interpretation, and generalization of artificial intelligent systems. Similarly, inducing the underlying structures and grammars from raw sensory inputs, *e.g.*, vision and language [11, 51, 39, 23, 38, 5, 40,
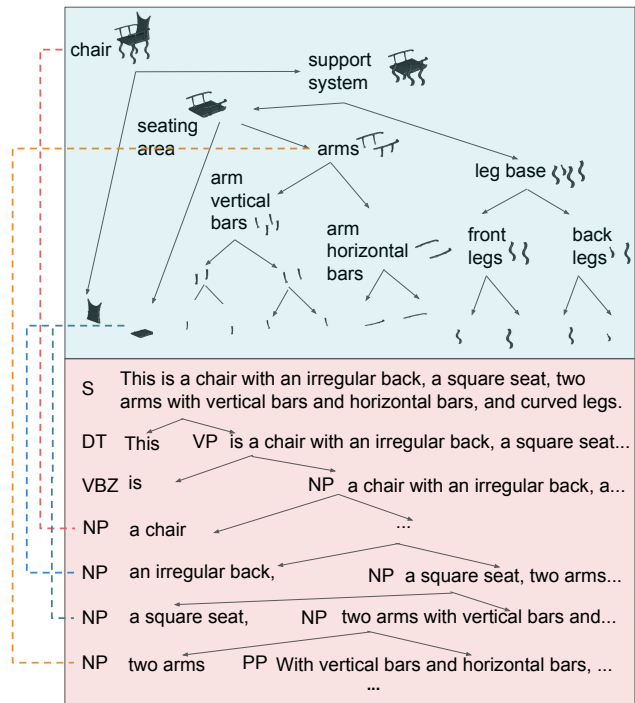

Figure 1: **An example of a sentence parse tree aligned with an image parse tree.** The arrow lines represent production rules of the image grammar and the language grammar. The dashed lines represent alignment between the constituents of two modalities.

54, 48, 32, 14, 26, 15], has been a long-standing challenge in the field of artificial intelligence (AI).

With the development of unsupervised learning techniques, the unsupervised grammar induction for natural language [33, 34, 21, 20] has recently made satisfying progress. These works formulate the grammar induction of language as a self-contained system that relies solely on textual corpora. Following this trend, [35, 53] propose the visually grounded grammar induction. They empirically show that if the constituents in a sentence's parse tree are well aligned with the image that the sentence describes, the induced grammar will be more accurate.

Visually grounded grammar induction takes one step further towards *cognitive grammar* [24, 25], a concept from

linguistic theory. Cognitive grammar argues that it is pointless to analyze grammatical units without reference to their semantics, which is grounded and structured by patterns of perception, such as vision. However, previous works ground all the constituents of a sentence with the embedding of a single image [35, 53]. They focus on aligning the image feature to language grammar but miss the hierarchical structures in the image. This is inconsistent with cognitive grammar's notion that a constituent's semantic value does not reside in one individual image base, but rather in the relationship between the substructure and the base.

Part-whole relationships are crucial in semantic structures [16]. For example, the constituent "two arms" in Fig. 1 does not simply refer to a chair, but instead refers to the chair's arms. Thus, it is necessary to align the language grammar with the hierarchical structures in physical objects. As shown in Fig. 1, a visual object can be parsed into parts with hierarchical structures, and constituents that describe parts of an object can be naturally grounded with the parts at different hierarchies.

While the study of the hierarchical structure of images has a long history [11, 51, 32, 14, 13, 45, 54], the structure is mainly pre-defined by human and static across images. Therefore, challenges remain as: (1) how to represent flexible part-whole hierarchies that vary with images using an identical network [14], and (2) how to learn structure automatically without pre-defined templates. One possible way is to learn the image grammar that parses an object into parts. Instead of allocating neurons to represent nodes in the parse graph, we can use neurons to represent grammar rules. The grammar rules are general for all the images and can be recursively re-used to handle arbitrarily complicated objects (*e.g.*, a chair can have an arbitrary number of legs).

Inspired by the above ideas, we present VLGrammar, a framework that jointly learns image and language grammar. Specifically, we use compound probabilistic context-free grammars (compound PCFGs), which parameterize a PCFG's rule probabilities with neural networks and relax the context-free constraints with a latent compound variable. To achieve grounded learning, we calculate an alignment score between the image parse tree and the language parse tree, and use a contrastive loss to learn the compound PCFGs for both image and language jointly.

To obtain data that contains multi-modal part-whole information for learning grounded grammars, we collect a large-scale dataset, PARTIT, which contains 10,613 manually annotated descriptive sentences paired with the images of objects and parts. The sentences collected via Amazon Mechanic Turk (AMT) describe the detailed object and part semantics for 3D objects.

Experiments on the proposed PARTIT dataset show that our proposed VLGrammar outperforms all baselines in both image grammar induction and language grammar induction.

Moreover, it naturally benefits related downstream tasks, for example, improving the accuracy of unsupervised part clustering from ∼40% to ∼70%, and achieving better performance in the image-text retrieval tasks. Our image grammar trained on `chair` and `table` can be easily generalized to unseen categories such as `bed` and `bag`. Qualitative studies also show that our method is capable of predicting part-whole hierarchies and recursive structures of objects, as well as constituency parsing of sentences.

Our contributions can be summarized as follows:
- To benchmark the grounded grammar induction problem, we collect a large-scale dataset, PARTIT, which contains human-written sentences describing both object-level and part-level semantics for 3D objects.
- We propose VLGrammar, which utilizes compound PCFGs to induce grounded grammars for both vision and language by enforcing the cross-modal alignment.
- We conduct extensive experiments on the PARTIT dataset. The results demonstrate the superiority of the proposed VLGrammar on the grammar induction and downstream tasks, such as unsupervised part clustering and image-text retrieval. VLGrammar also shows great generalization ability on unseen object categories.

2. Related Work

2.1. Grammar Induction of Language

Grammar induction has a long history in natural language processing [39, 23, 38, 5, 40]. Recently, researchers focus on using neural networks to induce parse trees solely from sentences [34, 33, 8, 21, 20].

