

**UCLA**

**UCLA Electronic Theses and Dissertations**

**Title**

Task-oriented Visual Understanding for Scenes and Events

**Permalink**

<https://escholarship.org/uc/item/0qb0005c>

**Author**

Qi, Siyuan

**Publication Date**

2019

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Task-oriented Visual Understanding for Scenes and Events

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Computer Science

by

Siyuan Qi

2019

© Copyright by  
Siyuan Qi  
2019

## ABSTRACT OF THE DISSERTATION

Task-oriented Visual Understanding for Scenes and Events

by

Siyuan Qi

Doctor of Philosophy in Computer Science

University of California, Los Angeles, 2019

Professor Song-Chun Zhu, Chair

Scene understanding and event understanding of humans correspond to the spatial and temporal aspects of computer vision. Such abilities serve as a foundation for humans to learn and perform tasks in the world we live in, thus motivating a task-oriented representation for machines to interpret observations of this world.

Toward the goal of *task-oriented scene understanding*, I begin this thesis by presenting a human-centric scene synthesis algorithm. Realistic synthesis of indoor scenes is more complicated than neatly aligning objects; the scene needs to be functionally plausible, which requires the machine to understand the tasks that could be performed in the scene.

Instead of directly modeling the object-object relationships, the algorithm learns the human-object relations and generate scene configurations by imagining the hidden human factors in the scene. I analyze the realism of the synthesized scenes, as well as its usefulness for various computer vision tasks. This framework is useful for backward inference of 3D scenes structures from images in an analysis-by-synthesis fashion; it is also useful for generating data to train various algorithms.

Moving forward, I introduce a *task-oriented event understanding* framework for event

parsing, event prediction, and task planning. In the computer vision literature, event understanding usually refers to action recognition from videos, *i.e.*, “what is the action of the person”. Task-oriented event understanding goes beyond this definition to find out the underlying driving forces of other agents. It answers questions such as intention recognition (“what is the person trying to achieve”), and intention prediction (“how the person is going to achieve the goal”), from a planning perspective.

The core of this framework lies in the temporal representation for tasks that is appropriate for humans, robots, and the transfer between these two. In particular, inspired by natural language modeling, I represent the tasks by stochastic context-free grammars, which are natural choices to capture the semantics of tasks, but traditional grammar parsers (*e.g.*, Earley parser) only take symbolic sentences as inputs. To overcome this drawback, I generalize the Earley parser to parse sequence data which is neither segmented nor labeled. This generalized Earley parser integrates a grammar parser with a classifier to find the optimal segmentation and labels. It can be used for event parsing, future predictions, as well as incorporating top-down task planning with bottom-up sensor inputs.

The dissertation of Siyuan Qi is approved.

Ying Nian Wu

Demetri Terzopoulos

Kai-Wei Chang

Song-Chun Zhu, Committee Chair

University of California, Los Angeles

2019

*To my parents and Tian.*

## TABLE OF CONTENTS

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Human-centric Indoor Scene Synthesis Using Stochastic Grammar</b>	<b>4</b>
2.1	Introduction	6
2.2	Related Work	11
2.3	Representation and Formulation	15
2.3.1	Representation: Attributed Spatial And-Or Graph	15
2.4	Probabilistic Formulation of S-AOG	19
2.5	Learning, Sampling and Synthesis	24
2.5.1	Learning the S-AOG	25
2.5.2	Sampling Scene Geometry Configurations	28
2.5.3	Scene Instantiation using 3D Object Datasets	33
2.5.4	Scene Attribute Configurations	34
2.6	Photorealistic Scene Rendering	35
2.7	Experiments	40
2.7.1	Realistic of Sampled Scene Configurations	40
2.7.2	Synthesized Indoor Scene Data for Scene Understanding	47
2.8	Discussion	56
2.9	Conclusion and Future Work	60
2.10	More Results	62



<b>3</b>	<b>Human Activity Prediction Using Stochastic Grammar</b>	<b>71</b>
3.1	Related Work	76
3.2	Representation: Probabilistic Context-Free Grammars	79
3.3	Earley Parser	80
3.4	Generalized Earley Parser	85
3.4.1	Parsing Operations	88
3.4.2	Parsing & Prefix Probability Formulation	90
3.4.3	Incorporating Grammar Prior	91
3.4.4	Segmentation and Labeling	96
3.4.5	Future Label Prediction	96
3.5	Human Activity Parsing and Prediction	99
3.5.1	Grammar Induction	100
3.5.2	Experiment on CAD-120 Dataset	101
3.5.3	Experiment on Watch-n-Patch Dataset	103
3.5.4	Experiment on Breakfast Dataset	106
3.5.5	Discussion	108
3.6	Conclusion	111
<b>4</b>	<b>Conclusion</b>	<b>112</b>
	<b>References</b>	<b>114</b>

## LIST OF FIGURES

2.1	An example of synthesized indoor scene (bedroom) with affordance heatmap. The joint sampling of a scene is achieved by alternative sampling of humans and objects according to the joint probability distribution. . . . .	5
2.2	(Top Left) An example automatically-generated 3D bedroom scene, rendered as a photorealistic RGB image, along with per-pixel ground truth of surface normal, depth, and object identity. (Top Right) Another synthesized bedroom scene. Synthesized scenes include fine details—objects ( <i>e.g.</i> , the duvet and pillows on beds) and their textures are changeable, by sampling physical parameters of materials (reflectance, roughness, glossiness, <i>etc.</i> ), and illumination parameters are sampled from continuous spaces of possible positions, intensities, and colors. (Bottom) Rendered images of 4 example synthetic indoor scenes. . . . .	9

2.3	Scene grammar as an attributed S-AOG. A scene of different types is decomposed into a room, furniture, and supported objects. Attributes of terminal nodes are internal attributes (sizes), external attributes (positions and orientations), and a human position that interacts with this entity. Furniture and object nodes are combined by an address terminal node and a regular terminal node. A furniture node ( <i>e.g.</i> , a chair) is grouped with another furniture node ( <i>e.g.</i> , a desk) pointed by its address terminal node. An object ( <i>e.g.</i> , a monitor) is supported by the furniture ( <i>e.g.</i> , a desk) it is pointing to. If the value of the address node is null, the furniture is not grouped with any furniture, or the object is put on the floor. Contextual relations are defined between the room and furniture, between a supported object and supporting furniture, among different pieces of furniture, and among functional groups. . . . .	14
2.4	(a) A simplified example of a parse graph of a bedroom. The terminal nodes of the parse graph form an MRF in the terminal layer. Cliques are formed by the contextual relations projected to the terminal layer. Examples of the four types of cliques are shown in (b)-(e), representing four different types of contextual relations. . . . .	20
2.5	The learning-based pipeline for synthesizing images of indoor scenes. . . . .	21
2.6	Given a scene configuration, we use bi-directional RRT to plan from every piece of furniture to another, generating a human activity probability map. . .	23

2.7	Examples of the learned affordance maps. Given the object positioned in the center facing upwards, <i>i.e.</i> , coordinate of $(0, 0)$ facing direction $(0, 1)$ , the maps show the distributions of human positions. The affordance maps accurately capture the subtle differences among desks, coffee tables, and dining tables. Some objects are orientation sensitive, <i>e.g.</i> , books, laptops, and night stands, while some are orientation invariant, <i>e.g.</i> , fruit bowls and vases. . . .	29
2.8	MCMC sampling process (from left to right) of scene configurations with simulated annealing. . . . .	31
2.9	Synthesis for different values of $\beta$ . Each image shows a typical configuration sampled from a Markov chain. . . . .	31
2.10	Qualitative results in different types of scenes. . . . .	35
2.11	We can configure the scene with different (a) illumination intensities, (b) illumination colors, (c) materials, and (d) even on each object part. We can also control (e) the number of light source and their positions, (f) camera lenses ( <i>e.g.</i> , fish eye), (g) depths of field, or (h) render the scene as a panorama for virtual reality and other virtual environments. (i) 7 different background wall textures. Note how the background affects the overall illumination. . . . .	36
2.12	(Continue:) we can configure the scene with different (a) illumination intensities, (b) illumination colors, (c) materials, and (d) even on each object part. We can also control (e) the number of light source and their positions, (f) camera lenses ( <i>e.g.</i> , fish eye), (g) depths of field, or (h) render the scene as a panorama for virtual reality and other virtual environments. (i) 7 different background wall textures. Note how the background affects the overall illumination. . . .	37

2.13	Examples of scenes in ten different categories. Top: top-view. Middle: a side-view. Bottom: affordance heatmap. . . . .	42
2.14	(Continue:) examples of scenes in ten different categories. Top: top-view. Middle: a side-view. Bottom: affordance heatmap. . . . .	43
2.15	Top-view segmentation maps for classification. . . . .	44
2.16	<b>Top:</b> previous methods [YYT11] only re-arranges a given input scene with a fixed room size and a predefined set of objects. <b>Bottom:</b> our method samples a large variety of scenes. . . . .	45
2.17	Examples of normal estimation results predicted by the model trained with our synthetic data. . . . .	48
2.18	Examples of depth estimation results predicted by the model trained with our synthetic data. . . . .	51
2.19	We can render the scenes as (a) a sequence of video frames after setting a camera trajectory, (b) which can be used to evaluate SLAM reconstruction [WLS15] results. The top row shows a successful reconstruction case, while the middle and bottom rows show two failure cases due to a fast moving camera and a plain, untextured surface, respectively. . . . .	55
2.20	Benchmark results. (a) Given a set of generated RGB images rendered with different illuminations and object material properties (top to bottom: original settings, with high illumination, with blue illumination, and with metallic material properties), we evaluate (b)–(d) three depth prediction algorithms, (e)–(f) two surface normal estimation algorithms, (g) a semantic segmentation algorithm, and (h) an object detection algorithm. . . . .	57

3.1	What is he going to do? (a)(b) Input RGB-D video frames. (c) Activity prediction: human action with interacting objects (how the agent will perform the task). The red skeleton is the current observation. The magenta, green and blue skeletons and interacting objects are possible future states. . . . .	73
3.2	The input of the generalized Earley parser is a matrix of probabilities of each label for each frame, given by an arbitrary classifier. The parser segments and labels the sequence data into a label sentence in the language of a given grammar. Future predictions can be made based on the grammar. . . . .	77
3.3	An example of a temporal grammar representing the activity “making cereal”. The green and yellow nodes are And-nodes ( <i>i.e.</i> , production rules that represents combinations) and Or-nodes ( <i>i.e.</i> , productions rules that represents alternatives), respectively. The numbers on branching edges of Or-nodes represent the branching probability. The circled numbers on edges of And-nodes indicates the temporal order of expansion. . . . .	81
3.4	A simplified example illustrating the symbolic parsing and prediction process based on the Earley parser and detected actions. In the first two figures, the red edges and blue edges indicates two different parse graphs for the past observations. The purple edges indicate the overlap of the two possible explanations. The red parse graph is eliminated from the third figure. For the terminal nodes, yellow indicates the current observation and green indicates the next possible state(s). . . . .	82
3.5	An illustrative example of the original Earley parser. . . . .	84

3.6	Grammar prefix probabilities computed according to the grammar in Figure 3.5. The numbers next to the tree nodes are prefix probabilities according to the grammar. The transition probabilities can be easily computed from this tree, <i>e.g.</i> , $p("1" "0+", G) = p("0+1" \dots  G) / p("0+" \dots  G) = 0.126 / 0.18 = 0.7$ .	92
3.7	An example of the generalized Earley parser. A classifier is applied to a 5-frame signal and outputs a probability matrix (a) as the input to our algorithm. The proposed algorithm expands a grammar prefix tree (c), where "e" represents termination. It finally outputs the best label "0 + 1" with probability 0.054. The probabilities of children nodes do not sum to 1 since the grammatically incorrect nodes are eliminated. . . . .	93
3.8	Confusion matrices for prediction results on CAD-120. . . . .	103
3.9	Qualitative results of segmentation results. In each group of four segmentations, the rows from the top to the bottom show the results of: 1) ground-truth, 2) ST-AOG + Earley, 3) Bi-LSTM, and 4) Bi-LSTM + generalized Earley parser. The results show (a) corrections and (b) insertions by our algorithm on the initial segment-wise labels given by the classifier (Bi-LSTM). . . . .	109

## LIST OF TABLES

2.1	Comparisons of rendering time vs quality. The first column tabulates the reference number and rendering results used in this work, the second column lists all the criteria, and the remaining columns present comparative results. The color differences between the reference image and images rendered with various parameters are measured by LAB Delta E standard [SB02] tracing back to Helmholtz and Hering [BKW98, Val07]. . . . .	39
2.2	Classification results on segmentation maps of synthesized scenes using different methods vs. SUNCG. . . . .	40
2.3	Comparison between affordance maps computed from our samples and real data	41
2.4	Human subjects' ratings (1-5) of the sampled layouts based on functionality (top) and naturalness (bottom) . . . . .	41
2.5	Performance of normal estimation for the NYU-Depth V2 dataset with different training protocols. . . . .	47
2.6	Depth estimation performance on the NYU-Depth V2 dataset with different training protocols. . . . .	50
2.7	Depth estimation. Intensity, color, and material represent the scene with different illumination intensities, colors, and object material properties, respectively.	52
2.8	Surface Normal Estimation. Intensity, color, and material represent the setting with different illumination intensities, illumination colors, and object material properties. . . . .	54
3.1	Summary of notations used for parsing & prefix probability formulation. . .	90



3.2 Detection results on CAD-120. . . . . 104

3.3 Future 3s prediction results on CAD-120. . . . . 104

3.4 Segment prediction results on CAD-120. . . . . 105

3.5 Detection results on Watch-n-Patch. . . . . 106

3.6 Future 3s prediction results on Watch-n-Patch. . . . . 106

3.7 Segment prediction results on Watch-n-Patch. . . . . 106

3.8 Detection results on Breakfast. . . . . 107

## ACKNOWLEDGMENTS

Foremost, I would like to express my sincere gratitude to my advisor Dr. Song-Chun Zhu for the continuous support of my research, for his motivation, enthusiasm, and vision.

I also owe deep gratitude to Dr. Ying Nian Wu, for his warm encouragements, his down-to-earth but inspiring and insightful thinking toward great research.

I would also like to thank Dr. Demetri Terzopoulos and Dr. Kai-Wei Chang for their supportive advice and valuable discussions for both on-going research and future career.

I am fortunate to be grateful for many people during this wonderful journey.

Dr. Yibiao Zhao, for his patient guidance at the very beginning of my research path. Reading only shows me what research is, he showed me how.

Dr. Yixin Zhu, for his generous help from all aspects. He is always the go-to person whenever there are difficulties, whether they are academic or technical.

Dr. Ping Wei, for his lead in the early stage — the first project of my own.

Siyuan Huang, for his great resolution to push the boundary of scene understanding.

Baoxiong Jia, for his precious curiosity, constructive suggestions, and quick execution of ideas.

Feng Gao, Mark Edmonds, Xu Xie, and Hangxin Liu, my previous office mates and reliable friends. Many thanks for them taking the lead to work out big projects and support the entire lab.

Qingyi Zhao and Fan Hin Fung, for their determination to challenge the very hard topic of theory-of-mind planning.

Tianmin Shu, Yuanlu Xu, and Yang Liu, my Ph.D. peers that share memorable days of

the lab and graduate together with me.

Finally, those to whom this thesis is dedicated — my parents and Tian Ye. I would like to express my deepest gratitude to their love and never-ending support for my life.

## VITA

- 2009–2010 Tsinghua University, Department of Precision Instrument
- 2010–2013 University of Hong Kong, B.Eng. in Computer Engineering, Faculty of Engineering
- 2013–2015 University of California, Los Angeles, M.S. in Computer Science, School of Engineering and Applied Science
- 2015–2019 University of California, Los Angeles, Ph.D. in Computer Science, School of Engineering and Applied Science

## PUBLICATIONS

*Configurable 3D Scene Synthesis and 2D Image Rendering with Per-Pixel Ground Truth Using Stochastic Grammars.*

Chenfanfu Jiang, **Siyuan Qi**, Yixin Zhu, Siyuan Huang, Jenny Lin, Lap-Fai Yu, Demetri Terzopoulos, and Song-Chun Zhu

*International Journal of Computer Vision (IJCV)*, 2018

*Cooperative Holistic Scene Understanding: Unifying 3D Object, Layout, and Camera Pose Estimation.*

Siyuan Huang, **Siyuan Qi**, Yinxue Xiao, Yixin Zhu, Ying Nian Wu, and Song-Chun Zhu

*Advances in Neural Information Processing Systems (NeurIPS)*, 2018

*Learning Human-Object Interactions by Graph Parsing Neural Networks.*

**Siyuan Qi**, Wenguan Wang, Baoxiong Jia, Jianbing Shen, and Song-Chun Zhu

*European Conference on Computer Vision (ECCV), 2018*

*Holistic 3D Scene Parsing and Reconstruction from a Single RGB Image.*

Siyuan Huang, **Siyuan Qi**, Yixin Zhu, Yinxue Xiao, Yuanlu Xu, and Song-Chun Zhu

*European Conference on Computer Vision (ECCV), 2018*

*Generalized Earley Parser: Bridging Symbolic Grammars and Sequence Data for Future Prediction.*

**Siyuan Qi**, Baoxiong Jia, and Song-Chun Zhu

*International Conference on Machine Learning (ICML), 2018*

*Human-centric Indoor Scene Synthesis Using Stochastic Grammar.*

**Siyuan Qi**, Yixin Zhu, Siyuan Huang, Chenfanfu Jiang, and Song-Chun Zhu

*Conference on Computer Vision and Pattern Recognition (CVPR), 2018*

*Intent-aware Multi-agent Reinforcement Learning.*

**Siyuan Qi** and Song-Chun Zhu

*International Conference on Robotics and Automation (ICRA), 2018*

*Predicting Human Activities Using Stochastic Grammar.*

**Siyuan Qi**, Siyuan Huang, Ping Wei, and Song-Chun Zhu

*International Conference on Computer Vision (ICCV), 2017*

# CHAPTER 1

## Introduction

“Perhaps the composition and layout of surfaces constitute what they afford. If so, to perceive them is to perceive what they afford. ”

— James J. Gibson, 1979 [Gib79]

The success of human species could be partly attributed to our remarkable capability to perceive and survive the physical world. Some computer scientists and psychologists believe that perception is more about perceiving the affordance of the environment than recognizing the geometric structure of it. Affordances of the environment, first proposed by Gibson [Gib79], means what it offers the animals. For example, if a surface is horizontal, flat, extended, and rigid, then it might provide the affordance of support to an animal.

Such affordances are highly related to the *task* at hand, and the perception of affordances is a foundation for humans to learn and perform tasks in the world we live in. To mimic human intelligence, we need machines that can understand human tasks and their relations with the environment. In two of the most important aspects of computer vision, scene understanding, and event understanding, this inspires a task-oriented representation for machines to interpret observations of this world.

Toward the goal of *task-oriented scene understanding*, I begin this thesis by presenting

a human-centric scene synthesis algorithm in Chapter 2. Realistic synthesis of indoor scenes is more complicated than neatly aligning objects; the scene needs to demonstrate feasible affordances to humans, *i.e.*, be functionally plausible. It addresses the question of “how the scene is related to humans”. This requires the machine to understand the human tasks that could be performed in the scene.

Instead of directly modeling the object-object relationships, the algorithm learns the human-object relations and generate scene configurations by imagining the hidden human factors in the scene. The scenes are modeled by spatial And-Or graphs (S-AOGs). An S-AOG is a probabilistic grammar model, in which the terminal nodes are object entities including room, furniture, and supported objects. Human contexts as contextual relations are encoded by Markov Random Fields (MRF) on the terminal nodes. Synthesis of indoor scenes is achieved by sampling from this distribution via Markov chain Monte Carlo.

I analyze the realism of the synthesized scenes, as well as its usefulness for various computer vision tasks. This framework is useful for backward inference of 3D scenes structures from images in an analysis-by-synthesis fashion; it is also useful for generating data to train various algorithms.

Moving forward, in Chapter 3 I introduce a *task-oriented event understanding* framework for event parsing, event prediction, and task planning. In the computer vision literature, event understanding usually refers to action recognition from videos, *i.e.* “what is the action of the person”. Task-oriented event understanding goes beyond this definition to find out the underlying driving forces of other agents. It answers questions such as intention recognition (“what is the person trying to achieve”), and intention prediction (“how the person is going to achieve the goal”), from a planning perspective.

The core of this framework lies in the temporal representation for tasks that is ap-

appropriate for humans, robots, and the transfer between these two. In particular, inspired by natural language modeling, I represent the tasks by stochastic context-free grammars, which are natural choices to capture the semantics of tasks, but traditional grammar parsers (e.g., Earley parser) only take symbolic sentences as inputs. To overcome this drawback, I generalize the Earley parser to parse sequence data which is neither segmented nor labeled. This generalized Earley parser integrates a grammar parser with a classifier to find the optimal segmentation and labels. It can be used for event parsing, future predictions, as well as incorporating top-down task planning with bottom-up sensor inputs.

This thesis aims to make progress in the direction of task-oriented perception, for laying down foundations for developing human-like robots in the sense of task learning and planning. Although still far from being comprehensive, it develops frameworks for both the spatial and temporal aspects of visual understanding. Finally, I conclude the thesis in Chapter 4 with a summary of these two frameworks and insights for future research under the theme of task-oriented representations.



## CHAPTER 2

# Human-centric Indoor Scene Synthesis Using Stochastic Grammar

In this chapter, we present a human-centric method to sample and synthesize 3D room layouts and 2D images thereof, to obtain large-scale 2D/3D image data with the perfect per-pixel ground truth, for the purposes of training, benchmarking, and diagnosing learning-based computer vision and robotics algorithms.

We propose an attributed spatial And-Or graph (S-AOG) to represent indoor scenes. The S-AOG is a probabilistic grammar model, in which the terminal nodes are object entities including room, furniture, and supported objects. Human contexts as contextual relations are encoded by Markov Random Fields (MRF) on the terminal nodes. We learn the distributions from an indoor scene dataset and sample new layouts using Monte Carlo Markov Chain. Our pipeline is capable of synthesizing scene layouts with high diversity, and it is *configurable* in that it enables the precise customization and control of important attributes of the generated scenes. It renders photorealistic RGB images of the generated scenes while automatically synthesizing detailed, per-pixel ground truth data, including visible surface depth and normal, object identity, and material information (detailed to object parts), as well as environments (*e.g.*, illumination and camera viewpoints).

Experiments demonstrate that the proposed method can robustly sample a large va-

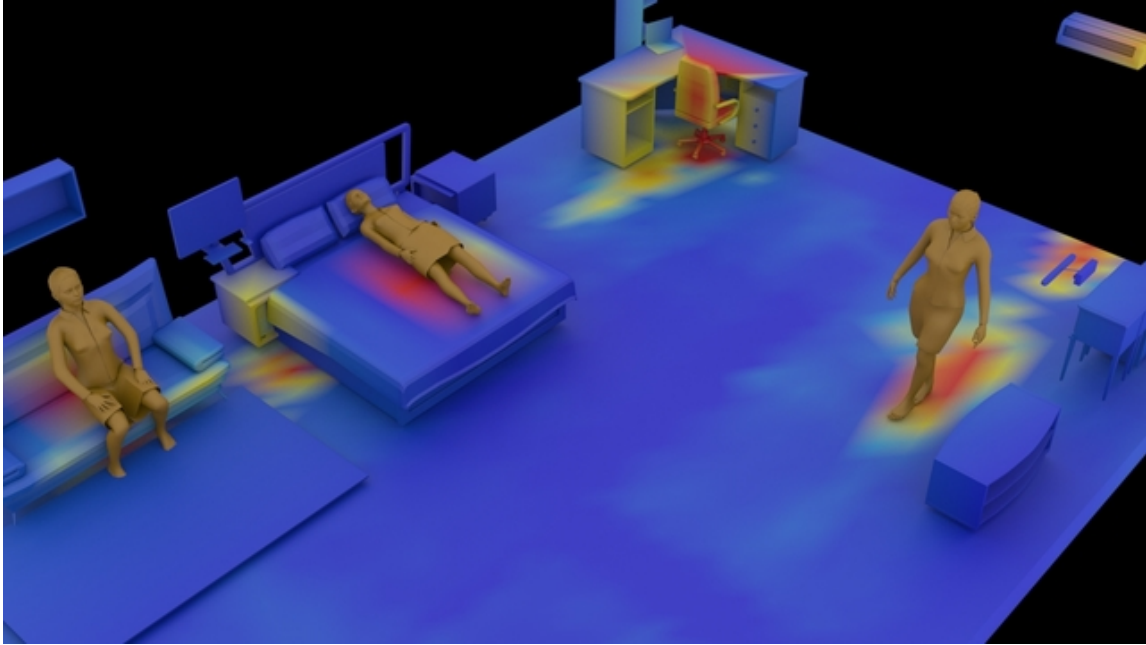


Figure 2.1: An example of synthesized indoor scene (bedroom) with affordance heatmap. The joint sampling of a scene is achieved by alternative sampling of humans and objects according to the joint probability distribution.

riety of realistic room layouts based on three criteria: (i) visual realism comparing to a state-of-the-art room arrangement method, (ii) accuracy of the affordance maps with respect to ground-truth, and (ii) the functionality and naturalness of synthesized rooms evaluated by human subjects. We also demonstrate the value of our dataset, by improving performance in certain machine-learning-based scene understanding tasks—*e.g.*, depth and surface normal prediction, semantic segmentation, reconstruction, *etc.*—and by providing benchmarks for and diagnostics of trained models by modifying object attributes and scene properties in a controllable manner.

## 2.1 Introduction

Recent advances in visual recognition and classification through machine-learning-based vision algorithms have yielded similar or even better than human performance (*e.g.*, [HZR15, EEV15]) by leveraging large-scale, ground-truth-labeled RGB datasets [DDS09, LMB14]. However, indoor scene understanding remains a largely unsolved challenge due in part to the limitations of appropriate RGB-D datasets available for training purposes. To date, the most commonly used RGB-D dataset for scene understanding is the NYU-Depth V2 dataset [SHK12], which comprises only 464 scenes with only 1449 labeled RGB-D pairs provided while the remaining 407,024 pairs are unlabeled. This is clearly insufficient for the supervised training of modern computer vision methods, especially those based on deep learning.

Furthermore, traditional methods of 2D/3D image data collection and ground-truth labeling have evident limitations. i) High-quality ground truths are hard to obtain, as depth and surface normal obtained from sensors are always noisy. ii) It is impossible to label certain ground truth information, *e.g.*, 3D objects sizes in 2D images. iii) Manual labeling of massive ground-truth is tedious and error-prone even if possible. To provide training data for modern machine learning algorithms, an approach to generate large-scale, high-quality data with the perfect per-pixel ground truth is in need.

To address this deficiency, recent years have seen the increased use of synthetic image datasets as training data. In fact, recent advances in computer vision and robotics community [ZSY17, HWM14] have shown that synthetic datasets are beneficial for either improving data-driven methods or analyzing problems that are difficult to obtain accurate ground truth.

However, synthesizing indoor scenes is a non-trivial task. It is often difficult to prop-

erly model either the relations between furniture of a functional group, or the relations between the supported objects and the supporting furniture. Specifically, we argue there are four major difficulties. (i) In a functional group such as a dining set, the number of pieces may vary. How should we model the relationship between a dining table and a chair so that we can generate such functional groups? Using multi-modal probability distributions for position relations would be restricted to simple and rigid configurations, disallowing a large variety of possible layouts. (ii) Even if we only consider pair-wise relations, there is already a quadratic number of object-object relations. (iii) What makes it worse is that most object-object relations are not obviously meaningful. For example, it is unnecessary to model the relation between a pen and a monitor, even though they are both placed on a desk. (iv) Due to the previous difficulties, an excessive number of constraints are generated. Many of the constraints contain loops, making the final layout hard to sample and optimize.

To date, little effort has been devoted to the learning-based systematic generation of massive quantities of sufficiently complex synthetic indoor scenes for the purposes of training scene understanding algorithms. This is also partially due to the difficulties other than modeling the object relations in the scenes: (i) devising sampling processes capable of generating diverse scene configurations, and (ii) the intensive computational costs of photorealistically rendering large-scale scenes. Aside from a few efforts, reviewed in Section 2.2, in generating small-scale synthetic scenes, the most notable work was recently reported by Song *et al.* [SYZ17a], in which a large scene layout dataset was downloaded from the Planner5D website.

To address these challenges, we propose a human-centric approach to model indoor scene layout, from which we can render 2D images with pixel-wise ground-truth of the surface normal, depth, and segmentation, *etc.*. It integrates human activities and func-

tional grouping/supporting relations as illustrated in Figure 2.1. This method not only captures the human context but also simplifies the scene structure. Specifically, we use a probabilistic grammar model for images and scenes [ZM07] – an attributed spatial And-Or graph (S-AOG), including vertical hierarchy and horizontal contextual relations. The contextual relations encode functional grouping relations and supporting relations modeled by object affordances [Gib79]. For each object, we learn the affordance distribution, *i.e.*, an object-human relation, so that a human can be sampled based on that object. Besides static object affordance, we also consider dynamic human activities in a scene, constraining the layout by planning trajectories from one piece of furniture to another.

The proposed algorithm is useful for tasks including but not limited to: i) learning and inference for various computer vision tasks; ii) 3D content generation for 3D modeling and games; iii) 3D reconstruction and robot mappings problems; iv) benchmarking of both low-level and high-level task-planning problems in robotics. The proposed algorithm especially benefit scene understanding tasks, including a) 3D scene completion using partially observed 3D scenes, b) various scene understanding tasks such as depth and surface normal prediction, semantic segmentation, *etc.*, and c) fundamental computer vision problems like object detection.

By comparison, our work is also unique in that we devise a complete learning-based pipeline for synthesizing large scale *learning-based configurable* scene layouts via stochastic sampling, as well as the photorealistic physics-based rendering of these scenes with associated per-pixel ground truth to serve as training data. Our pipeline has the following characteristics:

- By utilizing a stochastic grammar model, one represented by an attributed Spatial And-Or Graph (S-AOG), our sampling algorithm combines hierarchical composi-



Figure 2.2: (Top Left) An example automatically-generated 3D bedroom scene, rendered as a photorealistic RGB image, along with per-pixel ground truth of surface normal, depth, and object identity. (Top Right) Another synthesized bedroom scene. Synthesized scenes include fine details—objects (*e.g.*, the duvet and pillows on beds) and their textures are changeable, by sampling physical parameters of materials (reflectance, roughness, glossiness, *etc.*), and illumination parameters are sampled from continuous spaces of possible positions, intensities, and colors. (Bottom) Rendered images of 4 example synthetic indoor scenes.

tions and contextual constraints to enable the systematic generation of 3D scenes with high variability, not only at the scene level (*e.g.*, control of size of the room and the number of objects within), but also at the object level (*e.g.*, control of the material properties of individual object parts).

- As Figure 2.2 shows, we employ state-of-the-art physics-based rendering, yielding photorealistic synthetic images. Our advanced rendering enables the systematic

sampling of an infinite variety of environmental conditions and attributes, including illumination conditions (positions, intensities, colors, *etc.*, of the light sources), camera parameters (Kinect, fisheye, panorama, camera models and depth of field, *etc.*), and object properties (color, texture, reflectance, roughness, glossiness, *etc.*).

Since our synthetic data is generated in a forward manner—by rendering 2D images from 3D scenes of detailed geometric object models—ground truth information is naturally available without the need for any manual labeling. Hence, not only are our rendered images highly realistic, but they are also accompanied by accurate, per-pixel ground truth color, depth, surface normals, and object labels.

In our experimental study, we first demonstrate the sampled room configurations are realistic based on three criteria: (i) visual realism comparing to a state-of-the-art room arrangement method, (ii) accuracy of the affordance maps with respect to ground-truth, and (iii) the functionality and naturalness of synthesized rooms evaluated by human subjects.

Then we further demonstrate the usefulness of our dataset by improving the performance in certain scene understanding tasks, showcasing the prediction of surface normals from RGB images, as well as the depth prediction from RGB images. Furthermore, by modifying object attributes and scene properties in a controllable manner, we provide benchmarks and diagnostics of trained models for common scene understanding tasks; *e.g.*, depth and surface normal prediction, semantic segmentation, reconstruction, *etc.*

Our work makes the following contributions:

1. We introduce a *learning-based configurable* pipeline for generating massive quantities of photorealistic images of indoor scenes with perfect per-pixel ground truth, including color, surface depth, surface normal, and object identity. The parameters and constraints are automatically learned from the SUNCG [SYZ17a] and

ShapeNet [CFG15] datasets.

