Explain What You See

Argumentation-Based Learning &
Robotic Vision

Hamed Ayoobi

Printing: Unknown

ISBN printed version: Unknown
ISBN electronic version: Unknown

© Hamed Ayoobi, Groningen, the Netherlands, 2022.



university of groningen

Explain What You See

Argumentation-Based Learning &
Robotic Vision

PhD thesis

To obtain the degree of PhD at the
University of Groningen
on the authority of the
Rector Magnificus Prof. C. Wijmenga.
and in accordance with
the decision by the College of Deans.

This thesis will be defended in public on

Unknown

by

Hamed Ayoobi

born on 04 March 1990
in Tehran

Supervisors
Prof. H.B. Verheij
Prof. M. Cao
Prof. L.C. Verbrugge
Dr. H. Kasaei

Assessment committee
Prof. K.J. Batenburg
Prof. D. Grossi
Prof. C. Seifert

Acknowledgements

To be added

DSSC COFUND

This thesis is conducted at the center of Data Science and Systems Complexity (DSSC) and sponsored with a Marie Skłodowska-Curie COFUND grant, agreement no. 754315.

Contents

1 Introduction **1**
- 1.1 Machine learning and robotics . . . 2
- 1.2 Background . . . 3
  - 1.2.1 Online Incremental Machine Learning . . . 3
  - 1.2.2 Open-Ended Class-Incremental Machine Learning . . . 4
  - 1.2.3 Argumentation in Artificial Intelligence . . . 4
  - 1.2.4 3D Object Category Recognition and Parts Segmentation 5
- 1.3 General Purpose Service Robots: Desirable Features . . . 6
- 1.4 Thesis Structure . . . 8
- 1.5 Publications per Chapter . . . 10
- 1.6 Additional Resources . . . 11

2 Argumentation Online Incremental Learning: An Explainable Machine Learning Model **13**
- 2.1 Introduction . . . 15
  - 2.1.1 Argumentation . . . 15
  - 2.1.2 Argumentation in Machine Learning . . . 16
  - 2.1.3 The Expansions . . . 16
- 2.2 Background . . . 17
  - 2.2.1 Formal Definition of Abstract Argumentation Framework 17
  - 2.2.2 Formal Definition of an Abstract Bipolar Argumentation Framework . . . 19
  - 2.2.3 Definition of Online Incremental Machine Learning Algorithms . . . 20
- 2.3 Scenarios . . . 21
  - 2.3.1 Recovery Behaviors . . . 22
  - 2.3.2 Test Scenario 1 . . . 22
  - 2.3.3 Test Scenario 2 . . . 23
  - 2.3.4 Test Scenario 3 . . . 23
  - 2.3.5 Test Scenario 4 . . . 23
  - 2.3.6 Argumentation-Based Learning (ABL) . . . 24

2.3.7 Example . . . 26
2.3.8 Hypotheses Generation Unit (BAF Unit) . . . 29
2.3.9 Formal Representation of Updating Procedure of the Hypothesis Generation Unit (Algorithm 2.2) . . . 29
2.3.10 Formal Representation of Generating the Second Guesses using *BAF* (Algorithm 2.3) . . . 35
2.3.11 Formal Representation of Hypotheses Generation (Algorithm 2.4) . . . 36
2.3.12 Hypotheses Argumentation Unit using *AF* . . . 37
2.3.13 Formal Representation of Updating Procedure of Hypotheses Argumentation Unit (Algorithm 2.5) . . . 38
2.3.14 Why not Reinforcement Learning? . . . 39
2.3.15 Contextual Bandits . . . 40
2.3.16 Generalizing *ABL* to Other Real-Word Scenarios . . . 41
2.4 Experiments . . . 42
2.4.1 Performance Measure . . . 42
2.4.2 Comparison criteria . . . 43
2.4.3 Comparison Methods . . . 43
2.4.4 Results . . . 44
2.5 Discussion . . . 47
2.6 Conclusion . . . 50

3 Argue to Learn: Accelerated Argumentation-Based Learning 51
3.1 Introduction . . . 53
3.1.1 Argumentation in Machine Learning . . . 53
3.1.2 The Expansions . . . 53
3.2 Background . . . 54
3.2.1 Argumentation-Based Learning . . . 54
3.2.2 Formal Definition of an Abstract Bipolar Argumentation Framework . . . 54
3.2.3 Online Incremental Machine Learning Algorithms . . . 55
3.3 Scenarios . . . 55
3.3.1 Recovery Behaviors . . . 56
3.3.2 Test Scenario 1 . . . 57
3.3.3 Test Scenario 2 . . . 57
3.3.4 Test Scenario 3 . . . 57
3.4 Method . . . 57
3.4.1 Explanation of the Method with an Illustrating Example 58
3.4.2 Accelerated Argumentation-Based Learning . . . 61
3.4.3 Complexity Analysis . . . 64
3.5 Experiments . . . 64

3.5.1 Experimental Setup . . . 64
3.5.2 Experimental Results . . . 64
3.6 Conclusion . . . 67

4 Local-HDP: Interactive Open-ended 3D Object Category Recognition in Real-Time Robotic Scenarios 69
4.1 Introduction . . . 71
4.2 Related Work . . . 73
4.3 Method . . . 75
4.3.1 Pre-Processing Layers . . . 75
4.3.2 Local Hierarchical Dirichlet Process . . . 75
4.3.3 Dictionary of Visual Words . . . 77
4.3.4 Local Online Variational Inference . . . 77
4.3.5 Object Category Learning and Recognition . . . 79
4.4 Experimental Results . . . 80
4.4.1 Datasets and Baselines for Comparison . . . 81
4.4.2 Offline Evaluation . . . 82
4.4.3 Open-Ended Evaluation . . . 85
4.5 Real-time Robotic Application . . . 87
4.6 Conclusion . . . 89

5 Inference in a Probabilistic Graphical Model: Swift Distance Transformed Loopy Belief Propagation using a Novel Dynamic Label Pruning Method 91
5.1 Introduction . . . 93
5.2 Background and Related Works . . . 95
5.2.1 Loopy belief propagation . . . 95
5.2.2 Distance transform . . . 96
5.2.3 Priority-BP . . . 97
5.2.4 Measuring efficiency . . . 97
5.2.5 Image inpainting and Image completion . . . 97
5.2.6 Other related works . . . 98
5.3 Swift Distance Transformed Belief Propagation . . . 99
5.3.1 Dynamic label pruning . . . 99
5.3.2 Constant versus dynamic B_{conf} . . . 102
5.3.3 Early stopping criterion for belief propagation . . . 104
5.3.4 Incorporating dynamic label pruning in DT-BP . . . 104
5.4 Experiments . . . 107
5.5 Future works . . . 115
5.6 Conclusion . . . 116