These approaches mostly suggest that language is an autonomous system that does not rely on perceptions and semantics. This notion departs from the cognitive grammar concept in linguistic theory which emphasizes the role of semantic structure in grammar induction. To address this issue, visually-grounded grammar induction is proposed [35, 53]. However, they use a single image to estimate the *concreteness* of language spans (*i.e.*, to define concrete words as those referring to perception) [46, 19], which is insufficient to represent the full semantic structure. Given that a language constituent is typically associated with a specific part of the image, we propose to align a constituent with a specific part in the vision structure.

2.2. Hierarchical Structure of Images

The study of the hierarchical structure of images has been the interest of researchers for decades, ranging from syntactic pattern recognition [11, 51], graph grammars [10, 13], to and-or graphs [54, 48, 44], capsule networks [32, 14], and hierarchical shape segmentation [52, 30, 50]. Grammar models are frequently used to model hierarchical relations and build structured representations.

However, the vision structure and grammar for these works are mostly pre-defined or learned with supervision. Previous works attempt to induce image grammar in an unsupervised manner [44, 36], but also use dense pre-defined operations on the nodes. In this paper, we propose joint grammar induction of image and language via compound PCFGs [20] in a self-supervised manner, which eliminates most pre-defined structures.

2.3. Grounded Vision and Language Learning

In recent years, there have been lots of efforts and advances on exploiting the cross-modality alignment between vision and language for various tasks, such as image-text retrieval [22, 18], image captioning [17, 29, 49], and visual question answering [28, 1]. These works align the objects in images and the words in sentences either explicitly by building the visual-word mapping [22, 18, 17], or implicitly by modeling the cross-modality attention [49, 28, 1]. Most recently, there has been a surge of interest in multi-modal pre-training for representation learning in vision-and-language tasks [42, 41, 27, 43, 4]. These works extend BERT [6], a popular pre-training framework for natural language understanding, to multi-modalities by pre-training on large-scale image/video and text pairs, then fine-tuning on downstream tasks. These multimodal BERT's success greatly relies on encoding the alignment between words and image regions into attention flows in the Transformer architecture [47].

In this work, we share a similar spirit of structurally aligning visual and textual elements to facilitate grammar induction in both vision and language. The intuition behind this practice is that forcing the multimodal alignment can reduce the inherent ambiguity of grammar induction for individual modalities, and the induced grammar can be more effective for downstream tasks with its structured representation.

3. The PARTIT Dataset

We present PARTIT, a large-scale dataset of manually annotated sentences that describe both the object-level and the part-level features of an object. To the best of our knowledge, it is the first dataset with annotated natural language sentences that describe both object semantics and fine-grained part semantics paired with images.

We use AMT to collect such sentences. Given an image of an object together with the images of highlighted parts of the object, a worker is asked to use one sentence to describe all parts of the object. The workers can describe the shape, size, and amount of the parts as well as the type of the object (*e.g.*, a chair can be a folding chair, office chair, sofa, *etc.*). The annotating interface, detailed instructions, and examples that we provide for workers can be found in the *supplementary material*.

We obtain ∼10,000 3D CAD models and their part annotations from the PartNet dataset [31]. We choose four categories of objects: chair, table, bed, and bag. These categories are picked because they are geometrically complex, highly diverse, and have rich grammar hierarchies. While the PartNet dataset provides part annotations at multiple levels (coarse, middle, and fine-grained) based on and-or grammar, we propose to learn the grammar in an unsupervised manner without annotation. We only take the fine-grained parts from the PartNet dataset, following their original order of decomposition. We combine certain minuscule parts (*e.g.*, knob and connector) with their parents for simplicity. Based on the and-or templates provided by PartNet, we generate ground-truth grammar rules of each object category for evaluation only, which are listed in the *supplementary material*.

Table 1 shows the statistics of our dataset. We observe that the median number of grammar rules used per object is 8, which suggests that the part grammar is complex enough to be learned. For language, the median length of the sentences is 16, which is much longer comparing to existing image captioning datasets (*e.g.*, previous visually-grounded grammar induction models [35, 53] use MSCOCO, which only has an average length of 10 words per sentence). Apart from grammar induction, the dataset can be used in related downstream tasks, *e.g.*, image captioning, language-guided part segmentation, 3D reconstruction and so on. Examples of the PARTIT dataset are shown in Fig. 2.

Table 1: **The statistics of the PARTIT dataset.** #PS is the number of part semantics, and #G is the number of grammar rules. P_{med} and P_{max} denote the median and maximum numbers of part instances per object, respectively. G_{med} and G_{max} denote the median and maximum number of grammar rules used per object, respectively. LG_{med} and LG_{max} denote the median and maximum length of sentences, respectively, and $Vocab$ denote the size of the language vocabulary.

	All	Chair	Table	Bed	Bag
#I	10613	5031	5290	185	109
#PS	120110	13	10	8	3
#G	75	23	34	18	4
P_{med}	7	8	6	9	3
P_{max}	136	38	136	28	6
G_{med}	8	8	8	7	3
G_{max}	18	12	18	15	4
LG_{med}	16	19	13	19	15
LG_{max}	98	98	68	42	21
$Vocab$	2007	1634	903	176	61

4. Grounded Grammar Induction

In this section, we introduce the proposed VLGrammar for grounded grammar induction in both vision and language. Our model starts from the compound PCFG for in-
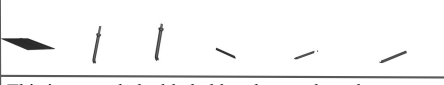
Figure 2: **Examples from our PARTIT dataset.** The annotator is asked to use one sentence to describe all the parts of an object.

ducing the language grammar [20] and generalizes this idea to vision, which are jointly optimized by a contrastive loss.

4.1. Compound PCFG for Language

A context-free grammar (CFG) can be 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 nonterminal nodes, $\mathcal{P}$ is a finite set of preterminal nodes, Σ is a finite set of terminal nodes, and $\mathcal{R}$ is a set of production rules in the Chomsky normal form:

$$
\begin{aligned}
S &\to A, & A &\in \mathcal{N} \\
A &\to BC, & A &\in \mathcal{N}, B, C \in \mathcal{N} \cup \mathcal{P} \\
T &\to w, & T &\in \mathcal{P}, w \in \Sigma
\end{aligned}
\tag{1}
$$

In natural language, nonterminals $\mathcal{N}$ are constituent labels and preterminals $\mathcal{P}$ are part-of-speech tags. A terminal node w is a word from a sentence, and Σ is the vocabulary. During implementation, we do not have the ground truth constituent labels and part-of-speech tags. Therefore, nonterminals and preterminals are sets of nodes (or clusters) which implicitly represent their functions.