2. For scene generation, we propose the use of a stochastic grammar model in the form of an attributed Spatial And-Or graph (S-AOG). It jointly models objects, affordances, and activity planning for indoor scene configurations. Our model supports the arbitrary addition and deletion of objects and modification of their categories, yielding significant variation in the resulting collection of synthetic scenes.
3. By precisely customizing and controlling important attributes of the generated scenes, we provide a set of diagnostic benchmarks of previous work on several common computer vision tasks. To our knowledge, this is the first work to provide comprehensive diagnostics with respect to algorithm stability and sensitivity to certain scene attributes.
4. We demonstrate that the sampled configurations are realistic. We also demonstrate the effectiveness of the proposed synthesized scene dataset by advancing the state-of-the-art in the prediction of surface normals and depth.

## 2.2 Related Work

**3D content generation** is one of the largest communities in the game industry and we refer readers to a recent survey [HMV13] and book [STN16, QZH18]. In this work, we focus on approaches related to our work [JQZ18] using probabilistic inference. Yu [YYT11] and Handa [HPS16] optimize the layout of rooms given a set of furniture using MCMC, while Talton [TLL11] and Yeh [YYW12] consider an open world layout using RJMCMC. These 3D room re-arrangement algorithms optimize room layouts based on constraints to generate new room layouts using a given set of objects. In contrast, the proposed method



is capable of adding or deleting objects without fixing the number of objects. Some literature focused on fine-grained room arrangement for specific problems, *e.g.*, small objects arrangement using user-input examples [FRS12, YYT16], optimizing the number of objects in scenes using LARJ-MCMC [YYW12], and procedural modeling of objects to encourage volumetric similarity to a target shape [RMG15]. [FSL15] synthesizes 3D scenes given a 3D scan of a room. To achieve better realism, Merrell [MSL11] introduced an interactive system providing suggestions following interior design guidelines. Jiang [JKS16] uses a mixture of conditional random field (CRF) to model the hidden human context and arrange new small objects based on existing furniture in a room. However, it cannot directly sampling/synthesizing an indoor scene, since the CRF is intrinsically a discriminative model for structured classification instead of generation.

**Synthetic image datasets** have recently been a source of training data for object detection and correspondence matching [SGS10, SS14, SX14, FKI14, DFI15, PSA15, ZKA16, GWC16, MKS16, QSN16], single-view reconstruction [HWK15], view-point estimation [MSB14, SQL15], 2D human pose estimation [PJA12, RLB15, Qiu16], 3D human pose estimation [SSK13, SVD03, YIK16, DWL16, GKS16, RS16, ZZL16, CWL16, VRM17], depth prediction [SHM14], pedestrian detection [MVG10, PJW11, VLM14, HNK15], action recognition [RM15, RM16, SGC17], semantic segmentation [RVR16], scene understanding [HPS16, KIX16, QY16, HPB16, ZBK17, HQX18], autonomous pedestrians and crowd [OPO10, QZ18, ST05], VQA [JHM17], training autonomous vehicles [CSK15, DRC17, SDL17], human utility learning [YQK17, ZJZ16], and in benchmark data sets [HWM14]. Previously, synthetic imagery, generated on the fly, online, had been used in visual surveillance [QT08] and active vision / sensorimotor control [TR95]. Although prior work demonstrates the potential of synthetic imagery to advance computer vision research, to

our knowledge no large synthetic RGB-D dataset of *learning-based configurable* indoor scenes has yet been released.

**Image synthesis** has been attempted using various deep neural network architectures, including recurrent neural networks (RNN) [GDG15], generative adversarial networks (GAN) [WG16, RMC15], inverse graphics networks [KWK15], and generative convolutional networks [LZW16, XLZ16b, XLZ16a]. However, images of indoor scenes synthesized by these models often suffer from glaring artifacts, such as blurred patches. More recently, some applications of general purpose inverse graphics solutions using probabilistic programming languages have been reported [MKP13, LB14, KKT15]. However, the problem space is enormous, and the quality of inverse graphics “renderings” is disappointingly low and slow.

**Stochastic grammar model** has been used for parsing the hierarchical structures from images of indoor [LZZ14, ZZ13, HQZ18] and outdoor scenes [LZZ14], and images/videos involving humans [QHW17, WXS18]. In this work, instead of using stochastic grammar for parsing, we forward sample from a grammar model to generate large variations of indoor scenes.

**Domain adaptation** Although the presented work does not directly involve domain adaptation, this plays an important role in learning from synthetic data, as the goal of using synthetic data is to transfer the learned knowledge and apply it to real-world scenarios. A review of existing work in this area is beyond the scope of this work; we refer the reader to a recent comprehensive survey [Csu17]. Traditionally, the widely used techniques for domain adaptation can be divided into four categories: i) covariate shift

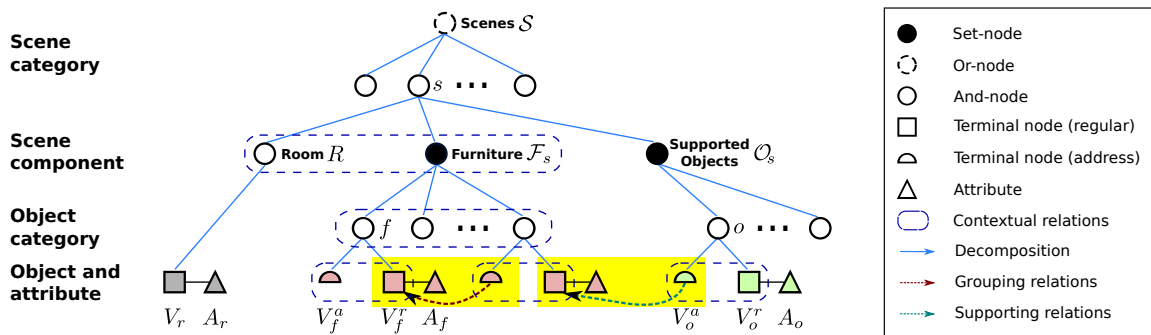


Figure 2.3: Scene grammar as an attributed S-AOG. A scene of different types is decomposed into a room, furniture, and supported objects. Attributes of terminal nodes are internal attributes (sizes), external attributes (positions and orientations), and a human position that interacts with this entity. Furniture and object nodes are combined by an address terminal node and a regular terminal node. A furniture node (*e.g.*, a chair) is grouped with another furniture node (*e.g.*, a desk) pointed by its address terminal node. An object (*e.g.*, a monitor) is supported by the furniture (*e.g.*, a desk) it is pointing to. If the value of the address node is null, the furniture is not grouped with any furniture, or the object is put on the floor. Contextual relations are defined between the room and furniture, between a supported object and supporting furniture, among different pieces of furniture, and among functional groups.

with shared support [Hec77, GSH09, CMR08, BBS09], ii) learning shared representations [BMP06, BBC07, MMR09], iii) feature-based learning [EP04, Dau07, WDL09], and iv) parameter-based learning [CH05b, YTS05, XLC07, Dau09]. With the recent boost of deep learning, researchers have started to apply deep features to domain adaptation (*e.g.*, [GL15, THD15]).

## 2.3 Representation and Formulation

### 2.3.1 Representation: Attributed Spatial And-Or Graph

A scene model should be capable of: (i) representing the compositional/hierarchical structure of indoor scenes, and (ii) capturing the rich contextual relationships between different components of the scene. Specifically,

- *Compositional hierarchy* of the indoor scene structure is embedded in a graph-based model to model the decomposition into sub-components and the switch among multiple alternative sub-configurations. In general, an indoor scene can first be categorized into different indoor settings (*i.e.*, bedrooms, bathrooms, *etc.*), each of which has a set of walls, furniture, and supported objects. Furniture can be decomposed into functional groups that are composed of multiple pieces of furniture; *e.g.*, a “work” functional group consists of a desk and a chair.
- *Contextual relations* between pieces of furniture are helpful in distinguishing the functionality of each furniture item and furniture pairs, providing a strong constraint for representing natural indoor scenes. In this work, we consider four types of contextual relations: (i) relations between furniture and walls; (ii) relations among furniture; (iii) relations between supported objects and their supporting objects (*e.g.*, monitor and desk); and (iv) relations between objects of a functional pair (*e.g.*, sofa and TV).

**Representation:** We represent the hierarchical structure of indoor scenes by an attributed Spatial And-Or Graph (S-AOG), which is a Stochastic Context-Sensitive Grammar (SCSG)

with attributes on the terminal nodes. An example is shown in Figure 2.3. This representation combines (i) a stochastic context-free grammar (SCFG) and (ii) contextual relations defined on a Markov random field (MRF); *i.e.*, the horizontal links among the terminal nodes. The S-AOG represents the hierarchical decompositions from scenes (top level) to objects (bottom level), whereas contextual relations encode the spatial and functional relations through horizontal links between nodes.

**Definitions:** Formally, an S-AOG is denoted by a 5-tuple:  $\mathcal{G} = \langle S, V, R, P, E \rangle$ , where  $S$  is the root node of the grammar,  $V = V_{NT} \cup V_T$  is the vertex set including non-terminal nodes  $V_{NT}$  and terminal nodes  $V_T$ ,  $R$  stands for the production rules,  $P$  represents the probability model defined on the attributed S-AOG, and  $E$  denotes the contextual relations represented as horizontal links between nodes in the same layer.

**Vertex Set**  $V$  can be decomposed into a finite set of non-terminal and terminal nodes:

$$V = V_{NT} \cup V_T.$$

- $V_{NT} = V^{And} \cup V^{Or} \cup V^{Set}$ . The non-terminal nodes consists of three subsets. i) A set of **And-nodes**  $V^{And}$ , in which each node represents a decomposition of a larger entity (*e.g.*, a bedroom) into smaller components (*e.g.*, walls, furniture and supported objects). ii) A set of **Or-nodes**  $V^{Or}$ , in which each node branches to alternative decompositions (*e.g.*, an indoor scene can be a bedroom or a living room), enabling the algorithm to reconfigure a scene. iii) A set of **Set nodes**  $V^{Set}$ , in which each node represents a nested And-Or relation: a set of Or-nodes serving as child branches are grouped by an And-node, and each child branch may include different numbers of objects.

- $V_T = V_T^r \cup V_T^a$ . The terminal nodes consists of two subsets of nodes: regular nodes and address nodes. i) A **regular terminal node**  $v \in V_T^r$  represents a spatial entity in

a scene (*e.g.*, an office chair in a bedroom) with attributes. In this work, the attributes include internal attributes  $A_{int}$  of object sizes  $(w, l, h)$ , external attributes  $A_{ext}$  of object position  $(x, y, z)$  and orientation ( $x - y$  plane)  $\theta$ , and sampled human positions  $A_h$ . ii) To avoid excessively dense graphs, an **address terminal node**  $v \in V_T^a$  is introduced to encode interactions that only occur in a certain context but are absent in all others [Fri03]. It is a pointer to regular terminal nodes, taking values in the set  $V_T^r \cup \{\text{nil}\}$ , representing supporting or grouping relations as shown in Figure 2.3. For instance, an address node connected with a “monitor” node from the “supported objects” node points to a “desk” node, meaning a monitor is supported by a desk; an address node of a “chair” from the “furniture” node points to a “desk” node, meaning the chair is associated with the desk as a functional group.

**Production Rules:** Corresponding to three different types of non-terminal nodes, three types of production rules are defined:

- And rules for an And-node  $v \in V^{\text{And}}$ , are defined as a deterministic decomposition

$$v \rightarrow u_1 \cdot u_2 \cdots u_{n(v)}. \quad (2.1)$$

- Or rules for an Or-node  $v \in V^{\text{Or}}$ , are defined as a switch

$$v \rightarrow u_1 | u_2 \cdots | u_{n(v)}, \quad (2.2)$$

with  $\rho_1 | \rho_2 \cdots | \rho_{n(v)}$ .

- Set rules for a Set-node  $v \in V^{\text{Set}}$  are defined as

$$v \rightarrow (\text{nil} | u_1^1 | u_1^2 | \cdots) \cdots (\text{nil} | u_{n(v)}^1 | u_{n(v)}^2 | \cdots), \quad (2.3)$$

with  $(\rho_{1,0} | \rho_{1,1} | \rho_{1,2} | \cdots) \cdots (\rho_{n(v),0} | \rho_{n(v),1} | \rho_{n(v),2} | \cdots)$ , where  $u_i^k$  denotes the case that object  $u_i$  appears  $k$  times, and the probability is  $\rho_{i,k}$ .

**Terminal Nodes:** The set of terminal nodes can be divided into two types: (i) regular terminal nodes  $v \in V_T^r$  representing spatial entities in a scene, with attributes  $A$  divided into internal  $A_{in}$  (size) and external  $A_{ex}$  (position and orientation) attributes, and (ii) address terminal nodes  $v \in V_T^a$  that point to regular terminal nodes and take values in the set  $V_T^r \cup \{\text{nil}\}$ . These latter nodes avoid excessively dense graphs by encoding interactions that occur only in a certain context [Fri03].

**Contextual Relations:** The contextual relations  $E = E_w \cup E_f \cup E_o \cup E_g$  among nodes are represented by horizontal links in the AOG. The relations are divided into four subsets:

- relations between furniture and walls  $E_w$ ;
- relations among furniture  $E_f$ ;
- relations between supported objects and their supporting objects  $E_o$  (e.g., monitor and desk); and
- relations between objects of a functional pair  $E_g$  (e.g., sofa and TV).

Accordingly, the cliques formed in the terminal layer may also be divided into four subsets:

$$C = C_w \cup C_f \cup C_o \cup C_g.$$

Note that the contextual relations of nodes will be inherited from their parents; hence, the relations at a higher level will eventually collapse into cliques  $C$  among the terminal nodes. These contextual relations also form an MRF on the terminal nodes. To encode the contextual relations, we define different types of potential functions for different kinds of cliques.

**Parse Tree:** A hierarchical parse tree  $pt$  instantiates the S-AOG by selecting a child node for the Or-nodes as well as determining the state of each child node for the Set-nodes. A parse graph  $pg$  consists of a parse tree  $pt$  and a number of contextual relations  $E$  on the parse tree:  $pg = (pt, E_{pt})$ . Figure 2.4 illustrates a simple example of a parse graph and four types of cliques formed in the terminal layer.

## 2.4 Probabilistic Formulation of S-AOG

The purpose of representing indoor scenes using an S-AOG is to bring the advantages of compositional hierarchy and contextual relations to the generation of realistic and diverse novel/unseen scene configurations from a learned S-AOG. In this section, we introduce the related probabilistic formulation.

**Prior:** We define the prior probability of a scene configuration generated by an S-AOG with the parameter set  $\Theta$ . A scene configuration is represented by  $pg$ , including objects in the scene and their attributes. The prior probability of  $pg$  generated by an S-AOG parameterized by  $\Theta$  is formulated as a Gibbs distribution

$$p(pg|\Theta) = \frac{1}{Z} \exp\{-\mathcal{E}(pg|\Theta)\} \quad (2.4)$$

$$= \frac{1}{Z} \exp\{-\mathcal{E}(pt|\Theta) - \mathcal{E}(E_{pt}|\Theta)\}, \quad (2.5)$$

where  $\mathcal{E}(pg|\Theta)$  is the energy function of the parse graph,  $\mathcal{E}(pt|\Theta)$  is the energy function of a parse tree, and  $\mathcal{E}(E_{pt}|\Theta)$  is the energy function of the contextual relations. Here,  $\mathcal{E}(pt|\Theta)$  is defined as combinations of probability distributions with closed-form expressions, and  $\mathcal{E}(E_{pt}|\Theta)$  is defined as potential functions relating to the external attributes of the terminal nodes.



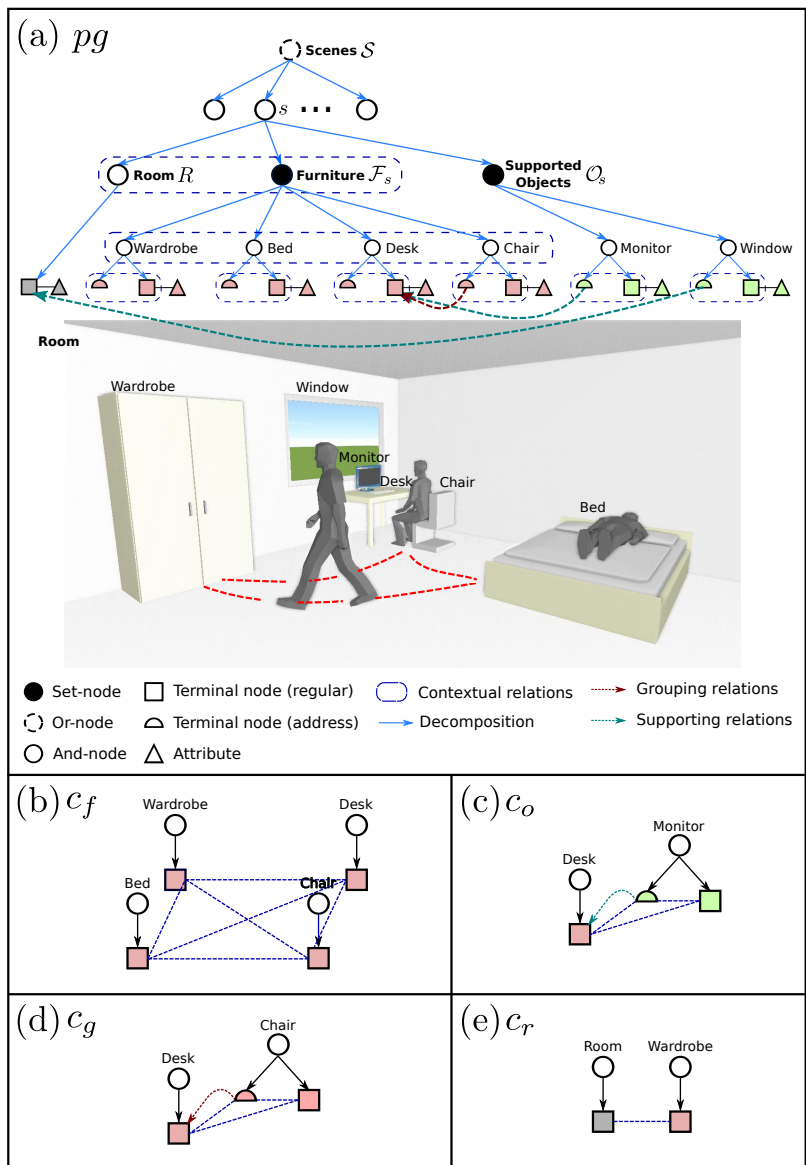


Figure 2.4: (a) A simplified example of a parse graph of a bedroom. The terminal nodes of the parse graph form an MRF in the terminal layer. Cliques are formed by the contextual relations projected to the terminal layer. Examples of the four types of cliques are shown in (b)-(e), representing four different types of contextual relations.

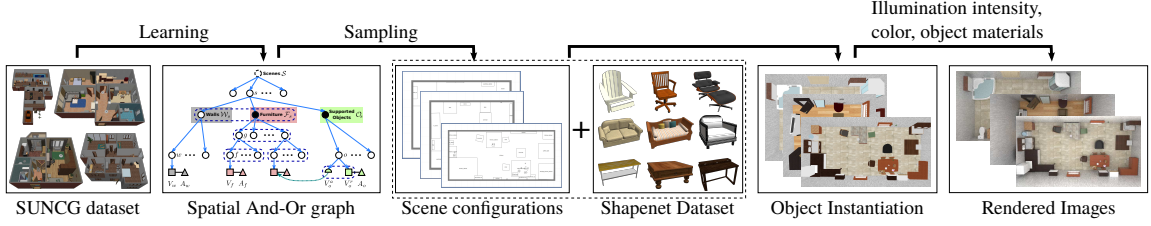


Figure 2.5: The learning-based pipeline for synthesizing images of indoor scenes.

**Energy of Parse Tree:** Energy  $\mathcal{E}(pt|\Theta)$  is further decomposed into energy functions of different types of non-terminal nodes, and energy functions of internal attributes of both regular and address terminal nodes:

$$\mathcal{E}(pt|\Theta) = \underbrace{\sum_{v \in V^{\text{Or}}} \mathcal{E}_{\Theta}^{\text{Or}}(v) + \sum_{v \in V^{\text{Set}}} \mathcal{E}_{\Theta}^{\text{Set}}(v)}_{\text{non-terminal nodes}} + \underbrace{\sum_{v \in V_T^T} \mathcal{E}_{\Theta}^{A_{\text{in}}}(v)}_{\text{terminal nodes}}, \quad (2.6)$$

where the choice of child node of an Or-node  $v \in V^{\text{Or}}$  follows a multinomial distribution, and each child branch of a Set-Node  $v \in V^{\text{Set}}$  follows a Bernoulli distribution. Note that the And-nodes are deterministically expanded; hence, (Eq. 2.6) lacks an energy term for the And-nodes. The internal attributes  $A_{\text{in}}$  (size) of terminal nodes follows a non-parametric probability distribution learned via kernel density estimation.

**Energy of Contextual Relations:**  $\mathcal{E}(E_{pt}|\Theta)$  combines the potentials of the four types of cliques formed in the terminal layer, integrating human attributes and external attributes of regular terminal nodes:

$$p(E_{pt}|\Theta) = \frac{1}{Z} \exp\{-\mathcal{E}(E_{pt}|\Theta)\} \quad (2.7)$$

$$= \prod_{c \in C_f} \phi_f(c) \prod_{c \in C_o} \phi_o(c) \prod_{c \in C_g} \phi_g(c) \prod_{c \in C_r} \phi_r(c). \quad (2.8)$$

### Human Centric Potential Functions:

- Potential function  $\phi_f(c)$  is defined on relations between furniture (Figure 2.4(b)).

The clique  $c = \{f_i\} \in C_f$  includes all the terminal nodes representing furniture:

$$\phi_f(c) = \frac{1}{Z} \exp\{-\lambda_f \cdot \langle \sum_{f_i \neq f_j} l_{\text{col}}(f_i, f_j), l_{\text{ent}}(c) \rangle\}, \quad (2.9)$$

where  $\lambda_f$  is a weight vector,  $\langle \cdot, \cdot \rangle$  denotes a vector, and the cost function  $l_{\text{col}}(f_i, f_j)$  is the overlapping volume of the two pieces of furniture, serving as the penalty of collision. The cost function  $l_{\text{ent}}(c) = -H(\Gamma) = -\sum_i p(\gamma_i) \log p(\gamma_i)$  yields better utility of the room space by sampling human trajectories, where  $\Gamma$  is the set of planned trajectories in the room, and  $H(\Gamma)$  is the entropy. The trajectory probability map is first obtained by planning a trajectory  $\gamma_i$  from the center of every piece of furniture to another one using bi-directional rapidly-exploring random tree (RRT) [LaV98], which forms a heatmap. The entropy is computed from the heatmap as shown in Figure 2.6.

- Potential function  $\phi_o(c)$  is defined on relations between a supported object and the supporting furniture (Figure 2.4(c)). A clique  $c = \{f, a, o\} \in C_o$  includes a supported object terminal node  $o$ , the address node  $a$  connected to the object, and the furniture terminal node  $f$  pointed by  $a$ :

$$\phi_o(c) = \frac{1}{Z} \exp\{-\lambda_o \cdot \langle l_{\text{hum}}(f, o), l_{\text{add}}(a) \rangle\}, \quad (2.10)$$

where the cost function  $l_{\text{hum}}(f, o)$  defines the human usability cost—a favorable human position should enable an agent to access or use both the furniture and the object. To compute the usability cost, human positions  $h_i^o$  are first sampled based on position, orientation, and the affordance map of the supported object. Given a piece of furniture, the probability

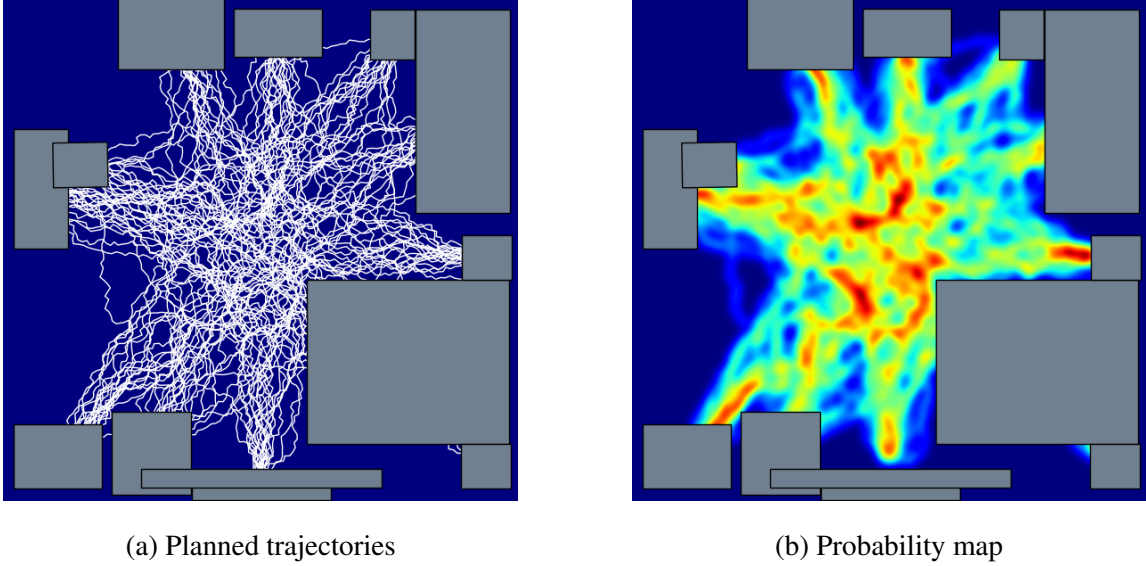


Figure 2.6: Given a scene configuration, we use bi-directional RRT to plan from every piece of furniture to another, generating a human activity probability map.

of the human positions is then computed by:

$$l_{\text{hum}}(f, o) = \max_i p(h_i^o | f). \quad (2.11)$$

The cost function  $l_{\text{add}}(a)$  is the negative log probability of an address node  $v \in V_T^a$ , treated as a certain regular terminal node, following a multinomial distribution.

- Potential function  $\phi_g(c)$  is defined on functional grouping relations between furniture (Figure 2.4(d)). A clique  $c = \{f_i, a, f_j\} \in C_g$  consists of terminal nodes of a core functional furniture  $f_i$ , pointed by the address node  $a$  of an associated furniture  $f_j$ . The grouping relation potential is defined similarly to the supporting relation potential

$$\phi_g(c) = \frac{1}{Z} \exp\{-\lambda_c \cdot \langle l_{\text{hum}}(f_i, f_j), l_{\text{add}}(a) \rangle\}. \quad (2.12)$$

**Other Potential Functions:**

- Potential function  $\phi_r(c)$  is defined on relations between the room and furniture (Figure 2.4(e)). A clique  $c = \{f, r\} \in C_r$  includes a terminal node  $f$  and  $r$  representing a piece of furniture and a room, respectively. The potential is defined as

$$\phi_r(c) = \frac{1}{Z} \exp\{-\lambda_r \cdot \langle l_{\text{dis}}(f, r), l_{\text{ori}}(f, r) \rangle\}, \quad (2.13)$$

where the distance cost function is defined as  $l_{\text{dis}}(f, r) = -\log p(d|\Theta)$ , in which  $d \sim \ln \mathcal{N}(\mu, \sigma^2)$  is the distance between the furniture and the nearest wall modeled by a log normal distribution. The orientation cost function is defined as  $l_{\text{ori}}(f, r) = -\log p(\theta|\Theta)$ , where  $\theta \sim p(\mu, \kappa) = \frac{e^{\kappa \cos(x-\mu)}}{2\pi I_0(\kappa)}$  is the relative orientation between the model and the nearest wall modeled by a von Mises distribution.

## 2.5 Learning, Sampling and Synthesis

Before introducing the algorithm for learning all the parameters associated with an S-AOG, in Section 2.5.1, note that our configurable scene synthesis pipeline includes the following components:

- A sampling algorithm based on the learned S-AOG for synthesizing realistic scene geometric configurations. This sampling algorithm controls the size of the individual objects as well as their pair-wise relations. More complex relations are recursively formed using pair-wised relations. The details are found in Section 2.5.2.
- An attribute assignment process, which sets different material attributes to each object part, as well as various camera parameters and illuminations of the environment. The details are found in Section 2.5.4.

The above two components are the essence of *configurable* scene synthesis; the first gener-

ates the structure of the scene while the second controls its detailed attributes. In between these two components, a scene instantiation process is applied to generate a 3D mesh of the scene based on the sampled scene layout. This step is described in Section 2.5.3. Figure 2.5 illustrates the pipeline. At the end of this section, we showcase several examples of synthesized scenes with different configurable attributes.

### 2.5.1 Learning the S-AOG

We use the SUNCG dataset [SYZ17b] as training data. It contains over 45K different scenes with manually created realistic room and furniture layouts. We collect the statistics of room types, room sizes, furniture occurrences, furniture sizes, relative distances, orientations between furniture and walls, furniture affordance, grouping occurrences, and supporting relations. The parameters  $\Theta$  of a probability model can be learned in a supervised way from a set of  $N$  observed parse trees  $\{pt_n, n = 1, 2, \dots, N\}$  by maximum likelihood estimation (MLE)

$$\Theta^* = \arg \max_{\Theta} \prod_{n=1}^N p(pt_n | \Theta). \quad (2.14)$$

We now describe how to learn all the parameters  $\Theta$ , with the focus on learning the weights of the loss functions.

**Weights of Loss Functions:** Recall that the probability distribution of cliques formed in the terminal layer is

$$p(E_{pt} | \Theta) = \frac{1}{Z} \exp\{-\mathcal{E}(E_{pt} | \Theta)\} \quad (2.15)$$

$$= \frac{1}{Z} \exp\{-\langle \lambda, l(E_{pt}) \rangle\}, \quad (2.16)$$

where  $\lambda$  is the weight vector and  $l(E_{pt})$  is the loss vector given by four different types of potential functions.

To learn the weight vector, the standard MLE maximizes the average log-likelihood:

$$\mathcal{L}(E_{pt}|\Theta) = -\frac{1}{N} \sum_{n=1}^N \langle \lambda, l(E_{pt_n}) \rangle - \log Z. \quad (2.17)$$

This is usually maximized by following the gradient:

$$\frac{\partial \mathcal{L}(E_{pt}|\Theta)}{\partial \lambda} = -\frac{1}{N} \sum_{n=1}^N l(E_{pt_n}) - \frac{\partial \log Z}{\partial \lambda} \quad (2.18)$$

$$= -\frac{1}{N} \sum_{n=1}^N l(E_{pt_n}) - \frac{\partial \log \sum_{pt} \exp\{-\langle \lambda, l(E_{pt}) \rangle\}}{\partial \lambda} \quad (2.19)$$

$$= -\frac{1}{N} \sum_{n=1}^N l(E_{pt_n}) + \sum_{pt} \frac{1}{Z} \exp\{-\langle \lambda, l(E_{pt}) \rangle\} l(E_{pt}) \quad (2.20)$$

$$= -\frac{1}{N} \sum_{n=1}^N l(E_{pt_n}) + \frac{1}{\tilde{N}} \sum_{\tilde{n}=1}^{\tilde{N}} l(E_{pt_{\tilde{n}}}), \quad (2.21)$$

where  $\{E_{pt_{\tilde{n}}}\}_{\tilde{n}=1, \dots, \tilde{N}}$  is the set of synthesized examples from the current model.

It is usually computationally infeasible to sample a Markov chain that burns into an *equilibrium distribution* at every iteration of gradient ascent. Hence, instead of waiting for the Markov chain to converge, we adopt the contrastive divergence (CD) learning that follows the gradient of difference of two divergences [Hin02]

$$\text{CD}_{\tilde{N}} = \text{KL}(p_0||p_{\infty}) - \text{KL}(p_{\tilde{n}}||p_{\infty}), \quad (2.22)$$

where  $\text{KL}(p_0||p_{\infty})$  is the Kullback-Leibler divergence between the data distribution  $p_0$  and the model distribution  $p_{\infty}$ , and  $p_{\tilde{n}}$  is the distribution obtained by a Markov chain started at the data distribution and run for a small number  $\tilde{n}$  of steps. In this work, we set  $\tilde{n} = 1$ .

Contrastive divergence learning has been applied effectively to addressing various problems; one of the most notable work is in the context of Restricted Boltzmann Machines [HS06]. Both theoretical and empirical evidences shows its efficiency while keeping bias typically very small [CH05a]. The gradient of the contrastive divergence is given

by

$$\frac{\partial \text{CD}_{\tilde{N}}}{\partial \lambda} = \frac{1}{N} \sum_{n=1}^N l(E_{pt_n}) - \frac{1}{\tilde{N}} \sum_{\tilde{n}=1}^{\tilde{N}} l(E_{pt_{\tilde{n}}}) - \frac{\partial p_{\tilde{n}}}{\partial \lambda} \frac{\partial \text{KL}(p_{\tilde{n}} \| p_{\infty})}{\partial p_{\tilde{n}}}. \quad (2.23)$$

Extensive simulations [Hin02] showed that the third term can be safely ignored since it is small and seldom opposes the resultant of the other two terms.