6 Explain What You See: 3D Object Recognition and Parts Segmentation using Local-HDP and Argumentation **117**
6.1 Introduction . . . 119
6.2 Related Work . . . 122
6.3 Background . . . 124
6.3.1 Hierarchical Dirichlet Process . . . 124
6.3.2 Local Hierarchical Dirichlet Process for 3D Object Category Recognition . . . 124
6.3.3 Argumentation-Based Learning . . . 125
6.4 Method . . . 126
6.4.1 Processing Layers . . . 127
6.4.2 Oriented Local to Global 3D Object Descriptor . . . 128
6.4.3 3D Object Segmentation using Local-HDP . . . 130
6.4.4 Local Online Variational Inference . . . 131
6.5 Experimental Results . . . 134
6.5.1 Offline Evaluation of the 3D Object Parts Segmentation 134
6.5.2 Open-Ended Evaluation of the 3D Object Parts Segmentation . . . 135
6.5.3 Object Category Recognition with Argumentation-Based Learning . . . 137
6.6 Conclusion . . . 141

7 Discussion and Conclusion **143**
7.1 Contributions . . . 144
7.2 Future research . . . 147
7.2.1 Argumentation-Based Learning . . . 147
7.2.2 Local Hierarchical Dirichlet Process . . . 147
7.2.3 Inference Algorithms . . . 148
7.2.4 Argumentative Explanations for Machine Learning Models 148
7.2.5 Explanations for Debugging the Model and Human-Robot Interaction . . . 149
7.3 Conclusion . . . 149

References **153**

List of Publications **173**

CHAPTER 1

Introduction

Imagine that you have a robot that wants to operate in your home environment like a human and accomplish several different tasks. Do you think that it is possible to hard-code such a robot to work in every home-like environment to do several tasks without any failure? This seems impossible since even the programmer cannot predict all the possible conditions with which a robot might be confronted while operating. Therefore, the robot should be able to continuously learn and adapt to its surrounding environment. For example, assume that such a robot wants to recognize different kinds of objects and move them or give them to the user. What if the robot is limited to be able to recognize only a predefined category of objects? It can detect all the objects that the programmer considered in the model but cannot learn new object categories. This makes the robot less and less useful after a while when you bring new unforeseen objects to your house. What is needed is a robot that is able to learn new object categories with no limitation on the number of possible kinds of objects in the environment.

Let us think of a robot having such a capability of open-ended learning but not being able to learn by only watching a few objects of the same category and needing a lot of training data. This makes the model not sufficiently flexible to adapt to the changes in a dynamic environment. Therefore, learning with a smaller number of learning data is preferable.

Assume that a robot can learn an open-ended number of objects using only a small number of learning instances. However, suppose we find that the robot has wrongly categorized a teapot as a mug. Even knowing that a teapot and a mug might look similar, we still would like to know precisely why the model makes such a mistake. Otherwise, debugging the model to prevent the mistake would require that many different mugs and teapots are shown to the robot while we don't have so many in house. Therefore, it would be ideal to have explanations for each prediction of the model and to know why a certain object is categorized as a certain category. If we have such a robot that can provide such explanations, we can debug the model more easily. Moreover, we could

trust the robot more since every decision of the robot would be transparent to us.

Summarizing, in this thesis, methods are developed for continuous learning, for learning new object categories using limited data, and for explaining categorization outcomes. The methods can for instance be used in robot scenarios.

1.1 Machine learning and robotics

The focus in most current machine learning approaches is on improving the learning precision of the algorithm using a large number of learning instances. Deep neural networks typically need days of training on a fast Graphics Processing Unit (GPU). Using these processing units together with the learning capacity of the neural architectures enables the state-of-the-art techniques to achieve the highest learning precision. These approaches often follow an end-to-end learning paradigm that can directly map the input to the desired output without the need for manual feature-extraction and a dedicated feature engineering phase. Deep learning approaches usually have a separate initial learning phase followed by a testing phase. These methods also have some limitations which make them an unsuitable choice for open-ended robotic applications.

For instance, General Purpose Service Robots (GPSRs) need to operate in dynamic environments that are prone to changes. Therefore, a robotic agent should be able to quickly adapt to these changes in the environment in an online incremental manner. Handling unforeseen failure conditions is also required for GPSRs robots since pre-programming an agent to handle unforeseen conditions is not possible. Therefore, a GPSR should be able to learn in an online incremental manner for handling unforeseen failure conditions.

In robotic scenarios with both planning and vision components, a GPSR has a limited source of inputs and should operate in an open-ended dynamic scenario. In open-ended scenarios, the number of class labels is neither limited nor known in advance. This means that the robot should be able to learn new classes over time as they are needed for a general purpose agent in a dynamic environment. Therefore, the model should be class-incremental and learn in an online incremental manner without each time fully retraining the model. A GPSR acquires data by interacting with the environment. Therefore, it does not have access to a large amount of data in the initial phase and should learn from a few instances. Moreover, we need a model to be explainable for human users so that they can interact with the model to explain the underlying reasons for each decision so that the human users can both understand and

debug the model using a set of correcting feedback. Obtaining an explainable model leads to a more trustworthy robot that is easier to interact with [124]. Therefore, proposing a machine learning technique that can learn with a small number of training instances and that can operate in open-ended scenarios while producing explanations for each choice is required for robotic applications. Furthermore, this model can be used for robotic vision applications to recognize 3D object categories and segment 3D objects to semantic parts.

Developing methods for explainable online incremental machine learning that can operate in open-ended scenarios is the aim of this thesis. Towards this goal, different approaches in argumentation theory and topic modeling has been proposed. Consequently, the combination of the proposed argumentation-based approaches and the topic modeling techniques have been used to achieve the desired model.

1.2 Background

In this section, we discuss the research background of the studies in this thesis: Online Incremental Machine Learning, Open-Ended Class-Incremental Machine Learning, Argumentation-based Learning, 3D object category recognition, and parts segmentation. Subsequently, we will list the desired features of a machine learning technique for a GPSR. The thesis structure is discussed next, and the list of publications for each chapter concludes this chapter.

1.2.1 Online Incremental Machine Learning

Machine learning methods are employed to mine the collected data for relevant information and to predict future developments by generated models. However, classical batch machine learning approaches in which all data is simultaneously accessed, do not meet the requirements whenever the data is gradually gathered. Also systems should be able to work based on a small set of currently gathered data. Furthermore, these models do not continuously integrate new information into already constructed models but instead new models are regularly reconstructed from scratch. Such circumstances do not only imply very time consuming tasks but also lead to potentially outdated models when a needed reconstruction is delayed.