Probabilistic context-free grammars (PCFGs) extend CFGs by assigning a probability π_r to each production rule $r \in \mathcal{R}$ such that $\sum_{r:A\to\gamma} \pi_r = 1$, *i.e.*, the probabilities of production rules with the same left-hand-side nonterminal sum to 1. Kim et al. [20] propose a neural parameterization where rule probabilities are based on distributed representations. To mitigate the strong context-free assumption, they extend neural PCFGs to compound PCFGs by assuming that rule probabilities follow a compound probability distribution [3]:

$$
\pi_r = g_r(\mathbf{z}; \theta), \quad \mathbf{z} \sim p(\mathbf{z})
\tag{2}
$$

where $p(\mathbf{z})$ is a prior distribution of the latent variable $\mathbf{z}$, and rule probability π_r is parameterized by θ. π_r takes one

of the following forms:

$$
\pi_{S\to A} = \frac{\exp\left(\mathbf{u}_A^T f_s\left([\mathbf{w}_S; \mathbf{z}]\right)\right)}{\sum_{A'\in\mathcal{N}} \exp\left(\mathbf{u}_{A'}^T f_s\left([\mathbf{w}_S; \mathbf{z}]\right)\right)}
\tag{3}
$$

$$
\pi_{A\to BC} = \frac{\exp\left(\mathbf{u}_{BC}^T [\mathbf{w}_A; \mathbf{z}]\right)}{\sum_{B',C'\in\mathcal{N}\cup\mathcal{P}} \exp\left(\mathbf{u}_{B'C'}^T [\mathbf{w}_A; \mathbf{z}]\right)}
\tag{4}
$$

$$
\pi_{T\to w} = \frac{\exp\left(\mathbf{u}_w^T f_t\left([\mathbf{w}_T; \mathbf{z}]\right)\right)}{\sum_{w'\in\Sigma} \exp\left(\mathbf{u}_{w'}^T f_t\left([\mathbf{w}_T; \mathbf{z}]\right)\right)}
\tag{5}
$$

where $\mathbf{u}$ is a parameter vector, $\mathbf{w}_N$ ($N \in \{S\} \cup \mathcal{N} \cup \mathcal{P}$) is a symbol embedding. $[\cdot; \cdot]$ indicates vector concatenation, and $f_s(\cdot)$ and $f_t(\cdot)$ are feedforward neural networks that encode the inputs.

In the compound PCFG, the log marginal likelihood $\log p_\theta(\boldsymbol{w})$ of the observed sentence $\boldsymbol{w} = w_1 w_2 \ldots w_n$ can be obtained by summing out the latent tree structure using the inside algorithm [2]:

$$
\log p_\theta(\boldsymbol{w}) = \log \int_{\mathbf{z}} \sum_{t\in\mathcal{T}_\mathcal{G}(\boldsymbol{w})} p_\theta(t \mid \mathbf{z}) p(\mathbf{z}) d\mathbf{z}
\tag{6}
$$

where $\mathcal{T}_\mathcal{G}$ consists of all parses of the sentence $\boldsymbol{w}$ under a grammar $\mathcal{G}$. Since the integral over z makes this likelihood intractable, Compound PCFGs use amortized variational inference and compute the loss based on the evidence lower bound (ELBO):

$$
\begin{aligned}
\mathcal{L}_g(\boldsymbol{w}; \phi, \theta) &= -\operatorname{ELBO}(\boldsymbol{w}; \phi, \theta) \\
&= -\mathbb{E}_{q_\phi(\mathbf{z}|\boldsymbol{w})}\left[\log p_\theta(\boldsymbol{w} \mid \mathbf{z})\right] + \operatorname{KL}\left[q_\phi(\mathbf{z} \mid \boldsymbol{w}) \| p(\mathbf{z})\right]
\end{aligned}
\tag{7}
$$

where $q_\phi(\mathbf{z} \mid \boldsymbol{w})$ is a variational posterior modeled by a neural network parameterized by ϕ.

4.2. Compound PCFG for Imgae

Compound PCFGs can be naturally extended to image grammar. In a compound PCFG for image, S denotes an object, *e.g.*, a chair. Nonterminals $\mathcal{N}$ are types of middle-level coarse parts. Preterminals $\mathcal{P}$ are types of fine-grained leaf-parts. The middle-level parts can be further decomposed into sub-parts which are either middle-level parts or leaf-parts; for example, the base of a chair is decomposed into the central support and the leg system, and the leg system is further decomposed into several legs.

Eq. (3) and Eq. (4) can be directly applied to represent the compound PCFG for image. However, Eq. (5) does not work for image, since we do not have a fixed vocabulary for images, and terminal nodes are varied *w.r.t* pixels. To address this problem, we design a bottom-up perception module to substitute the top-down generation in Eq. (5).

4.2.1 Bottom-Up Perception

Preterminals T can be viewed as a set of clusters that group the terminal nodes since we do not have ground-truth labels. Therefore, instead of inducing the top-down grammar,
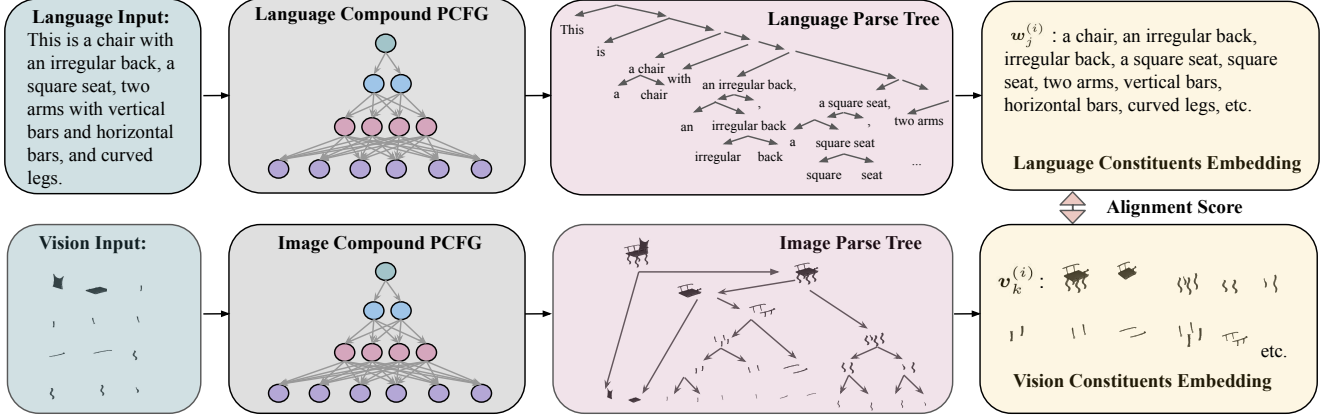
Figure 3: **Our proposed VLGrammar framework.** We implement image grammar induction and language grammar induction via compound PCFGs. Parse trees are derived from the grammars. We compute alignment scores between the vision and language constituents in the parse trees to guide the joint learning procedure.

we use a bottom-up perception module to propose terminal nodes for T.