Finally, the weight vector is learned by gradient descent computed by generating a small number  $\tilde{N}$  of examples from the Markov chain

$$\lambda_{t+1} = \lambda_t - \eta_t \frac{\partial \text{CD}_{\tilde{N}}}{\partial \lambda} \quad (2.24)$$

$$= \lambda_t + \eta_t \left( \frac{1}{\tilde{N}} \sum_{\tilde{n}=1}^{\tilde{N}} l(E_{pt_{\tilde{n}}}) - \frac{1}{N} \sum_{n=1}^N l(E_{pt_n}) \right). \quad (2.25)$$

**Branching Probabilities:** The MLE of the branch probabilities  $\rho_i$  of Or-nodes, Set-nodes and address terminal nodes is simply the frequency of each alternative choice [ZM07]:

$$\rho_i = \frac{\#(v \rightarrow u_i)}{\sum_{j=1}^{n(v)} \#(v \rightarrow u_j)} \quad (2.26)$$

**Grouping Relations:** The grouping relations are hand-defined (*i.e.*, nightstands are associated with beds, chairs are associated with desks and tables). The probability of occurrence is learned as a multinomial distribution, and the supporting relations are automatically extracted from SUNCG.

**Room Size and Object Sizes:** The distribution of the room size and object size among all the furniture and supported objects is learned as a non-parametric distribution. We first extract the size information from the 3D models inside SUNCG dataset, and then fit a non-parametric distribution using kernel density estimation. The distances and relative



orientations of the furniture and objects to the nearest wall are computed and fitted into a log normal and a mixture of von Mises distributions, respectively.

**Affordances:** We learn the affordance maps of all the furniture and supported objects by computing the heatmap of possible human positions. These position include annotated humans, and we assume that the center of chairs, sofas, and beds are positions that humans often visit. By accumulating the relative positions, we get reasonable affordance maps as non-parametric distributions as shown in Figure 2.7.

### 2.5.2 Sampling Scene Geometry Configurations

Based on the learned S-AOG, we sample scene configurations (parse graphs) based on the prior probability  $p(pg|\Theta)$  using a Markov Chain Monte Carlo (MCMC) sampler. The sampling process comprises two major steps:

1. Top-down sampling of the parse tree structure  $pt$  and internal attributes of objects. This step selects a branch for each Or-node as well as chooses a child branch for every Set-node. In addition, internal attributes (sizes) of each regular terminal node are also sampled. Note that this can be easily done by sampling from closed-form distributions.
2. MCMC sampling of the external attributes (positions and orientations) of objects as well as the values of the address nodes. Samples are proposed by Markov chain dynamics, and are taken after the Markov chain converges to the prior probability. These attributes are constrained by multiple potential functions, hence it is difficult to directly sample from the true underlying probability distribution.

Algorithm 1 provides an overview of the sampling process. Some qualitative results are

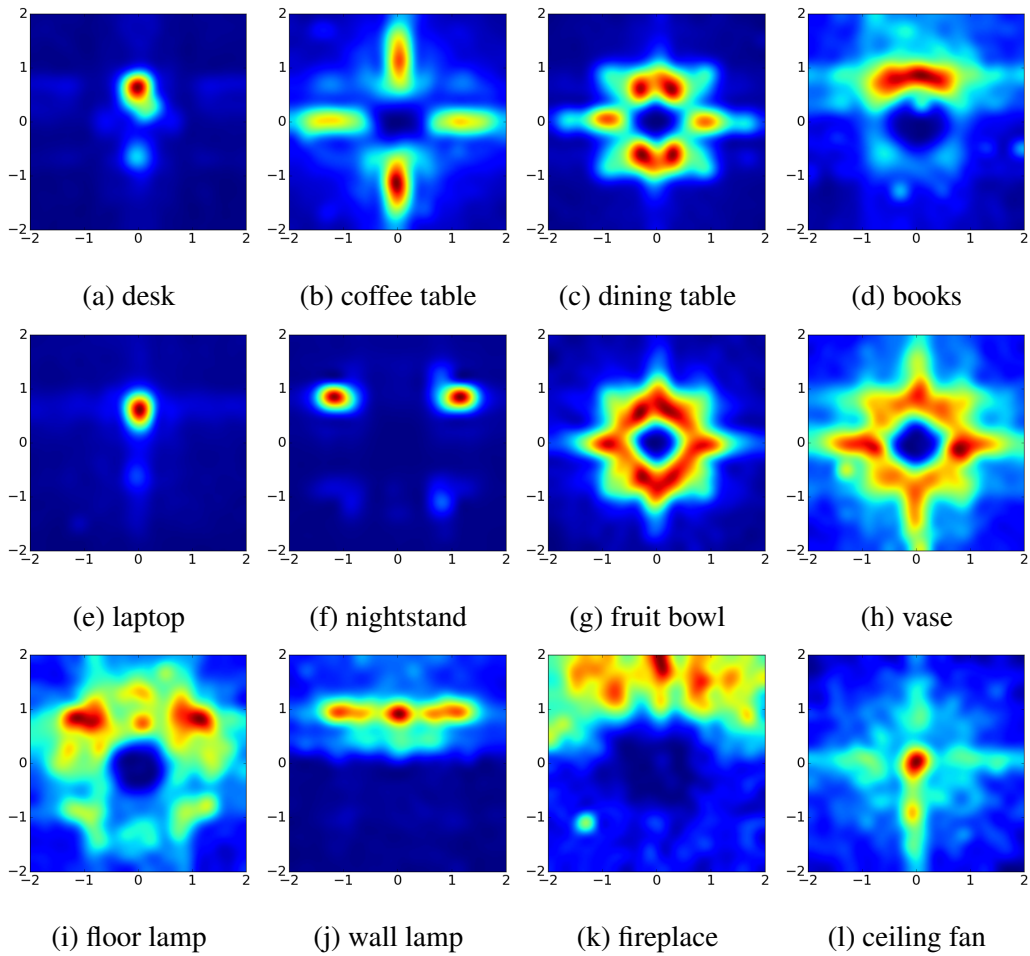


Figure 2.7: Examples of the learned affordance maps. Given the object positioned in the center facing upwards, *i.e.*, coordinate of  $(0,0)$  facing direction  $(0,1)$ , the maps show the distributions of human positions. The affordance maps accurately capture the subtle differences among desks, coffee tables, and dining tables. Some objects are orientation sensitive, *e.g.*, books, laptops, and night stands, while some are orientation invariant, *e.g.*, fruit bowls and vases.

---

**Algorithm 1: Sampling Scene Configurations**

---

**Input** : Attributed S-AOG  $\mathcal{G}$

Landscape parameter  $\beta$

sample number  $n$

**Output:** Synthesized room layouts  $\{pg_i\}_{i=1,\dots,n}$

```
1 for  $i = 1$  to  $n$  do
2   Sample the child nodes of the Set nodes and Or nodes from  $\mathcal{G}$  directly to get the
   structure of  $pg_i$ .
3   Sample the sizes of room, furniture  $f$  and objects  $o$  in  $pg_i$  directly.
4   Sample the address nodes  $V^a$ .
5   Randomly initialize positions and orientations of furniture  $f$  and objects  $o$  in
    $pg_i$ .
6    $iter = 0$ 
7   while  $iter < iter_{\max}$  do
8     Propose a new move and get proposal  $pg'_i$ .
9     Sample  $u \sim \text{unif}(0, 1)$ .
10    if  $u < \min(1, \exp(\beta(\mathcal{E}(pg_i|\Theta) - \mathcal{E}(pg'_i|\Theta))))$  then
11      |  $pg_i = pg'_i$ 
12    end
13     $iter += 1$ 
14  end
15 end
```

---

shown in Figure 2.10.

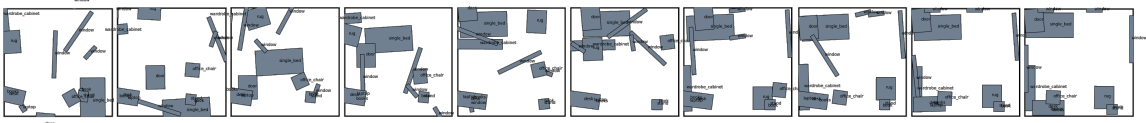


Figure 2.8: MCMC sampling process (from left to right) of scene configurations with simulated annealing.

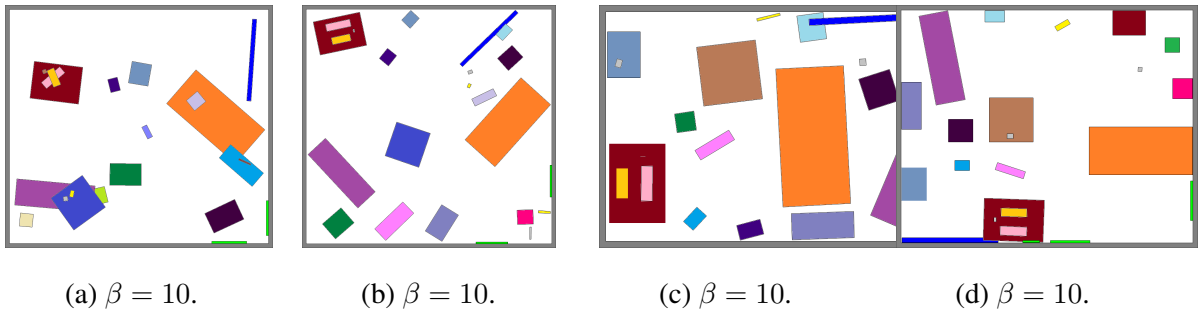


Figure 2.9: Synthesis for different values of  $\beta$ . Each image shows a typical configuration sampled from a Markov chain.

**Markov Chain Dynamics:** Four types of Markov chain dynamics  $q_i, i = 1, 2, 3, 4$  are designed to be chosen randomly with probabilities to propose moves. Specifically, the dynamics  $q_1$  and  $q_2$  are diffusion, while  $q_3$  and  $q_4$  are reversible jumps:

1. *Translation of Objects.* Dynamic  $q_1$  chooses a regular terminal node and samples a new position based on the current position of the object

$$\text{pos} \rightarrow \text{pos} + \delta\text{pos}, \quad (2.27)$$

where  $\delta\text{pos}$  follows a bivariate normal distribution.

2. *Rotation of Objects.* Dynamic  $q_2$  chooses a regular terminal node and samples a new

orientation based on the current orientation of the object

$$\theta \rightarrow \theta + \delta\theta, \quad (2.28)$$

where  $\delta\theta$  follows a normal distribution.

3. *Swapping of Objects.* Dynamic  $q_3$  chooses two regular terminal nodes and swaps the positions and orientations of the objects.
4. *Swapping of Supporting Objects.* Dynamic  $q_4$  chooses an address terminal node and samples a new regular furniture terminal node pointed to. We sample a new 3D location  $(x, y, z)$  for the supported object:
  - Randomly sample  $x = u_x * w_p$ , where  $u_x \sim \text{unif}(0, 1)$ , and  $w_p$  is the width of the supporting object.
  - Randomly sample  $y = u_y * l_p$ , where  $u_y \sim \text{unif}(0, 1)$ , and  $l_p$  is the length of the supporting object.
  - The height  $z$  is simply the height of the supporting object.

Adopting the Metropolis-Hastings algorithm, a newly proposed parse graph  $pg'$  is accepted according to the following acceptance probability:

$$\alpha(pg'|pg, \Theta) = \min\left(1, \frac{p(pg'|\Theta)p(pg|pg')}{p(pg|\Theta)p(pg'|pg)}\right) \quad (2.29)$$

$$= \min\left(1, \frac{p(pg'|\Theta)}{p(pg|\Theta)}\right) \quad (2.30)$$

$$= \min\left(1, \exp(\mathcal{E}(pg|\Theta) - \mathcal{E}(pg'|\Theta))\right). \quad (2.31)$$

The proposal probabilities are canceled since the proposed moves are symmetric in probability.

**Convergence:** To test if the Markov chain has converged to the prior probability, we keep a histogram of the energy of the last  $w$  samples. When the difference between two histograms at a distance of  $s$  sampling steps is smaller than a threshold  $\epsilon$ , the Markov chain is considered to have converged.

**Tidiness of Scenes:** During the sampling process, a typical state is drawn from the distribution. We can easily control the level of tidiness of the sampled scenes by adding an extra parameter  $\beta$  to control the landscape of the prior distribution:

$$p(pg|\Theta) = \frac{1}{Z} \exp\{-\beta\mathcal{E}(pg|\Theta)\}. \quad (2.32)$$

Some examples are shown in Figure 2.9.

Note that the parameter  $\beta$  is analogous albeit differs from the temperature in simulated annealing optimization—the temperature in simulated annealing is time-variant; *i.e.*, it changes during the simulated annealing process. In our model, we simulate a Markov chain under one specific  $\beta$  to get typical samples at a certain level of tidiness. When  $\beta$  is small, the distribution is “smooth”; *i.e.*, the differences between local minima and local maxima are small. A simulated annealing scheme is adopted to obtain samples with high probability as shown in Figure 2.8.

### 2.5.3 Scene Instantiation using 3D Object Datasets

Given a generated 3D scene layout, the 3D scene is instantiated by assembling objects into it using 3D object datasets. In this work, we incorporate both ShapeNet dataset [CFG15] and SUNCG dataset [SYZ17a] as our 3D model dataset. Scene instantiation includes five steps:

1. For each object in the scene layout, find the model has the closest the length/width ratio to the dimension specified in the scene layout.
2. Align the orientations of selected models according to the orientation specified in the scene layout.
3. Transform the models to the specified positions, and scales the models according to the generated scene layout.
4. Since we only fit the length and width in Step 1, an extra step to adjust object position along the gravity direction is needed, eliminating all the floating models and the models that penetrated into each other.
5. Add the floor, walls, and ceiling to complete the instantiated scene.

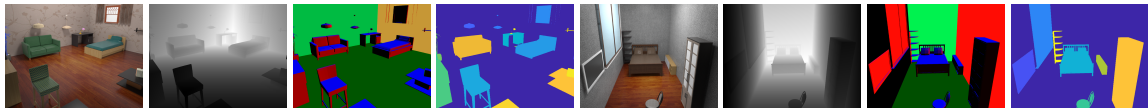
#### **2.5.4 Scene Attribute Configurations**

As we generate scenes in a forward fashion, our pipeline enables the precise customization and control of important attributes of the generated scenes. Some configurations are shown in Figure 2.11. The rendered images are determined by combinations of the following four factors:

- Illuminations, including light source positions, intensities, colors, and the number of light sources.
- Material and texture of the environment; *i.e.*, the walls, floor and ceiling.
- Cameras, such as fisheye, panorama, and Kinect cameras, have different focal lengths and apertures, yielding dramatically different rendered images. By virtue of physics-



(a) Different categories of the scenes using default attributes of object material, the lighting conditions, and camera parameters. Top row: top view. Bottom row: a random view.



(b) Additional examples of two bedrooms, with corresponding depth map, surface normal, and semantic segmentation.

Figure 2.10: Qualitative results in different types of scenes.

based rendering, our pipeline can even control the F-stop and focal distance, resulting in different depths of field.

- Different object materials and textures will have various properties, represented by roughness, metallicness, and reflectivity.

## 2.6 Photorealistic Scene Rendering

We adopt Physics-Based Rendering (PBR) [PH04] to generate the photorealistic 2D images. PBR has become the industry standard in computer graphics applications in recent years, and has been widely adopted for both offline and real-time rendering. Unlike traditional rendering techniques where heuristic shaders are used to control how light is scattered by a surface, PBR simulates the physics of real-world light by computing the bidirectional scattering distribution function (BSDF) [BDW81] of the surface.





(a) Lighting intensity: half and double

(b) Lighting color: purple and blue



(c) Different object materials: metal, gold, chocolate, and clay

Figure 2.11: We can configure the scene with different (a) illumination intensities, (b) illumination colors, (c) materials, and (d) even on each object part. We can also control (e) the number of light source and their positions, (f) camera lenses (*e.g.*, fish eye), (g) depths of field, or (h) render the scene as a panorama for virtual reality and other virtual environments. (i) 7 different background wall textures. Note how the background affects the overall illumination.

**Formulation:** Following the law of conservation of energy, PBR solves the rendering equation for the total spectral radiance of outgoing light in direction  $\mathbf{w}$  from point  $\mathbf{x}$  on a surface

$$L_o(\mathbf{x}, \mathbf{w}) = L_e(\mathbf{x}, \mathbf{w}) + \int_{\Omega} f_r(\mathbf{x}, \mathbf{w}', \mathbf{w}) L_i(\mathbf{x}, \mathbf{w}') (-\mathbf{w}' \cdot \mathbf{n}) d\mathbf{w}', \quad (2.33)$$

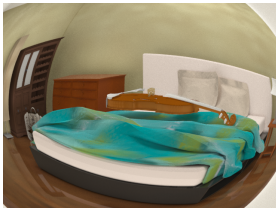
where  $L_o$  is the outgoing light,  $L_e$  is the emitted light (from a light source),  $\Omega$  is the unit hemisphere uniquely determined by  $\mathbf{x}$  and its normal,  $f_r$  is the bidirectional reflectance distribution function (BRDF),  $L_i$  is the incoming light from direction  $\mathbf{w}'$ , and  $\mathbf{w}' \cdot \mathbf{n}$  ac-



(a) Different materials in each object part



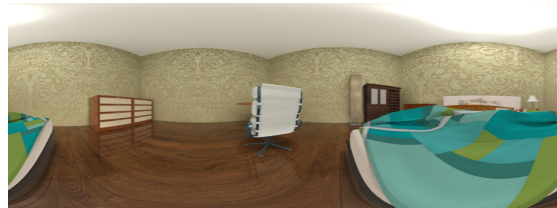
(b) Using multiple light sources



(c) Fish eye lens



(d) Depth of field



(e) Panorama images



(f) [Different background materials affect the rendering results










Figure 2.12: (Continue:) we can configure the scene with different (a) illumination intensities, (b) illumination colors, (c) materials, and (d) even on each object part. We can also control (e) the number of light source and their positions, (f) camera lenses (*e.g.*, fish eye), (g) depths of field, or (h) render the scene as a panorama for virtual reality and other virtual environments. (i) 7 different background wall textures. Note how the background affects the overall illumination.

counts for the attenuation of the incoming light.

**Advantages:** In path tracing, the rendering equation is often solved with Monte Carlo methods. Contrasting what happens in the real world, the paths of photons in a scene are traced backwards from the camera (screen pixels) to the source lights. Objects in the scene receive lighting contributions as they interact with the photon paths. By computing both the reflected and transmitted components of rays in a physically accurate way while conserving energies and obeying the refraction equations, PBR photorealistically renders shadows, reflections, and refractions, thereby capturing unprecedented levels of detail compared to other existing shading techniques. Note PBR describes a shading process and does not dictate how images are rasterized in screen space. In this work we use the *Mantra*<sup>®</sup> PBR engine to render synthetic image data with raytracing for its accurate calculation of lighting and shading as well as its physically intuitive parameter configuration.

Indoor scenes are typically closed rooms. Various reflective and diffusive surfaces may exist throughout the space. Therefore the effect of secondary rays is particularly important in achieving realistic lighting. PBR robustly samples both direct lighting contributions on surfaces from light sources and indirect lighting from rays reflected and diffused by other surfaces. The BSDF shader on a surface manages and modifies its color contribution when hit by a secondary ray. Doing so results in more secondary rays being sent out from the surface in evaluation. The reflection limit (the number of times a ray can be reflected) and the diffuse limit (the number of times diffuse rays bounce on surfaces) need to be chosen wisely to balance the final image quality and the rendering time. Decreasing indirect lighting samples will likely yield a nice rendering time reduction, but at the cost of significantly diminished visual realism.

Table 2.1: Comparisons of rendering time vs quality. The first column tabulates the reference number and rendering results used in this work, the second column lists all the criteria, and the remaining columns present comparative results. The color differences between the reference image and images rendered with various parameters are measured by LAB Delta E standard [SB02] tracing back to Helmholtz and Hering [BKW98, Val07].

Ref.	Criteria	Comparisons							
$3 \times 3$	Baseline pixel samples	$2 \times 2$	$1 \times 1$	$3 \times 3$	$3 \times 3$	$3 \times 3$	$3 \times 3$	$3 \times 3$	$3 \times 3$
0.001	Noise level	0.001	0.001	<b>0.01</b>	<b>0.1</b>	0.001	0.001	0.001	0.001
22	Maximum additional rays	22	22	22	22	<b>10</b>	<b>3</b>	22	22
6	Bounce limit	6	6	6	6	6	6	<b>3</b>	<b>1</b>
203	Time (second)	131	45	196	30	97	36	198	178
	LAB Delta E difference								

**Rendering Time vs Rendering Quality:** In summary, we use the following control parameters to adjust the render quality and speed:

- *Baseline pixel samples.* This is the minimum number of rays sent per pixel. Each pixel is typically divided evenly along both directions. Common values for this parameter are  $3 \times 3$  and  $5 \times 5$ . The higher pixel sample counts are usually required to produce motion blur and depth of field effects.
- *Noise level.* Different rays sent from each pixel will not yield identical paths. This parameter determines the maximum allowed variance among the different results. If necessary, additional rays (in addition to baseline pixel sample count) will be generated to decrease the noise.
- *Maximum additional rays.* This parameter is the upper limit of the additional rays

sent for satisfying the noise level.

- *Bounce limit.* The maximum number of secondary ray bounces. We use this parameter to restrict both diffuse and reflected rays. Note that in PBR the diffuse ray is one of the most significant contributors to realistic global illumination, while the other parameters are more important in controlling the Monte Carlo sampling noise.

Table 2.1 summarizes our analysis of how these parameters affect the render time and image quality.

## 2.7 Experiments

### 2.7.1 Realistic of Sampled Scene Configurations

First, we design three experiments to test if our algorithm can generate realistic scenes. It is based on different criteria: i) visual similarity to manually constructed scenes, ii) the accuracy of affordance maps for the synthesized scenes, and iii) functionalities and naturalness of the synthesized scenes. The first experiment compares our method with a state-of-the-art room arrangement method; the second experiment measures the synthesized affordances; the third one is an ablation study. Overall, the experiments show that our

Table 2.2: Classification results on segmentation maps of synthesized scenes using different methods vs. SUNCG.

<b>Method</b>	Yu <i>et al.</i> [YYT11]	SUNCG Perturbed	Ours
<b>Accuracy(%)</b> ↓	87.49	63.69	76.18

Table 2.3: Comparison between affordance maps computed from our samples and real data

Metric	Bathroom	Bedroom	Dining Room	Garage	Guest Room	Gym	Kitchen	Living Room	Office	Storage
Total variation	0.431	0.202	0.387	0.237	0.175	0.278	0.227	0.117	0.303	0.708
Hellinger distance	0.453	0.252	0.442	0.284	0.212	0.294	0.251	0.158	0.318	0.703

Table 2.4: Human subjects' ratings (1-5) of the sampled layouts based on functionality (top) and naturalness (bottom)

Method	Bathroom	Bedroom	Dining Room	Garage	Guest Room	Gym	Kitchen	Living Room	Office	Storage
no-context	1.12 ± 0.33	1.25 ± 0.43	1.38 ± 0.48	1.75 ± 0.66	1.50 ± 0.50	3.75 ± 0.97	2.38 ± 0.48	1.50 ± 0.87	1.62 ± 0.48	1.75 ± 0.43
object	3.12 ± 0.60	3.62 ± 1.22	2.50 ± 0.71	3.50 ± 0.71	2.25 ± 0.97	3.62 ± 0.70	3.62 ± 0.70	3.12 ± 0.78	1.62 ± 0.48	4.00 ± 0.71
Yu <i>et al.</i> [YYT11]	3.61 ± 0.52	4.15 ± 0.25	3.15 ± 0.40	3.59 ± 0.51	2.58 ± 0.31	2.03 ± 0.56	3.91 ± 0.98	4.62 ± 0.21	3.32 ± 0.81	2.58 ± 0.64
ours	4.58 ± 0.86	4.67 ± 0.90	3.33 ± 0.90	3.96 ± 0.79	3.25 ± 1.36	4.04 ± 0.79	4.21 ± 0.87	4.58 ± 0.86	3.67 ± 0.75	4.79 ± 0.58
no-context	1.00 ± 0.00	1.00 ± 0.00	1.12 ± 0.33	1.38 ± 0.70	1.12 ± 0.33	1.62 ± 0.86	1.00 ± 0.00	1.25 ± 0.43	1.12 ± 0.33	1.00 ± 0.00
object	2.88 ± 0.78	3.12 ± 1.17	2.38 ± 0.86	3.00 ± 0.71	2.50 ± 0.50	3.38 ± 0.86	3.25 ± 0.66	2.50 ± 0.50	1.25 ± 0.43	3.75 ± 0.66
Yu <i>et al.</i> [YYT11]	4.00 ± 0.52	3.85 ± 0.92	3.27 ± 1.01	2.99 ± 0.25	3.52 ± 0.93	2.14 ± 0.63	3.89 ± 0.90	3.31 ± 0.29	2.77 ± 0.67	2.96 ± 0.41
ours	4.21 ± 0.71	4.25 ± 0.66	3.08 ± 0.70	3.71 ± 0.68	3.83 ± 0.80	4.17 ± 0.75	4.38 ± 0.56	3.42 ± 0.70	3.25 ± 0.72	4.54 ± 0.71

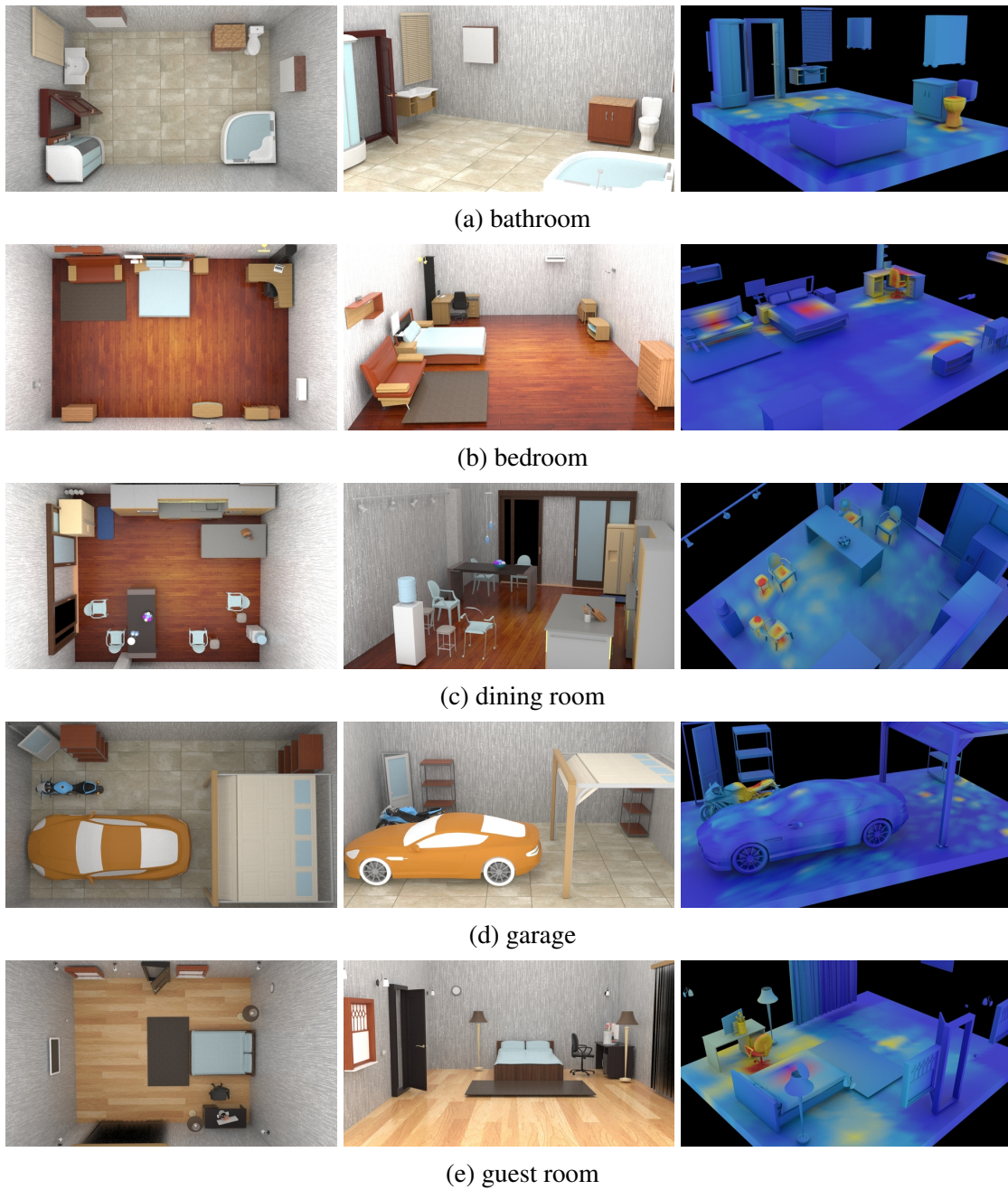


Figure 2.13: Examples of scenes in ten different categories. Top: top-view. Middle: a side-view. Bottom: affordance heatmap.

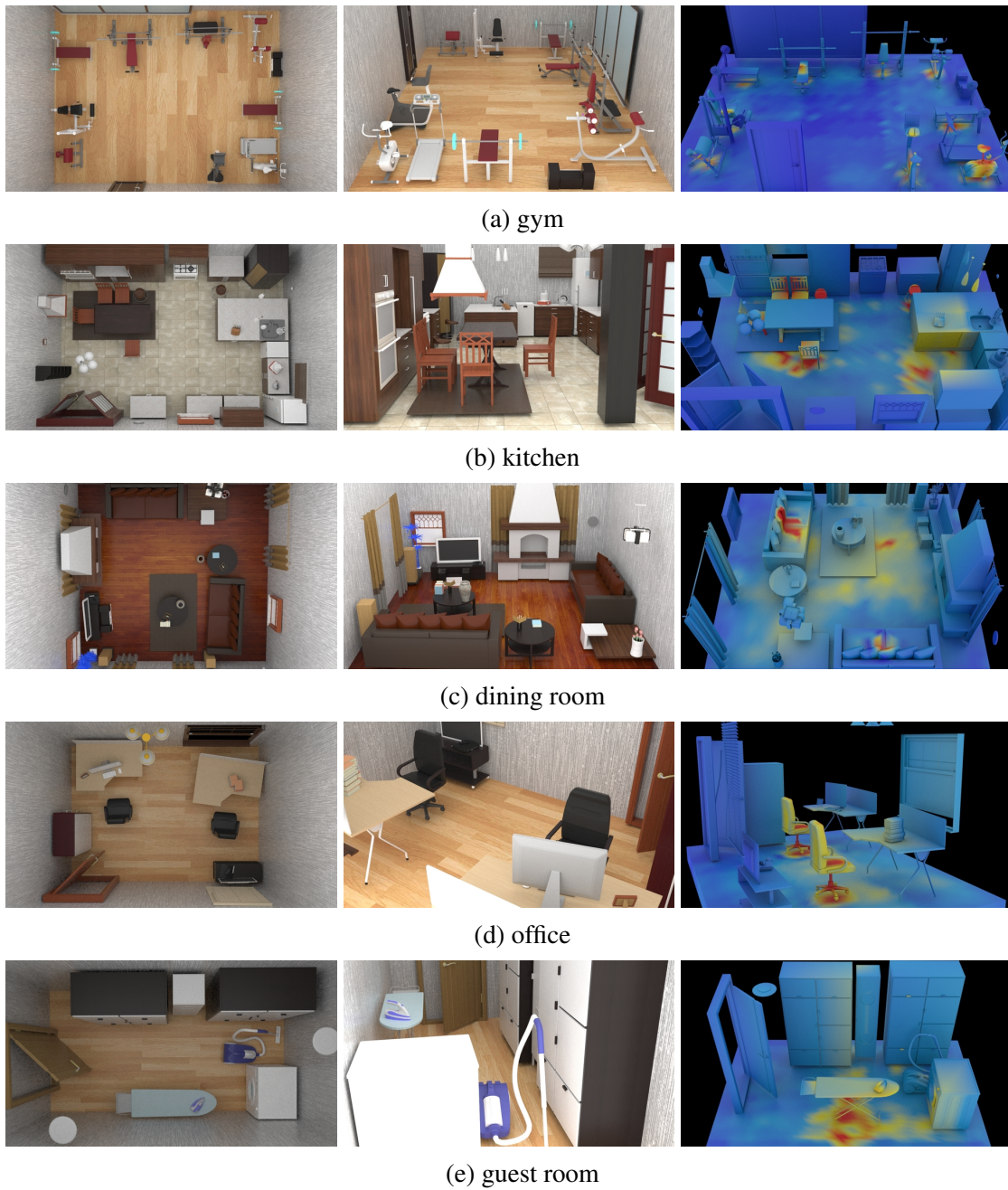


Figure 2.14: (Continue:) examples of scenes in ten different categories. Top: top-view. Middle: a side-view. Bottom: affordance heatmap.