Overcoming such limitations requires a paradigm shift to sequential data processing in a streaming scheme. This does not only allow to use information as soon as it is available to guarantee up-to-date models, but also reduces the costs for data storage and maintenance.

Online incremental learning aims to develop models that can continuously learn new tasks from new data while preserving knowledge learned from the previously learned tasks [134]. Incremental and online algorithms fit naturally to this scheme, since they continuously incorporate information into their model, and traditionally aim for minimal processing time and space.

A recent study on the comparison of state-of-the-art methods for incremental online machine learning [112] shows that three methods are outperforming others, namely Incremental Support Vector Machines (ISVM) [38] together with LASVM [34], which is an online approximate SVM solver, and Online Random Forest (ORF) [145].

1.2.2 Open-Ended Class-Incremental Machine Learning

Incremental learning is related to different research topics, including continual learning and lifelong open-ended learning. These learning schemes are often used interchangeably [50, 163]. Incremental learning can be considered as a continual learning type in which tasks are presented in supervised data chunks. In contrast, continual learning is not limited to supervised learning. Lifelong open-ended learning [42, 20] requires models to be capable of learning during the whole life-time of the model, in which new information must be acquired and incorporated continuously into the existing model to optimize and update task performance. In these circumstances, a streaming data source might have a non-stationary distribution. In order to handle this variance, a desirable model should be able to continuously adapt to the streaming data in an incremental manner. This model should handle a large number of tasks using a limited computational and memory capacity. Open-ended machine learning techniques are not restricted to learn from a fixed number of class labels and they can handle a growing number of class labels in run-time [91].

1.2.3 Argumentation in Artificial Intelligence

Argumentation is a reasoning and discussion model based on interaction between supporting and attacking arguments [164, 22]. Argumentation has been used in various applications such as non-monotonic reasoning [166], inconsistency handling in knowledge bases [8, 26], and decision making [10, 33, 66]. In [57], Dung defined an Abstract Argumentation Framework (AF) as a pair with as members a set of arguments (of which the inner structure remains unspecified) and a binary relation representing the attack relation among the arguments. Extending Dung's idea, some arguments can also support a conclusion while others attack the conclusion. This has been formalized in Bipolar Argumentation Frameworks (BAF) [9].

There is a growing literature on argumentation-based machine learning. The survey by Cocarascu et al. [46] lists the following works using argumentation in supervised learning. Argumentation-Based Machine Learning (ABML) [120] uses the CN2 classification approach [44]. This method uses experts' arguments to improve classification results. Amgoud and colleagues [11] explicitly use argumentation in the context of classification. There are other approaches for improving classification using argumentation in the literature [36].

In contrast with the aforementioned methods, we are not using argumentation for improving the current machine learning approaches or resolving conflicting decisions between current classification methods; instead, we focus on the development of a supervised incremental learning method.

1.2.4 3D Object Category Recognition and Parts Segmentation

The topic of 3D object category recognition and classification experienced an increasing interest in recent years since 3D sensors became popular and different 3D object datasets have become publicly available. These methods have different applications in robotics, namely in robotic manipulation, navigation, and security, for instance for detecting dangerous objects [37].

Most recent object recognition and detection techniques are based on deep neural networks [79, 80, 88, 109, 136, 146, 111]. These methods typically need a large labeled dataset for a long training process. Typically, the number of object categories (class labels) should be predefined in advance for such methods. However, in some real-time robotic scenarios, an agent can face new object categories while operating in the environment. Therefore, the model should get updated in real-time in an open-ended manner without completely retraining the model [16].

Object parts segmentation is one of the challenging problems in 3D shape analysis. Data-driven part-segmentation methods typically outperform traditional geometrical methods [179]. In recent years, deep learning approaches have been widely exploited among researchers in this field [182]. Although these techniques show promising results in some applications, they are not well-suited for open-ended learning scenarios where the number of object categories and part-segments are not predefined and can be extended over time.

1.2.4.1 Limitations of 3D Semantic Segmentation Techniques

The majority of existing models for 3D shape segmentation have the following five limitations when they are used in open-ended dynamic environments.

First, most of these models are trained with a fixed set of labels, which greatly limits their flexibility and adaptivity. For instance, a model trained to segment a table into three semantic parts cannot be used to correctly segment a table with four parts. Second, using a fixed set of labels limits the number of object categories that the model can segment. For example, a model which previously learned how to segment a cup and a table cannot learn to segment a new object such as an airplane unless the model is retrained. Third, for state-of-the-art techniques, a good accuracy requires a long training time. This prevents the model to quickly adapt to the changes in the open-ended dynamic environment. Fourth, the object parts segmentation and object category recognition methods in the literature typically use a large training set, while learning with a lower number of learning instances is required for quick adaptation of the model to changes. Fifth, 3D object category recognition techniques are typically not robust to a high degree of occlusion, while encountering occluded objects is common in the real-word dynamic environments.

These limitations motivated us to design an open-ended 3D object parts segmentation model which can learn with higher accuracy and a lower number of learning instances.

1.3 General Purpose Service Robots: Desirable Features

In this section, we list desirable features as they are needed for General Purpose Service Robots, and are studied in this thesis. We relate the features to current key topics of research in artificial intelligence: *adaptability*, *explainability*, *collaboration* and *responsibility*, as suggested in [5].

1. **Learning in an online incremental manner** (Chapters 2, 3, 4, and 6). This means that the model should learn incrementally based on the streaming observations of the agent and adapt the model without a need for retraining the model. This requires the model to save previously acquired knowledge and adapt it when a contradicting concept is encountered. The desired model should be robust to the problems of catastrophic forgetting [67, 98] and concept-shift [173]. This feature is related to the research topic *adaptability*.

2. **Learning from a small number of learning instances** (Chapters 2, 3, 4, and 6). This feature is strongly required when an agent should learn and adapt to the environment as quickly as possible after observing only a few data instances. This topic has recently received a lot of attention

that is typically called few-shot learning [171]. This feature is related to the research topic *adaptability*.

3. **Class-incremental (open-ended or lifelong) learning** (Chapters 4, and 6). Lifelong learning is a key feature for robots operating in a dynamic environment where new unforeseen classes of objects can be observed any time. In these cases, the robot should be able to extend the model while preserving all the previously learned knowledge. Lifelong (open-ended or class-incremental) learning has recently been addressed in many research papers [185, 24, 157]. This feature is related to the research topic *adaptability*.