We consider the terminal nodes to be a sequence of leaf-parts of an object $v = v_1v_2...v_n$. We want to assign a tag T to each leaf part v_i.

$$s(T, v_i) = \mathbf{u}_T^T f_t \left(\psi(v_i) \right) \tag{8}$$

where ψ is a perception module, *i.e.*, ResNet-18 in our model. f_t is a clustering model, which is a single-layer feed-forward neural network that gives the score of clustering leaf-part v_i to the tag T and $\mathbf{u}_T$ is a parameter vector for the tag T. The rule probability of a preterminal to a leaf-part is thus:

$$\pi_{T \to v_i} = \frac{exp(s(T, v_i))}{\sum_{v' \in \Sigma} exp(s(T, v'))} \tag{9}$$

All leaf parts in a training batch constitute Σ.

We maximize the log-likelihood of the part sequence with ELBO:

$$\mathcal{L}_g(\boldsymbol{v}; \phi, \theta) = -\mathbb{E}_{q_\phi(\mathbf{z}|\boldsymbol{v})} \left[\log p_\theta(\boldsymbol{v} \mid \mathbf{z}) \right] - \text{KL} \left[q_\phi(\mathbf{z} \mid \boldsymbol{v}) \| p(\mathbf{z}) \right] \tag{10}$$

where $q_\phi(\mathbf{z} \mid \boldsymbol{v})$ is a variational posterior.

Note that the image sequence $\boldsymbol{v}$ is independent of z given the tags $\boldsymbol{T} = T_1 T_2 ... T_n$ of $\boldsymbol{v}$. Therefore,

$$p_\theta(\boldsymbol{v} \mid \mathbf{z}) = \sum_{\boldsymbol{T}} p_{\theta_\psi}(\boldsymbol{v}|\boldsymbol{T}) p_{\theta_\mathcal{G}}(\boldsymbol{T}|\mathbf{z})$$
$$\propto \sum_{\boldsymbol{T}} p_{\theta_\psi}(\boldsymbol{T}|\boldsymbol{v}) p_{\theta_\mathcal{G}}(\boldsymbol{T}|\mathbf{z}) \tag{11}$$

where we sum over all possible tags for the part. θ_ψ denotes the parameters of the clustering module, and $\theta_\mathcal{G}$ denotes the parameters of Eq. (3) and Eq. (4) in the image grammar.

We notice that if $\boldsymbol{T}$ has higher probability given by the grammar module, $p_{\theta_\mathcal{G}}(\boldsymbol{T}|\boldsymbol{z})$ has a larger value, thus gives larger weight for $p_{\theta_\psi}(\boldsymbol{T}|\boldsymbol{v})$. This means $\boldsymbol{T}$ is more likely to

be the accurate clusters over the images if it conforms to the current grammar. Therefore, the grammar module can boost the training of the clustering module, and *vice versa*. This is demonstrated in Section 5.2.2. In practice, a pre-trained clustering module can speed up the training.

4.3. Joint Learning by Alignment

We propose to jointly learn the grammars for image and language by aligning the paired image and sentence. Similar to [35] and [53], we use an end-to-end contrastive learning framework. While they align each language constituent with a single image, we compute an alignment score between each language constituent and each visual constituent.

Given a sentence $\boldsymbol{w} = w_1 \ldots w_m$ where m is the total number of words, a language constituent is defined as a span over this sentence, denoted as $\mathbf{w}_j = w_a \ldots w_b \in [\boldsymbol{w}]$ where $0 < a < b \leqslant m$ and $[\boldsymbol{w}]$ denotes the set of all possible spans over $\boldsymbol{w}$. We use a Bi-LSTM to obtain the embedding of a language constituent:

$$\boldsymbol{w}_j = f_w \left(\frac{1}{b - a + 1} \sum_{l=a}^{b} \mathbf{h}_l \right) \tag{12}$$

where $\mathbf{h}_l$ is the hidden state of the Bi-LSTM, and f_w is an affine transformation. We average the label-specific representations like in [53].

Given an object $\boldsymbol{v} = v_1 \ldots v_n$ where n is the total number of parts, a visual constituent is defined as a span over this part sequence, denoted as $\mathbf{v}_k = v_c \ldots v_d \in [\boldsymbol{v}]$ where $0 < c < d \leqslant n$ and $[\boldsymbol{v}]$ denotes the set of all possible sub-parts over $\boldsymbol{v}$. We define the embedding of a visual constituent as:

$$\boldsymbol{v}_k = f_v \left(\frac{1}{d - c + 1} \sum_{l=c}^{d} \psi(v_l) \right) \tag{13}$$

where ψ is the perception module from Eq. (8) and f_v is an affine transformation.