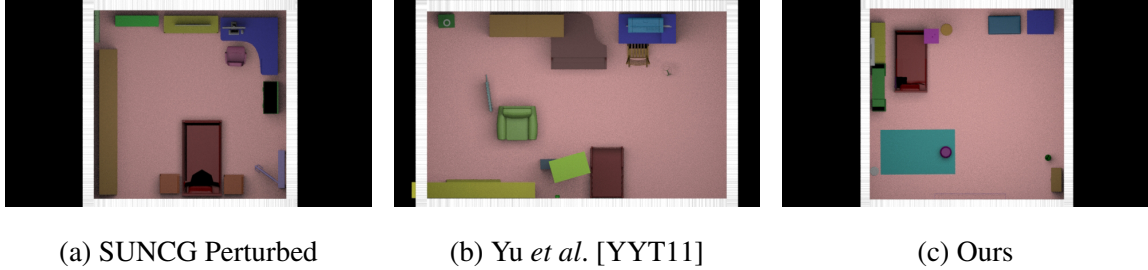


Figure 2.15: Top-view segmentation maps for classification.

algorithm can robustly sample a large variety of realistic scenes that exhibits naturalness and functionality.

### 2.7.1.1 Layout Classification

To quantitatively evaluate the visual realism, we trained a classifier on the top-view segmentation maps of synthesized scenes and SUNCG scenes. Specifically, we train a ResNet-152 [HZR16] to classify top view layout segmentation maps (synthesized vs. SUNCG). Examples of top-view segmentation maps are shown in Figure 2.15. The reason to use segmentation maps is that we want to evaluate the room layout excluding rendering factors such as object materials. We use two methods for comparison: i) a state-of-the-art furniture arrangement optimization method proposed by Yu *et al.* [YYT11], and ii) slight perturbation of SUNCG scenes by adding small Gaussian noise (*e.g.*  $\mu = 0, \sigma = 0.1$ ) to the layout. The room arrangement algorithm proposed by [YYT11] takes one pre-fixed input room and re-organizes the room. 1500 scenes are randomly selected for each method and SUNCG: 800 for training, 200 for validation, and 500 for testing. As shown in Table 2.2, the classifier successfully distinguishes Yu *et al.* vs. SUNCG with an accuracy of 87.49%. Our method achieves a better performance of 76.18%, exhibiting a higher realism



Figure 2.16: **Top:** previous methods [YYT11] only re-arranges a given input scene with a fixed room size and a predefined set of objects. **Bottom:** our method samples a large variety of scenes.

and larger variety. This result indicates our method is much more visually similar to real scenes than the comparative scene optimization method. Qualitative comparisons of Yu *et al.* and our method are shown in Figure 2.16.

### 2.7.1.2 Affordance Maps Comparison

We sample 500 rooms of 10 different scene categories summarized in Table 2.3. For each type of room, we compute the affordance maps of the objects in the synthesized samples, and calculate both the total variation distances and Hellinger distances between the affordance maps computed from the synthesized samples and the SUNCG dataset. The two distributions are similar if the distance is close to 0. Most sampled scenes using the proposed method show similar affordance distributions to manually created ones from SUNCG. Some scene types (*e.g.* Storage) show a larger distance since they do not exhibit

clear affordances. Overall, the results indicate that affordance maps computed from the synthesized scenes are reasonably close to the ones computed from manually constructed scenes by artists.

### **2.7.1.3 Functionality and naturalness**

Three methods are used for comparison: (i) direct sampling of rooms according to the statistics of furniture occurrence without adding contextual relation, (ii) an approach that only models object-wise relations by removing the human constraints in our model, and (iii) the algorithm proposed by Yu *et al.* [YYT11]. We showed the sampled layouts using three methods to 4 human subjects. Subjects were told the room category in advance, and instructed to rate given scene layouts without knowing the method used to generate the layouts. For each of the 10 room categories, 24 samples were randomly selected using our method and [YYT11], and 8 samples were selected using both the object-wise modeling method and the random generation. The subjects evaluated the layouts based on two criteria: (i) functionality of the rooms, *e.g.*, can the “bedroom” satisfies a human’s needs for daily life; and (ii) the naturalness and realism of the layout. Scales of responses range from 1 to 5, with 5 indicating perfect functionality or perfect naturalness and realism. The mean ratings and the standard deviations are summarized in Table 2.4. Our approach outperforms the three methods in both criteria, demonstrating the ability to sample a functionally reasonable and realistic scene layout. More qualitative results are shown in Figure 2.13.

Table 2.5: Performance of normal estimation for the NYU-Depth V2 dataset with different training protocols.

pre-train	fine-tune	mean↓	median↓	11.25° ↑	22.5° ↑	30° ↑
	NYUv2	27.30	21.12	27.21	52.61	64.72
	Eigen	22.2	15.3	38.6	64.0	73.9
[ZSY17]	NYUv2	21.74	14.75	39.37	66.25	76.06
ours+[ZSY17]	NYUv2	<b>21.47</b>	<b>14.45</b>	<b>39.84</b>	<b>67.05</b>	<b>76.72</b>

### 2.7.2 Synthesized Indoor Scene Data for Scene Understanding

In this section, we further demonstrate the usefulness of the generated synthetic indoor scenes from two perspectives:

1. Improving state-of-the-art computer vision models by training with our synthetic data. We showcase our results on the task of normal prediction and depth prediction from a single RGB image, demonstrating the potential of using the proposed dataset.
2. Benchmarking common scene understanding tasks with configurable object attributes and various environments, which evaluates the stabilities and sensitivities of the algorithms, providing directions and guidelines for their further improvement in various vision tasks.

The reported results use the reference parameters indicated in Table 2.1. We found that choosing parameters for lower-quality rendering via the Mantra renderer does not provide training images that suffice to outperform state-of-the-art methods using the experimental setup described below.



(a) RGB

(b) ground truth

(c) estimation

(d) error

Figure 2.17: Examples of normal estimation results predicted by the model trained with our synthetic data.

### 2.7.2.1 Normal Prediction

Predicting surface normals from a single RGB image is an essential task in scene understanding since it provides important information in recovering the 3D structure of the scenes. We train a neural network with our synthetic data to demonstrate that the perfect per-pixel ground truth generated using our pipeline could be utilized to improve upon

the state-of-the-art performance on a specific scene understanding task. Using the fully convolutional network model described by Zhang *et al.* [ZSY17], we compare the normal estimation results given by models trained under two different protocols: (i) the network is directly trained and tested on the NYU-Depth V2 dataset, and (ii) the network is first pre-trained using our synthetic data, then fine-tuned and tested on NYU-Depth V2.

Following the standard evaluation protocol [FGH13, BRG16], we evaluate a per-pixel error over the entire dataset. To evaluate the prediction error, we computed the mean, median, and RMSE of angular error between the predicted normals and ground truth normals. Prediction accuracy is given by calculating the fraction of pixels that are correct within a threshold  $t$ , where  $t = 11.25^\circ, 22.5^\circ, 30^\circ$ . Our experimental results are summarized in Table 2.5. By utilizing our synthetic data, the model achieves better performance. From the visualized results in Figure 2.17, we can see that the error mainly accrues in the area where the ground truth normal map is noisy. We argue that part of the reason is due to the sensor’s noise or sensing distance limit. Such results in turn imply the importance to have perfect per-pixel ground truth for training and evaluation.

### 2.7.2.2 Depth Estimation

Single-image depth estimation is a fundamental problem in computer vision, which has found broad applications in scene understanding, 3D modeling, and robotics. The problem is challenging since no reliable depth cues are available. In this task, the algorithms output a depth image based on a single RGB input image.

To demonstrate the efficacy of our synthetic data, we compare the depth estimation results provided by models trained following protocols similar to those we used in normal prediction with the network in [LSL15]. To perform a quantitative evaluation, we used the

Table 2.6: Depth estimation performance on the NYU-Depth V2 dataset with different training protocols.

Pre-Train	Fine-Tune	Error					Accuracy		
		Abs Rel	Sqr Rel	Log10	RMSE(linear)	RMSE(log)	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
NYUv2	-	0.233	0.158	0.098	0.831	0.117	0.605	0.879	0.965
Ours	-	0.241	0.173	0.108	0.842	0.125	0.612	0.882	0.966
Ours	NYUv2	<b>0.226</b>	<b>0.152</b>	<b>0.090</b>	<b>0.820</b>	<b>0.108</b>	<b>0.616</b>	<b>0.887</b>	<b>0.972</b>

metrics applied in previous work [EPF14]:

- Abs relative error:  $\frac{1}{N} \sum_p \frac{|d_p - d_p^{gt}|}{d_p^{gt}}$ ,
- Square relative difference:  $\frac{1}{N} \sum_p \frac{|d_p - d_p^{gt}|^2}{d_p^{gt}}$ ,
- Average  $\log_{10}$  error:  $\frac{1}{N} \sum_x |\log_{10}(d_p) - \log_{10}(d_p^{gt})|$ ,
- RMSE:  $\sqrt{\frac{1}{N} \sum_x |d_p - d_p^{gt}|^2}$ ,
- Log RMSE:  $\sqrt{\frac{1}{N} \sum_x |\log(d_p) - \log(d_p^{gt})|^2}$ ,
- Threshold: % of  $d_p$  s.t.  $\max(\frac{d_p}{d_p^{gt}}, \frac{d_p^{gt}}{d_p}) < \text{threshold}$ ,

where  $d_p$  and  $d_p^{gt}$  are the predicted depths and the ground truth depths at the pixel indexed by  $p$ , respectively, and  $N$  is the number of pixels in all the evaluated images. The first five metrics capture the error calculated over all the pixels; lower values are better. The threshold criteria capture the estimation accuracy; higher values are better.

Table 2.6 summarizes the results. We can see that the model pretrained on our dataset and fine-tuned on the NYU-Depth V2 dataset achieves the best performance, both in error and accuracy. Figure 2.18 shows qualitative results. This demonstrates the usefulness of our dataset in improving algorithm performance in scene understanding tasks.

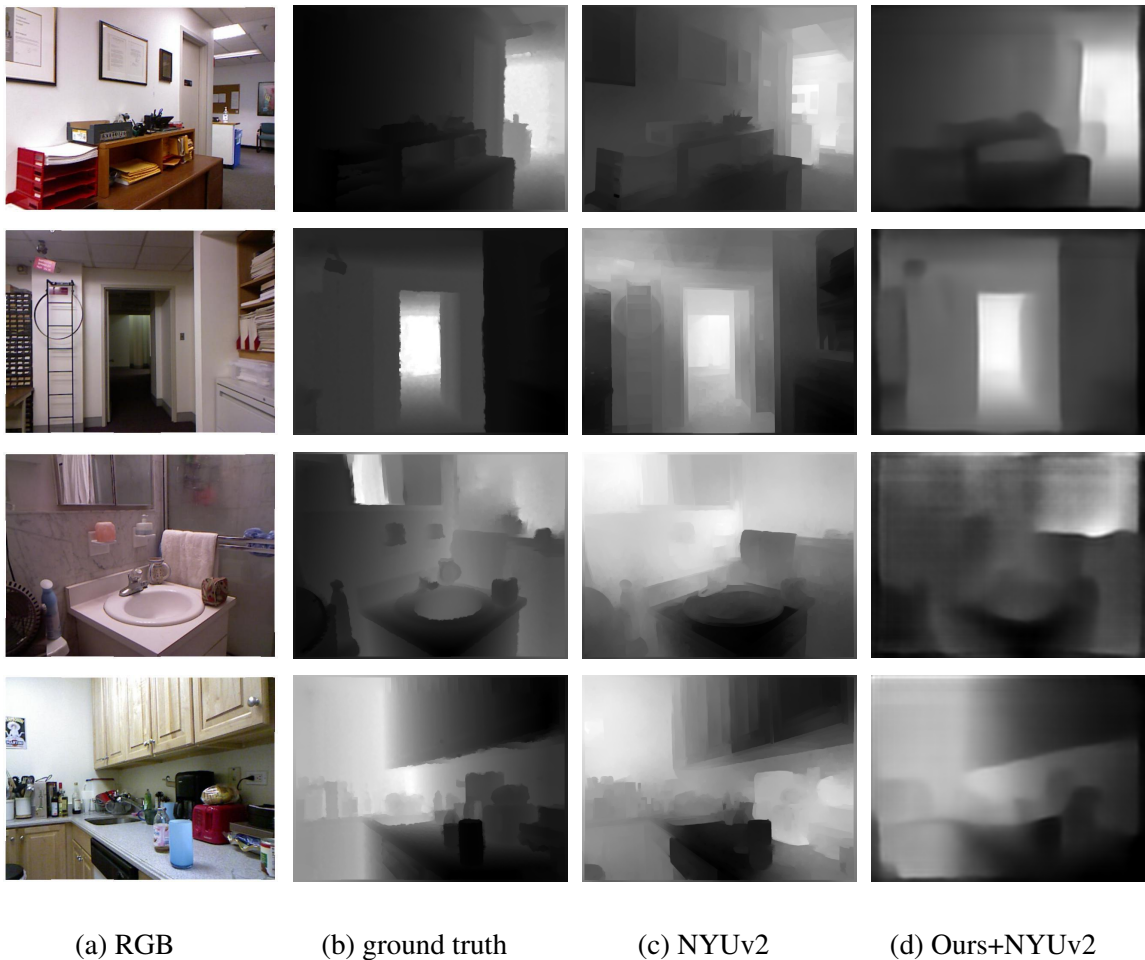


Figure 2.18: Examples of depth estimation results predicted by the model trained with our synthetic data.

### 2.7.2.3 Benchmark and Diagnosis

In this section, we show benchmark results and provide a diagnosis of various common computer vision tasks using our synthetic dataset.



Table 2.7: Depth estimation. Intensity, color, and material represent the scene with different illumination intensities, colors, and object material properties, respectively.

Setting	Method	Error					Accuracy		
		Abs Rel	Sqr Rel	Log10	RMSE(linear)	RMSE(log)	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Original	[LSL15]	0.225	0.146	0.089	0.585	0.117	0.642	0.914	0.987
	[EPF14]	0.373	0.358	0.147	0.802	0.191	0.367	0.745	0.924
	[EF15]	0.366	0.347	0.171	0.910	0.206	0.287	0.617	0.863
Intensity	[LSL15]	0.216	0.165	0.085	0.561	0.118	0.683	0.915	0.971
	[EPF14]	0.483	0.511	0.183	0.930	0.24	0.205	0.551	0.802
	[EF15]	0.457	0.469	0.201	1.01	0.217	0.284	0.607	0.851
Color	[LSL15]	0.332	0.304	0.113	0.643	0.166	0.582	0.852	0.928
	[EPF14]	0.509	0.540	0.190	0.923	0.239	0.263	0.592	0.851
	[EF15]	0.491	0.508	0.203	0.961	0.247	0.241	0.531	0.806
Material	[LSL15]	0.192	0.130	0.08	0.534	0.106	0.693	0.930	0.985
	[EPF14]	0.395	0.389	0.155	0.823	0.199	0.345	0.709	0.908
	[EF15]	0.393	0.395	0.169	0.882	0.209	0.291	0.631	0.889

**Depth Estimation.** In the presented benchmark, we evaluated three state-of-the-art single-image depth estimation algorithms due to Eigens *et al.* [EPF14, EF15] and Liu *et al.* [LSL15]. We evaluated those three algorithms with data generated from different settings including illumination intensities, colors, and object material properties. Table 2.7 shows a quantitative comparison. We see that both [EPF14] and [EF15] are very sensitive to illumination conditions, whereas [LSL15] is robust to illumination intensity, but sensitive to illumination color. All three algorithms are robust to different object materials. The reason may be that material changes do not alter the continuity of the surfaces. Note that [LSL15] exhibits nearly the same performance on both our dataset and the NYU-Depth V2 dataset, supporting the assertion that our synthetic scenes are suitable for algorithm evaluation and diagnosis.

**Normal Estimation.** Next, we evaluated two surface normal estimation algorithms due to Eigens *et al.* [EF15] and Bansal *et al.* [BRG16]. Table 2.8 summarizes our quantitative results. Compared with depth estimation, the surface normal estimation algorithms are stable to different illumination conditions as well as to different material properties. As in depth estimation, these two algorithms achieve comparable results on both our dataset and the NYU dataset.

**Semantic Segmentation.** Semantic segmentation has become one of the most popular tasks in scene understanding since the development and success of fully convolutional networks (FCNs). Given a single RGB image, the algorithm outputs a semantic label for every image pixel. We applied the semantic segmentation model described by Eigen *et al.* [EF15]. Since we have 129 classes of indoor objects whereas the model only has a maximum of 40 classes, we rearranged and reduced the number of classes to fit the

Table 2.8: Surface Normal Estimation. Intensity, color, and material represent the setting with different illumination intensities, illumination colors, and object material properties.

Setting	Method	Error			Accuracy		
		Mean	Median	RMSE	11.25°	22.5°	30°
Original	[EF15]	22.74	13.82	32.48	43.34	67.64	75.51
	[BRG16]	24.45	16.49	33.07	35.18	61.69	70.85
Intensity	[EF15]	24.15	14.92	33.53	39.23	66.04	73.86
	[BRG16]	24.20	16.70	32.29	32.00	62.56	72.22
Color	[EF15]	26.53	17.18	36.36	34.20	60.33	70.46
	[BRG16]	27.11	18.65	35.67	28.19	58.23	68.31
Material	[EF15]	22.86	15.33	32.62	36.99	65.21	73.31
	[BRG16]	24.15	16.76	32.24	33.52	62.50	72.17

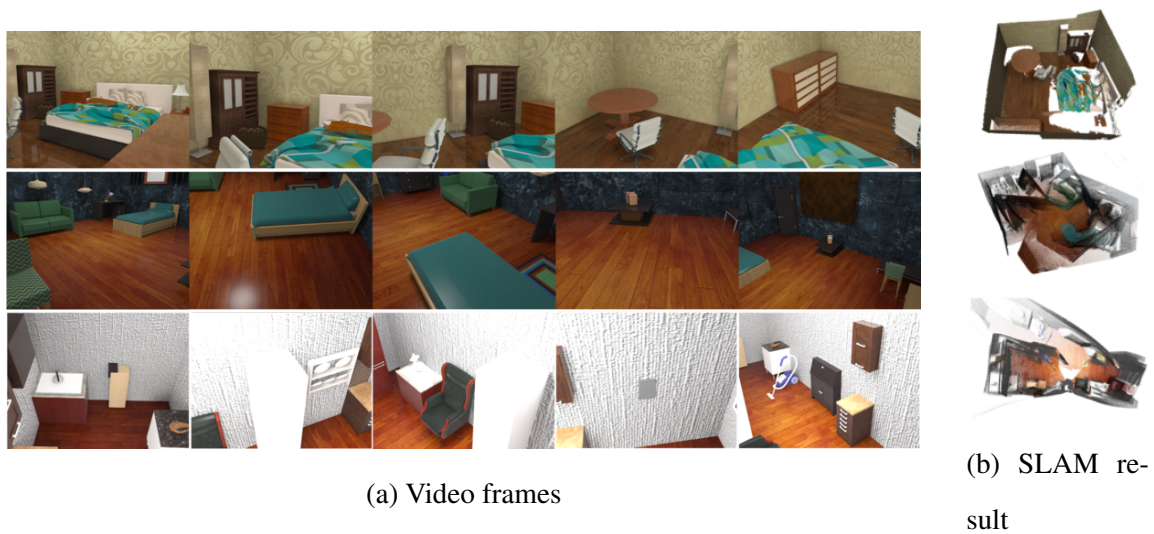


Figure 2.19: We can render the scenes as (a) a sequence of video frames after setting a camera trajectory, (b) which can be used to evaluate SLAM reconstruction [WLS15] results. The top row shows a successful reconstruction case, while the middle and bottom rows show two failure cases due to a fast moving camera and a plain, untextured surface, respectively.

prediction of the model. The algorithm achieves 60.5 pixel accuracy and 50.4 mIoU on our dataset.

**3D Reconstructions and SLAM.** We can evaluate 3D reconstruction and SLAM algorithms using images rendered from a sequence of different camera views. We generated different sets of images on diverse synthesized scenes with various camera motion paths and backgrounds to evaluate the effectiveness of the open-source SLAM algorithm ElasticFusion [WLS15]. A qualitative result is shown in Figure 2.19. Some scenes can be robustly reconstructed when we rotate the camera evenly and smoothly, as well as when both the background and foreground objects have rich textures. However, other recon-

structured 3D meshes are badly fragmented due to the failure to register the current frame with previous frames due to fast moving cameras or the lack of rich textures. Experiments indicate that our synthetic scenes with configurable attributes and background can be utilized to diagnose the SLAM algorithm since we have full control of both the scenes and the camera trajectories.

**Object Detection.** The performance of object detection algorithms have greatly improved in recent years with the appearance and development of region-based convolutional neural networks. We apply the Faster R-CNN Model [RHG15] to detect objects. We again need to rearrange and reduce the number of classes for evaluation. Figure 2.20 summarizes our qualitative results with a bedroom scene. Note that a change of material can adversely affect the output of the model—when the material of objects is changed to metal, the bed is detected as a “car”.

## 2.8 Discussion

**Complexity of synthesis.** The time complexity is hard to measure since MCMC sampling is adopted. Empirically, it takes about 20-40 minutes to sample an interior layout (20000 iterations of MCMC), and roughly 12-20 minutes to render a  $640 \times 480$  image on a normal PC. The rendering speed depends on settings related to illumination, environments, and the size of the scene, *etc.*.

**Configurable scene synthesis:** The most significant distinction between the our work and prior work reported in the literature is our ability to generate large-scale *configurable* 3D scenes. But why is configurable generation desirable, given the fact that SUNCG [SYZ17a]

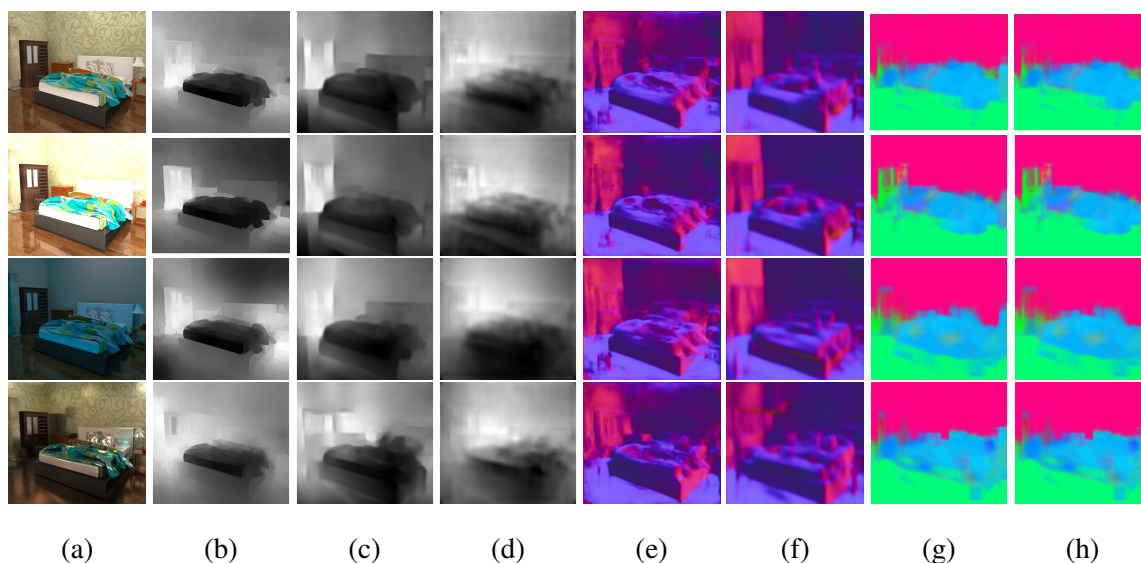


Figure 2.20: Benchmark results. (a) Given a set of generated RGB images rendered with different illuminations and object material properties (top to bottom: original settings, with high illumination, with blue illumination, and with metallic material properties), we evaluate (b)–(d) three depth prediction algorithms, (e)–(f) two surface normal estimation algorithms, (g) a semantic segmentation algorithm, and (h) an object detection algorithm.

already provided a large dataset of manually created 3D scenes?

A direct and obvious benefit is the potential to generate *unlimited* training data. As shown in a recent report by Sun *et al.* [SSS17], after introducing a dataset with 300 times of the size of ImageNet [DDS09], the performance of supervised learning appears to continue to increase linearly in proportion to the increased volume of labeled data. Such results indicate the usefulness of labeled datasets on a scale even larger than SUNCG. Although the SUNCG dataset is large by today’s standards, it is still a dataset limited by the manual specification of scene layouts.

A benefit of using configurable scene synthesis is to diagnose AI systems. Some pre-

liminary results were reported in this paper. In the future, we hope such methods can assist in building explainable AI. For instance, in the field of causal reasoning [Pea09], causal induction usually requires turning on and off specific conditions in order to draw a conclusion regarding whether or not a causal relation exists. Generating a scene in a controllable manner could provide a useful tool for studying these problems.

Furthermore, a configurable pipeline could be used to generate various virtual environment in a controllable manner in order to train virtual agents situated in virtual environments in order to learn task planning [LGS16, ZMK17] and control policy [HSL17, WMR17].

**The importance of the different energy terms:** In our experiments, the learned weights of the different energy terms indicate the importance of the terms. Based on the ranking from the largest weight to the smallest, the energy terms are 1) distances between furniture and the nearest wall, 2) relative orientations of furniture and the nearest wall, 3) supporting relations, 4) functional group relations, and 5) occlusions of the accessible space of furniture by other furniture. We can regard such rankings learned from training data as human preferences of various factors in indoor layout designs, which is important for sampling and generating realistic scenes. For example, one can imagine that it is more important to have a desk aligned with a wall (relative distance and orientation), than it is to have a chair close to a desk (functional group relations).

**Balancing rendering time and quality:** The advantage of physically accurate representation of shadows, colors, and reflections comes at the cost of computation. High quality rendering (*e.g.*, rendering for movies) requires tremendous amounts of CPU time and computer memory that is practical only with distributed render farms. Low quality settings are

prone to granular render noise due to stochastic sampling. Our comparisons between rendering time and rendering quality serve as a basic guideline for choosing the values of the rendering parameters. In practice, depending on the complexity of the scene (such as the number of light sources and reflective objects), manual adjustment is often needed in large-scale rendering (*e.g.*, an overview of a city) in order to achieve the best trade-off between rendering time and quality. Switching to GPU-based ray tracing engines is a promising alternative. This direction is especially useful for scenes with a small number of polygons and textures, which can fit into a modern GPU memory.

**The speed of the sampling process:** It takes roughly 3–5 minutes to render a  $640 \times 480$ -pixel image, depending on settings related to illumination, environments, and the size of the scene. By comparison, the sampling process consumes approximately 3 minutes with the current setup. Although the convergence speed of the Monte Carlo Markov chain is fast enough relative to photorealistic rendering, it is still desirable to accelerate the sampling process. In practice, to speed up the sampling and improve the synthesis quality, we split the sampling process into five stages: (i) Sample the objects on the wall, *e.g.*, windows, switches, paints and lights, (ii) sample the core functional objects in *functional groups* (*e.g.*, desks and beds), (iii) sample the objects that are associated with the core functional objects (*e.g.*, chairs and nightstands), (iv) sample the objects that are not paired with other objects (*e.g.*, wardrobes and bookshelves), and (v) Sample small objects that are supported by furniture (*e.g.*, laptops and books). By splitting the sampling process using functional groups, we effectively reduce the computational complexity, and different types of objects quickly converge to their final positions.



## 2.9 Conclusion and Future Work

Our proposed learning-based pipeline for generating and rendering configurable room layouts can synthesize massive quantities of images with detailed, per-pixel ground truth information for supervised training. We believe that the ability to generate room layouts in a controllable manner can benefit various vision areas, including but not limited to depth prediction [EPF14, EF15, LSL15, LRB16], surface normal prediction [WFG15, EF15, BRG16], semantic segmentation [LSD15, NHH15, CPK16], reasoning about object-supporting relations [FSH11, SHK12, ZZY15, LZZ16], material recognition [BUS13, BBS14, BUS15, WYL15], recovery of illumination conditions [NZI01, SSI03, KN09, ON14, BM15, HN05, ZDN15, ON16, LN16], inference of room layout and scene parsing [HEH05, HHH09, LHK09, GHK10, DBF12, XRT12, ZZ13, ML15, CCP15], determination of object functionality and affordance [SB91, BR06, GGV11, HRB11, ZZ13, GSE11, JKS13, ZFF14, MKF14, KS14, YDY15, KS16, RT16], and physical reasoning [ZZY13, ZZY15, ZZZ15, WYL15, ZJZ16, Wu16]. In addition, we believe that research on 3D reconstruction in robotics and on the psychophysics of human perception can also benefit from our work.

Our current approach has several limitations that we plan to address in future research. First, the scene generation process can be improved using a multi-stage sampling process; *i.e.*, sampling large furniture objects first and smaller objects later, which can potentially improve the scene layout. Second, we will consider modeling human activity inside the generated scenes, especially with regard to functionality and affordance. Third, we will consider the incorporation of moving virtual humans into the scenes, which can provide additional ground truth for human pose recognition, human tracking, and other human-related tasks. To model dynamic interactions, a Spatio-Temporal AOG (ST-AOG) representation is needed to extend the current spatial representation into the temporal domain.

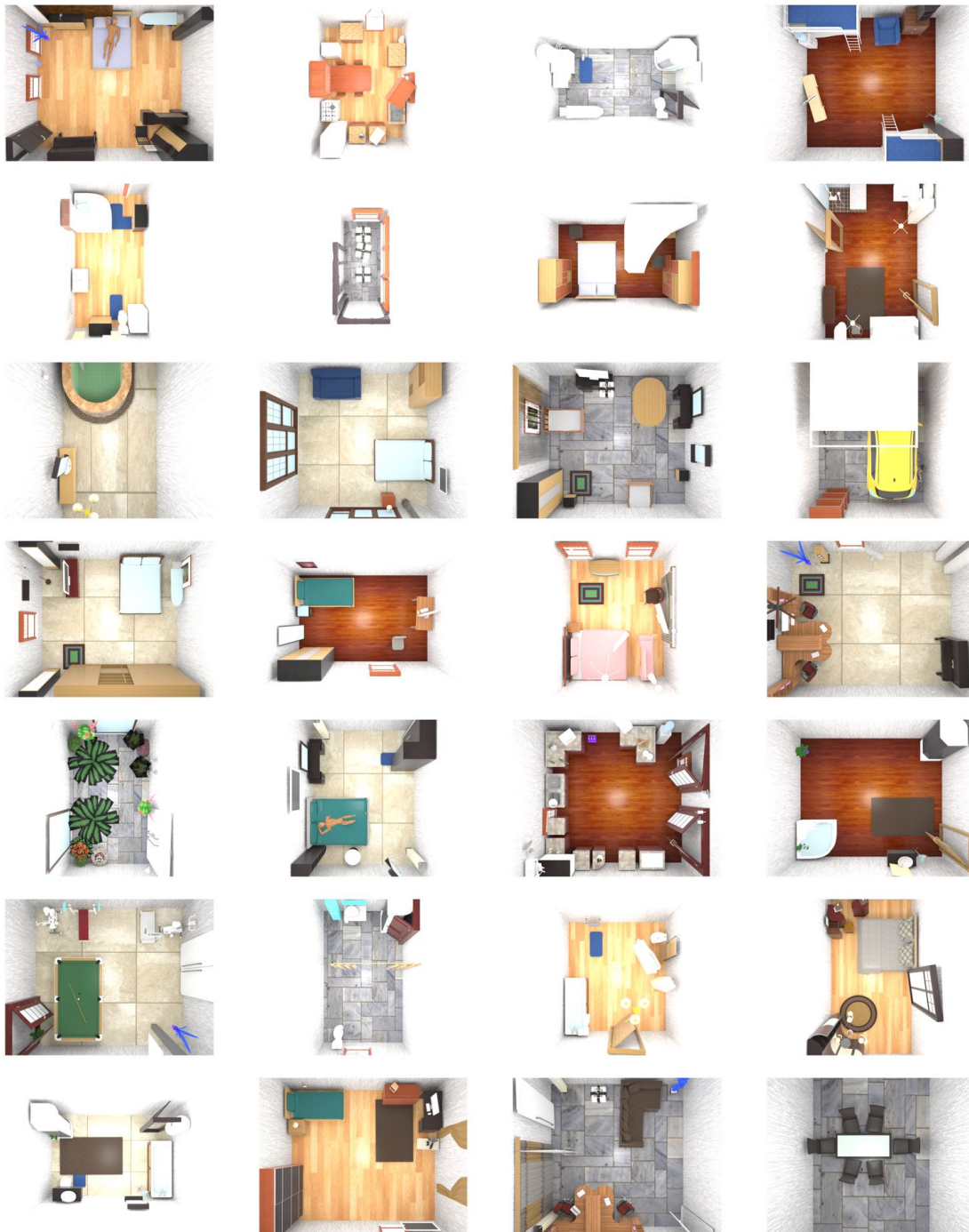
Such an extension would unlock the potential to further synthesize outdoor environments, although a large-scale, structured training dataset would be needed for learning-based approaches. Finally, domain adaptation has been shown to be important in learning from synthetic data [RSM16, LXG17, TE11]; hence, we plan to apply domain adaptation techniques to our synthetic dataset.

