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

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. 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

Inducing the underlying structures and grammars from raw sensory inputs, *e.g.*, vision and language [5, 23, 20, 11, 19, 3, 21, 15, 7], 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 [16, 17, 10, 9] 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, [18, 24] 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* [12, 13], 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 [18, 24]. 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 [8]. Thus, it is necessary to align the language grammar with the hierarchical structures in physical objects.

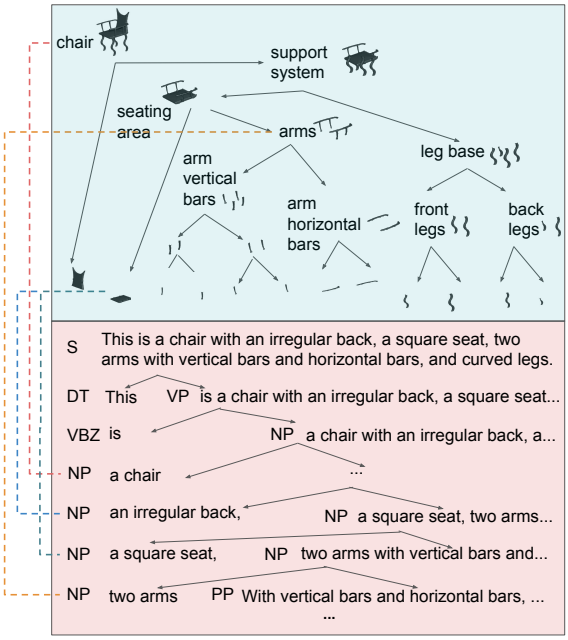

Figure 1: **An example of a sentence parse tree aligned with an image parse tree.** The arrow lines represent production rules of the image grammar and the language grammar. The dashed lines represent alignment between the constituents of two modalities.

While the study of the hierarchical structure of images has a long history [5, 23, 15, 7, 6, 22, 25], 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 [7], 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. 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. We collect a large-scale dataset, PARTIT, which contains 10,613 manually annotated descriptive sentences paired with the images of objects and parts. 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.

2. 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. 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. We obtain ∼10,000 3D CAD models and their part annotations from the PartNet dataset [14]. We choose four categories of objects: `chair`, `table`, `bed`, and `bag`. Based on the and-or templates provided by PartNet, we generate ground-truth grammar rules of each object category for evaluation only.

3. 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 inducing the language grammar [9, 24] and generalizes this idea to vision, which are jointly optimized by a contrastive loss.

3.1. Compound PCFG for Language

A context-free grammar (CFG) can be defined as a 5-tuple $\mathcal{G} = (S, \mathcal{N}, \mathcal{P}, \Sigma, \mathcal{R})$ as in [9]. 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.

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

A probabilistic context-free grammar (PCFG) extends a grammar $\mathcal{G}$ with rule probabilities $\boldsymbol{\pi} = \{\pi_r\}_{r \in \mathcal{R}}$, such that the rule r has probability π_r. Kim et al. extend neural PCFGs to compound PCFGs and mitigate the context-free assumptions by holding it conditioned on compound probability distribution [2]:

$$\mathbf{z} \sim p_\lambda(\mathbf{z}), \pi_{\mathbf{z}} = f_r(\mathbf{z}; E_{\mathcal{G}}) \tag{2}$$

where $E_{\mathcal{G}} = \mathbf{w}_N$ ($N \in \{S\} \cup \mathcal{N} \cup \mathcal{P}$) are the symbol embeddings. $p_\lambda(\mathbf{z})$ is a prior distribution of the latent variable $\mathbf{z}$ with spherical Gaussian λ, and the per-sentence rule probability $\pi_{\mathbf{z}}$ is parameterized by $E_{\mathcal{G}}$ with a neural network f_r. $\pi_{\mathbf{z}}$ takes one of the following forms [24, 9]:

$$\pi_{S \to A} = \frac{\exp\left(\mathbf{u}_A^T f_1\left([\mathbf{w}_S; \mathbf{z}]\right)\right)}{\sum_{A' \in \mathcal{N}} \exp\left(\mathbf{u}_{A'}^T f_1\left([\mathbf{w}_S; \mathbf{z}]\right)\right)} \tag{3}$$