The alignment score between a language constituent and a visual constituent is defined as their cosine similarity:

$$s(\boldsymbol{w}_j, \boldsymbol{v}_k) \triangleq cos(\boldsymbol{w}_j, \boldsymbol{v}_k) \qquad (14)$$

The alignment score between a sentence and an image is:

$$
\begin{aligned}
\mathcal{S}(\boldsymbol{w}, \boldsymbol{v}) &= \sum_{\substack{t_w \in \mathcal{T}_{\mathcal{G}_w}(\boldsymbol{w}) \\ t_v \in \mathcal{T}_{\mathcal{G}_v}(\boldsymbol{v})}} p(t_w|\boldsymbol{w}) p(t_v|\boldsymbol{v}) \sum_{\substack{\boldsymbol{w}_j \in t_w \\ \boldsymbol{v}_k \in t_v}} s(\boldsymbol{w}_j, \boldsymbol{v}_k) \\
&= \sum_{\substack{\boldsymbol{w}_j \in [\boldsymbol{w}] \\ \boldsymbol{v}_k \in [\boldsymbol{v}]}} \sum_{\substack{t_w \in \mathcal{T}_{\mathcal{G}_w}(\boldsymbol{w}) \\ t_v \in \mathcal{T}_{\mathcal{G}_v}(\boldsymbol{v})}} \mathbb{1}_{\{\boldsymbol{w}_j \in t_w\}} \mathbb{1}_{\{\boldsymbol{v}_k \in t_v\}} p(t_w|\boldsymbol{w}) p(t_v|\boldsymbol{v}) s(\boldsymbol{w}_j, \boldsymbol{v}_k) \\
&= \sum_{\substack{\boldsymbol{w}_j \in [\boldsymbol{w}] \\ \boldsymbol{v}_k \in [\boldsymbol{v}]}} p(\boldsymbol{w}_j|\boldsymbol{w}; \mathcal{G}_w) p(\boldsymbol{v}_k|\boldsymbol{v}; \mathcal{G}_v) s(\boldsymbol{w}_j, \boldsymbol{v}_k)
\end{aligned}
$$
$$(15)$$

where $p(\boldsymbol{w}_j|\boldsymbol{w}; \mathcal{G}_w) = \sum_{t_w \in \mathcal{T}_{\mathcal{G}_w}(\boldsymbol{w})} \mathbb{1}_{\{\boldsymbol{w}_j \in t_w\}} p(t_w|\boldsymbol{w})$ and $p(\boldsymbol{v}_k|\boldsymbol{v}; \mathcal{G}_v) = \sum_{t_v \in \mathcal{T}_{\mathcal{G}_v}(\boldsymbol{v})} \mathbb{1}_{\{\boldsymbol{v}_k \in t_v\}} p(t_v|\boldsymbol{v})$ are the conditional probabilities of a constituent given the sentence/object, marginalized over all possible parse trees under the current grammars. They can be efficiently computed with the inside algorithm and automatic differentiation [9].

Given a training batch $\mathcal{D} = \{\mathcal{W}, \mathcal{V}\} = \{(\boldsymbol{w}^{(i)}, \boldsymbol{v}^{(i)})\}$, the contrastive loss is defined as:

$$
\begin{aligned}
\mathcal{L}_C(\mathcal{W}, \mathcal{V}) = &\sum_{i, m \neq i} [\mathcal{S}(\boldsymbol{w}^{(m)}, \boldsymbol{v}^{(i)}) - \mathcal{S}(\boldsymbol{w}^{(i)}, \boldsymbol{v}^{(i)}) + \delta]_+ \\
+ &\sum_{i, m \neq i} [\mathcal{S}(\boldsymbol{w}^{(i)}, \boldsymbol{v}^{(m)}) - \mathcal{S}(\boldsymbol{w}^{(i)}, \boldsymbol{v}^{(i)}) + \delta]_+
\end{aligned}
$$
$$(16)$$

where δ is a constant margin, and $[\cdot]_+$ denotes $max(0, \cdot)$.

The overall training loss function is then:

$$\mathcal{L} = \lambda_w \mathcal{L}_{\mathcal{G}}(\mathcal{W}; \phi_w, \theta_w) + \lambda_v \mathcal{L}_{\mathcal{G}}(\mathcal{V}; \phi_v, \theta_v) + \lambda_C \mathcal{L}_C(\mathcal{W}, \mathcal{V}) \qquad (17)$$

where $\lambda_w, \lambda_v, \lambda_C$ are hyperparameters, and $\phi_t, \theta_t, \phi_v, \theta_v$ denote the parameters of the language and visual compound PCFGs, respectively.

5. Experiments and Results

5.1. Experimental Setup

5.1.1 Dataset

We evaluate our model and the baseline models on the PAR-TIT dataset that we collected. We obtain 2D images of the 3D objects via Blender[1]. If a part is occluded by other parts, the corresponding part image shows only the visible portion of the part. The final dataset is randomly divided into a training set of size 8,459 and a test set of size 2,154 (*i.e.*, approximately 80%/20% split).

5.1.2 Evaluation Tasks

Grammar Induction We evaluate the learned grammar of both image and language. For image, we manually parse the parts into parse trees based on production rules as ground-truth. For language, we apply Benepar [2] to obtain constituency parse trees as ground-truth. We report both the averaged corpus-level F1 score and the averaged instance-level[3] F1 score against these ground-truth parse trees.

Part Clustering We report the accuracy of the unsupervised part clustering module to examine whether the learned image grammar can improve the part clustering results.

Image-Text Retrieval We evaluate text-to-image retrieval and image-to-text retrieval. When presented a sentence and eight candidate images, a model chooses the image that has the highest alignment score with the given sentence, as defined by Eq. (15). The image-to-text retrieval is performed likewise.

5.1.3 Baselines

We compare the proposed VLGrammar with the following baselines:

Simple tree structures We use two simple baselines: left-branching binary trees and right-branching binary trees.

Ordered neurons (ON-LSTM) Shen et al. [34] use ON-LSTM cells to predict the syntactic distance between adjacent words to induce tree structures.

Compound PCFGs We use a language compound PCFG (L-PCFG) and a vision compound PCFG (V-PCFG) to induce language grammar and image grammar separately.

Grounded compound PCFGs Zhao et al. [53] propose to learn language compound PCFGs grounded in pretrained image features, denoted as L-PCFG-VG here. For a fair comparison, we take the average of all part embeddings as the feature of an image in our setting. Similarly, we train a language-grounded compound PCFG for vision which learns image grammar grounded in pretrained language features, denoted as V-PCFG-LG.

5.1.4 Implementation Details

For all models that induce image grammar, we use ResNet-18 to extract the features of part images. We pretrain the ResNet-18 over unlabeled part images using an unsupervised clustering method SCAN [12]. The part features are also used in L-PCFG-VG to ground the language grammar. For V-PCFG-LG, we use BERT [7] for pretrained language embedding to ground the image grammar.

Table 2: **The performance of grammar induction.** "C" and "I" denote corpus-level and instance-level F1 scores, respectively. "VLG w/o SCAN" denotes that we do not use SCAN to pretrain the unsupervised clustering module of VLGrammar.