4. **Learn to recognize the category of 3D objects** (Chapters 4, and 6). An agent should learn to recognize new object categories by observing only a few object views and object instances of the same category to adapt to the environment. This enables the agent to initiate different tasks in the environment and interact with these objects in a later stage. For instance, it can then learn to manipulate these 3D objects. This feature is related to the research topics *collaboration* and *adaptability*.

5. **Learn to semantically segment 3D object parts** (Chapter 6). This feature is specifically needed when an agent needs to manipulate different object categories or needs to explain why a specific object belongs to a certain category. This feature is related to the research topics *adaptability* and *explainability*.

6. **Accelerating a probabilistic inference technique to approximate the parameters of the models** (Chapters 4, 5, and 6). Using probabilistic machine learning approaches in different tasks typically needs an approximation algorithm for the inference process. These inference techniques should work as fast as possible to enable an agent to operate in real-time. This feature is related to the research topic *adaptability*.

7. **Handle high degrees of occlusion while recognizing 3D objects** (Chapter 6). In real-world robotic applications, occlusion typically happens, for instance since typically a single-source 3D camera is used. Therefore, part of an object might not be visible, hence the model should be able to handle a high degree of occlusion for 3D object category recognition. This feature is related to the research topic *adaptability*.

8. **Producing explanations for the predictions of the model** (Chapters 2, 3, and 6). Explainable Artificial Intelligence (XAI) aims to provide

explanations for a machine learning model to be understandable for humans. Some of the approaches in the XAI community try to interpret the underlying reasoning process of black-box approaches like Artificial Neural Networks (ANNs) [138, 119, 115]. Unlike these approaches, we would like to have a model which is interpretable without a need for a separate unboxing algorithm for finding out the reasoning process. Therefore, the desired model should provide explanations of the reasoning process in addition to the underlying learning task. This feature is related to the research topics *explainability* and *responsibility*.

9. **Interacting with human users to learn from them and enable them to debug the model** (Chapters 4, and 6). Having a model that can learn by interacting with a human user would be desirable in many applications. A desirable model should be easy to debug by a non-expert human user. This feature is related to the research topic *collaboration*.

1.4 Thesis Structure

In this thesis, we propose the following approaches to address the aforementioned challenges:

- Chapter 2 introduces an argumentation-based online incremental learning approach which can learn from a small number of learning instances and leads to explainable rule sets for the predictions of the model. The proposed model can be utilized for open-ended scenarios as well where the numbers of feature values is not fixed and pre-defined. Although the proposed model outperforms online incremental learning, (deep) reinforcement learning and contextual bandit techniques in the experiments, it has some limitations, namely, the high computational and space complexity.

  This chapter has been published in the IEEE 15^{th} International Conference on Automation Science and Engineering (CASE) in 2019 [16]. Moreover, it has been extended and published as a journal paper in IEEE Transactions on Automation Science and Engineering (TASE) in 2021 [18].

- Chapter 3 aims to accelerate the argumentation-based learning approach by lowering space and computational complexity. The model has been simplified and two new strategies have been utilized to accelerate the model. This leads to a new model that can handle higher dimensional datasets.

This chapter has been published in the 20^{th} International Conference on Machine Learning and Applications (ICMLA) in 2021 [17].

- Chapter 4 proposes an open-ended local hierarchical Dirichlet process for recognizing the object category of 3D objects in real-world robotic scenarios using a robotic arm. This technique can learn with a small number of learning instances and operate in real-time. A human user can interact with the model to teach new object categories to the model and send correcting feedback to the model in case of wrong predictions. Moreover, an artificially simulated teacher is developed to facilitate the extensive set of experiments for evaluating the model.

  This chapter has been published in the Robotics and Autonomous Systems (RAS) journal in 2022 [20].

- Chapter 5 is based on my master thesis research in the artificial intelligence program at Yazd University. We have extended this research in my PhD. It introduces an inference technique for the Markov Random Fields (MRF) probabilistic graphical model to accelerate the performance of loopy belief propagation. This algorithm is further utilized for the image completion task where a missing part of an image should be filled in with no training data other than the same testing image. This is a challenging task, since using the local information in a stand-alone image might not be adequate for filling in a large missing part from the same image.

  This chapter has been published in the IET Image Processing journal in 2020 [21].

- Chapter 6 adapts the previously proposed local hierarchical Dirichlet process technique for the open-ended 3D semantic object parts segmentation approach task. The resulting technique uses two local-to-global and global-to-local object descriptors to represent the 3D objects. This makes the model more suitable for open-ended applications, since constructing a pre-defined dictionary for the model is not required any more.

  This chapter also integrates argumentation-based learning with the local hierarchical Dirichlet process in order to obtain a 3D object category recognition technique that can handle a high degrees of occlusion. This technique can produce explanations for the predictions of the model using the parts of objects. Therefore, the resulting model can predict the category of an object even if some parts of the object are not visible due to occlusion.

  This chapter is being prepared for a submission [19].

1.5 Publications per Chapter

- Chapter 2: **Argumentation-Based Online Incremental Learning**
  - H. Ayoobi, M. Cao, R. Verbrugge, and B. Verheij, "Handling unforeseen failure conditions using argumentation-based learning," in *International Conference on Automation Science and Engineering*. IEEE, 2019
  - H. Ayoobi, M. Cao, R. Verbrugge, and B. Verheij, "Argumentation-based online incremental learning," *IEEE Transactions on Automation Science and Engineering (TASE)*, 2021
- Chapter 3: **Accelerated Argumentation-Based Learning**
  - H. Ayoobi, M. Cao, R. Verbrugge, and B. Verheij, "Argue to learn: Accelerated argumentation-based learning," in *20th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2021
- Chapter 4: **Local-HDP, Interactive Open-Ended 3D Object Category Recognition in Real-Time Robotic Scenarios**
  - H. Ayoobi, H. Kasaei, M. Cao, R. Verbrugge, and B. Verheij, "Local-HDP: Interactive open-ended 3D object category recognition in real-time robotic scenarios," *Robotics and Autonomous Systems (RAS)*, vol. 147, p. 103911, 2022
- Chapter 5: **Swift Distance Transformed Loopy Belief Propagation using a Novel Dynamic Label Pruning Method**
  - H. Ayoobi and M. Rezaeian, "Swift distance transformed belief propagation using a novel dynamic label pruning method," *IET Image Processing*, vol. 14, no. 9, pp. 1822–1831, 2020
- Chapter 6: **Explain What You See - 3D Object Recognition and Segmentation using Local-HDP and Argumentation**
  - H. Ayoobi, H. Kasaei, M. Cao, R. Verbrugge, and B. Verheij, "Explain What You See: 3D Object Recognition and Parts Segmentation," 2022, (Under Preparation)

1.6 Additional Resources

In this section, we have provided the online repositories for different chapters of this thesis. Moreover, two videos for the robotic experiments in Chapter 4 are provided.