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

$$\pi_{T \to w} = \frac{\exp\left(\mathbf{u}_w^T f_2\left([\mathbf{w}_T; \mathbf{z}]\right)\right)}{\sum_{w' \in \Sigma} \exp\left(\mathbf{u}_{w'}^T f_2\left([\mathbf{w}_T; \mathbf{z}]\right)\right)} \tag{5}$$

where $\mathbf{u}$ is a parameter vector, $\mathcal{M}$ denotes $(\mathcal{N} \cup \mathcal{P}) \times (\mathcal{N} \cup \mathcal{P})$. $[\cdot;\cdot]$ indicates vector concatenation, and $f_1(\cdot)$ and $f_2(\cdot)$ are feedforward neural networks that encode the inputs.

The log marginal likehood $\log p_\theta(\boldsymbol{w})$ of the observed sentence $\boldsymbol{w} = w_1 w_2 \dots w_n$ can be obtained by summing out the latent tree structure using the inside algorithm [1]:

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

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

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

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

3.2. 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). Instead of inducing the top-down grammar, 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 $\boldsymbol{v} = v_1 v_2 ... v_n$. 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:

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

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,

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

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 $p_{\theta_\psi}(\boldsymbol{T}|\boldsymbol{v})$. Therefore, the grammar module can boost the training of the clustering module, and *vice versa*. This is demonstrated in Section 4.2.

3.3. Joint Learning by Alignment

We propose to jointly learn the grammars for image and language by aligning the paired image and sentence. Given a sentence $\boldsymbol{w} = w_1 \dots w_m$ where m is the total number of words, a language constituent is defined as a span over this sentence, denoted as $\mathbf{w}_j = w_a \dots w_b \in [\boldsymbol{w}]$ where $0 < a < b \leqslant m$ and $[\boldsymbol{w}]$ denotes the set of all possible spans over $\boldsymbol{w}$. Given an object $\boldsymbol{v} = v_1 \dots v_n$ where n is the total number of parts, a visual constituent is defined as a span over this part sequence, denoted as $\mathbf{v}_k = v_c \dots v_d \in [\boldsymbol{v}]$ where $0 < c < d \leqslant n$ and $[\boldsymbol{v}]$ denotes the set of all possible sub-parts over $\boldsymbol{v}$. The embeddings of language and visual contituents are obtained via bi-LSTM and ResNet, respectively.

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

$$s(\boldsymbol{w}_j, \boldsymbol{v}_k) \triangleq cos(\boldsymbol{w}_j, \boldsymbol{v}_k) \tag{12}$$

The alignment score between a sentence and an image is:

$$\begin{aligned} \mathcal{S}(\boldsymbol{w}, \boldsymbol{v}) &= \sum_{\substack{t_w \in \mathcal{T}_{\mathcal{G}_w}(\boldsymbol{w}) \\ t_v \in \mathcal{T}_{\mathcal{G}_v}(\boldsymbol{v})}} p(t_w|\boldsymbol{w}) p(t_v|\boldsymbol{v}) \sum_{\substack{\boldsymbol{w}_j \in t_w \\ \boldsymbol{v}_k \in t_v}} s(\boldsymbol{w}_j, \boldsymbol{v}_k) \\ &= \sum_{\substack{\boldsymbol{w}_j \in [\boldsymbol{w}] \\ \boldsymbol{v}_k \in [\boldsymbol{v}]}} \sum_{\substack{t_w \in \mathcal{T}_{\mathcal{G}_w}(\boldsymbol{w}) \\ t_v \in \mathcal{T}_{\mathcal{G}_v}(\boldsymbol{v})}} \mathbb{1}_{\{\boldsymbol{w}_j \in t_w\}} \mathbb{1}_{\{\boldsymbol{v}_k \in t_v\}} p(t_w|\boldsymbol{w}) p(t_v|\boldsymbol{v}) s(\boldsymbol{w}_j, \boldsymbol{v}_k) \\ &= \sum_{\substack{\boldsymbol{w}_j \in [\boldsymbol{w}] \\ \boldsymbol{v}_k \in [\boldsymbol{v}]}} p(\boldsymbol{w}_j|\boldsymbol{w}; \mathcal{G}_w) p(\boldsymbol{v}_k|\boldsymbol{v}; \mathcal{G}_v) s(\boldsymbol{w}_j, \boldsymbol{v}_k) \end{aligned} \tag{13}$$

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 [4].