Model	Vision Grammar										Language Grammar									
	All		Chair		Table		Bed		Bag		All		Chair		Table		Bed		Bag	
	C	I	C	I	C	I	C	I	C	I	C	I	C	I	C	I	C	I	C	I
Left-Branch	16.4	20.2	9.9	11.5	21.1	26.3	**38.8**	59.4	54.2	60.0	16.2	17.6	19.2	19.8	13.7	15.8	10.5	12.0	8.4	8.9
Right-Branch	40.8	49.1	42.8	48.0	39.1	50.2	12.8	20.8	81.0	97.5	49.2	**53.5**	43.7	48.6	54.2	**58.1**	43.7	46.2	68.3	69.3
ON-LSTM	/	/	/	/	/	/	/	/	/	/	30.7	33.4	32.5	34.4	28.9	32.4	27.3	29.0	39.4	38.5
L-PCFG-P	/	/	/	/	/	/	/	/	/	/	47.8	49.4	41.4	44.9	53.6	53.5	44.9	44.3	63.7	63.5
L-PCFG	/	/	/	/	/	/	/	/	/	/	48.4	50.3	42.2	46.2	53.6	53.5	55.3	**55.1**	71.2	71.4
V-PCFG	47.5	59.3	51.6	59.0	43.3	59.2	36.2	48.2	82.4	91.3	/	/	/	/	/	/	/	/	/	/
L-PCFG-VG	/	/	/	/	/	/	/	/	/	/	49.0	49.6	42.3	44.0	**54.6**	54.3	56.0	54.6	73.0	73.0
V-PCFG-LG	44.2	52.7	42.0	47.5	45.6	56.6	**38.8**	54.3	88.2	95.7	/	/	/	/	/	/	/	/	/	/
VLGrammar	**51.4**	**63.4**	**56.4**	**65.9**	46.3	60.5	38.1	**59.7**	94.1	98.0	51.3	51.9	47.8	49.4	54.0	53.8	**56.2**	54.8	**73.6**	**73.6**
VLG w/o SCAN	44.7	55.5	30.5	33.6	**57.9**	**75.4**	29.0	56.4	88.2	95.7	49.0	49.8	43.4	45.3	53.7	53.5	55.1	54.0	72.6	72.6

Since the sentences describing different object categories share similar features in the language, we pretrain a category-agnostic language compound PCFG on the sentences across all types for 100 epochs, denoted as L-PCFG-P. Then we fine-tune the language grammar on each object category with L-PCFG, L-PCFG-VG and VLGrammar. All the models are trained for 100 epochs. The training hyperparameters are specified in the *supplementary material*.

5.2. Results

5.2.1 Grammar Induction

Table 2 shows the main results of grammar induction of vision and language. Our method outperforms all baselines by a large margin with regard to image F1 scores. Notably, for the image grammar on `table`, VLGrammar w/o SCAN for unsupervised clustering outperforms other models significantly. It shows that our proposed VLGrammar can learn unsupervised clustering and image grammar jointly from scratch. For a category that has simple structure like `bag`, VLGrammar can achieve nearly perfect performance.

For language grammar induction, our method is superior to all neural baselines, but slightly worse than right branching binary trees with regard to instance-level F1. The reason is that our dataset contains very long sentences, and humans tend to make right-branching sentences when the sentences are long. Similarly, the right-branching model is also a strong baseline in previous works on language grammar induction [33, 20]. The category-agnostic language compound PCFG (L-PCFG-P) obtains decent performance and fine-tuning it on each object category can further improves the F1 scores. One possible explanation is that while describing different objects, humans tend to use different language structures.

5.2.2 Part Clustering

Table 3 shows the accuracy of the unsupervised part clustering in the bottom-up module of the image compound PCFG. Overall, after training VLGrammar, the accuracy of the part label prediction boosts from 41.3% to 69.1%. This confirms

the argument derived from Eq. (11), that the induced grammar can benefit the part clustering in a top-down manner.

One surprising observation is that even without the SCAN pretraining, VLGrammar performs quite well in the part clustering. For the `table` category, VLGrammar w/o SCAN achieves even higher accuracy than VLGrammar. The overall clustering accuracy of VLGrammar w/o SCAN is 64.4%, which also outperforms the accuracy of SCAN (41.3%) significantly. This can be an inspiration for unsupervised clustering: while we do not have ground truth labels, modeling the underlying structure might provide a strong learning signal for boosting the clustering.

Table 3: **The accuracy of the unsupervised part clustering.**

Model	All	Chair	Table	Bed	Bag
SCAN	41.3	43.5	37.5	59.3	88.9
V-PCFG	61.6	68.3	58.3	69.9	88.9
V-PCFG-LG	65.4	66.8	63.2	71.8	**90.5**
VLGrammar	**69.1**	**71.6**	66.0	**75.1**	**90.5**
VLG w/o SCAN	64.4	62.0	**66.2**	60.4	**90.5**

5.2.3 Image-Text Retrieval

Since an alignment score is computed to measure the similarity between an image and a sentence, it's natural to use it for image-text retrieval. For text-to-image retrieval, given one descriptive sentence, the model chooses the answer among eight images. For image-to-text retrieval, the model chooses among eight sentences to pair with the given object. All models are trained using contrastive loss. The baseline model is a simple model that uses ResNet-18 as image encoder and BERT as sentence encoder. Table 4 shows the results. VLGrammar can outperform the baseline by a large margin and achieve satisfying performance, which is an extra bonus naturally earned with our grammar induction framework.

5.2.4 Cross-category Generalization

Different object categories share certain common structures among their parts, making it possible to generalize

Table 4: **The accuracy of image-text retrieval.** "IR" stands for text-to-image retrieval and "TR" is for image-to-text retrieval.

Model	Chair		Table		Bed		Bag	
	IR	TR	IR	TR	IR	TR	IR	TR
Baseline	24.1	28.5	29.8	31.2	20.1	20.1	19.1	24.5
L-PCFG-VG	**34.5**	36.9	39.3	42.0	35.5	**38.4**	23.0	28.7
V-PCFG-LG	25.9	27.8	38.8	41.8	29.6	25.7	23.8	24.9
VLGrammar	33.2	**39.0**	**39.8**	**42.5**	**39.6**	38.2	**24.6**	**29.3**

from learned categories to unseen categories. For example, `chair`, `table` and `bed` all have legs. To evaluate the model's generalization ability, we train a shared image compound PCFG for certain object categories, and then test on unseen categories. We merge the parts and production rules of `chair` and `table`, and train a compound PCFG model on these two categories. We then test the model on all categories including two unseen categories: `bed` and `bag`. The results shown in Table 5 indicate that the learned grammars can indeed be transferred to novel object categories.