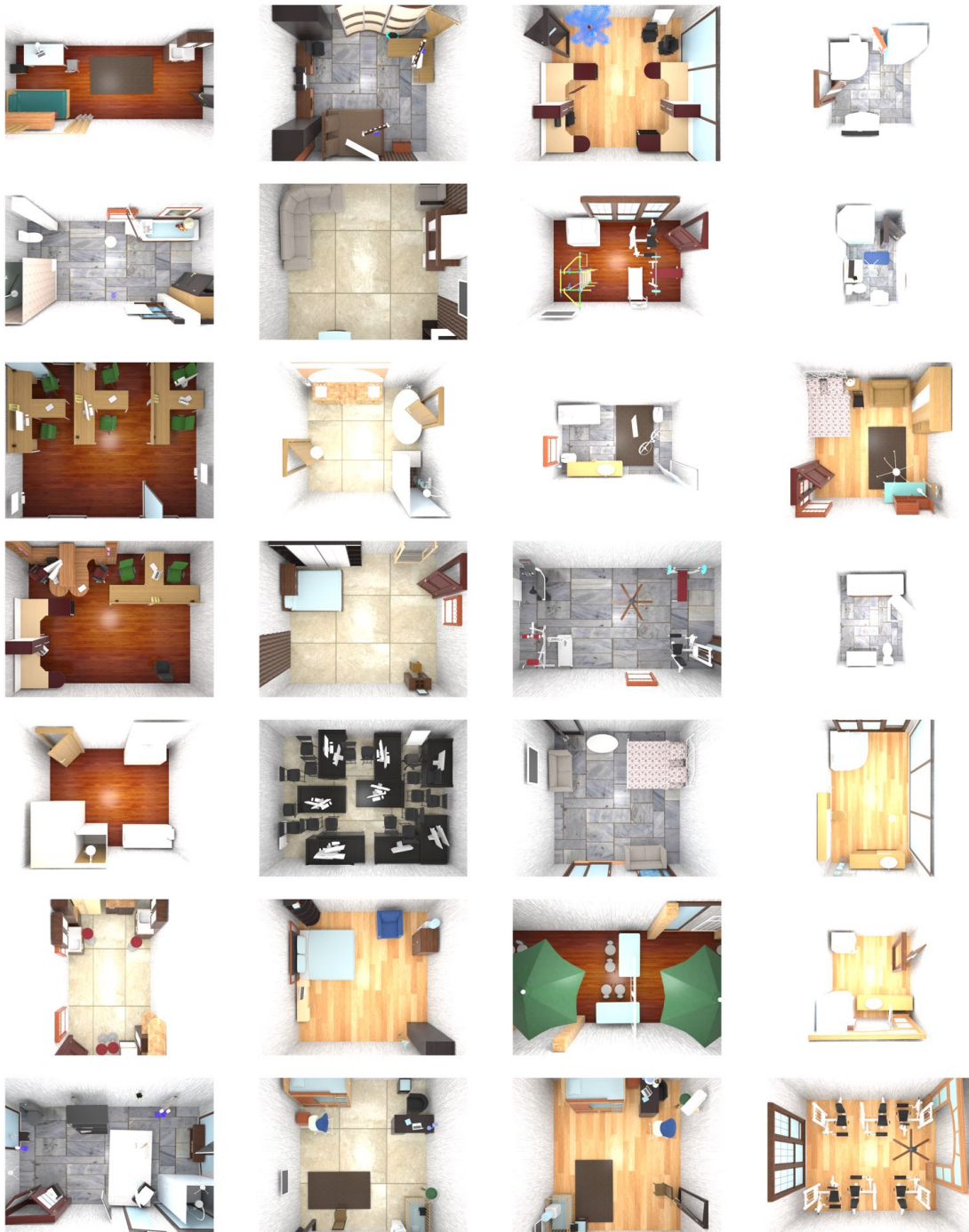
**2.10 More Results**





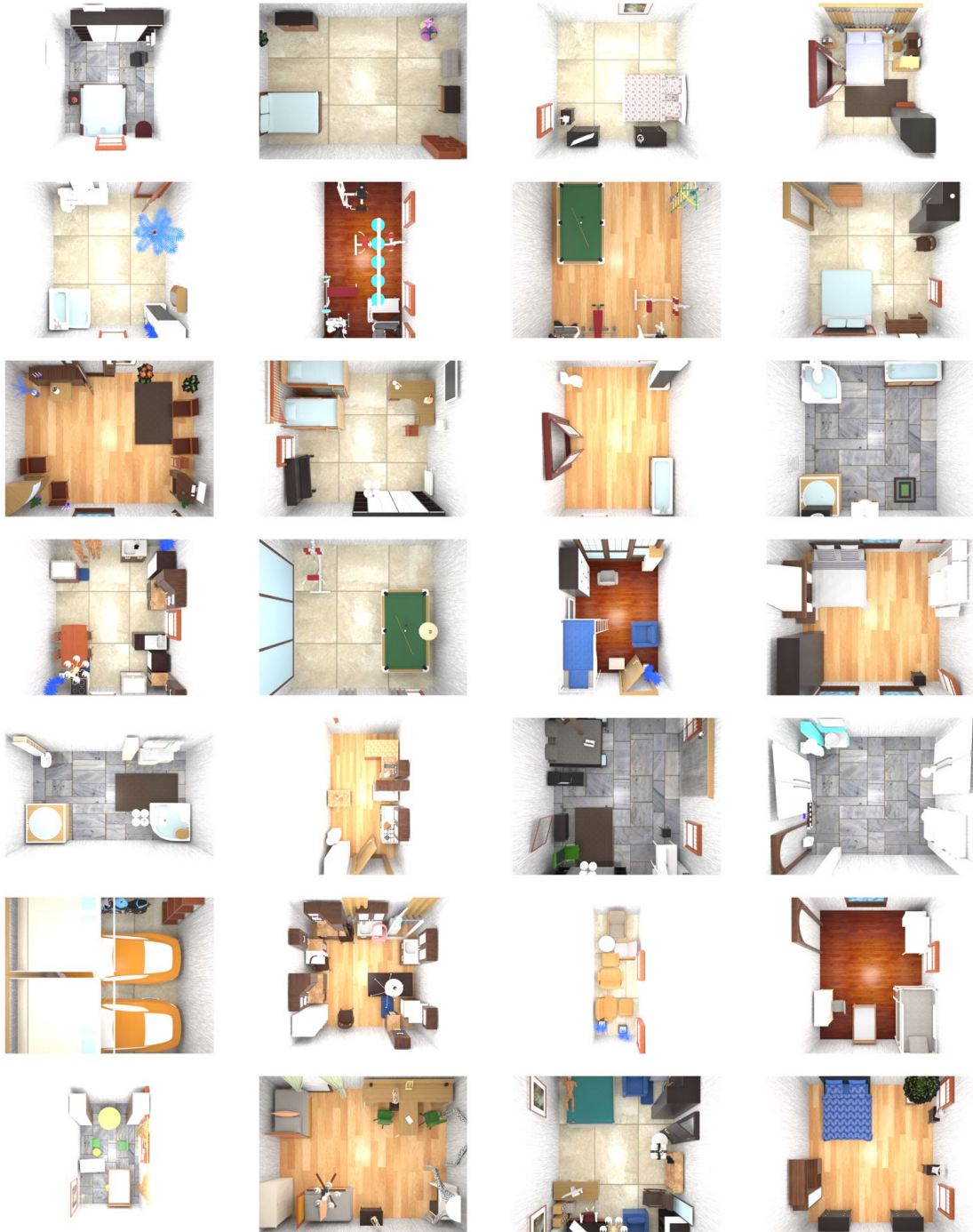














## CHAPTER 3

### Human Activity Prediction Using Stochastic Grammar

Consider the image from a video shown in Figure 3.1(a). A modern computer vision algorithm might reliably detect a human pose and some key objects in the scene: a chair, a monitor, a cup, a microwave and a water cooler. However, we as observers are able to reason beyond the current situation. We can predict what the possible future states are to some extent, and we can even evaluate how strong that belief is – a human can easily predict which state is the most likely future state from Figure 3.1(c).

We consider the task of understanding complex human activities from (partially-observed) videos from two important aspects: activity recognition and prediction. This is a ubiquitous problem driven by a wide range of applications in many perceptual tasks. Some scenarios further require the algorithm to have both recognition and prediction capabilities, *e.g.*, assistive robots need to recognize the current human activity and provide future-aware assistance.

Besides applications, a joint solution of recognition and prediction is also motivated from a modeling perspective. Activity prediction needs the observer to reason beyond appearance to find out many underlying factors: what happened, what is happening, what the goal of the agent is, what will happen/how the agent will perform the task. Activity recognition can benefit from answers to these questions. Attempts have been made to address activity prediction in both the computer vision [LF14, KZB12, WGH14, AGR16,

JZS16, PJZ11, Ryo11] and the robotics community [ZRG09, KKS12, WDA12, KS16, HZG16].

To find a good solution, we need to consider two questions: 1) what is a good representation for the structure of human activities/tasks, and 2) what is a good inference algorithm to cope with such a representation. A popular family of representations for events is the Markov models (*e.g.*, hidden Markov Model). However, Markov models are not expressive enough since human tasks often exhibit non-Markovian and compositional properties. Hence we argue that 1) a representation should reflect the hierarchical/compositional task structure of human activities, and 2) an inference algorithm should recover the hierarchical structure given the past observations, and be able to predict the future based on the understanding.

To choose a model to capture the hierarchical structure of the entire history, we refer to the Chomsky hierarchy, which is a containment hierarchy of classes of formal grammars in the formal languages of computer science and linguistics. The reason is that activities are analogous to languages: actions are like words and activities are like languages. The Chomsky hierarchy categorizes language models into four levels: 1) Turing machines, 2) context-sensitive grammars, 3) context-free grammars, and 4) regular grammars. Higher-level models contain lower-level models, and Markov models belongs to the lowest level (regular grammars). In this work, we propose to use *context-free grammars* to parse and predict human activities. In the definition of formal language theory, a grammar is a set of production rules for sentences in a formal language. In our case, the rules describe how to form sentences (activities) from the language's alphabet (actions) that are valid.

However, it has not been possible to directly use symbolic grammars to parse and label sequence data (*e.g.*, videos). Traditional grammar parsers take symbolic sentences

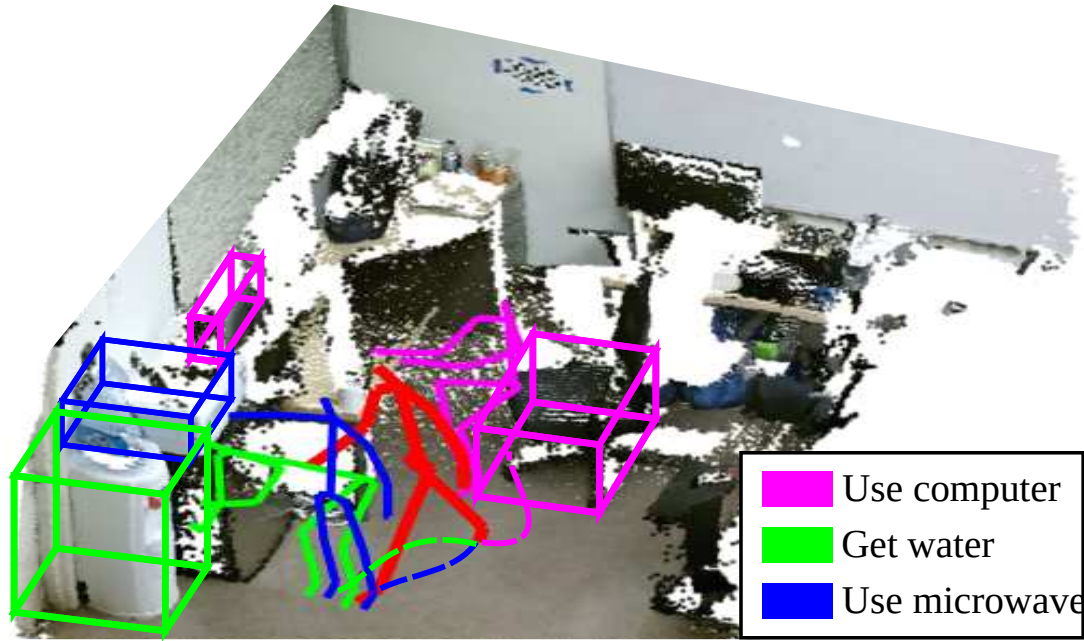
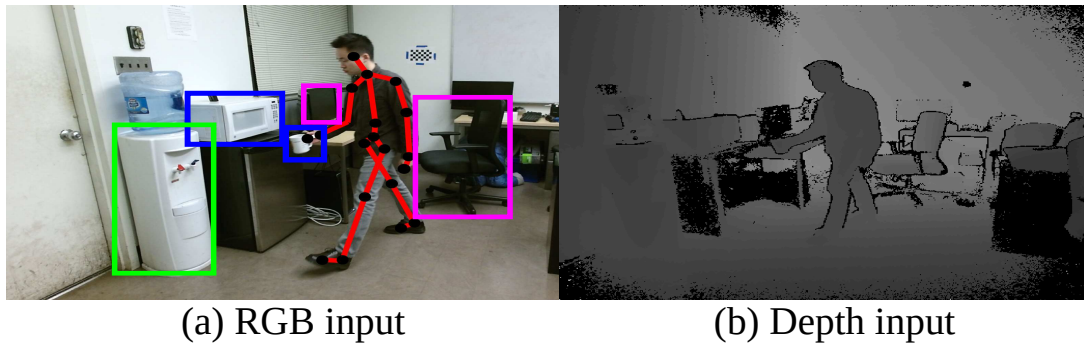


Figure 3.1: What is he going to do? (a)(b) Input RGB-D video frames. (c) Activity prediction: human action with interacting objects (how the agent will perform the task). The red skeleton is the current observation. The magenta, green and blue skeletons and interacting objects are possible future states.

as inputs instead of noisy sequence data. The data has to be i) segmented and ii) labeled to be parsed by existing grammar parsers. One naive solution is to first segment and label

the data using a detector and thus generating a label sentence. Then grammar parsers can be applied on top of it for parsing prediction. But this is apparently non-optimal, since the grammar rules are not considered in the detection/classification process. It may not even be possible to parse this label sentence, because the output from detectors are very often grammatically incorrect.

In this work, we design a grammar-based parsing algorithm that directly operates on sequence input data, which goes beyond the scope of symbolic string inputs. Specifically, we propose a generalized Earley parser to take *probabilistic sequence inputs* instead of deterministic symbolic inputs, based on the classic Earley parser [Ear70]. The algorithm finds the optimal segmentation and label sentence according to both a symbolic grammar and a classifier output of probabilities of labels for each frame as shown in Figure 3.2. Optimality here means maximizing the probability of the label sentence according to the classifier output while being *grammatically correct*.

The difficulty of achieving this optimality lies in the joint optimization of both the grammatical structure and the parsing likelihood of the output label sentence. For example, an expectation-maximization-type of algorithm will not work well since i) there is no guarantee for optimality, and ii) any grammatically incorrect sentence has a grammar prior of probability 0. The algorithm can easily get stuck in local minimums and fail to find the optimal solution that is grammatically correct.

The core idea of our algorithm is to directly and efficiently search for the optimal label sentence in the language defined by the grammar. The constraint of the search space ensures that the sentence is grammatically correct. Specifically, a heuristic search is performed on the prefix tree expanded according to the grammar, where the path from the root to a node represents a partial sentence (prefix). We search through the prefixes to find

the best sentence according to a heuristic. By carefully defining the heuristic as a prefix probability computed based on the classifier output, we can efficiently search through the tree to find the optimal label sentence.

The generalized Earley parser has four major **advantages**. **i)** The inference process highly integrates a high-level grammar with an underlying classifier; the grammar gives guidance for segmenting and labeling the sequence data and future predictions. **ii)** The only requirement for the underlying classifier is that the classifier should give probabilistic outputs. This made the algorithm widely applicable, since almost all statistical learning classifiers are probabilistic. **iii)** It generates semantically meaningful results (a grammar parse tree) for data sequence, and the process is highly explainable. **iv)** It is principled and generic, as it applies to most sequence data parsing and prediction problems (the data does not have to be videos).

We evaluate the proposed approach on three datasets of human activities in the computer vision domain. The first dataset CAD-120 [KGS13] consists of daily activities and most activity prediction methods are based on this dataset. Comparisons show that our method significantly outperforms state-of-the-art methods on future activity prediction. The second dataset Watch-n-Patch [WZS15] is designed for “action patching”, which includes daily activities that have action forgotten by people. Experiments on the second dataset show the robustness of our method on noisy data. The third dataset Breakfast [KAS14] consists of long videos of daily cooking activities. Results on this dataset show comparisons between our method and other structured modeling and language-inspired modeling methods. All experiments show that the generalized Earley parser performs well on both activity parsing and prediction tasks.

This work makes three major **contributions**.



- We design a parsing algorithm for symbolic context-free grammars that directly operates on sequence data. It can obtain the optimal segmentation and labels.
- We propose a prediction algorithm that naturally integrates with this parsing algorithm.
- We formulate an objective for future prediction for both grammar induction and classifier training. The generalized Earley parser serves as a concrete example for combining symbolic reasoning methods and connectionist approaches.

### 3.1 Related Work

**Activity recognition** refers to recognition of long-term and complicated activities from videos, whereas action recognition corresponds to short-term actions. They are two extensively studied topics in computer vision and we refer the readers to a survey [KF18] for a more comprehensive treatment. The main stream of work on activity recognition is to extend mid-level representations to high-level representations.

These extensions are designed in several different ways to model the complex activity structures. A number of methods have been proposed to model the high-level temporal structure of low-level features extracted from video [LLK07, LMS08, NCF10, GHS11, TFK12, JGR13]. Some other approaches represent complex activities as collections of attributes [LKS11, SC12, RRA12, FHX12]. Another important type of methods builds compositional/hierarchical models on actions [GSS09, WM10, SC12, SMD13, ZWY13, LZR15, HZG16]. [KGS13] proposed a model incorporating object affordances that detects and predicts human activities. [WZZ16] proposed a 4D human-object interaction model for event recognition. In some recent works, structural models are implicitly learned by

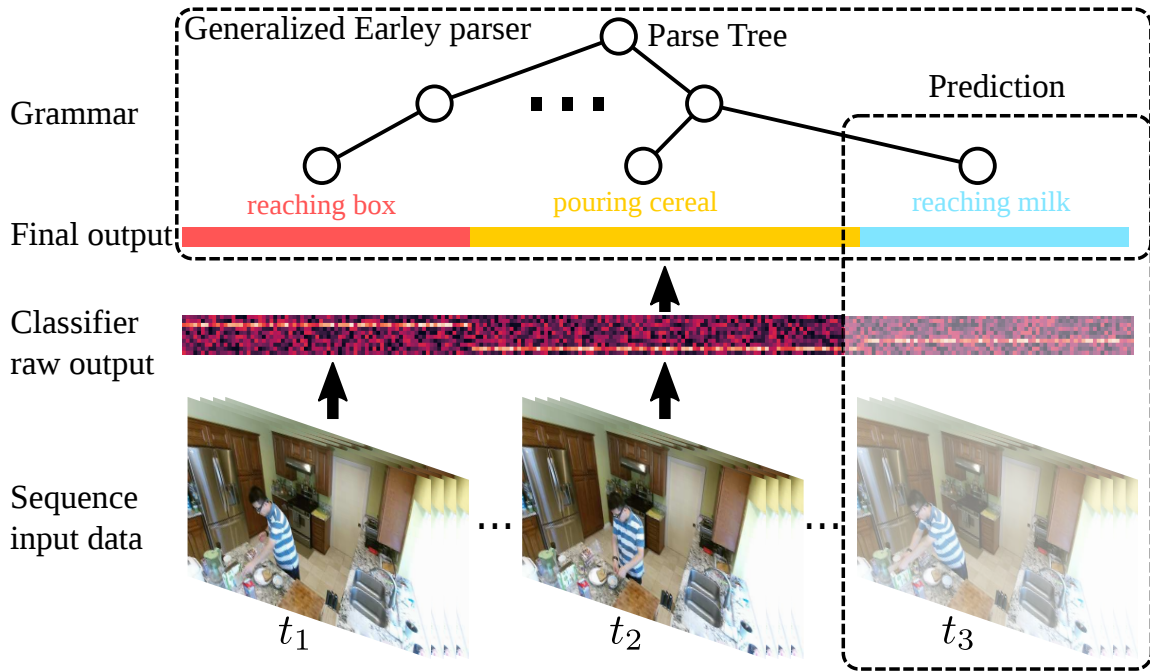


Figure 3.2: The input of the generalized Earley parser is a matrix of probabilities of each label for each frame, given by an arbitrary classifier. The parser segments and labels the sequence data into a label sentence in the language of a given grammar. Future predictions can be made based on the grammar.

neural networks [WFG16, CZ17, IM18, CSG18, ZTS19].

**Grammar models** falls into the category of compositional models for temporal structures. They have been applied to multiple other domains as well [RZ11, ZZ11, PZ15], here we focus methods for activity recognition. [IB00] proposed to first generate a discrete symbol stream from continuous low-level detectors, and then applied stochastic context-free parsing to incorporate prior knowledge of the temporal structure. [PJZ11] detects atomic actions and uses a stochastic context sensitive grammar for video parsing and intent prediction. [KAS14] models action units by hidden Markov models (HMMs), and mod-

els the higher-level action sequence by context-free grammars. [PR14] proposes segmental grammar for video parsing, which extends regular grammars to allow non-terminals to generate a segment of terminals of certain lengths. [VB14] generates a Bayes network, termed Sequential Interval Network (SIN), where the variable nodes correspond to the start and end times of component actions. This network then makes inference about start and end times for detected action primitives. [QHW17] proposed to integrate spatial-temporal attributes to terminal nodes of a context-free grammar. Based on Earley parser, an activity parsing and prediction algorithm is proposed. Overall, grammar-based methods have shown effectiveness on tasks that have compositional structures.

However, the above grammar-based algorithms (except [PR14]) take symbolic inputs like the traditional language parsers. They require the action primitives/atomic actions to be first detected, then a grammar is used for high-level parsing. This limits the applicability of these algorithms. Additionally, the parser does not provide guidance for either training the classifiers or segmenting the sequences. They also lack a good approach to handle grammatically incorrect label sentences. For example, [QHW17] finds in the training corpus the closest sentence to the recognized sentence and applies the language parser afterward. [PR14] ensures the results are grammatically correct, but it makes the grammar unnecessarily redundant (each possible segment length will make a new copy for each original grammar rule).

In our case, the proposed parsing algorithm takes sequence data of raw signals and a typical context-free grammar as input. It then generates the label sentence as well as the parse tree. All parsed label sentences are grammatically correct, and a learning objective is formulated for the classifier. Our work also serves as a bridge between connectionist and symbolic approaches, and it does not have any constraint on the low-level classifier.

**Future activity prediction** is a relatively new domain in computer vision. [ZRG09, YT10, Ryo11, KZB12, KKS12, WDA12, PSY13, WGH14, VOL14, LF14, WZZ16, HZG16, AGR16, XST18, EGX17, RK17, MHL17, QZ18, QWJ18, QJZ18] predict human trajectories/actions in various settings including complex indoor/outdoor scenes and crowded spaces. [LF14] builds a probabilistic suffix tree to model the Markov dependencies between action units and thus predict future events using a compositional model. [WGH14] predicted not only the future motions in the scene but also the visual appearances. In some recent work, [KS16] used an anticipatory temporal conditional random field to model the spatial-temporal relations through object affordances. [JZS16] proposed structural-RNN as a generic method to combine high-level spatial-temporal graphs and recurrent neural networks, which is a typical example that takes advantage of both graphical models and deep learning. [QHW17] proposed a spatial-temporal And-Or graph (ST-AOG) for activity prediction. In this work, we present a prediction algorithm based on the generalized Earley parser in which recognition and prediction are naturally and tightly integrated.

### 3.2 Representation: Probabilistic Context-Free Grammars

We model complex activities by grammars, where low-level actions are terminal symbols, *i.e.*, words in a language. In formal language theory, a *context-free grammar* (CFG) is a type of formal grammar, which contains a set of production rules that describe all possible sentences in a given formal language. In Chomsky Normal Form, a context-free grammar  $G$  is defined by a 4-tuple  $G = (V, \Sigma, R, \Gamma)$  where

- $V$  is a finite set of non-terminal symbols that can be replaced by/expanded to a sequence of symbols.

- $\Sigma$  is a finite set of terminal symbols that represent actual words in a language, which cannot be further expanded.
- $R$  is a finite set of production rules describing the replacement of symbols, typically of the form  $A \rightarrow BC$  or  $A \rightarrow \alpha$  for  $A, B, C \in V$  and  $\alpha \in \Sigma$ . A production rule replaces the left-hand side non-terminal symbol by the right-hand side expression. For example,  $A \rightarrow BC|\alpha$  means that  $A$  can be replaced by either  $BC$  or  $\alpha$ .
- $\Gamma \in V$  is the start symbol (root of the grammar).

*Probabilistic Context-Free Grammars* (PCFGs) augments CFGs by associating each production rule with a probability. Formally, it is defined by a 5-tuple  $G = (V, \Sigma, R, \Gamma, P)$ , where  $P$  is the set of probabilities on production rules. Figure 3.3 shows an example probabilistic temporal grammar of the activity “making cereal”.

Given a formal grammar, parsing is the process of analyzing a string of symbols, conforming to the production rules and generating a parse tree. A parse tree represents the syntactic structure of a string according to some context-free grammar. The root node of the tree is the grammar root. Other non-leaf nodes correspond to non-terminals in the grammar, expanded according to grammar production rules (could be expanding a combination or choosing alternatives). The leaf nodes are terminal symbols. All the leaf nodes together form a sentence in the language space described by the grammar.

### 3.3 Earley Parser

In this section, we briefly review the original Earley parser [Ear70], a classic grammar parsing algorithm with useful concepts that will be extended in the generalized Earley parser. An illustrative example is shown in Figure 3.5 to run through the algorithm. We

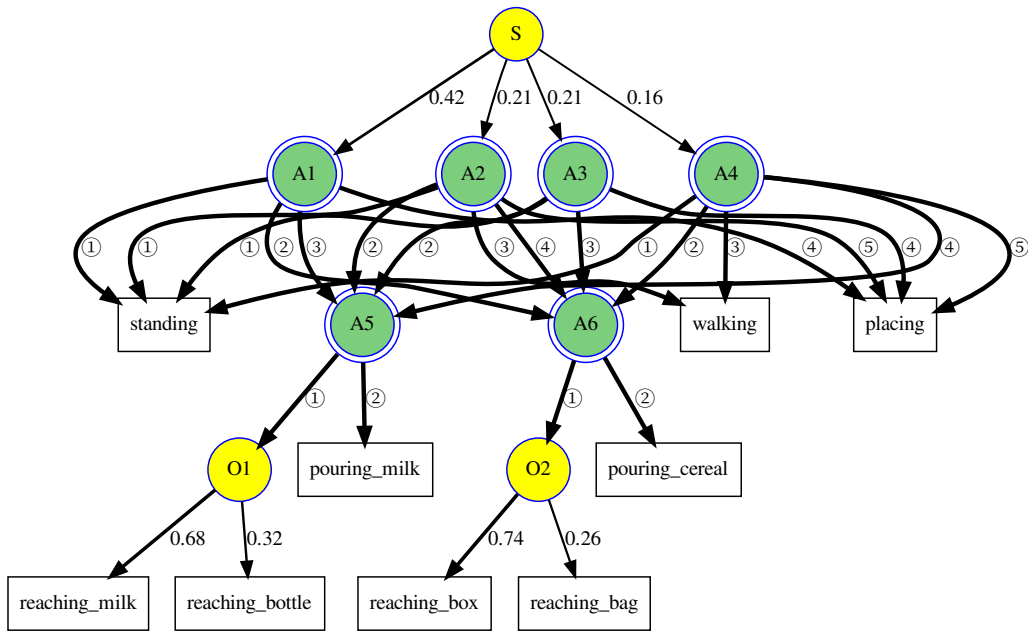


Figure 3.3: An example of a temporal grammar representing the activity “making cereal”. The green and yellow nodes are And-nodes (*i.e.*, production rules that represents combinations) and Or-nodes (*i.e.*, productions rules that represents alternatives), respectively. The numbers on branching edges of Or-nodes represent the branching probability. The circled numbers on edges of And-nodes indicates the temporal order of expansion.

then discuss how the original Earley parser can be applied to event parsing and its drawbacks.

Earley parser is an algorithm for parsing sentences of a given context-free language. In the following descriptions,  $\alpha$ ,  $\beta$ , and  $\gamma$  represent any string of terminals/nonterminals (including the empty string  $\epsilon$ ),  $A$  and  $B$  represent single nonterminals, and  $a$  represents a

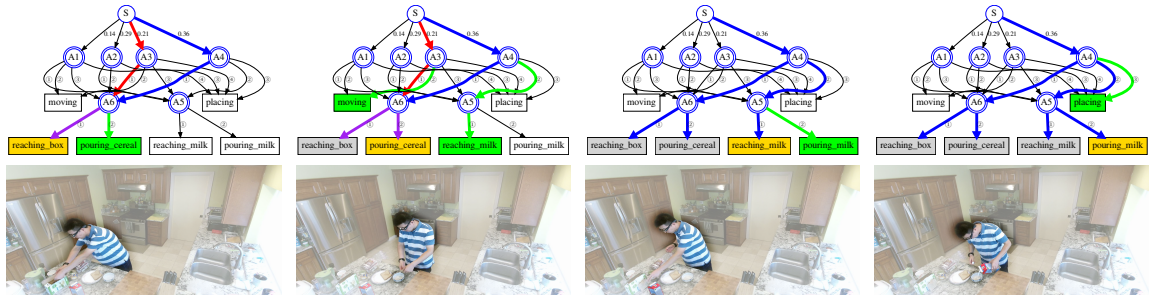


Figure 3.4: A simplified example illustrating the symbolic parsing and prediction process based on the Earley parser and detected actions. In the first two figures, the red edges and blue edges indicates two different parse graphs for the past observations. The purple edges indicate the overlap of the two possible explanations. The red parse graph is eliminated from the third figure. For the terminal nodes, yellow indicates the current observation and green indicates the next possible state(s).

terminal symbol. We adopt Earley’s dot notation: for production rule of form  $A \rightarrow \alpha\beta$ , the notation  $A \rightarrow \alpha \cdot \beta$  means  $\alpha$  has been parsed and  $\beta$  is expected.

Input position  $n$  is defined as the position after accepting the  $n$ th token, and input position 0 is the position prior to input. At each input position  $m$ , the parser generates a state set  $S(m)$ . Each state is a tuple  $(A \rightarrow \alpha \cdot \beta, i)$ , consisting of

- The production currently being matched ( $A \rightarrow \alpha\beta$ ).
- The dot: the current position in that production.
- The position  $i$  in the input at which the matching of this production began: the position of origin.

Seeded with  $S(0)$  containing only the top-level rule, the parser then repeatedly ex-

cuts three operations: prediction, scanning and completion:

- **Prediction:** for every state in  $S(m)$  of the form  $(A \rightarrow \alpha \cdot B\beta, i)$ , where  $i$  is the origin position as above, add  $(B \rightarrow \cdot\gamma, m)$  to  $S(m)$  for every production in the grammar with  $B$  on the left-hand side (*i.e.*,  $B \rightarrow \gamma$ ).
- **Scanning:** if  $a$  is the next symbol in the input stream, for every state in  $S(m)$  of the form  $(A \rightarrow \alpha \cdot a\beta, i)$ , add  $(A \rightarrow \alpha a \cdot \beta, i)$  to  $S(m + 1)$ .
- **Completion:** for every state in  $S(m)$  of the form  $(A \rightarrow \gamma \cdot, j)$ , find states in  $S(j)$  of the form  $(B \rightarrow \alpha \cdot A\beta, i)$  and add  $(B \rightarrow \alpha A \cdot \beta, i)$  to  $S(m)$ .