The source codes for Chapter 2, Chapter 3, Chapter 4, and Chapter 5 are available at https://github.com/H-Ayoobi/ABL, https://github.com/H-Ayoobi/Accelerated_ABL, https://github.com/H-Ayoobi/Local-HDP and https://github.com/H-Ayoobi/SDTBP written in Python and C++. Moreover, the code for chapter Chapter 6 will be available after publishing this chapter.

Two videos for the robotic demonstrations in Chapter 6 are available at https://youtu.be/YPsrBpqXWU4 and https://youtu.be/otxd8D8yYLc.

CHAPTER 2

Argumentation-Based Online Incremental Learning

An Explainable Machine Learning Model

This chapter is based on [18]. In this chapter, a new online incremental learning machine learning technique is proposed. This technique is based on argumentation theory and different argumentation formalisms, namely, abstract argumentation frameworks and bipolar argumentation frameworks. The resulting technique is a supervised structured machine learning method that can handle unforeseen failure conditions. This method has been compared to different online incremental techniques as well as (deep) reinforcement learning and contextual bandit (associative reinforcement learning) techniques. The results show that the proposed technique outperforms state-of-the-art methods in terms of learning speed and learning accuracy. However, it is limited to low-dimensional datasets since its computational complexity is high. Chapter 3 addresses this issue and proposes a simplified model with a lower computational complexity.

Argumentation-Based Online Incremental Learning

Chapter 2

Abstract

The environment around general-purpose service robots has a dynamic nature. Accordingly, even the robot's programmer cannot predict all the possible external failures which the robot may encounter. This research proposes an online incremental learning method that can be further used to autonomously handle external failures originating from a change in the environment. Existing research typically offers special-purpose solutions. Furthermore, the current incremental online learning algorithms cannot generalize well with just a few observations. In contrast, our method extracts a set of hypotheses, which can then be used for finding the best recovery behavior at each failure state. The proposed argumentation-based online incremental learning approach uses an abstract and bipolar argumentation framework to extract the most relevant hypotheses and model the defeasibility relation between them. This leads to a novel online incremental learning approach that overcomes the addressed problems and can be used in different domains including robotic applications. We have compared our proposed approach with state-of-the-art online incremental learning approaches, an approximation-based reinforcement learning method, and several online contextual bandit algorithms. The experimental results show that our approach learns more quickly with a lower number of observations and also has higher final accuracy than the other methods.

Note to Practitioners

This work proposes an online incremental learning method that learns faster by using a lower number of failure states than other state-of-the-art approaches. The resulting technique also has higher final learning accuracy than other methods. Argumentation-based online incremental learning generates an explainable set of rules which can be further used for human-robot interaction. Moreover, testing the proposed method using a publicly available dataset suggests wider applicability of the proposed incremental learning method outside the robotics field wherever an online incremental learner is required. The limitation of the proposed method is that it aims for handling discrete feature values.

2.1 Introduction

The development and application of domestic service robots are growing rapidly. Whereas basic household robots are already common practice [114], the study of General Purpose Domestic Service Robots (GPSR) able to do complex tasks is increasing [116, 148]. Due to the dynamic environment around GPSRs, they need to efficiently handle noise and uncertainty [83].

On the hardware level of GPSRs, any kind of system failure should be avoided. On a practical level, which involves persistent changes in the environment, it becomes much more difficult to account for all possible external failures at design time. Therefore, it is important to note that encountering unforeseen failures is mostly the default state for GPSRs, rather than an exceptional state as often described in the literature. There are some solutions for external failure recovery in the literature, which involve using simulations for the prediction of external failures [6] and logic-based reasoning to account for external failures [156, 147]. However, in most of these cases, the solutions are proposed for specific applications. In the following, we use the word "Failure" instead of the word "External Failure" for conciseness. This means that the focus of our research is not on system/hardware failures. In this thesis, we propose an argumentation-based incremental online learning method for recovering from unforeseen failures.

2.1.1 Argumentation

Argumentation is a reasoning model based on interaction between arguments [164]. Argumentation has been used in various applications such as non-monotonic reasoning [140], inconsistency handling in knowledge bases [165],

and decision making [13]. In [57], Dung has defined an Abstract Argumentation Framework (AF) as a pair of the arguments (whose inner structures are unknown) and a binary relation representing the attack relation among the arguments. Extending Dung's idea, some arguments can support a conclusion and others might be against (attacking) that conclusion in the bipolar argumentation framework [9]. Both the Bipolar Argumentation Framework (BAF) and the Abstract Argumentation Framework (AF) are used in the proposed argumentation-based learning approach.

2.1.2 Argumentation in Machine Learning

According to a recent survey by Cocarascu et al. [46], the works using argumentation in supervised learning are listed as follows. Argumentation-Based Machine Learning (ABML) [120] uses the CN2 classification approach [44]. This method uses experts' arguments to improve the classification results. The paper by Amgoud et al. [11] explicitly uses argumentation. There are other approaches for improving classification using argumentation in the literature [36].

Machine learning techniques have also been used for argumentation mining [103, 118, 59]. Bishop et al. combined argumentation with machine learning to prevent failure in deep neural network based break-the-glass access control systems [28].

In contrast with the aforementioned methods, we do not use argumentation for improving the current machine learning approaches or resolving conflicting decisions between current classification methods; instead, we focus on the development of an online incremental learning method. Moreover, the proposed method only uses class labels for the testing phase and not for the training. Therefore, it can be utilized in open-ended (class-incremental) scenarios as well [20].

2.1.3 The Expansions

This research is an expansion of the conference paper [16]. The specific expansions are listed as follows.

- The comparison of our proposed Argumentation-Based Learning (ABL) approach with multiple associative reinforcement learning approaches or contextual bandit algorithms has been added to the paper. Contextual bandit algorithms are the most relevant approach to study types of scenarios similar to those presented in the thesis.

- Formalizing the proposed method. This includes formalizing the updating procedure of the hypotheses generation unit and hypotheses argumentation unit, formalizing the process of generating hypotheses from the BAF and formalization of the first and second guess generation. In this way, the specification of the method is fully precise and non-ambiguous.

- Extending the proposed method to handle multiple successful recovery behaviors rather than only one successful recovery behavior at each state. Real-world robotic scenarios sometimes have multiple successful recovery behaviors for a failure state.

- Specifying the algorithms in the proposed method by adding pseudocodes to explain argumentation-based learning in more detail. In this way, the computational details of our implemented algorithms are fully explained.