Table 5: The performance of image grammars on all categories, while being trained on only `chair` and `table`.

Model	Seen				Unseen			
	Chair		Table		Bed		Bag	
	C	I	C	I	C	I	C	I
V-PCFG	43.9	52.7	38.1	54.5	20.7	33.1	82.4	91.3
V-PCFG-LG	44.3	**54.1**	38.5	54.8	25.6	**50.4**	**88.2**	**95.7**
VLGrammar	**44.8**	53.4	**41.1**	**56.7**	**29.4**	44.2	**88.2**	**95.7**

5.2.5 Qualitative Study

Fig. 4 visualizes several examples of parse trees predicted by VLGrammar. We summarize following observations from these examples:

Part-Whole Hierarchies Our VLGrammar can capture precise part-whole hierarchies of the images. The objects can be parsed into parts of various hierarchies. For instance, the chair can be parsed into the upper part and supporting system. The latter can be further divided into seating area and chair base. The seating area has a chair seat and arms with vertical bars and horizontal bars, which are grouped separately. In the table, the base can be divided into legs and leg bars.

Recursive Structures One interesting question is how VLGrammar deals with recursive structures. A chair can have an arbitrary number of legs, which shall be in the the same hierarchy. However, since context-free grammar is defined on binary trees, recursive grammar is used to group the parts of same functionality. We find that VLGrammar can at least learn three types of recursive structures: (1) Pair-wise grouping: VLGrammar first groups the parts into pairs according to positional information (*e.g.*, front legs and back legs, left horizontal arm bars and right horizontal arm bars, as shown in Fig. 4), and then group the pairs. (2) Right-branching: the vertical bars of the chair arms are grouped
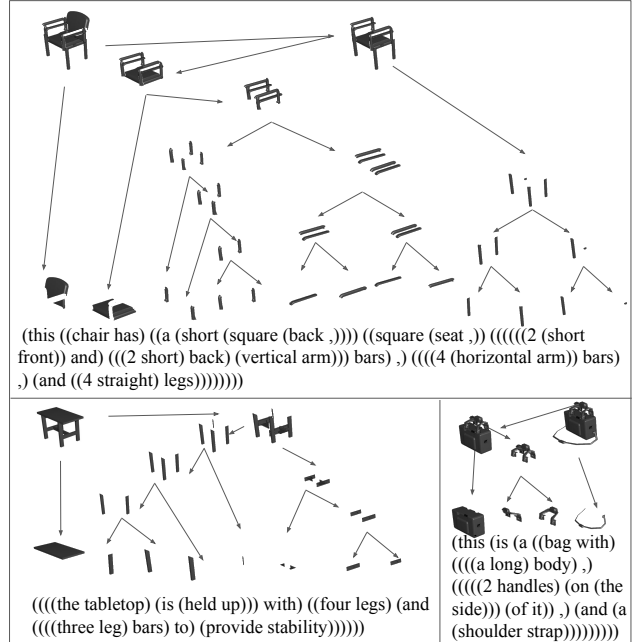

(this ((chair has) ((a (short (square (back ,)))) ((square (seat ,)) ((((((2 (short front)) and) (((2 short) back) (vertical arm))) bars) ,) ((((4 (horizontal arm)) bars) ,) (and ((4 (straight) legs)))))))))

(((((the tabletop) (is (held up))) with) ((four legs) (and ((((three leg) bars) to) (provide stability))))))

(this (is (a ((bag with) ((((a long) body) ,) ((((((2 handles) (on (the side))) (of it)) ,) (and (a (shoulder strap)))))))))

Figure 4: **Qualitative examples of parse trees predicted by VL-Grammar.** We visualize the image parse trees and the language parse trees derived by the VLGrammar. Since the language parse trees are large, we use a bracket form to represent them.

using right-branching binary trees. (3) Left-branching: *e.g.*, the grouping of the legs of the table. Right-branching and left-branching are effective when dealing with an arbitrary number of parts at the same level, and when there are no salient patterns to pair them. One example is the star leg base, where the legs are in arbitrary order and form a circle. **Language Phrases** VLGrammar excels at grouping phrases that refer to parts in the images. For example, VLGrammar can capture phrases such as "a short square back", "four legs", "three leg bars", "2 handles", "a shoulder strap", and so on. This merit comes from the learned alignment between the phrases and the referred visual parts.

6. Conclusion and Future Work

In this work, we propose VLGrammar, a framework that utilizes compound PCFGs to jointly induce the grammar of vision and language. We collect a large-scale dataset, PAR-TIT, for benchmarking this novel task. Experimental results show that VLGrammar performs well in grammar induction of vision and language, greatly benefits downstream tasks such as unsupervised part clustering and image-text retrieval, and easily generalizes to unseen categories.

One limitation of our work is that the image grammar is defined on part sequences. This practice eliminates the rich 2D structures of images. A possible solution is to define spatial grammars directly on 2D images and we leave it for future work.

References

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6077–6086, 2018. 3

[2] J. Baker. Trainable grammars for speech recognition. *Journal of the Acoustical Society of America*, 65, 1979. 4

[3] L. L. Cam. Asymptotic methods in statistical decision theory. 1986. 4

[4] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European Conference on Computer Vision (ECCV)*. Springer, 2020. 3

[5] Shay B. Cohen and Noah A. Smith. The shared logistic normal distribution for grammar induction. 2008. 1, 2

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3

[7] J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019. 6

[8] Andrew Drozdov, Pat Verga, Mohit Yadav, Mohit Iyyer, and A. McCallum. Unsupervised latent tree induction with deep inside-outside recursive autoencoders. *ArXiv*, abs/1904.02142, 2019. 2

[9] Jason Eisner. Inside-outside and forward-backward algorithms are just backprop (tutorial paper). In *SPNLP@EMNLP*, 2016. 6

[10] J. R EKERS and A. S CHÜRR. Defining and parsing visual languages with layered graph grammars 1. 1997. 2