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

$$\begin{aligned} \mathcal{L}_C(\mathcal{W}, \mathcal{V}) = &\sum_{i, m \neq i} \left[\mathcal{S}(\boldsymbol{w}^{(m)}, \boldsymbol{v}^{(i)}) - \mathcal{S}(\boldsymbol{w}^{(i)}, \boldsymbol{v}^{(i)}) + \delta\right]_+ \\ &+ \sum_{i, m \neq i} \left[\mathcal{S}(\boldsymbol{w}^{(i)}, \boldsymbol{v}^{(m)}) - \mathcal{S}(\boldsymbol{w}^{(i)}, \boldsymbol{v}^{(i)}) + \delta\right]_+ \end{aligned} \tag{14}$$

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

The overall training loss function is then:

$$\mathcal{L} = \lambda_w \mathcal{L}_\mathcal{G}(\mathcal{W}; \phi_w, \theta_w) + \lambda_v \mathcal{L}_\mathcal{G}(\mathcal{V}; \phi_v, \theta_v) + \lambda_C \mathcal{L}_C(\mathcal{W}, \mathcal{V}) \tag{15}$$

4. Experiments and Results

4.1. Grammar Induction

Table 1 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.

Table 1: **The performance of grammar induction.** "C" and "I" denote corpus-level and instance-level F1 scores, respectively. L-PCFG-P denotes a pretrained language PCFG of the sentences of all categories. L-PCFG and V-PCFG are language and visual PCFGs. L-PCFG-VG and V-PCFG-LG are visually-grounded and language-grounded PCFGS. SCAN is the unsupervised clustering module that we use to pretrain ResNet. "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

Table 2: **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**

4.2. Part Clustering

Table 2 shows the accuracy of the unsupervised part clustering in the bottom-up module of the image compound PCFG. 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.

4.3. 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. 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.

Table 3: **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**

4.4. Cross-category Generalization

To evaluate the model's generalization ability, we train a shared image compound PCFG for certain object categories, and then test on unseen categories. The results shown in Table 4 indicate that the learned grammars can indeed be transferred to novel object categories.

Table 4: 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**

4.5. Qualitative Study

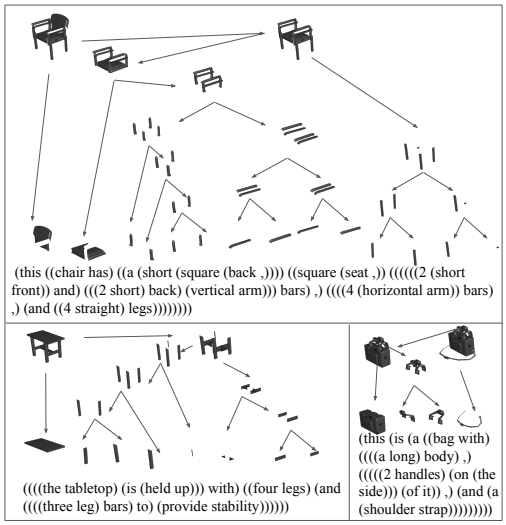

Figure 2: **Qualitative examples of parse trees predicted by VLGrammar.** We visualize the image parse trees and the language parse trees derived by the VLGrammar. Since the language parse trees are large, we use a bracket form to represent them.

Fig. 2 visualizes several examples of parse trees predicted by VLGrammar. From the examples, we can see that our model can capture capture precise part-whole hierarchies of the images. Moreover, it deals with repetitive parts with recursive structures. It also excels at grouping phrases that refer to parts in the images.

References

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

[2] L. L. Cam. Asymptotic methods in statistical decision theory. 1986. 2

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

[4] Jason Eisner. Inside-outside and forward-backward algorithms are just backprop (tutorial paper). In *SPNLP@EMNLP*, 2016. 3

[5] K. Fu. Syntactic pattern recognition and applications. 1968. 1

[6] 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. 1

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

[8] M. Johnson. The body in the mind: the bodily basis of meaning. 1987. 1

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

[10] 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

[11] 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

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

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

[14] 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. 2

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

[16] 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

[17] 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

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

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

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

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

[22] 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. 1

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

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

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