- Validating the argumentation-based learning method outside the robotics scenarios, using a publicly available machine learning dataset (from the UCI repository). This emphasizes the applicability of the proposed method as a general technique for online incremental learning and it shows that this method is not limited to robotics applications.

The rest of this chapter is organized as follows. The required background is presented in Section II. Section III introduces the scenarios used in this research. In Section IV, the proposed method has been explained in more detail. Section V presents the experiments and the results obtained from this research. The discussion is presented in Section VI. The conclusion is given in Section VII.

2.2 Background

The Abstract Argumentation Framework (AF) and Bipolar Argumentation Framework (BAF) are the building blocks of the online incremental learning approach. AF, BAF and online incremental machine learning algorithms are formally defined in this section.

2.2.1 Formal Definition of Abstract Argumentation Framework

An argumentation framework defined by Dung [57] is a pair $AF = \langle AR, R_{att} \rangle$ where AR is a set of arguments, and R_{att} is a binary relation on AR, i.e.

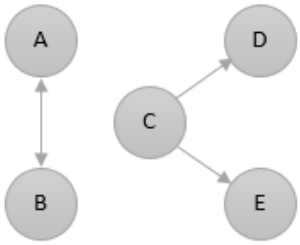


Figure 2.1: An abstract argumentation framework (AF)

$R_{att} \subseteq AR \times AR$. The meaning of $A\ R_{att}\ B$ is that A *attacks* B where A and B are two arguments. In order to define the *grounded extension* semantics in AF, which is used in the proposed learning method, some semantics should be defined first.

(**Conflict-Free**) Let $S \subseteq AR$. S is conflict-free iff there is no $B, C \in S$ such that B attacks C.

(**Acceptability**) An argument $A \in AR$ is *acceptable* with respect to a set S of arguments iff for each argument $B \in AR$: if B attacks A then B is attacked by at least one element of S.

(**Admissibility**) A conflict-free set of arguments S is *admissible* iff each argument in S is acceptable with respect to S.

(**Characteristic Function**) The *characteristic function* F_{AF} in an argumentation framework $AF = \langle\ AR,\ R_{att}\rangle$ is defined as follows:

$F_{AF} : 2^{AR} \rightarrow 2^{AR}$ and

$F_{AF}(S) = \{A | A \text{ is acceptable with respect to } S\}$.

(**Grounded Extension**) The *grounded extension* of an argumentation framework AF, denoted by GE_{AF}, is the least fixed point of F_{AF} with respect to set-inclusion [57]. Since F_{AF} is a monotonic function with respect to set inclusion [57], the existence of the fixed point for this function follows from the Knaster-Tarski theorem [158].

Example: Consider the argument set $AR = \{A, B, C, D, E\}$ and the attack relations given by $R_{att} = \{(A, B), (B, A), (C, D), (C, E)\}$ as demonstrated in Fig. 2.1. Then the conflict-free sets of arguments would be {}, {A}, {B}, {C}, {D}, {E},{A, C}, {A, D}, {A, E}, {B, C}, {B, D}, {B, E}, {D, E}, {A, D, E}, {B, D, E}. Among these, only the sets of {}, {A}, {B}, {C}, {A, C}, {B, C} are admissible. The grounded extension is {C}, which is the least fixed point of F_{AF}.

2.2.2 Formal Definition of an Abstract Bipolar Argumentation Framework

An Abstract Bipolar Argumentation Framework (BAF) [9] is an extension of Abstract Argumentation Framework by adding a support relationship. A BAF is a triple of the form $\langle AR, R_{att}, R_{sup}\rangle$ where AR is the finite set of arguments, $R_{att} \subseteq AR \times AR$ is the *attack* set and $R_{sup} \subseteq AR \times AR$ is the *support* set. Considering A_i and $A_j \in AR$, then $A_i \; R_{att} \; A_j$ means that A_i attacks A_j and $A_i \; R_{sup} \; A_j$ means that A_i supports the argument A_j.

The semantics of BAF are as follows:

(**Conflict-Free**) Let $S \subseteq AR$. S is conflict-free iff there is no $B, C \in S$ such that B attacks C.

(**Admissible set**) Let $S \subseteq AR$. S is admissible iff S is conflict-free, closed for R_{sup} (if $B \in S$ and $B \; R_{sup} \; C \Rightarrow C \in S$) and S defends all its elements. For instance in Fig. 2.3, $\{A, C, E, F\}$ is an admissible set since E defends C (i.e. E attacks D which itself is attacking C) and C defends A and no argument attacks F. Therefore, {A, C, E, F} defend all its elements.

(**Preferred extension**) The set $E \subseteq AR$ is a preferred extension iff E is inclusion-maximal among the admissible sets. An inclusion-maximal set among a collection of sets is a set that is not a subset of any other set in that collection.

(**Supporting Weights**) Like [126] the support relations in our model also have an assigned weight. Therefore, a node with higher sum of supporting weights can attack nodes with lower sum of supporting weights. For instance, Fig. 2.2 shows that the aggregated supporting weight of the argument A is $6 + 4 = 10$ and the corresponding supporting weight for the argument B is $2 + 3 + 4 = 9$. Therefore, argument A can attack and defeat B. The $\twoheadrightarrow$ arrows show attack relations and the $\rightarrow$ arrows demonstrate support relations in Fig. 2.2 and Fig. 2.3. The formal definition of the supporting weights function is defined in Eq. 8 in Section IV-D.

Figure 2.3 shows a bipolar argumentation framework. The admissible sets are {}, {E}, {A, C, E}, {A, C, E, F}. The preferred extension in this BAF is {A, C, E, F}.

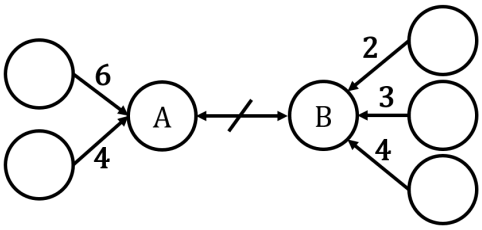


Figure 2.2: The supporting weights in the Bipolar Argumentation Framework (BAF).

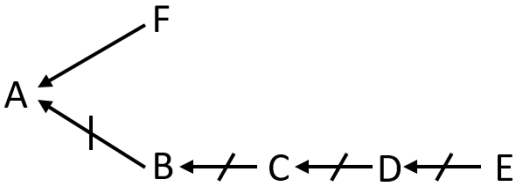


Figure 2.3: A Bipolar Argumentation Framework (BAF).