[11] K. Fu. Syntactic pattern recognition and applications. 1968. 1, 2

[12] Wouter Van Gansbeke, Simon Vandenhende, S. Georgoulis, M. Proesmans, and L. Gool. Scan: Learning to classify images without labels. In *European Conference on Computer Vision (ECCV)*, 2020. 6

[13] F. Han and S. Zhu. Bottom-up/top-down image parsing with attribute grammar. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 31:59–73, 2009. 2

[14] Geoffrey E. Hinton. How to represent part-whole hierarchies in a neural network. *ArXiv*, abs/2102.12627, 2021. 1, 2

[15] Siyuan Huang, Siyuan Qi, Yixin Zhu, Yinxue Xiao, Yuanlu Xu, and Song-Chun Zhu. Holistic 3d scene parsing and reconstruction from a single rgb image. In *European Conference on Computer Vision (ECCV)*, 2018. 1

[16] M. Johnson. The body in the mind: the bodily basis of meaning. 1987. 2

[17] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3

[18] Andrej Karpathy, Armand Joulin, and Li Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. *arXiv preprint arXiv:1406.5679*, 2014. 3

[19] Douwe Kiela, Felix Hill, A. Korhonen, and Stephen Clark. Improving multi-modal representations using image dispersion: Why less is sometimes more. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2014. 2

[20] Yoon Kim, Chris Dyer, and Alexander M. Rush. Compound probabilistic context-free grammars for grammar induction. *ArXiv*, abs/1906.10225, 2019. 1, 2, 3, 4, 7

[21] Yoon Kim, Alexander M. Rush, L. Yu, Adhiguna Kuncoro, Chris Dyer, and Gábor Melis. Unsupervised recurrent neural network grammars. *ArXiv*, abs/1904.03746, 2019. 1, 2

[22] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014. 3

[23] D. Klein and Christopher D. Manning. A generative constituent-context model for improved grammar induction. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002. 1, 2

[24] Ronald W. Langacker. Foundations of cognitive grammar. 1983. 1

[25] Ronald W. Langacker. An introduction to cognitive grammar. *Cogn. Sci.*, 10:1–40, 1986. 1

[26] Qing Li, Siyuan Huang, Yining Hong, Yixin Chen, Ying Nian Wu, and Song-Chun Zhu. Closed loop neural-symbolic learning via integrating neural perception, grammar parsing, and symbolic reasoning. In *International Conference on Machine Learning (ICML)*. PMLR, 2020. 1

[27] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *NIPS*, 2019. 3

[28] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical co-attention for visual question answering. *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2, 2016. 3

[29] Lin Ma, Zhengdong Lu, Lifeng Shang, and Hang Li. Multimodal convolutional neural networks for matching image and sentence. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3

[30] Kaichun Mo, P. Guerrero, L. Yi, H. Su, Peter Wonka, Niloy Mitra, and L. Guibas. Structurenet: Hierarchical graph networks for 3d shape generation. *ACM Transactions on Graphics (TOG)*, 38:242:1–242:19, 2019. 2

[31] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 909–918, 2019. 3

[32] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. In *NIPS*, 2017. 1, 2

[33] Yikang Shen, Zhouhan Lin, C. Huang, and Aaron C. Courville. Neural language modeling by jointly learning syntax and lexicon. *ArXiv*, abs/1711.02013, 2018. 1, 2, 7

[34] Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron C. Courville. Ordered neurons: Integrating tree structures into recurrent neural networks. *ArXiv*, abs/1810.09536, 2019. 1, 2, 6

[35] Haoyue Shi, Jiayuan Mao, Kevin Gimpel, and Karen Livescu. Visually grounded neural syntax acquisition. *ArXiv*, abs/1906.02890, 2019. 1, 2, 3, 5

[36] Zhangzhang Si and S. Zhu. Learning and-or templates for object recognition and detection. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35:2189–2205, 2013. 3

[37] HA Simon and Howard H Pattee. Hierarchy theory: The challenge of complex systems. *Hierarchy Theory: The Challenge of Complex Systems*, 1973. 1

[38] A. Simone and T. Jacques. Guiding new physics searches with unsupervised learning. *The European Physical Journal C*, 79:1–15, 2018. 1, 2

[39] Vladimir Solmon. The estimation of stochastic context-free grammars using the inside-outside algorithm. 2003. 1, 2

[40] Valentin I. Spitkovsky, H. Alshawi, Dan Jurafsky, and Christopher D. Manning. Viterbi training improves unsupervised dependency parsing. In *CoNLL*, 2010. 1, 2

[41] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *ICLR*, 2020. 3

[42] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7464–7473, 2019. 3

[43] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *EMNLP*, 2019. 3

[44] K. Tu, Maria Pavlovskaia, and S. Zhu. Unsupervised structure learning of stochastic and-or grammars. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2013. 2, 3

[45] Zhuowen Tu, Xiangrong Chen, A. Yuille, and S. Zhu. Image parsing: unifying segmentation, detection, and recognition. *International Conference on Computer Vision (ICCV)*, pages 18–25 vol.1, 2003. 2

[46] Peter D. Turney, Y. Neuman, Dan Assaf, and Yohai Cohen. Literal and metaphorical sense identification through concrete and abstract context. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2011. 2

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NIPS*, 2017. 3

[48] Tianfu Wu and S. Zhu. A numerical study of the bottom-up and top-down inference processes in and-or graphs. *International Journal of Computer Vision (IJCV)*, 93:226–252, 2010. 1, 2

[49] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning (ICML)*. PMLR, 2015. 3

[50] L. Yi, L. Guibas, Aaron Hertzmann, Vladimir G. Kim, H. Su, and Ersin Yumer. Learning hierarchical shape segmentation and labeling from online repositories. *ACM Transactions on Graphics (TOG)*, 36:1 – 12, 2017. 2

[51] K. C. You and K. Fu. Syntactic shape recognition using attributed grammars. 1978. 1, 2

[52] Fenggen Yu, Kun Liu, Yan Zhang, Chenyang Zhu, and K. Xu. Partnet: A recursive part decomposition network for fine-grained and hierarchical shape segmentation. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[53] Yanpeng Zhao and Ivan Titov. Visually grounded compound pcfgs. *ArXiv*, abs/2009.12404, 2020. 1, 2, 3, 5, 6

[54] Song-Chun Zhu and David Mumford. A stochastic grammar of images. *Found. Trends. Comput. Graph. Vis.*, page 259–362, 2006. 1, 2