In this process, duplicate states are not added to the state set. These three operations are repeated until no new states can be added to the set. The Earley parser executes in  $O(n^2)$  for unambiguous grammars regarding the string length  $n$ , and  $O(n)$  for almost all  $LR(k)$  grammars.

The original Earley parser inspires a way to do event parsing and prediction from videos [QHW17]. The video can be first processed by a classifier to be segmented and labeled by actions, thus generating a label sentence. We can apply the Earley parser to parse the sentence to get a partial parse tree. The tree can be partial, since the sentence representing the activity might not be complete. Then action prediction can naturally be accomplished by looking that the open Earley states generated by the “prediction” operation. An example is shown in Figure 3.4.

However, this process can be problematic. Since the Earley parser takes symbols as input, it has little guidance to help the segmentation process that happens in the frame level. A more severe problem is that the segmentation and labeling process often generates sentences that are grammatically incorrect, *i.e.*, not in the language space described



$\Gamma \rightarrow R$	1.0	$N \rightarrow "0"$	0.3
$R \rightarrow N$	0.4	$N \rightarrow "1"$	0.7
$R \rightarrow N "+" N$	0.6		

(a) The input grammar. It contains a root symbol  $\Gamma$ , two non-terminal symbols  $R$  and  $N$ , three terminal symbols 0, 1 and +. The number to the right of each production rule is the corresponding probability.

state	rule	comment
$S(0)$		
(0)	$\Gamma \rightarrow \cdot R$	start rule
(1)	$R \rightarrow \cdot N$	predict: (0)
(2)	$R \rightarrow \cdot N + N$	predict: (0)
(3)	$N \rightarrow \cdot 0$	predict: (1)
(4)	$N \rightarrow \cdot 1$	predict: (1)
$S(1)$		
(0)	$N \rightarrow 0 \cdot$	scan: $S(0)(3)$
(1)	$R \rightarrow N \cdot$	complete: (0) and $S(0)(1)$
(2)	$R \rightarrow N \cdot + N$	complete: (0) and $S(0)(2)$
(3)	$\Gamma \rightarrow R \cdot$	complete: (1) and $S(0)(0)$
$S(2)$		
(0)	$R \rightarrow N + \cdot N$	scan: $S(1)(2)$
(1)	$N \rightarrow \cdot 0$	predict: (0)
(2)	$N \rightarrow \cdot 1$	predict: (0)
$S(3)$		
(0)	$N \rightarrow 1 \cdot$	scan: $S(2)(2)$
(1)	$R \rightarrow N + N \cdot$	complete: (0) and $S(2)(0)$
(2)	$\Gamma \rightarrow R \cdot$	complete: (1) and $S(0)(0)$

(b) A run-through for input string "0 + 1".

Figure 3.5: An illustrative example of the original Earley parser.

by the grammar. Thus the sentence cannot be parsed by the parser. In such cases, extra efforts are needed to modify the label sentence. One way to address that is sampling sentences from the language and find the closest alternatives [QHW17]. There also exist work in computational linguistics [Par11, Wag12, WF09] that address the problem of parsing grammatically incorrect sentences. However, these methods still operates in the symbolic space and does not provide much guidance for frame-level inference. To solve these problems, we propose the generalized Earley parser (detailed in Section 3.4) that directly takes sequence data as input and generates symbolic parse trees and predictions.

### 3.4 Generalized Earley Parser

In this section, we introduce the proposed generalized Earley parser. Instead of taking symbolic sentences as input, we aim to design an algorithm that can parse raw sequence data  $\mathbf{x}$  of length  $T$  (*e.g.*, videos or audios) into a sentence  $l$  of labels (*e.g.*, actions or words) of length  $|l| \leq T$ , where each label  $k \in \{0, 1, \dots, K\}$  corresponds to a segment of a sequence.

To achieve that, a classifier (*e.g.*, a neural network) is first applied to each sequence  $\mathbf{x}$  to get a  $T \times K$  probability matrix  $\mathbf{y}$  (*e.g.*, softmax activations of the neural network), with  $y_t^k$  representing the probability of frame  $t$  being labeled as  $k$ . The proposed generalized Earley parser takes  $\mathbf{y}$  as input and outputs the sentence  $l^*$  that best explains the data according to a grammar  $G$  of Chomsky normal form.

Now we discuss how we generalize the Earley parser to run on the output of a classifier, *i.e.*, the probability matrix. The core idea is to use the original Earley parser to help construct a prefix tree according to the grammar. The best solution is found by performing a heuristic search in this tree, where the heuristic is computed based on the probability

matrix given by the classifier.

Figure 3.6 and Figure 3.7c shows example prefix trees for the grammar in Figure 3.5. A prefix tree is composed of three types of nodes. 1) The root node of the “empty” symbol  $\epsilon$  represents the start of a sentence. 2) The non-leaf nodes (except the root node) correspond to terminal symbols in the grammar. A path from the root node to any non-leaf node represents a partial sentence (prefix). 3) The leaf nodes  $e$  are terminations that represent ends of sentences.

To find the best label sentence for a probability matrix, we perform a heuristic search in the prefix expanded according to the grammar: each node in the tree is associated with a probability, and the probabilities prioritize the nodes to be expanded in the prefix tree. The parser finds the best solution when it expands a termination node in the tree. It then returns the current prefix string as the best solution.

We compute two different heuristic probabilities for non-leaf nodes and leaf nodes. For non-leaf nodes, the heuristic is a prefix probability  $p(l...|x_{0:T})$ : the probability that the current path is the prefix for the label sentence. In other words, it measures the probability that  $\exists t \in [0, T]$ , the current path  $l$  is the label for frame  $x_{0:t}$ . For leaf nodes  $e$ , the heuristic  $p(l|x_{0:T})$  is a parsing probability: the probability that the current path  $l$  is the label sentence for  $x_{0:T}$ . The computation for  $p(l|x_{0:T})$  and  $p(l...|x_{0:T})$  are based on the input probability matrix  $\mathbf{y}$ . The formulation is derived in details in Section 3.4.2.

This heuristic search generalizes the Earley parser to parse the probability matrix. Specifically, the scan operation in the Earley parser essentially expands a new node in the grammar prefix tree. We organize the states into state sets by the partial sentence (prefix) each state represents. Instead of matching the sentence to the symbolic input, we now process state sets according to their prefix probabilities.

---

**Algorithm 2:** Generalized Earley Parser

---

**Input :** Grammar  $G$ , probability matrix  $y$ **Output:** Best label string  $l^*$ 

```
/* For brevity, we denote  $p(\cdot; y)$  as  $p(\cdot)$  */
/* Initialization */
1  $S(0, 0) = \{(\Gamma \rightarrow \cdot R, 0, 0, \epsilon, 1.0)\}$ 
2  $q = \text{priorityQueue}()$ 
3  $q.\text{push}(1.0, (0, 0, \epsilon, S(0, 0)))$ 
4 while  $(m, n, l^-, \text{currentSet}) = q.\text{pop}()$  do
5   for  $s = (r, i, j, l, p(l\dots)) \in \text{currentSet}$  do
6     if  $p(l) > p(l^*)$ :  $l^* = l$  then  $l^* = l$ 
7     if  $r$  is  $(A \rightarrow \alpha \cdot B\beta)$  then // predict
8       for each  $(B \rightarrow \Gamma)$  in  $G$  do
9          $r' = (B \rightarrow \cdot \Gamma)$ 
10         $s' = (r', m, n, l, p(l\dots))$ 
11         $S(m, n).\text{add}(s')$ 
12      end
13    end
14    else if  $r$  is  $(A \rightarrow \alpha \cdot a\beta)$  then // scan
15       $r' = (A \rightarrow \alpha a \cdot \beta)$ 
16       $m' = m + 1, n' = |S(m + 1)|$ 
17       $s' = (r', i, j, l + a, p((l + a)\dots))$ 
18       $S(m', n').\text{add}(s')$ 
19       $q.\text{push}(p((l + a)\dots), (m', n', S(m', n')))$ 
20    end
21    else if  $r$  is  $(B \rightarrow \Gamma \cdot)$  then // complete
22      for each  $((A \rightarrow \alpha \cdot B\beta), i', j')$  in  $S(i, j)$  do
23         $r' = (A \rightarrow \alpha B \cdot \beta)$ 
24         $s' = (r', i', j', l, p(l\dots))$ 
25         $S(m, n).\text{add}(s')$ 
26      end
27    end
28    if  $p(l^-) > p(l), \forall$  un-expanded  $l$  then return  $l^*$ 
29  end
30 end
31 return  $l^*$ 
```

---

### 3.4.1 Parsing Operations

We now describe the details of the parsing operations. Each scan operation will create a new state set  $S(m, n) \in S(m)$ , where  $m$  is the length of the scanned string,  $n$  is the total number of the terminals that have been scanned at position  $m$ . This can be thought of as creating a new node in the prefix tree, and  $S(m)$  is the set of all created nodes at level  $m$ . A priority queue  $q$  is kept for state sets for prefix search. Scan operations will push the newly created set into the queue with priority  $p(l...)$ , where  $l$  is the parsed string of the state being scanned. For brevity, we use  $p(l...)$  as a shorthand for  $p(l...|x_{0:t})$  when describing the algorithm.

Each state is a tuple  $(A \rightarrow \alpha \cdot \beta, i, j, l, p(l...))$  augmented from the original Earley parser by adding  $j, l, p(l...)$ . Here  $l$  is the parsed string of the state, and  $i, j$  are the indices of the set that this rule originated. The parser then repeatedly executes three operations: prediction, scanning, and completion modified from Earley parser:

- **Prediction:** for every state in  $S(m, n)$  of the form  $(A \rightarrow \alpha \cdot B\beta, i, j, l, p(l...))$ , add  $(B \rightarrow \cdot\Gamma, m, n, l, p(l...))$  to  $S(m, n)$  for every production in the grammar with  $B$  on the left-hand side.
- **Scanning:** for every state in  $S(m, n)$  of the form  $(A \rightarrow \alpha \cdot a\beta, i, j, l, p(l...))$ , append the new terminal  $a$  to  $l$  and compute the probability  $p((l + a)...)$ . Create a new set  $S(m + 1, n')$  where  $n'$  is the current size of  $S(m + 1)$ . Add  $(A \rightarrow \alpha a \cdot \beta, i, j, l + a, p((l + a)...) )$  to  $S(m + 1, n')$ . Push  $S(m + 1, n')$  into  $q$  with priority  $p((l + a)...)$ .
- **Completion:** for every state in  $S(m, n)$  of the form  $(A \rightarrow \Gamma \cdot, i, j, l, p(l...))$ , find states in  $S(i, j)$  of the form  $(B \rightarrow \alpha \cdot A\beta, i', j', l', p(l'...))$  and add  $(B \rightarrow \alpha A \cdot \beta, i', j', l, p(l...))$  to  $S(m, n)$ .

This parsing process is efficient since we do not need to search through the entire tree. As shown in Figure 3.7 and Algorithm 2, the best label sentence  $l$  is returned when the probability of termination is larger than any other prefix probabilities. As long as the parsing and prefix probabilities are computed correctly, it is guaranteed to return the best solution.

The original Earley parser is a special case of the generalized Earley parser. Intuitively, for any input sentence to Earley parser, we can always convert it to one-hot vectors and apply the proposed algorithm. On the other hand, the original Earley parser cannot be applied to segmented one-hot vectors since the labels are often grammatically incorrect. Hence we have the following proposition.

**Proposition 1.** *Earley parser is a special case of the generalized Earley parser.*

*Proof.* Let  $L(G)$  denote the language of grammar  $G$ ,  $h(\cdot)$  denote a one-to-one mapping from a label to a one-hot vector.  $L(G)$  is the input space for Earley parser.  $\forall l \in L(G)$ , the generalized Earley parser accepts  $h(l)$  as input. Therefore the proposition follows.  $\square$

Here we emphasize two important distinctions of our method to traditional probabilistic parsers with prefix probabilities. i) In traditional parsers, the prefix probability is the probability of a string being a prefix of some strings generated by a grammar (top-down grammar prior). In our case, the parser computes the bottom-up data likelihood. We further extend this to a posterior that integrates these two in Section 3.4.3. ii) Traditional parsers only maintain a parse tree, while our algorithm maintains both a parse tree and a prefix tree. The introduction of the prefix tree into the parser enables us to efficiently search in the grammar according to a desired heuristic.

Table 3.1: Summary of notations used for parsing & prefix probability formulation.

---

$x_{0:t}$	input frames from time 0 to $t$
$l$	a label sentence
$k$	the last label in $l$
$l^-$	the label sentence obtained by removing the last label $k$ from the label sentence $l$
$y_t^k$	the probability for frame $t$ to be labelled as $k$
$p(l x_{0:t})$	parsing probability of $l$ for $x_{0:t}$
$p(l... x_{0:t})$	prefix probability of $l$ for $x_{0:t}$

---

### 3.4.2 Parsing & Prefix Probability Formulation

The parsing probability  $p(l|x_{0:T})$  is computed in a dynamic programming fashion. Let  $k$  be the last label in  $l$ . For  $t = 0$ , the probability is initialized by:

$$p(l|x_0) = \begin{cases} y_0^k & l \text{ contains only } k, \\ 0 & \text{otherwise.} \end{cases} \quad (3.1)$$

Let  $l^-$  be the label sentence obtained by removing the last label  $k$  from the label sentence  $l$ . For  $t > 0$ , the last frame  $t$  must be classified as  $k$ . The previous frames can be labeled as either  $l$  or  $l^-$ . Then we have:

$$p(l|x_{0:t}) = y_t^k(p(l|x_{0:t-1}) + p(l^-|x_{0:t-1})), \quad (3.2)$$

where  $p(l|x_{0:t-1})$  corresponds to the possibility that frame  $t - 1$  is also labelled as  $k$ , and  $p(l^-|x_{0:t-1})$  accounts for the possibility that label  $k$  starts from frame  $t$ . It is worth mentioning that when  $y_t^k$  is wrongly given as 0, the dynamic programming process will have trouble correcting the mistake. Even if  $p(l^-|x_{0:t-1})$  is high, the probability  $p(l|x_{0:t})$

will be 0. Fortunately, since the softmax function is usually adopted to compute  $y$ ,  $y_t^k$  will not be 0 and the solution will be kept for further consideration.

Then we compute the prefix probability  $p(l...|x_{0:T})$  based on  $p(l^-|x_{0:t})$ . For  $l$  to be the prefix, the transition from  $l^-$  to  $l$  can happen at any frame  $t \in \{0, \dots, T\}$ . Once the label  $k$  is observed (the transition happens),  $l$  becomes the prefix and the rest frames can be labeled arbitrarily. Hence the probability of  $l$  being the prefix is:

$$p(l...|x_{0:T}) = p(l|x_0) + \sum_{t=1}^T y_t^k p(l^-|x_{0:t-1}). \quad (3.3)$$

In practice, the probability  $p(l|x_{0:t})$  decreases exponentially as  $t$  increases and will soon lead to numeric underflow. To avoid this, the probabilities need to be computed in log space:

$$\begin{aligned} \log p(l|x_{0:t}) &= \log(y_t^k) + d + \\ &\log(\exp(\log p(l|x_{0:t-1}) - d) + \exp(\log p(l^-|x_{0:t-1}) - d)), \end{aligned} \quad (3.4)$$

where  $d$  is a constant number and is usually set to be  $\max(\log(y_t^k), \log p(l|x_{0:t-1}), \log p(l^-|x_{0:t-1}))$ .

The time complexity of computing the probabilities is  $O(T)$  for each sentence  $l$  because  $p(l^-|x_{0:t})$  are cached. The worst case complexity of the entire parsing is  $O(T|G|)$ .

### 3.4.3 Incorporating Grammar Prior

For PCFGs, we can integrate the grammar prior of the sentence  $l$  into the above formulation to obtain a posterior parsing probability. The basic idea is that we can compute a “transition probability” of appending a new symbol to the current sentence. This probability will be multiplied to the parsing probability when we append a new symbol.

To compute a transition probability  $p(k|l^-, G)$ , we can first compute the prefix probabilities  $p(l^-|G)$  and  $p(l...|G)$  according to the grammar. Then the transition probability is



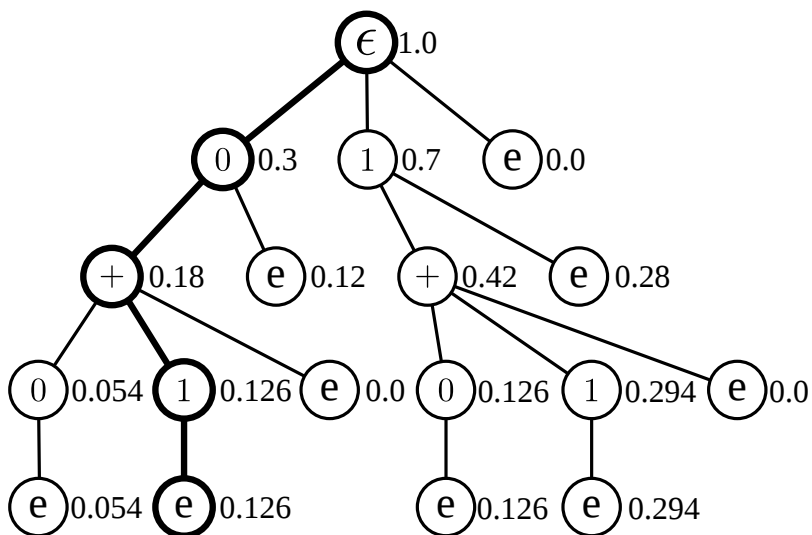


Figure 3.6: Grammar prefix probabilities computed according to the grammar in Figure 3.5. The numbers next to the tree nodes are prefix probabilities according to the grammar. The transition probabilities can be easily computed from this tree, *e.g.*,  $p("1"|"0 + ", G) = p("0 + 1" \dots |G) / p("0 + " \dots |G) = 0.126 / 0.18 = 0.7$ .

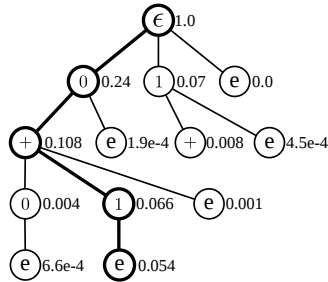
$\Gamma \rightarrow R$	1.0
$R \rightarrow N$	0.4
$R \rightarrow N \text{“+”} N$	0.6
$N \rightarrow \text{“0”}$	0.3
$N \rightarrow \text{“1”}$	0.7

frame	“0”	“1”	“+”
0	0.8	0.1	0.1
1	0.8	0.1	0.1
2	0.1	0.1	0.8
3	0.1	0.8	0.1
4	0.1	0.8	0.1

(a) Left: input grammar. Right: input probability matrix.

Frame	$\epsilon$	0	1	0+	1+	0+0	0+1
0	0.000	0.240	0.070	0.000	0.000	0.000	0.000
1	0.000	0.192	0.007	0.014	0.004	0.000	0.000
2	0.000	0.019	7.0e-04	0.104	0.007	4.3e-04	0.001
3	0.000	0.002	5.6e-04	0.012	7.1e-04	0.003	0.059
4	0.000	1.9e-04	4.5e-04	0.001	1.1e-04	6.6e-04	0.054
prefix	1.000	0.240	0.070	0.108	0.008	0.004	0.066

(b) Cached probabilities



(c) Prefix tree

state #	rule	$\mu$	$\nu$	prefix	comment
$S(0, 0) : l = \epsilon, p(l G) = 1.000, p(l x, G) = 0.000, p(l_{...} x, G) = 1.000$					
(0)	$\Gamma \rightarrow \cdot R$	1.000	1.000	“ $\epsilon$ ”	start rule
(1)	$R \rightarrow \cdot N$	0.400	0.400	“ $\epsilon$ ”	predict: (0)
(2)	$R \rightarrow \cdot N + N$	0.600	0.600	“ $\epsilon$ ”	predict: (0)
(3)	$N \rightarrow \cdot 0$	0.300	0.300	“ $\epsilon$ ”	predict: (1),(2)
(4)	$N \rightarrow \cdot 1$	0.700	0.700	“ $\epsilon$ ”	predict: (1),(2)

$S(1, 0) : l = \text{“0”}, p(l G) = 0.300, p(l x, G) = 1.9e - 04, p(l_{...} x, G) = 0.240$					
(0)	$N \rightarrow 0 \cdot$	0.300	0.300	“0”	scan: S(0, 0)(3)
(1)	$R \rightarrow N \cdot$	0.120	0.120	“0”	complete: (0) and S(0, 0)(1)
(2)	$R \rightarrow N \cdot + N$	0.180	0.180	“0”	complete: (0) and S(0, 0)(2)
(3)	$\Gamma \rightarrow R \cdot$	0.120	0.120	“0”	complete: (1) and S(0, 0)(0)

$S(1, 1) : l = \text{“1”}, p(l G) = 0.700, p(l x, G) = 4.5e - 04, p(l_{...} x, G) = 0.070$					
(0)	$N \rightarrow 1 \cdot$	0.700	0.700	“1”	scan: S(0, 0)(4)
(1)	$R \rightarrow N \cdot$	0.280	0.280	“1”	complete: (0) and S(0, 0)(1)
(2)	$R \rightarrow N \cdot + N$	0.420	0.420	“1”	complete: (0) and S(0, 0)(2)
(3)	$\Gamma \rightarrow R \cdot$	0.280	0.280	“1”	complete: (1) and S(0, 0)(0)

$S(2, 0) : l = \text{“0+”}, p(l G) = 0.180, p(l x, G) = 0.001, p(l_{...} x, G) = 0.108$					
(0)	$R \rightarrow N + \cdot N$	0.180	0.180	“0+”	scan: S(1, 0)(2)
(1)	$N \rightarrow \cdot 0$	0.054	0.300	“0+”	predict: (0)
(2)	$N \rightarrow \cdot 1$	0.126	0.700	“0+”	predict: (0)

$S(2, 1) : l = \text{“1+”}, p(l G) = 0.420, p(l x, G) = 1.1e - 04, p(l_{...} x, G) = 0.008$					
(0)	$R \rightarrow N + \cdot N$	0.420	0.420	“1+”	scan: S(1, 1)(2)

$S(3, 0) : l = \text{“0+0”}, p(l G) = 0.054, p(l x, G) = 6.6e - 04, p(l_{...} x, G) = 0.004$					
(0)	$N \rightarrow 0 \cdot$	0.054	0.300	“0+0”	scan: S(2, 0)(1)

$S(3, 1) : l = \text{“0+1”}, p(l G) = 0.126, p(l x, G) = 0.054, p(l_{...} x, G) = 0.066$					
(0)	$N \rightarrow 1 \cdot$	0.126	0.700	“0+1”	scan: S(2, 0)(2)
(1)	$R \rightarrow N + N \cdot$	0.126	0.126	“0+1”	complete: (0) and S(2, 0)(0)
(2)	$\Gamma \rightarrow R \cdot$	0.126	0.126	“0+1”	complete: (1) and S(0, 0)(0)

Final output:  $l^* = \text{“0+1”}$  with probability 0.054

(d) A run-through of the algorithm

Figure 3.7: An example of the generalized Earley parser. A classifier is applied to a 5-frame signal and outputs a probability matrix (a) as the input to our algorithm. The proposed algorithm expands a grammar prefix tree (c), where “e” represents termination. It finally outputs the best label “0+1” with probability 0.054. The probabilities of children nodes do not sum to 1 since the grammatically incorrect nodes are eliminated.

given by:

$$p(k|l^-, G) = \frac{p(l...|G)}{p(l^-|G)}. \quad (3.5)$$

An example is shown in Figure 3.6 for a better intuition. The computation of this grammar prefix probability will be detailed in subsection 3.4.3.1. There are two important remarks to make here. 1) This prior prefix probability is different from the prefix probability based on the likelihood. The prior is the probability that a string is the prefix of a sentence in the language defined by the grammar, without seeing any data; the likelihood is the probability that a string is the prefix of a video's label. 2) This grammar-based transition probability is non-Markovian, since the new symbol is conditioned on the entire history string that has a variable length.

Now, incorporating the grammar transition probability, for  $t = 0$ , the probability is initialized by:

$$p(l|x_0, G) \propto \begin{cases} p(k|\epsilon, G) y_0^k & l \text{ contains only } k, \\ 0 & \text{otherwise,} \end{cases} \quad (3.6)$$

where  $p(k|\epsilon, G)$  is the probability of appending  $k$  to the empty string  $\epsilon$ , which is equivalent to  $p(k...|G)$  or  $p(l...|G)$ . Notice that the equal sign is replaced by  $\propto$  since the right hand side should be normalized by the prior  $p(x_0)$  to get the correct posterior.

Whenever we append a new symbol to our sentence, we multiply the probability by the transition probability. Hence for  $t > 0$  we have:

$$p(l|x_{0:t}, G) \propto y_t^k (p(l|x_{0:t-1}, G) + p(k|l^-, G)p(l^-|x_{0:t-1}, G)). \quad (3.7)$$

Comparing to Eq. 3.2, we multiply the second term by  $p(k|l^-, G)$  to account for the transition to symbol  $k$ .

Finally the posterior probability of  $l$  being the prefix of the label sentence for data  $x$  is:

$$p(l\dots|x_{0:T}, G) = p(l|x_0, G) + \sum_{t=1}^T p(k|l^-, G) y_t^k p(l^-|x_{0:t-1}, G). \quad (3.8)$$

### 3.4.3.1 Grammar Prefix Probability

The derivation of the grammar prefix probability with Earley parser [Sto95] can be achieved by augmenting the Earley states with two additional variables: forward probability  $\mu$  and inner probability  $\nu$ . For a state  $S$ , the forward probability  $\mu$  is the probability of all parses that lead to  $S$ , the inner probability  $\nu$  is the probability of all parses expanded from  $S$ . In other words,  $\mu$  is the probability of the prefix before  $S$ , and  $\nu$  is the probability of the partial string parsed by  $S$ . Assuming that the grammar is not left-recursive, these two terms can be computed efficiently during the Earley parsing process:

- **Prediction.** For  $(A \rightarrow \alpha \cdot B\beta, i, [\mu, \nu]) \Rightarrow (B \rightarrow \cdot\gamma, m, [\mu', \nu'])$ , the new probabilities are given by

$$\mu'_+ = \alpha \cdot P(B \rightarrow \gamma), \nu' = P(B \rightarrow \gamma).$$

- **Scanning.** For  $(A \rightarrow \alpha \cdot a\beta, i, [\mu, \nu]) \Rightarrow (A \rightarrow \alpha a \cdot \beta, i, [\mu', \nu'])$ , we have

$$\mu' = \mu, \nu' = \nu.$$

- **Completion.** For  $(A \rightarrow \gamma\cdot, j, [\mu'', \nu''])$  and  $(B \rightarrow \alpha \cdot A\beta, i, [\mu, \nu]) \Rightarrow (B \rightarrow \alpha A \cdot \beta, i, [\mu', \nu'])$ , we have

$$\mu'_+ = \mu \cdot \nu'', \nu' = \nu \cdot \nu''.$$

Finally, the prefix probability of a string is given by the sum of forward probabilities over all scanned states. A run-through example of the generalized Earley parser with grammar prior is shown in Figure 3.7.

### 3.4.4 Segmentation and Labeling

The generalized Earley parser gives us the best grammatically correct label sentence  $l$  to explain the sequence data, which takes all possible segmentations into consideration. Therefore the probability  $p(l|x_{0:T})$  is the summation of probabilities of all possible segmentations. Let  $p(l|y_{0:e})$  be the probability of the best segmentation based on the classifier output  $y$  for sentence  $l$ . We perform a maximization over different segmentations by dynamic programming to find the best segmentation:

$$p(l|y_{0:e}) = \max_{b < e} p(l^- | y_{0:b}) \prod_{t=b}^e y_t^k, \quad (3.9)$$

where  $e$  is the time frame that  $l$  ends and  $b$  is the time frame that  $l^-$  ends. The best segmentation can be obtained by backtracing the above probability. Similar to the previous probabilities, this probability needs to be computed in log space as well. The time complexity of the segmentation and labeling is  $O(T^2)$ .

### 3.4.5 Future Label Prediction

We consider two types of future label predictions: 1) segment-wise prediction that predicts the next segment label at each time  $t$ , and 2) frame-wise prediction that predicts the labels for the future  $\delta t$  frames.

#### 3.4.5.1 Segment-wise Prediction

Given the parsing result  $l$ , we can make grammar-based top-down predictions for the next label  $z$  to be observed. The predictions are naturally obtained by the predict operation in the generalized Earley parser, and it is inherently an online prediction algorithm.

To predict the next possible symbols at current position  $(m, n)$ , we search through the states  $S(m, n)$  of the form  $(X \rightarrow \alpha \cdot z\beta, i, j, l, p(l\dots))$ , where the first symbol  $z$  after the current position is a terminal node. The predictions  $\Sigma$  are then given by the set of all possible  $z$ :

$$\Sigma = \{z : \exists s \in S(m, n), s = (X \rightarrow \alpha \cdot z\beta, i, j, l, p(l\dots))\}. \quad (3.10)$$

The probability of each prediction is then given by the parsing likelihood of the sentence constructed by appending the predicted label  $z$  to the current sentence  $l$ . Assuming that the best prediction corresponds to the best parsing result, the goal is to find the best prediction  $z^*$  that maximizes the following conditional probability as parsing likelihood:

$$z^* = \operatorname{argmax}_{z \in \Sigma} p(z, l|G). \quad (3.11)$$

For a grammatically complete sentence  $u$ , the parsing likelihood is simply the Viterbi likelihood [Vit67] given by the probabilistic context-free grammar. For an incomplete sentence  $l$  of length  $|l|$ , the parsing likelihood is given by the grammar prefix probability. Hence they are both the forward probability computed in subsection 3.4.3.1. We can also integrate top-down and bottom-up inference for segment-wise prediction. A classifier can be trained to predict the next segment label, and it can be combined with the grammar prior probability for better predictions.

### 3.4.5.2 Frame-wise Prediction

Frame-wise future label prediction is rather straightforward using the generalized Earley parser. We first run activity detection on the input videos, and we sample the duration of the current action. Based on the segment-wise prediction, we can further sample the duration for future segments, thus obtaining frame-wise future predictions according to the prediction range.

Another way to do frame-wise prediction is treating the problem as a parsing problem. Besides the classifier for detection, we train another classifier for prediction, *i.e.*, for each frame  $x_t$  the classifier predicts the label for frame  $x_{t+\delta t}$ . By concatenating the output from detection classifier and the prediction classifier, we can obtain a  $n \times (T + \delta t)$  probability matrix. Then the prediction can be obtained by running the generalized Earley parser on the concatenated probability matrix.

### 3.4.5.3 Maximum Likelihood Estimation for Prediction

We are interested in finding the best grammar and classifier that give us the most accurate segment-wise predictions based on the generalized Earley parser. Let  $G$  be the grammar,  $f$  be the classifier, and  $D$  be the set of training examples. The training set consists of pairs of complete or partial data sequence  $\mathbf{x}$  and the corresponding label sequence for all the frames in  $\mathbf{x}$ . By merging consecutive labels that are the same, we can obtain partial label sentences  $l$  and predicted labels  $z$ . Hence we have  $D = \{(\mathbf{x}, l, z)\}$ . The best grammar  $G^*$  and the best classifier  $f^*$  together minimizes the prediction loss:

$$G^*, f^* = \underset{G, f}{\operatorname{argmin}} \mathcal{L}_{pred}(G, f), \quad (3.12)$$

where the prediction loss is given by the negative log likelihood of the predictions over the entire training set:

$$\begin{aligned} \mathcal{L}_{pred}(G, f) &= - \sum_{(\mathbf{x}, l, z) \in D} \log(p(z|\mathbf{x})) \\ &= - \sum_{(\mathbf{x}, l, z) \in D} \left( \underbrace{\log(p(z|l, G))}_{\text{grammar}} + \underbrace{\log(p(l|\mathbf{x}))}_{\text{classifier}} \right). \end{aligned} \quad (3.13)$$

Given the intermediate variable  $l$ , the loss is decomposed into two parts that correspond to the induced grammar and the trained classifier, respectively. Let  $u \in \{l\}$  be the complete

label sentences in the training set (*i.e.*, the label sentence for a complete sequence  $\mathbf{x}$ ). The best grammar maximizes the following probability:

$$\prod_{(z,l) \in D} p(z|l, G) = \prod_{(z,l) \in D} \frac{p(z, l|G)}{p(l|G)} = \prod_{u \in D} p(u|G), \quad (3.14)$$

where denominators  $p(l|G)$  are canceled by the previous numerator  $p(z, l|G)$ , and only the likelihood of the complete sentences remain. Therefore inducing the best grammar that gives us the most accurate future prediction is equivalent to the maximum likelihood estimation (MLE) of the grammar for complete sentences in the dataset. This finding lets us to turn the problem (induce the grammar that gives the best future prediction) into a standard grammar induction problem, which can be solved by existing algorithms, *e.g.*, [SHR05] and [TPZ13].