2.2.3 Definition of Online Incremental Machine Learning Algorithms

We define an incremental learning approach that uses a sequence of data instances $d_1, d_2, ..., d_t$ for generating the corresponding models $M_1, M_2, ..., M_t$. In case of incremental online learning, each data instance d_i incrementally updates the model and $M_i : \mathbb{R}^n \rightarrow \{1, ..., C\}$, where C is the number of class labels, is representing the model which depends on M_{i-1}. The online learning is then defined as an incremental learning which is also able to continuously learn. Incremental learning approaches have the following properties:

- The model should adapt gradually, i.e. M_i is updated using M_{i-1}.
- The previously learned knowledge should be preserved.

A recent study on the comparison of the state-of-the-art methods for incremental online machine learning [112] shows that Incremental Support Vector Machines ($ISVM$) [38, 152] together with LASVM [34], which is an online approximate SVM solver, and Online Random Forest (ORF) [145] outperform the other methods. The comparison methods used in our paper have been chosen based on the aforementioned survey [112].

The proposed argumentation-based incremental learning approach uses the bipolar argumentation framework to model the visited data instances and generate relevant hypotheses. Subsequently, the abstract argumentation frame-

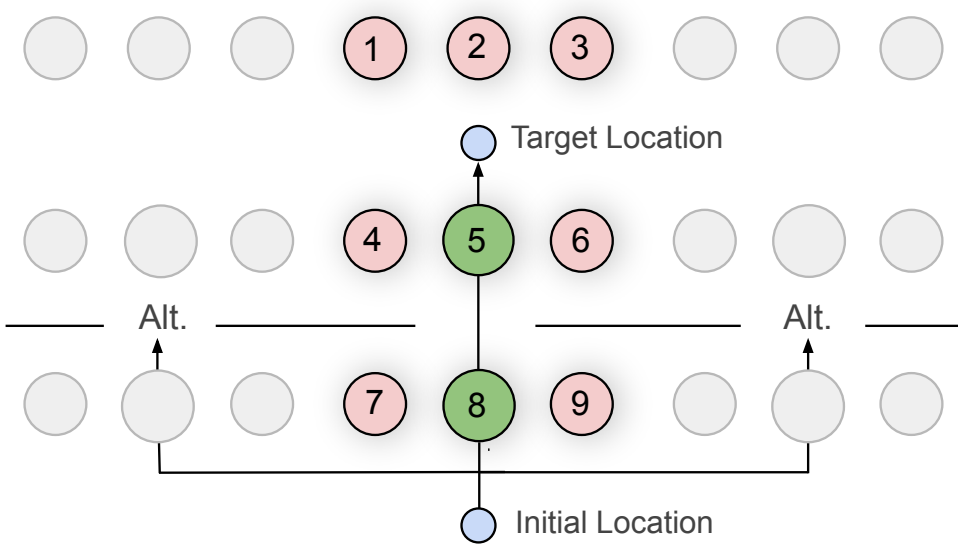


Figure 2.4: Schematic overview of the possible failure state scenarios. Only the green location is relevant for finding the best recovery behavior. Alt. stands for the Alternative Route recovery behavior.

work is used to model the defeasibility relations (i.e. the attack relations) between the current set of generated hypotheses and predict the best action (recovery behavior) for an unforeseen incoming data instance. Furthermore, the model incrementally gets updated as new data instances enter the model.

2.3 Scenarios

The performance of the different methods is tested using three test scenarios. The aim of the first two test scenarios is to model a situation where a programmer has provided an initial solution (e.g., a top level behavior such as entering the room), while (s)he has not accounted for all possible failures (e.g., objects and persons blocking the entrance), but allows the robot to find new solutions whenever a (previously unseen) failure occurs.

The basic setup of the first two test scenarios is illustrated in Fig. 2.4. The high-level behavior of the robot aims to proceed from the initial location to the target location using three entrances. Different obstacles might be on its way to the target location. In these scenarios, an agent observes all the obstacle locations at once and chooses a single recovery behavior (action) for recovering from that failure state. The agent can reach the goal if it chooses the best recovery behavior; otherwise, it fails to reach the goal.

In order to make the explanation of the method simpler, we first concentrate on finding only the best recovery behavior for each failure state. In the method Section IV, we will also explain how to generalize the method to scenarios where multiple recovery behaviors might be successful in a failure state.

2.3.1 Recovery Behaviors

Whenever the robot is confronted with a failure state, it may use any of the following recovery behaviors to resolve the issue. The run-time of each recovery behavior in seconds is presented in parentheses in front of each recovery behavior:

- Continue (2s): This solution is only useful if the failure has resolved itself (e.g., the obstacle moved away just after the failure).
- Push (5s): The robot can try pushing any obstacle.
- Ask (4s): The robot can try to ask any type to move.
- Alternative Route (Alt) (10s): The robot can move to another entrance to reach the target location.

It is important to note that choosing Alternative Route as the best recovery behavior may not always lead to success, because the robot may again encounter new obstacles (Fig. 2.4). Moreover, the best recovery behavior not only depends on the run-time of each recovery behavior, but also on the type, the color and the location of the obstacles.

2.3.2 Test Scenario 1

In this scenario, three types of obstacles (ball, box or person) with four colors (red, blue, green or yellow) can be presented in one of the locations 1 to 6 (Fig. 2.4). Locations 7 to 9 play no role in this scenario. There can be either zero or one combination of color-type in each location. Only location number 5, marked in green (Fig. 2.4), is relevant for choosing the best recovery behavior. It is important to notice that the robot does not know this fact and it should infer that the only effective location is location number 5 by observing different failure states in the environment.The agent observes all the obstacle locations at once and chooses a single recovery behavior (action) at each state. A new state is generated randomly at each time step. The number of possible combinations of the color-type in each location is 13 (3 types $\times$ 4 colors + "no obstacle" = 13). Since there are 6 locations in this scenario, the number of all possible states in this scenario is $13^6 = 4,826,809$.

Notice that colors can have meaningful interpretations for each type of obstacle. For instance, the red object might be heavy and cannot be pushed, while green ones are light. On the other hand, red people can be more cooperative and move out of the robot's way when being asked. Therefore, the colors can represent any realistic feature for the people and the objects. Using the colors instead of these realistic features simplifies the scenarios with fewer features.

2.3.3 Test Scenario 2

This scenario is more complex than the first scenario since each color-type combination can be presented in any of the nine possible locations. Here, only the green locations 5 and 8 are required for determining the best recovery behavior. Once again, the robot does not know this fact and it should infer that the only effective locations are the location number 5 and 8 by observing different failure states in the environment. The agent predicts a single recovery behavior (action) while it can observe all the obstacle locations at once at each state. The number of all possible states in this scenario is $13^9 = 10,604,499,373$.