The best classifier minimizes the second term of Eq. 3.13:

$$\begin{aligned} f^* &= \operatorname{argmin}_f - \sum_{(\mathbf{x}, l, z) \in D} \log(p(l|\mathbf{x})) \\ &\approx \operatorname{argmin}_f - \sum_{(\mathbf{x}, \mathbf{y}) \in D} \sum_k y_k \log(\hat{y}_k), \end{aligned} \quad (3.15)$$

where  $p(l|\mathbf{x})$  can be maximized by the CTC loss [GFG06]. In practice, it can be substituted by the commonly adopted cross entropy loss for efficiency. Therefore we can directly apply generalized Earley parser to outputs of general detectors/classifiers for parsing and prediction.

### 3.5 Human Activity Parsing and Prediction

We evaluate our method on the task of human activity detection and prediction. We present and discuss our experiment results on three datasets, CAD-120 [KGS13], Watch-n-Patch [WZS15], and Breakfast [KAS14], for comparisons with state-of-the-art methods



and evaluation of the robustness of our approach. CAD-120 is the dataset that most existing prediction algorithms are evaluated on. It contains videos of daily activities that are long sequences of sub-activities. Watch-n-Patch is a daily activity dataset that features forgotten actions. Breakfast is a dataset that contains long videos of daily cooking activities. Results show that our method performs well on both activity detection and activity prediction.

### 3.5.1 Grammar Induction

In both experiments, we used a modified version of the ADIOS (automatic distillation of structure) [SHR05] grammar induction algorithm to learn the event grammar. The algorithm learns the production rules by generating significant patterns and equivalent classes. The significant patterns are selected according to a context-sensitive criterion defined regarding local flow quantities in the graph: two probabilities are defined over a search path. One is the right-moving ratio of fan-through (through-going flux of path) to fan-in (incoming flux of paths). The other one, similarly, is the left-going ratio of fan-through to fan-in. The criterion is described in detail in [SHR05].

The algorithm starts by loading the corpus of activity onto a graph whose vertices are sub-activities, augmented by two special symbols, begin and end. Each event sample is represented by a separate path over the graph. Then it generates candidate patterns by traversing a different search path. At each iteration, it tests the statistical significance of each subpath to find significant patterns. The algorithm then finds the equivalent classes that are interchangeable. At the end of the iteration, the significant pattern is added to the graph as a new node, replacing the subpaths it subsumes. We favor shorter patterns in our implementation.

### 3.5.2 Experiment on CAD-120 Dataset

**Dataset** The CAD-120 dataset [KGS13] is a standard dataset for human activity prediction. It contains 120 RGB-D videos of four different subjects performing 10 high-level activities, where each high-level activity was performed three times with different objects. It includes a total of 61,585 total video frames. Each video is a sequence of sub-activities involving 10 different sub-activity labels. The videos vary from subject to subject regarding the lengths and orders of the sub-activities as well as the way they executed the task.

**Evaluation metrics** We use the following metrics to evaluate and compare the algorithms. 1) Frame-wise detection accuracy of sub-activity labels for all frames. 2) Frame-wise (future 3s) online prediction accuracy. We compute the frame-wise accuracy of prediction of the sub-activity labels of the future 3s (using the frame rate of 14Hz as reported in [KGS13]). The predictions are made online at each frame  $t$ , *i.e.*, the algorithms only sees frame 0 to  $t$  and predicts the labels of frame  $t + 1$  to  $t + \delta t$ . 3) Segment-wise online prediction accuracy. At each frame  $t$ , the algorithm predicts the sub-activity label of the next video segment.

We consider the overall micro accuracy (P/R), macro precision, macro recall and macro F1 score for all evaluation metrics. Micro accuracy is the percentage of correctly classified labels. Macro precision and recall are the average of precision and recall respectively for all classes.

**Comparative methods** We compare our method with four state-of-the-art methods for activity prediction:

1. KGS [KGS13]: a Markov random field model where the nodes represent objects and sub-activities, and the edges represent the spatial-temporal relationships. Future frames

are predicted based on the transition probabilities given the inferred label of the last frame.

2. Anticipatory temporal CRF (ATCRF) [KS16]: an anticipatory temporal conditional random field that models the spatial-temporal relations through object affordances. Future frames are predicting by sampling a spatial-temporal graph.

3. ST-AOG [QHW17]: a spatial-temporal And-Or graph (ST-AOG) that uses a symbolic context-free grammar to model activity sequences. This sets up a comparison between our proposed method and methods that use traditional probabilistic parsers. Since traditional parsers operate on symbolic data, extra efforts need to be done first to extract symbols from sequence data. In this comparative method, the videos are first segmented and labeled by classifiers; the predictions are then made by the original Earley parser.

4. Bidirectional LSTM (Bi-LSTM): a two-layer Bi-LSTM with a hidden size of 256. For the detection task, the output for each frame input is the sub-activity label. For the future 3s prediction, the LSTM is trained to output the label for frame  $t + 3s$  for an input frame at time  $t$ . For future segment prediction, it outputs the label of the next segment for an input frame. All three tasks use the same training schemes.

5. Bi-LSTM + generalized Earley parser: the proposed generalized Earley parser applied to the classifier output of the above detection Bi-LSTM. The predictions for the next segments are made according to Section 3.4.5. The lengths of unobserved segments are sampled from a log-normal distribution for the future 3s prediction.

**Feature extraction** All methods in the experiment use the same publicly available features from KGS [KGS13]. These features include the human skeleton features and human-object interaction features for each frame. The human skeleton features are location and distance features (relative to the subjects head location) for all upper-skeleton joints of a

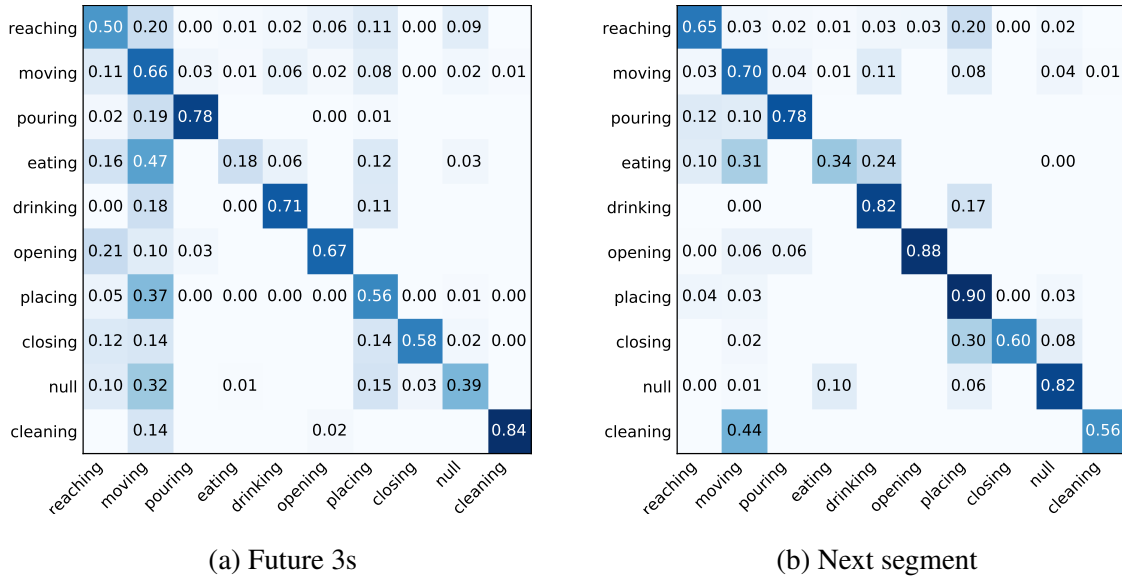


Figure 3.8: Confusion matrices for prediction results on CAD-120.

subject. The human-object features are spatial-temporal, containing the distances between object centroids and skeleton joint locations as well as the temporal changes.

**Experiment results** We follow the convention in KGS [KGS13] to train on three subjects and test on a new subject with a 4-fold validation. The results for the three evaluation metrics are summarized in Table 3.2, Table 3.3 and Table 3.4, respectively. Our method outperforms the comparative methods on all three tasks. Specifically, the generalized Earley parser on top of a Bi-LSTM performs better than ST-AOG, while ST-AOG outperforms the Bi-LSTM. More discussions are highlighted in Section 3.5.5.

### 3.5.3 Experiment on Watch-n-Patch Dataset

**Dataset** Watch-n-Patch [WZS15] is an RGB-D dataset that features forgotten actions. For example, a subject might fetch milk from a fridge, pour milk, and leave. The typical action

Table 3.2: Detection results on CAD-120.

Method	Micro	Macro		
	P/R	Prec.	Recall	F1-score
KGS [KGS13]	68.2	71.1	62.2	66.4
ATCRF [KS16]	70.3	74.8	66.2	70.2
Bi-LSTM	76.2	78.5	74.5	74.9
ST-AOG + Earley [QHW17]	76.5	77.0	75.2	76.1
Bi-LSTM + Generalized Earley	<b>79.4</b>	<b>87.4</b>	<b>77.0</b>	<b>79.7</b>

Table 3.3: Future 3s prediction results on CAD-120.

Method	Micro	Macro		
	P/R	Prec.	Recall	F1-score
KGS [KGS13]	28.6	–	–	11.1
LSTM	49.4	40.9	37.3	37.8
ATCRF [KS16]	49.6	–	–	40.6
ST-AOG + Earley [QHW17]	55.2	<b>56.5</b>	<b>56.6</b>	<b>56.6</b>
Bi-LSTM + Generalized Earley	<b>57.1</b>	52.3	54.1	52.3

“putting the milk back into the fridge” is forgotten. The dataset contains 458 videos with a total length of about 230 minutes, in which people forgot actions in 222 videos. Each video in the dataset contains 2-7 actions interacted with different objects. 7 subjects are asked to perform daily activities in 8 offices and 5 kitchens with complex backgrounds. It consists of 21 types of fully annotated actions (10 in the office, 11 in the kitchen) interacted with 23 types of objects.

**Feature extraction** We extract the same features as described in [WZS15] for all

Table 3.4: Segment prediction results on CAD-120.

Method	Micro	Macro		
	P/R	Prec.	Recall	F1-score
LSTM	52.8	52.5	52.8	47.6
ST-AOG + Earley [QHW17]	54.3	61.4	39.2	45.4
Bi-LSTM + Generalized Earley	<b>70.6</b>	<b>72.1</b>	<b>70.6</b>	<b>70.1</b>

methods. Similar to the previous experiment, the features are composed of skeleton features and human-object interaction features extracted from RGB-D images. The skeleton features include angles between connected parts, the change of joint positions and angles from previous frames. Each frame is segmented into super-pixels, and foreground masks are detected. We extract features from the image segments with more than 50% in the foreground mask and within a distance to the human hand joints in both 3D points and 2D pixels. Six kernel descriptors [WLS14] are extracted from these image segments: gradient, color, local binary pattern, depth gradient, spin, surface normals, and KPCA/self-similarity.

**Experiment results** We use the same evaluation metrics as the previous experiment and compare our method to ST-AOG [QHW17] and Bi-LSTM. We use the train/test split in [WZS15]. The results for the three evaluation metrics are summarized in Table 3.5, Table 3.6 and Table 3.7, respectively. Our method slightly improves the detection results over the Bi-LSTM outputs, and outperforms the other methods on both prediction tasks. In general, the algorithms make better predictions on CAD-120, since Watch-n-Patch features forgotten actions and the behaviors are more unpredictable. More details are discussed in Section 3.5.5.

Table 3.5: Detection results on Watch-n-Patch.

Method	Micro	Macro		
	P/R	Prec.	Recall	F1-score
ST-AOG + Earley [QHW17]	79.3	71.5	73.5	71.9
Bi-LSTM	84.0	79.7	82.2	80.3
Bi-LSTM + Generalized Earley	<b>84.8</b>	<b>80.7</b>	<b>83.4</b>	<b>81.5</b>

Table 3.6: Future 3s prediction results on Watch-n-Patch.

Method	Micro	Macro		
	P/R	Prec.	Recall	F1-score
LSTM	43.9	28.3	26.6	24.9
ST-AOG + Earley [QHW17]	48.9	43.1	39.3	39.3
Bi-LSTM + Generalized Earley	<b>58.7</b>	<b>50.5</b>	<b>49.9</b>	<b>49.4</b>

Table 3.7: Segment prediction results on Watch-n-Patch.

Method	Micro	Macro		
	P/R	Prec.	Recall	F1-score
ST-AOG + Earley [QHW17]	29.4	28.5	18.9	19.9
LSTM	44.6	43.6	44.6	40.4
Bi-LSTM + Generalized Earley	<b>49.5</b>	<b>50.1</b>	<b>49.4</b>	<b>45.5</b>

### 3.5.4 Experiment on Breakfast Dataset

**Dataset** Breakfast [KAS14] is a dataset of daily cooking activities. The dataset includes 52 unique participants, each conducting 10 distinct cooking activities captured in 18 different

Table 3.8: Detection results on Breakfast.

Method	Micro	Macro		
	P/R	Prec.	Recall	F1-score
HOGHOF+HTK [KAS14]	28.8	–	–	–
ED-TCN [LFV17]*	43.3	–	–	–
Bi-LSTM	45.6	29.2	25.4	25.6
TCFPN [DX18]	52.0	–	–	–
Fisher+HTK [KGS16]	56.3	38.1	–	–
Bi-LSTM + Generalized Earley	<b>59.7</b>	<b>45.8</b>	<b>36.3</b>	<b>38.5</b>

\*The results for [LFV17] is obtained from [DX18].

kitchens, It has  $\sim 77$  hours of video footage containing different camera views (3 to 5 depending on the location). For data annotations, 48 different coarse action units are identified with 11,267 samples (segments) in total including  $\sim 3,600$  silence samples.

**Comparative methods** Besides Bi-LSTM, we compare Bi-LSTM + Generalized Earley with state-of-art methods for activity detection on the Breakfast dataset:

1. HOGHOF+HTK [KAS14]: a action-grammar-based method. The authors proposed to use the hidden Markov model (HMM) for modeling individual action units in the sequence recognition problem. These action units then form the building blocks to model complex human activities as sentences using an action grammar. A speech recognition engine (the HTK toolkit [YEG02]) is used for recognition on top of the extracted HOGHOF features [LMS08].

2. ED-TCN [LFV17]: an end-to-end method. Encoder-Decoder Temporal Convolutional Network is proposed to tackle the action classification problem. Under the model’s



setting, predictions at each frame are a function of a fixed-length period of time, which is referred to by the authors as the receptive field.

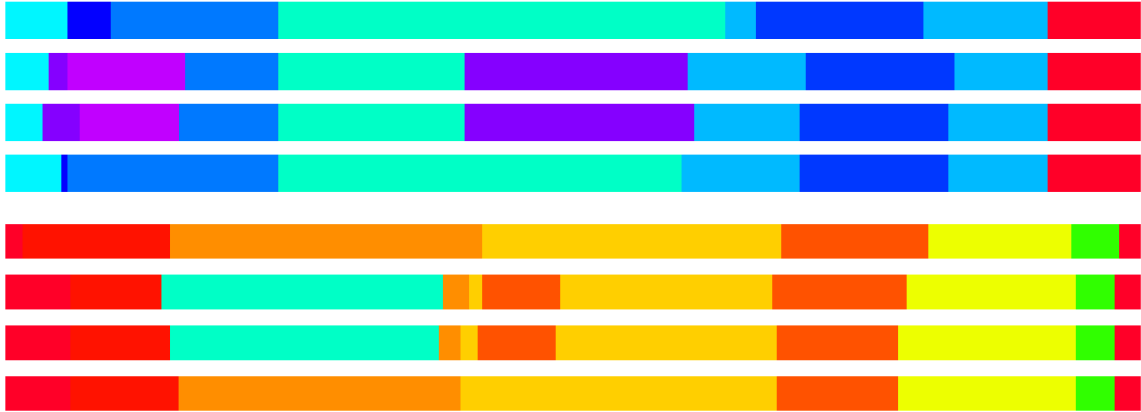
3. TCFPN [DX18]: one of the end-to-end state-of-the-art methods. The Temporal Convolutional Feature Pyramid retains the the encoder-decoder architecture and adapt the lateral connection mechanism proposed in Feature Pyramid networks to the task of action segmentation.

4. Fisher+HTK [KGS16]: one of the grammar-based state-of-the-art methods. Similar to [KAS14], the action units are modeled by HMM, and high-level activities are modeled by action grammars. The main difference is that a different type of feature (Fisher kernels [JH99]) is proposed for action recognition.

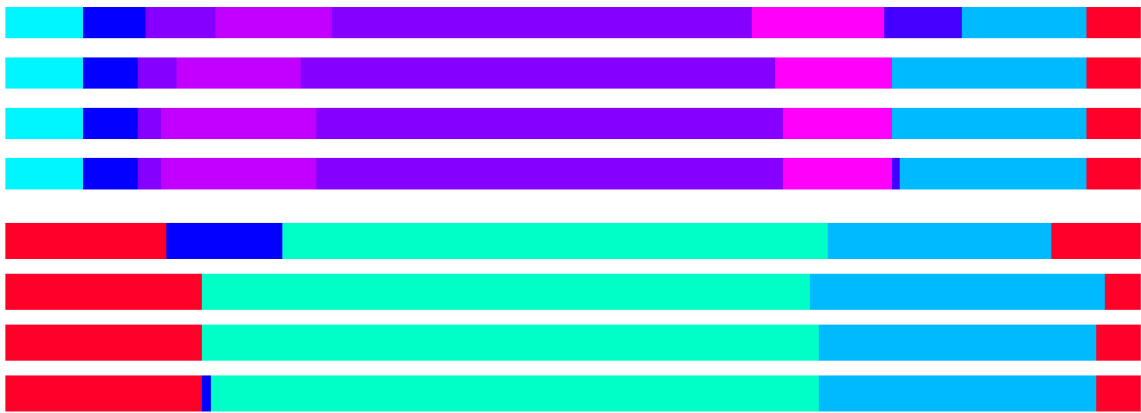
**Experiment results** To eliminate the factors of feature extraction for fair comparison, we use the pre-computed feature provided by [KGS16] to train the underlying Bi-LSTM classifier. The results (Table 3.8) show that even though a simple Bi-LSTM is far from state-of-the-art methods (an absolute difference of 10.7%), our full algorithm Bi-LSTM + Generalized Earley still outperforms the state-of-the-art by 3.6%. This shows that our explicit grammar regularization is effective in correcting the mistakes of the underlying classifier.

### 3.5.5 Discussion

**How different are the classifier outputs and the final outputs for detection?** Figure 3.9 shows some qualitative examples of the ground truth segmentations and results given by different methods. The segmentation results show that the refined outputs are similar with the classifier outputs since the confidence given by the classifiers are often very high.



(a) Correction



(b) Insertion

Figure 3.9: Qualitative results of segmentation results. In each group of four segmentations, the rows from the top to the bottom show the results of: 1) ground-truth, 2) ST-AOG + Earley, 3) Bi-LSTM, and 4) Bi-LSTM + generalized Earley parser. The results show (a) corrections and (b) insertions by our algorithm on the initial segment-wise labels given by the classifier (Bi-LSTM).

**How does the generalized Earley parser refine the classifier detection outputs?**

When the classifier outputs violate the grammar, two types of refinements occur: i) cor-

rection and deletion of wrong labels as shown in Figure 3.9a; ii) insertion of new labels as shown in Figure 3.9b. The inserted segments are usually very short to accommodate both the grammar and the classifier outputs. Most boundaries of the refined results are well aligned with the classifier outputs.

**Why do we use two metrics for future prediction?** The future 3s prediction is a standard evaluation criterion set up by KGS and ATCRF. However, this criterion does not tell how well the algorithm predicts the next segment label. i) At any time frame, part of the future 3s involves the current sub-activity for most of the times. ii) If the predicted length of the current sub-activity is inaccurate, the frame-wise inaccuracy drops proportionally, even when the future segment label prediction is accurate. Therefore we also compare against the future segment label prediction because it is invariant to variations in activity lengths.

**How well does the generalized Earley parser perform for activity detection and prediction?** From the results we can see that it slightly improves over the classifier outputs for detection, but significantly outperforms the classifier for predictions. The modifications on classifier outputs (corrections and insertions in Figure 3.9) are minor but important to make the sentences grammatically correct, thus high-quality predictions can be made.

**How useful is the grammar for activity modeling?** From Table 3.3, Table 3.4, Table 3.6 and Table 3.7 we can see that both ST-AOG and generalized Earley parser outperforms Bi-LSTM for prediction. Prediction algorithms need to give different outputs for similar inputs based on the observation history. Hence the non-Markovian property of grammars is useful for activity modeling, especially for future prediction.

**How robust is the generalized Earley parser?** Comparing Table 3.4 and Table 3.7 we can see that there is a performance drop when the action sequences are more unpredictable

(in the Watch-n-Patch dataset). But it is capable of improving over the noisy classifier inputs and significantly outperforms the other methods. It is also robust in the sense that it can always find the best sentence in a given language that best explains the classifier outputs.

### **3.6 Conclusion**

We proposed a generalized Earley parser for parsing sequence data according to symbolic grammars. Detections and predictions are made by the parser given the probabilistic outputs from any classifier. We are optimistic about and interested in further applications of the generalized Earley parser. In general, we believe this is a step towards the goal of integrating the connectionist and symbolic approaches.

## CHAPTER 4

### Conclusion

In this thesis, I developed task-oriented frameworks for visual understanding of scenes and events. Incorporating object affordances, I synthesized realistic indoor scenes that are both useful for back inference of 3D scenes from images and generating data for training of vision algorithms. For event understanding, I introduce a framework to interpret videos from the perspective of task planning, in which the tasks are represented by stochastic context-free grammars. To achieve this, I propose the generalized Earley parser to bridge the raw sequence data and symbolic grammar. Hence we are able to do event parsing, event prediction, and task planning via the combination of top-down plans and bottom-up sensor inputs.

This thesis has made progress in both the spatial and temporal aspects of task-oriented visual understanding. However, there are still unexplored aspects. Both aforementioned aspects do not involve actual interaction with the environment, in the sense of either physical or social interactions.

In particular, for individual agents to maximize their values in such environments, they must learn to interact with and against others, as well as understand the consequences of their actions. To learn to interact with other agents, it is essential to reason about their behaviors. This motivates an interactive task-oriented framework to perceive both the environment (spatially and temporally) and the other agents.

To develop a truly comprehensive task-oriented representation, these need to be incorporated in the future to account for not only the physical tasks, but also the mental states of other agents (beliefs, desires, and intentions). On top of that, a perception and task planning system could possibly be built to learn and survive in this world.

## REFERENCES

- [AGR16] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. “Social lstm: Human trajectory prediction in crowded spaces.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [BBC07] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. “Analysis of representations for domain adaptation.” In *Advances in Neural Information Processing Systems (NeurIPS)*, 2007.
- [BBS09] Steffen Bickel, Michael Brückner, and Tobias Scheffer. “Discriminative learning under covariate shift.” *Journal of Machine Learning Research*, **10**(Sep):2137–2155, 2009.
- [BBS14] Sean Bell, Kavita Bala, and Noah Snavey. “Intrinsic images in the wild.” *ACM Transactions on Graphics (TOG)*, **33**(4), 2014.
- [BDW81] FO Bartell, EL Dereniak, and WL Wolfe. “The theory and measurement of bidirectional reflectance distribution function (BRDF) and bidirectional transmittance distribution function (BTDF).” In *Radiation scattering in optical systems*, volume 257, pp. 154–161. International Society for Optics and Photonics, 1981.
- [BKW98] Werner GK Backhaus, Reinhold Kliegl, and John S Werner. *Color vision: Perspectives from different disciplines*. Walter de Gruyter, 1998.
- [BM15] Jonathan T Barron and Jitendra Malik. “Shape, illumination, and reflectance from shading.” *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, **37**(8), 2015.
- [BMP06] John Blitzer, Ryan McDonald, and Fernando Pereira. “Domain adaptation with structural correspondence learning.” In *Empirical Methods in Natural Language Processing (EMNLP)*, 2006.
- [BR06] Ezer Bar-Aviv and Ehud Rivlin. “Functional 3D Object Classification Using Simulation of Embodied Agent.” In *British Machine Vision Conference (BMVC)*, 2006.

- [BRG16] Aayush Bansal, Bryan Russell, and Abhinav Gupta. “Marr revisited: 2d-3d alignment via surface normal prediction.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [BUS13] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. “OpenSurfaces: A richly annotated catalog of surface appearance.” *ACM Transactions on Graphics (TOG)*, **32**(4), 2013.
- [BUS15] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. “Material recognition in the wild with the materials in context database.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [CCP15] Wongun Choi, Yu-Wei Chao, Caroline Pantofaru, and Silvio Savarese. “Indoor scene understanding with geometric and semantic contexts.” *International Journal of Computer Vision (IJCV)*, **112**(2), 2015.
- [CFG15] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. “ShapeNet: An Information-Rich 3D Model Repository.” *arXiv preprint arXiv:1512.03012*, 2015.
- [CH05a] Miguel A Carreira-Perpinan and Geoffrey E Hinton. “On contrastive divergence learning.” In *AI Stats*, 2005.
- [CH05b] Olivier Chapelle and Zaid Harchaoui. “A machine learning approach to conjoint analysis.” In *Advances in Neural Information Processing Systems (NeurIPS)*, 2005.
- [CMR08] Corinna Cortes, Mehryar Mohri, Michael Riley, and Afshin Rostamizadeh. “Sample selection bias correction theory.” In *International Conference on Algorithmic Learning Theory*, 2008.
- [CPK16] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs.” *arXiv preprint arXiv:1606.00915*, 2016.
- [CSG18] Anoop Cherian, Suvrit Sra, Stephen Gould, and Richard Hartley. “Non-linear temporal subspace representations for activity recognition.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.



- [CSK15] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. “Deepdriving: Learning affordance for direct perception in autonomous driving.” In *International Conference on Computer Vision (ICCV)*, 2015.
- [Csu17] Gabriela Csurka. “Domain adaptation for visual applications: A comprehensive survey.” *arXiv preprint arXiv:1702.05374*, 2017.
- [CWL16] Wenzheng Chen, Huan Wang, Yangyan Li, Hao Su, Dani Lischinsk, Daniel Cohen-Or, Baoquan Chen, et al. “Synthesizing Training Images for Boosting Human 3D Pose Estimation.” In *International Conference on 3D Vision (3DV)*, 2016.
- [CZ17] Joao Carreira and Andrew Zisserman. “Quo vadis, action recognition? a new model and the kinetics dataset.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [Dau07] Hal Daumé III. “Frustratingly Easy Domain Adaptation.” In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2007.
- [Dau09] Hal Daumé III. “Bayesian multitask learning with latent hierarchies.” In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2009.
- [DBF12] Luca Del Pero, Joshua Bowdish, Daniel Fried, Bonnie Kermgard, Emily Hartley, and Kobus Barnard. “Bayesian geometric modeling of indoor scenes.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [DDS09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. “Imagenet: A large-scale hierarchical image database.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.
- [DFI15] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. “Flownet: Learning optical flow with convolutional networks.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [DRC17] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. “CARLA: An Open Urban Driving Simulator.” In *Conference on Robot Learning*, 2017.
- [DWL16] Yu Du, Yongkang Wong, Yonghao Liu, Feilin Han, Yilin Gui, Zhen Wang, Mohan Kankanhalli, and Weidong Geng. “Marker-less 3D human motion

- capture with monocular image sequence and height-maps.” In *European Conference on Computer Vision (ECCV)*, 2016.
- [DX18] Li Ding and Chenliang Xu. “Weakly-supervised action segmentation with iterative soft boundary assignment.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [Ear70] Jay Earley. “An efficient context-free parsing algorithm.” *Communications of the ACM*, 1970.
- [EEV15] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. “The pascal visual object classes challenge: A retrospective.” *International Journal of Computer Vision (IJCV)*, **111**(1), 2015.
- [EF15] David Eigen and Rob Fergus. “Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture.” In *International Conference on Computer Vision (ICCV)*, 2015.
- [EGX17] Mark Edmonds, Feng Gao, Xu Xie, Hangxin Liu, Siyuan Qi, Yixin Zhu, Brandon Rothrock, and Song-Chun Zhu. “Feeling the Force: Integrating Force and Pose for Fluent Discovery through Imitation Learning to Open Medicine Bottles.” In *International Conference on Intelligent Robots and Systems (IROS)*, 2017.
- [EP04] Theodoros Evgeniou and Massimiliano Pontil. “Regularized multi-task learning.” In *International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2004.
- [EPF14] David Eigen, Christian Puhersch, and Rob Fergus. “Depth map prediction from a single image using a multi-scale deep network.” In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.
- [FGH13] David F Fouhey, Abhinav Gupta, and Martial Hebert. “Data-driven 3D primitives for single image understanding.” In *International Conference on Computer Vision (ICCV)*, 2013.
- [FHX12] Yanwei Fu, Timothy M Hospedales, Tao Xiang, and Shaogang Gong. “Attribute learning for understanding unstructured social activity.” In *European Conference on Computer Vision (ECCV)*, 2012.

- [FKI14] Sean Ryan Fanello, Cem Keskin, Shahram Izadi, Pushmeet Kohli, David Kim, David Sweeney, Antonio Criminisi, Jamie Shotton, Sing Bing Kang, and Tim Paek. “Learning to be a depth camera for close-range human capture and interaction.” *ACM Transactions on Graphics (TOG)*, **33**(4):86, 2014.
- [Fri03] Arthur Fridman. “Mixed markov models.” *Proceedings of the National Academy of Sciences (PNAS)*, 2003.
- [FRS12] Matthew Fisher, Daniel Ritchie, Manolis Savva, Thomas Funkhouser, and Pat Hanrahan. “Example-based synthesis of 3D object arrangements.” *ACM Transactions on Graphics (TOG)*, 2012.
- [FSH11] Matthew Fisher, Manolis Savva, and Pat Hanrahan. “Characterizing structural relationships in scenes using graph kernels.” *ACM Transactions on Graphics (TOG)*, **30**(4), 2011.
- [FSL15] Matthew Fisher, Manolis Savva, Yangyan Li, Pat Hanrahan, and Matthias Nießner. “Activity-centric scene synthesis for functional 3D scene modeling.” *ACM Transactions on Graphics (TOG)*, 2015.
- [GDG15] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. “DRAW: A recurrent neural network for image generation.” *arXiv preprint arXiv:1502.04623*, 2015.
- [GFG06] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks.” In *International Conference on Machine Learning (ICML)*, 2006.
- [GGV11] Helmut Grabner, Juergen Gall, and Luc Van Gool. “What makes a chair a chair?” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.
- [GHK10] Abhinav Gupta, Martial Hebert, Takeo Kanade, and David M Blei. “Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces.” In *Advances in Neural Information Processing Systems (NeurIPS)*, 2010.
- [GHS11] Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid. “Action sequence models for efficient action detection.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

- [Gib79] James Jerome Gibson. *The ecological approach to visual perception*. Houghton, Mifflin and Company, 1979.
- [GKS16] Mona Fathollahi Ghezelghieh, Rangachar Kasturi, and Sudeep Sarkar. “Learning camera viewpoint using CNN to improve 3D body pose estimation.” In *International Conference on 3D Vision (3DV)*, 2016.
- [GL15] Yaroslav Ganin and Victor Lempitsky. “Unsupervised domain adaptation by backpropagation.” In *International Conference on Machine Learning (ICML)*, 2015.
- [GSE11] Abhinav Gupta, Scott Satkin, Alexei A Efros, and Martial Hebert. “From 3d scene geometry to human workspace.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.
- [GSH09] Arthur Gretton, Alex J Smola, Jiayuan Huang, Marcel Schmittfull, Karsten M Borgwardt, and Bernhard Schölkopf. “Covariate Shift by Kernel Mean Matching.” In *Dataset shift in machine learning*, pp. 131–160. MIT Press, 2009.
- [GSS09] Abhinav Gupta, Praveen Srinivasan, Jianbo Shi, and Larry S Davis. “Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.
- [GWC16] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. “Virtual worlds as proxy for multi-object tracking analysis.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [Hec77] James J Heckman. “Sample selection bias as a specification error (with an application to the estimation of labor supply functions).”, 1977.
- [HEH05] Derek Hoiem, Alexei A Efros, and Martial Hebert. “Automatic photo pop-up.” *ACM Transactions on Graphics (TOG)*, **24**(3), 2005.
- [HHF09] Varsha Hedau, Derek Hoiem, and David Forsyth. “Recovering the spatial layout of cluttered rooms.” In *International Conference on Computer Vision (ICCV)*, 2009.
- [Hin02] Geoffrey E Hinton. “Training products of experts by minimizing contrastive divergence.” *Neural Computation*, 2002.

- [HMV13] Mark Hendrikx, Sebastiaan Meijer, Joeri Van Der Velden, and Alexandru Iosup. “Procedural content generation for games: A survey.” *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2013.
- [HN05] Kenji Hara, Ko Nishino, et al. “Light source position and reflectance estimation from a single view without the distant illumination assumption.” *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, **27**(4), 2005.
- [HNC15] Hironori Hattori, Vishnu Naresh Boddeti, Kris M Kitani, and Takeo Kanade. “Learning scene-specific pedestrian detectors without real data.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [HPB16] Ankur Handa, Viorica Pătrăucean, Vijay Badrinarayanan, Simon Stent, and Roberto Cipolla. “Understanding real world indoor scenes with synthetic data.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [HPS16] Ankur Handa, Viorica Patraucean, Simon Stent, and Roberto Cipolla. “SceneNet: an Annotated Model Generator for Indoor Scene Understanding.” In *International Conference on Robotics and Automation (ICRA)*, 2016.
- [HQX18] Siyuan Huang, Siyuan Qi, Yinxue Xiao, Yixin Zhu, Ying Nian Wu, and Song-Chun Zhu. “Cooperative Holistic Scene Understanding: Unifying 3D Object, Layout, and Camera Pose Estimation.” In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.
- [HQZ18] 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.
- [HRB11] Tucker Hermans, James M Rehg, and Aaron Bobick. “Affordance prediction via learned object attributes.” In *International Conference on Robotics and Automation (ICRA)*, 2011.
- [HS06] Geoffrey E Hinton and Ruslan R Salakhutdinov. “Reducing the dimensionality of data with neural networks.” *Science*, 2006.
- [HSL17] Nicolas Heess, Srinivasan Sriram, Jay Lemmon, Josh Merel, Greg Wayne, Yuval Tassa, Tom Erez, Ziyu Wang, Ali Eslami, Martin Riedmiller, et al. “Emergence of Locomotion Behaviours in Rich Environments.” *arXiv preprint arXiv:1707.02286*, 2017.

- [HWK15] Qixing Huang, Hai Wang, and Vladlen Koltun. “Single-view reconstruction via joint analysis of image and shape collections.” *ACM Transactions on Graphics (TOG)*, 2015.
- [HWM14] Ankur Handa, Thomas Whelan, John McDonald, and Andrew J Davison. “A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM.” In *International Conference on Robotics and Automation (ICRA)*, 2014.
- [HZG16] Steven Holtzen, Yibiao Zhao, Tao Gao, Joshua B Tenenbaum, and Song-Chun Zhu. “Inferring human intent from video by sampling hierarchical plans.” In *International Conference on Intelligent Robots and Systems (IROS)*, 2016.
- [HZR15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification.” In *International Conference on Computer Vision (ICCV)*, 2015.
- [HZR16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [IB00] Yuri A Ivanov and Aaron F Bobick. “Recognition of visual activities and interactions by stochastic parsing.” *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2000.
- [IM18] Mostafa S Ibrahim and Greg Mori. “Hierarchical relational networks for group activity recognition and retrieval.” In *European Conference on Computer Vision (ECCV)*, 2018.
- [JGR13] Arpit Jain, Abhinav Gupta, Mikel Rodriguez, and Larry S Davis. “Representing videos using mid-level discriminative patches.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.
- [JH99] Tommi Jaakkola and David Haussler. “Exploiting generative models in discriminative classifiers.” In *Advances in Neural Information Processing Systems (NeurIPS)*, 1999.
- [JHM17] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. “CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning.” *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

- [JKS13] Yun Jiang, Hema Koppula, and Ashutosh Saxena. “Hallucinated humans as the hidden context for labeling 3d scenes.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.
- [JKS16] Yun Jiang, Hema S Koppula, and Ashutosh Saxena. “Modeling 3D environments through hidden human context.” *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2016.
- [JQZ18] Chenfanfu Jiang, Siyuan Qi, Yixin Zhu, Siyuan Huang, Jenny Lin, Lap-Fai Yu, and Zhu Song-Chun Terzopoulos, Demetri. “Configurable 3D Scene Synthesis and 2D Image Rendering with Per-Pixel Ground Truth Using Stochastic Grammars.” *International Journal of Computer Vision (IJCV)*, 2018.
- [JZS16] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. “Structural-RNN: Deep learning on spatio-temporal graphs.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [KAS14] Hilde Kuehne, Ali Arslan, and Thomas Serre. “The language of actions: Recovering the syntax and semantics of goal-directed human activities.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [KF18] Yu Kong and Yun Fu. “Human Action Recognition and Prediction: A Survey.” *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018.
- [KGS13] Hema Swetha Koppula, Rudhir Gupta, and Ashutosh Saxena. “Learning human activities and object affordances from rgb-d videos.” *The International Journal of Robotics Research*, 2013.
- [KGS16] Hilde Kuehne, Juergen Gall, and Thomas Serre. “An end-to-end generative framework for video segmentation and recognition.” In *Winter Conference on Applications of Computer Vision (WACV)*, 2016.
- [KIX16] Yinda Zhang Mingru Bai Pushmeet Kohli, Shahram Izadi, and Jianxiong Xiao. “DeepContext: Context-Encoding Neural Pathways for 3D Holistic Scene Understanding.” *arXiv preprint arXiv:1603.04922*, 2016.
- [KKS12] Markus Kuderer, Henrik Kretschmar, Christoph Sprunk, and Wolfram Burgard. “Feature-Based Prediction of Trajectories for Socially Compliant Navigation.” In *Robotics: Science and Systems (RSS)*, 2012.

- [KKT15] Tejas D Kulkarni, Pushmeet Kohli, Joshua B Tenenbaum, and Vikash Mansinghka. “Picture: A probabilistic programming language for scene perception.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [KN09] Louis Kratz and Ko Nishino. “Factorizing scene albedo and depth from a single foggy image.” In *International Conference on Computer Vision (ICCV)*, 2009.
- [KS14] Hema S Koppula and Ashutosh Saxena. “Physically grounded spatio-temporal object affordances.” In *European Conference on Computer Vision (ECCV)*, 2014.
- [KS16] Hema S Koppula and Ashutosh Saxena. “Anticipating human activities using object affordances for reactive robotic response.” *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, **38**(1), 2016.
- [KWK15] Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. “Deep convolutional inverse graphics network.” In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.
- [KZB12] Kris M Kitani, Brian D Ziebart, James Andrew Bagnell, and Martial Hebert. “Activity forecasting.” In *European Conference on Computer Vision (ECCV)*, 2012.
- [LaV98] Steven M LaValle. “Rapidly-Exploring Random Trees: A New Tool for Path Planning.” Technical report, Iowa State University, 1998.
- [LB14] Matthew M Loper and Michael J Black. “OpenDR: An approximate differentiable renderer.” In *European Conference on Computer Vision (ECCV)*, 2014.
- [LF14] Kang Li and Yun Fu. “Prediction of human activity by discovering temporal sequence patterns.” *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2014.
- [LFV17] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. “Temporal convolutional networks for action segmentation and detection.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [LGS16] Jenny Lin, Xingwen Guo, Jingyu Shao, Chenfanfu Jiang, Yixin Zhu, and Song-Chun Zhu. “A virtual reality platform for dynamic human-scene interaction.” In *SIGGRAPH ASIA 2016 Virtual Reality meets Physical Reality: Modelling and Simulating Virtual Humans and Environments*. ACM, 2016.



- [LHK09] David C Lee, Martial Hebert, and Takeo Kanade. “Geometric reasoning for single image structure recovery.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.
- [LKS11] Jingen Liu, Benjamin Kuipers, and Silvio Savarese. “Recognizing human actions by attributes.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.
- [LLK07] Benjamin Laxton, Jongwoo Lim, and David Kriegman. “Leveraging temporal, contextual and ordering constraints for recognizing complex activities in video.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.
- [LMB14] 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 *European Conference on Computer Vision (ECCV)*, 2014.
- [LMS08] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. “Learning realistic human actions from movies.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.
- [LN16] Stephen Lombardi and Ko Nishino. “Reflectance and illumination recovery in the wild.” *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, **38**(1), 2016.
- [LRB16] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. “Deeper Depth Prediction with Fully Convolutional Residual Networks.” *arXiv preprint arXiv:1606.00373*, 2016.
- [LSD15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully convolutional networks for semantic segmentation.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [LSL15] Fayao Liu, Chunhua Shen, and Guosheng Lin. “Deep convolutional neural fields for depth estimation from a single image.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [LXG17] Antonio M López, Jiaolong Xu, José L Gómez, David Vázquez, and Germán Ros. “From Virtual to Real World Visual Perception Using Domain Adaptation The DPM as Example.” In *Domain Adaptation in Computer Vision Applications*, pp. 243–258. Springer, 2017.

- [LZR15] Tian Lan, Yuke Zhu, Amir Roshan Zamir, and Silvio Savarese. “Action recognition by hierarchical mid-level action elements.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [LZW16] Yang Lu, Song-Chun Zhu, and Ying Nian Wu. “Learning FRAME Models Using CNN Filters.” In *AAAI Conference on Artificial Intelligence (AAAI)*, 2016.
- [LZZ14] Xiaobai Liu, Yibiao Zhao, and Song-Chun Zhu. “Single-view 3d scene parsing by attributed grammar.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [LZZ16] Wei Liang, Yibiao Zhao, Yixin Zhu, and Song-Chun Zhu. “What is Where: Inferring Containment Relations from Videos.” In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.
- [MHL17] Wei-Chiu Ma, De-An Huang, Namhoon Lee, and Kris M Kitani. “Forecasting interactive dynamics of pedestrians with fictitious play.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [MKF14] Austin Myers, Angjoo Kanazawa, Cornelia Fermuller, and Yiannis Aloimonos. “Affordance of object parts from geometric features.” In *Workshop on Vision meets Cognition, CVPR*, 2014.
- [MKP13] Vikash Mansinghka, Tejas D Kulkarni, Yura N Perov, and Josh Tenenbaum. “Approximate bayesian image interpretation using generative probabilistic graphics programs.” In *Advances in Neural Information Processing Systems (NeurIPS)*, 2013.
- [MKS16] Yair Movshovitz-Attias, Takeo Kanade, and Yaser Sheikh. “How useful is photo-realistic rendering for visual learning?” In *European Conference on Computer Vision (ECCV)*, 2016.
- [ML15] Arun Mallya and Svetlana Lazebnik. “Learning Informative Edge Maps for Indoor Scene Layout Prediction.” In *International Conference on Computer Vision (ICCV)*, 2015.
- [MMR09] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. “Domain adaptation: Learning bounds and algorithms.” In *Annual Conference on Learning Theory (COLT)*, 2009.

- [MSB14] Yair Movshovitz-Attias, Yaser Sheikh, Vishnu Naresh Boddeti, and Zijun Wei. “3D Pose-by-Detection of Vehicles via Discriminatively Reduced Ensembles of Correlation Filters.” In *British Machine Vision Conference (BMVC)*, 2014.
- [MSL11] Paul Merrell, Eric Schkufza, Zeyang Li, Maneesh Agrawala, and Vladlen Koltun. “Interactive furniture layout using interior design guidelines.” *ACM Transactions on Graphics (TOG)*, 2011.
- [MVG10] Javier Marin, David Vázquez, David Gerónimo, and Antonio M López. “Learning appearance in virtual scenarios for pedestrian detection.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.
- [NCF10] Juan Carlos Niebles, Chih-Wei Chen, and Li Fei-Fei. “Modeling temporal structure of decomposable motion segments for activity classification.” In *European Conference on Computer Vision (ECCV)*, 2010.
- [NHH15] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. “Learning deconvolution network for semantic segmentation.” In *International Conference on Computer Vision (ICCV)*, 2015.
- [NZI01] Ko Nishino, Zhengyou Zhang, and Katsushi Ikeuchi. “Determining reflectance parameters and illumination distribution from a sparse set of images for view-dependent image synthesis.” In *International Conference on Computer Vision (ICCV)*, 2001.
- [ON14] Geoffrey Oxholm and Ko Nishino. “Multiview shape and reflectance from natural illumination.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [ON16] Geoffrey Oxholm and Ko Nishino. “Shape and Reflectance Estimation in the Wild.” *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, **38**(2), 2016.
- [OPO10] Jan Ondřej, Julien Pettré, Anne-Hélène Olivier, and Stéphane Donikian. “A synthetic-vision based steering approach for crowd simulation.” *ACM Transactions on Graphics (TOG)*, 2010.
- [Par11] Deep Parsing. “A Grammar Correction Algorithm.” In *Formal Grammar: 14th International Conference*, 2011.
- [Pea09] Judea Pearl. *Causality*. Cambridge university press, 2009.

- [PH04] Matt Pharr and Greg Humphreys. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2004.
- [PJA12] Leonid Pishchulin, Arjun Jain, Mykhaylo Andriluka, Thorsten Thormählen, and Bernt Schiele. “Articulated people detection and pose estimation: Reshaping the future.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [PJW11] Leonid Pishchulin, Arjun Jain, Christian Wojek, Mykhaylo Andriluka, Thorsten Thormählen, and Bernt Schiele. “Learning people detection models from few training samples.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.
- [PJZ11] Mingtao Pei, Yunde Jia, and Song-Chun Zhu. “Parsing video events with goal inference and intent prediction.” In *International Conference on Computer Vision (ICCV)*, 2011.
- [PR14] Hamed Pirsiavash and Deva Ramanan. “Parsing videos of actions with segmental grammars.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [PSA15] Xingchao Peng, Baochen Sun, Karim Ali, and Kate Saenko. “Learning deep object detectors from 3D models.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [PSY13] Mingtao Pei, Zhangzhang Si, B Yao, and Song-Chun Zhu. “Video event parsing and learning with goal and intent prediction.” *Computer Vision and Image Understanding (CVIU)*, 2013.
- [PZ15] Seyoung Park and Song-Chun Zhu. “Attributed grammars for joint estimation of human attributes, part and pose.” In *International Conference on Computer Vision (ICCV)*, 2015.
- [QHW17] Siyuan Qi, Siyuan Huang, Ping Wei, and Song-Chun Zhu. “Predicting Human Activities Using Stochastic Grammar.” In *International Conference on Computer Vision (ICCV)*, 2017.
- [Qiu16] Weichao Qiu. *Generating Human Images and Ground Truth using Computer Graphics*. PhD thesis, UNIVERSITY OF CALIFORNIA, LOS ANGELES, 2016.

- [QJZ18] Siyuan Qi, Baoxiong Jia, and Song-Chun Zhu. “Generalized Earley Parser: Bridging Symbolic Grammars and Sequence Data for Future Prediction.” In *International Conference on Machine Learning (ICML)*, 2018.
- [QSN16] Charles R Qi, Hao Su, Matthias Niessner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. “Volumetric and Multi-View CNNs for Object Classification on 3D Data.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [QT08] Faisal Qureshi and Demetri Terzopoulos. “Smart camera networks in virtual reality.” *Proceedings of the IEEE*, **96**(10):1640–1656, 2008.
- [QWJ18] Siyuan Qi, Wenguan Wang, Baoxiong Jia, Jianbing Shen, and Song-Chun Zhu. “Learning Human-Object Interactions by Graph Parsing Neural Networks.” In *European Conference on Computer Vision (ECCV)*, 2018.
- [QY16] Weichao Qiu and Alan Yuille. “UnrealCV: Connecting Computer Vision to Unreal Engine.” *ACM Multimedia Open Source Software Competition*, 2016.
- [QZ18] Siyuan Qi and Song-Chun Zhu. “Intent-aware Multi-agent Reinforcement Learning.” In *International Conference on Robotics and Automation (ICRA)*, 2018.
- [QZH18] Siyuan Qi, Yixin Zhu, Siyuan Huang, Chenfanfu Jiang, and Song-Chun Zhu. “Human-centric Indoor Scene Synthesis Using Stochastic Grammar.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [RHG15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. “Faster R-CNN: Towards real-time object detection with region proposal networks.” In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.
- [RK17] Nicholas Rhinehart and Kris M Kitani. “First-person activity forecasting with online inverse reinforcement learning.” In *International Conference on Computer Vision (ICCV)*, 2017.
- [RLB15] Javier Romero, Matthew Loper, and Michael J Black. “FlowCap: 2D human pose from optical flow.” In *German Conference on Pattern Recognition*, 2015.
- [RM15] Hossein Rahmani and Ajmal Mian. “Learning a non-linear knowledge transfer model for cross-view action recognition.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

- [RM16] Hossein Rahmani and Ajmal Mian. “3d action recognition from novel viewpoints.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [RMC15] Alec Radford, Luke Metz, and Soumith Chintala. “Unsupervised representation learning with deep convolutional generative adversarial networks.” *arXiv preprint arXiv:1511.06434*, 2015.
- [RMG15] Daniel Ritchie, Ben Mildenhall, Noah D Goodman, and Pat Hanrahan. “Controlling procedural modeling programs with stochastically-ordered sequential monte carlo.” *ACM Transactions on Graphics (TOG)*, 2015.
- [RRA12] Marcus Rohrbach, Michaela Regneri, Mykhaylo Andriluka, Sikandar Amin, Manfred Pinkal, and Bernt Schiele. “Script data for attribute-based recognition of composite activities.” In *European Conference on Computer Vision (ECCV)*, 2012.
- [RS16] Grégory Rogez and Cordelia Schmid. “MoCap-guided data augmentation for 3D pose estimation in the wild.” In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.
- [RSM16] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. “The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [RT16] Anirban Roy and Sinisa Todorovic. “A Multi-scale CNN for Affordance Segmentation in RGB Images.” In *European Conference on Computer Vision (ECCV)*, 2016.
- [RVR16] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. “Playing for data: Ground truth from computer games.” In *European Conference on Computer Vision (ECCV)*, 2016.
- [Ryo11] Michael S Ryoo. “Human activity prediction: Early recognition of ongoing activities from streaming videos.” In *International Conference on Computer Vision (ICCV)*, 2011.
- [RZ11] Brandon Rothrock and Song-Chun Zhu. “Human parsing using stochastic and-or grammars and rich appearances.” In *International Conference on Computer Vision (ICCV)*, 2011.

- [SB91] Louise Stark and Kevin Bowyer. “Achieving generalized object recognition through reasoning about association of function to structure.” *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, **13**(10), 1991.
- [SB02] Gaurav Sharma and Raja Bala. *Digital color imaging handbook*. CRC press, 2002.
- [SC12] Sreemananath Sadanand and Jason J Corso. “Action bank: A high-level representation of activity in video.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [SDL17] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. “Aerial Informatics and Robotics Platform.” Technical report, Microsoft Research, 2017.
- [SGC17] Cesar Roberto de Souza, Adrien Gaidon, Yohann Cabon, and Antonio Manuel Lopez. “Procedural Generation of Videos to Train Deep Action Recognition Networks.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [SGS10] Michael Stark, Michael Goesele, and Bernt Schiele. “Back to the Future: Learning Shape Models from 3D CAD Data.” In *British Machine Vision Conference (BMVC)*, 2010.
- [SHK12] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. “Indoor segmentation and support inference from RGBD images.” In *European Conference on Computer Vision (ECCV)*, 2012.
- [SHM14] Hao Su, Qixing Huang, Niloy J Mitra, Yangyan Li, and Leonidas Guibas. “Estimating image depth using shape collections.” *ACM Transactions on Graphics (TOG)*, 2014.
- [SHR05] Zach Solan, David Horn, Eytan Ruppin, and Shimon Edelman. “Unsupervised learning of natural languages.” *Proceedings of the National Academy of Sciences (PNAS)*, 2005.
- [SMD13] Yale Song, Louis-Philippe Morency, and Randall Davis. “Action recognition by hierarchical sequence summarization.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.
- [SQL15] Hao Su, Charles R Qi, Yangyan Li, and Leonidas J Guibas. “Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views.” In *International Conference on Computer Vision (ICCV)*, 2015.

- [SS14] Baochen Sun and Kate Saenko. “From Virtual to Reality: Fast Adaptation of Virtual Object Detectors to Real Domains.” In *British Machine Vision Conference (BMVC)*, 2014.
- [SSI03] Imari Sato, Yoichi Sato, and Katsushi Ikeuchi. “Illumination from shadows.” *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, **25**(3), 2003.
- [SSK13] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. “Real-time human pose recognition in parts from single depth images.” *Communications of the ACM*, **56**(1):116–124, 2013.
- [SSS17] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. “Revisiting Unreasonable Effectiveness of Data in Deep Learning Era.” *International Conference on Computer Vision (ICCV)*, 2017.
- [ST05] Wei Shao and Demetri Terzopoulos. “Autonomous pedestrians.” In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA)*, 2005.
- [STN16] Noor Shaker, Julian Togelius, and Mark J Nelson. *Procedural Content Generation in Games*. Springer, 2016.
- [Sto95] Andreas Stolcke. “An efficient probabilistic context-free parsing algorithm that computes prefix probabilities.” *Computational linguistics*, 1995.
- [SVD03] Gregory Shakhnarovich, Paul Viola, and Trevor Darrell. “Fast pose estimation with parameter-sensitive hashing.” In *International Conference on Computer Vision (ICCV)*, 2003.
- [SX14] Shuran Song and Jianxiong Xiao. “Sliding shapes for 3d object detection in depth images.” In *European Conference on Computer Vision (ECCV)*, 2014.
- [SYZ17a] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. “Semantic Scene Completion from a Single Depth Image.” *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [SYZ17b] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. “Semantic Scene Completion from a Single Depth Image.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.



- [TE11] Antonio Torralba and Alexei A Efros. “Unbiased look at dataset bias.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.
- [TFK12] Kevin Tang, Li Fei-Fei, and Daphne Koller. “Learning latent temporal structure for complex event detection.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [THD15] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. “Simultaneous deep transfer across domains and tasks.” In *International Conference on Computer Vision (ICCV)*, 2015.
- [TLL11] Jerry O Talton, Yu Lou, Steve Lesser, Jared Duke, Radomír Měch, and Vladlen Koltun. “Metropolis procedural modeling.” *ACM Transactions on Graphics (TOG)*, 2011.
- [TPZ13] Kewei Tu, Maria Pavlovskaja, and Song-Chun Zhu. “Unsupervised structure learning of stochastic and-or grammars.” In *Advances in Neural Information Processing Systems (NeurIPS)*, 2013.
- [TR95] Demetri Terzopoulos and Tamer F Rabić. “Animat vision: Active vision in artificial animals.” In *International Conference on Computer Vision (ICCV)*, 1995.
- [Val07] Arne Valberg. *Light vision color*. John Wiley & Sons, 2007.
- [VB14] Nam N Vo and Aaron F Bobick. “From stochastic grammar to bayes network: Probabilistic parsing of complex activity.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [Vit67] Andrew Viterbi. “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm.” *IEEE transactions on Information Theory*, 1967.
- [VLM14] David Vázquez, Antonio M Lopez, Javier Marin, Daniel Ponsa, and David Geronimo. “Virtual and real world adaptation for pedestrian detection.” *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, **36**(4):797–809, 2014.
- [VOL14] Tuan-Hung Vu, Catherine Olsson, Ivan Laptev, Aude Oliva, and Josef Sivic. “Predicting actions from static scenes.” In *European Conference on Computer Vision (ECCV)*, 2014.

- [VRM17] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael Black, Ivan Laptev, and Cordelia Schmid. “Learning from Synthetic Humans.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [Wag12] Joachim Wagner. *Detecting grammatical errors with treebank-induced, probabilistic parsers*. PhD thesis, Dublin City University, 2012.
- [WDA12] Zhikun Wang, Marc Peter Deisenroth, Heni Ben Amor, David Vogt, Bernhard Schölkopf, and Jan Peters. “Probabilistic modeling of human movements for intention inference.” *Robotics: Science and Systems (RSS)*, 2012.
- [WDL09] Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. “Feature hashing for large scale multitask learning.” In *International Conference on Machine Learning (ICML)*, 2009.
- [WF09] Joachim Wagner and Jennifer Foster. “The effect of correcting grammatical errors on parse probabilities.” In *Proceedings of the 11th International Conference on Parsing Technologies*. Association for Computational Linguistics, 2009.
- [WFG15] Xiaolong Wang, David Fouhey, and Abhinav Gupta. “Designing deep networks for surface normal estimation.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [WFG16] Xiaolong Wang, Ali Farhadi, and Abhinav Gupta. “Actions~transformations.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [WG16] Xiaolong Wang and Abhinav Gupta. “Generative Image Modeling using Style and Structure Adversarial Networks.” *arXiv preprint arXiv:1603.05631*, 2016.
- [WGH14] Jacob Walker, Abhinav Gupta, and Martial Hebert. “Patch to the future: Unsupervised visual prediction.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [WLS14] Chenxia Wu, Ian Lenz, and Ashutosh Saxena. “Hierarchical Semantic Labeling for Task-Relevant RGB-D Perception.” In *Robotics: Science and Systems (RSS)*, 2014.
- [WLS15] Thomas Whelan, Stefan Leutenegger, Renato F Salas-Moreno, Ben Glocker, and Andrew J Davison. “ElasticFusion: Dense SLAM without a pose graph.” In *Robotics: Science and Systems (RSS)*, 2015.

- [WM10] Yang Wang and Greg Mori. “Hidden part models for human action recognition: Probabilistic versus max margin.” *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2010.
- [WMR17] Ziyu Wang, Josh Merel, Scott Reed, Greg Wayne, Nando de Freitas, and Nicolas Heess. “Robust Imitation of Diverse Behaviors.” *arXiv preprint arXiv:1707.02747*, 2017.
- [Wu16] Jiajun Wu. *Computational perception of physical object properties*. PhD thesis, Massachusetts Institute of Technology, 2016.
- [WXS18] Wenguan Wang, Yuanlu Xu, Jianbing Shen, and Song-Chun Zhu. “Attentive Fashion Grammar Network for Fashion Landmark Detection and Clothing Category Classification.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [WYL15] Jiajun Wu, Ilker Yildirim, Joseph J Lim, Bill Freeman, and Josh Tenenbaum. “Galileo: Perceiving physical object properties by integrating a physics engine with deep learning.” In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.
- [WZS15] Chenxia Wu, Jiemi Zhang, Silvio Savarese, and Ashutosh Saxena. “Watch-n-Patch: Unsupervised Understanding of Actions and Relations.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [WZZ16] Ping Wei, Yibiao Zhao, Nanning Zheng, and Song-Chun Zhu. “Modeling 4d human-object interactions for joint event segmentation, recognition, and object localization.” *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2016.
- [XLC07] Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram. “Multi-task learning for classification with dirichlet process priors.” *Journal of Machine Learning Research*, **8**(Jan):35–63, 2007.
- [XLZ16a] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Ying Nian Wu. “Cooperative Training of Descriptor and Generator Networks.” *arXiv preprint arXiv:1609.09408*, 2016.
- [XLZ16b] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Ying Nian Wu. “A theory of generative convnet.” In *International Conference on Machine Learning (ICML)*, 2016.

- [XRT12] Jianxiong Xiao, Bryan Russell, and Antonio Torralba. “Localizing 3D cuboids in single-view images.” In *Advances in Neural Information Processing Systems (NeurIPS)*, 2012.
- [XST18] Dan Xie, Tianmin Shu, Sinisa Todorovic, and Song-Chun Zhu. “Modeling and Inferring Human Intents and Latent Functional Objects for Trajectory Prediction.” *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018.
- [YDY15] Lap-Fai Yu, Noah Duncan, and Sai-Kit Yeung. “Fill and transfer: A simple physics-based approach for containability reasoning.” In *International Conference on Computer Vision (ICCV)*, 2015.
- [YEG02] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, et al. “The HTK book.” *Cambridge university engineering department*, 2002.
- [YIK16] Hashim Yasin, Umar Iqbal, Björn Krüger, Andreas Weber, and Juergen Gall. “A Dual-Source Approach for 3D Pose Estimation from a Single Image.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [YQK17] Tian Ye, Siyuan Qi, James Kubricht, Yixin Zhu, Hongjing Lu, and Song-Chun Zhu. “The Martian: Examining Human Physical Judgments across Virtual Gravity Fields.” *IEEE Transactions on Visualization and Computer Graph (TVCG)*, 2017.
- [YT10] Jenny Yuen and Antonio Torralba. “A data-driven approach for event prediction.” In *European Conference on Computer Vision (ECCV)*, 2010.
- [YTS05] Kai Yu, Volker Tresp, and Anton Schwaighofer. “Learning Gaussian processes from multiple tasks.” In *International Conference on Machine Learning (ICML)*, 2005.
- [YYT11] Lap Fai Yu, Sai Kit Yeung, Chi Keung Tang, Demetri Terzopoulos, Tony F Chan, and Stanley J Osher. “Make it home: automatic optimization of furniture arrangement.” *ACM Transactions on Graphics (TOG)*, 2011.
- [YYT16] Lap-Fai Yu, Sai-Kit Yeung, and Demetri Terzopoulos. “The clutterpalette: An interactive tool for detailing indoor scenes.” *IEEE Transactions on Visualization and Computer Graph (TVCG)*, **22**(2):1138–1148, 2016.

- [YYW12] Yi-Ting Yeh, Lingfeng Yang, Matthew Watson, Noah D Goodman, and Pat Hanrahan. “Synthesizing open worlds with constraints using locally annealed reversible jump mcmc.” *ACM Transactions on Graphics (TOG)*, 2012.
- [ZBK17] Yinda Zhang, Mingru Bai, Pushmeet Kohli, Shahram Izadi, and Jianxiong Xiao. “DeepContext: Context-Encoding Neural Pathways for 3D Holistic Scene Understanding.” In *International Conference on Computer Vision (ICCV)*, 2017.
- [ZDN15] Hang Zhang, Kristin Dana, and Ko Nishino. “Reflectance hashing for material recognition.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [ZFF14] Yuke Zhu, Alireza Fathi, and Li Fei-Fei. “Reasoning about object affordances in a knowledge base representation.” In *European Conference on Computer Vision (ECCV)*, 2014.
- [ZJZ16] Yixin Zhu, Chenfanfu Jiang, Yibiao Zhao, Demetri Terzopoulos, and Song-Chun Zhu. “Inferring forces and learning human utilities from videos.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [ZKA16] Tinghui Zhou, Philipp Krähenbühl, Mathieu Aubry, Qixing Huang, and Alexei A. Efros. “Learning Dense Correspondence via 3D-guided Cycle Consistency.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [ZM07] Song-Chun Zhu and David Mumford. “A stochastic grammar of images.” *Foundations and Trends® in Computer Graphics and Vision*, 2007.
- [ZMK17] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. “Target-driven visual navigation in indoor scenes using deep reinforcement learning.” In *International Conference on Robotics and Automation (ICRA)*, 2017.
- [ZRG09] Brian D Ziebart, Nathan Ratliff, Garratt Gallagher, Christoph Mertz, Kevin Peterson, J Andrew Bagnell, Martial Hebert, Anind K Dey, and Siddhartha Srinivasa. “Planning-based prediction for pedestrians.” In *International Conference on Intelligent Robots and Systems (IROS)*, 2009.
- [ZSY17] Yinda Zhang, Shuran Song, Ersin Yumer, Manolis Savva, Joon-Young Lee, Hailin Jin, and Thomas Funkhouser. “Physically-Based Rendering for Indoor

- Scene Understanding Using Convolutional Neural Networks.” *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [ZTS19] Yubo Zhang, Pavel Tokmakov, Cordelia Schmid, and Martial Hebert. “A Structured Model For Action Detection.” *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [ZWY13] Jun Zhu, Baoyuan Wang, Xiaokang Yang, Wenjun Zhang, and Zhuowen Tu. “Action recognition with actons.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.
- [ZZ11] Yibiao Zhao and Song-Chun Zhu. “Image parsing with stochastic scene grammar.” In *Advances in Neural Information Processing Systems (NeurIPS)*, 2011.
- [ZZ13] Yibiao Zhao and Song-Chun Zhu. “Scene parsing by integrating function, geometry and appearance models.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.
- [ZZL16] Xiaowei Zhou, Menglong Zhu, Spyridon Leonardos, Konstantinos G Derpanis, and Kostas Daniilidis. “Sparseness meets deepness: 3D human pose estimation from monocular video.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [ZZY13] Bo Zheng, Yibiao Zhao, Joey C Yu, Katsushi Ikeuchi, and Song-Chun Zhu. “Beyond point clouds: Scene understanding by reasoning geometry and physics.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.
- [ZZY15] Bo Zheng, Yibiao Zhao, Joey Yu, Katsushi Ikeuchi, and Song-Chun Zhu. “Scene understanding by reasoning stability and safety.” *International Journal of Computer Vision (IJCV)*, **112**(2), 2015.
- [ZZZ15] Yixin Zhu, Yibiao Zhao, and Song-Chun Zhu. “Understanding tools: Task-oriented object modeling, learning and recognition.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.