2.3.4 Test Scenario 3

The third scenario has a different purpose and context. It shows the applicability of the proposed method outside the robotics field. The recent study on online incremental machine learning techniques [112] used the publicly available datasets from the UCI machine learning repository [56]. We also used the SPECT heart dataset from the UCI machine learning repository. This dataset represents the diagnosis of cardiac Single Proton Emission Computed Tomography (SPECT) images. Each of the images (patients) is even classified as normal or abnormal. The database of 267 SPECT image sets has been processed for extracting features that summarize the original SPECT images. We randomly selected 40 out of 267 data instances and fed them incrementally to the incremental learning approaches in order to compare the results.

The SPECT heart dataset has recently been used in various researches [149, 69, 53, 128].

2.3.5 Test Scenario 4

In addition to the three scenarios introduced in this thesis, we have also included the mushroom dataset from the UCI machine learning repository [56] that has been used in contextual bandits research [172, 139, 31]. The

mushroom dataset includes descriptions of hypothetical samples corresponding to 23 species of mushrooms divided into two classes (edible and poisonous). For this experiment, the dataset has been randomly shuffled in each iteration and the first 500 instances of the shuffled data have been chosen.

Notice that we do not use class labels in the training and that the label for each class is determined autonomously based on a trial and error procedure in our proposed method. Class labels are only used for testing the performance of the model for prediction on an unforeseen data instance.
In this section, we will discuss the proposed argumentation-based learning method for recovering from an unforeseen failure state.

2.3.6 Argumentation-Based Learning (*ABL*)

In order to explain *ABL*, we first use a simplified version of the previous test scenarios where there is only one location ahead of the robot (instead of 6 or 9). When there is no obstacle ahead of the robot, the best recovery behavior is "Continue".

Assume that the robot encounters a blue-ball blocking the entrance. Since there is no pre-trained model yet, the robot tests different recovery behaviors in order of their run-time to find the best one. Supposing that pushing the ball was successful in this case, the robot should learn from this experience.

However, unlike the traditional tabular reinforcement learning techniques, only learning the best recovery behaviors (actions) for exactly the same experiences (states) is not enough. We need a learning approach capable of inferring the correlated feature values (each feature value is the color or type of the obstacle at each location or an empty location with no color and type) for choosing the best recovery behavior. This is known as *generalization* in the machine learning literature. For instance, encountering a red ball and a green ball with the same recovery behavior of pushing, the robot should make a new hypothesis *push a ball*. Therefore, the next time the robot encounters the yellow ball, it can easily infer that *Push* is the best recovery behavior.

Encountering a yellow ball with *Alternative Route* as the best recovery behavior contradicts the previous hypothesis. Therefore, a new hypothesis is made: *Push a ball unless it's yellow*. From an argumentation perspective, we can see each hypothesis as an argument. Therefore, the second generated hypothesis can attack and defeat the first argument. This is inspired by human agents who make new hypotheses from their perceptions and reason about the best course of action at each state.

The architecture of the proposed argumentation-based learning method is shown in Fig. 2.5. A bipolar argumentation framework is used as hypotheses generator unit and an abstract argumentation framework models the defeasibility relation between these generated hypotheses.

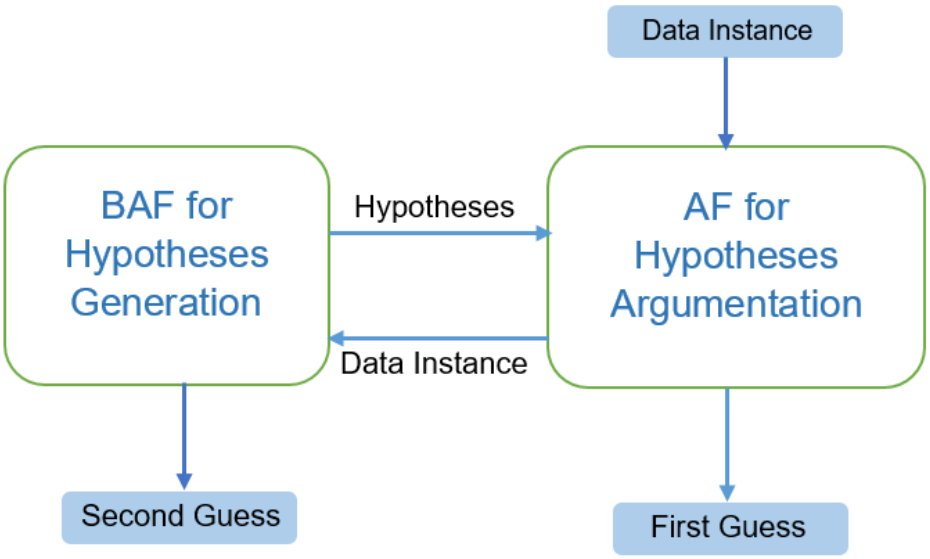


Figure 2.5: Architecture of the proposed Argumentation-based learning method.

Algorithm 2.4 presents the pseudocode of argumentation-based learning. When a new data instance enters the model, all the combinations of its feature-values and the set of nodes in the grounded extension of the AF will be extracted. Each node (argument) in the AF unit is of the form *precondition* $\rightarrow$ *post-condition: weight*. According to the similarity between the *preconditions* of the arguments in the grounded extension and the feature values combinations, there will be three possible cases. Either there will be a unique similarity, multiple similarities or no similarity. In case of unique similarity, the post-condition of the argument (which is a recovery behavior) will be used as the first guess and will be applied to the environment to see the result. On the occasion that there exist multiple similarities, the recovery behavior with the highest weight among the arguments will be chosen and its post-condition will be applied to the environment. A successful recovery from the failure state will update the BAF unit using Algorithm 2.2. On the other hand, failure from recovery will lead to generating the second guess (Algorithm 2.3), updating the BAF unit (Algorithm 2.2), generating hypotheses from BAF unit (Algorithm 2.4) and updating the AF unit (Algorithm 2.5), respectively.

Notice that the graphs that are the outcome of the argumentation-based learning algorithm are a disjoint union of complete irreflexive graphs. The grounded extensions of the graphs that result from the algorithm all have the

Algorithm 2.1: Argumentation-Based Learning pseudocode

input: Current **BAF** Graph, Current **AF** Graph, Data Instance **DI** entering the argumentation-based learning model
output: List of best recovery behaviors called **BRB-list**

- Extract all feature-value combinations **DI** and add them to a list called **Combs**.
- Find the set of nodes in grounded extension (**GE** of *AF*)
for *(all **gx** in **GE**)* **do**
  for *(all **comb** in **Combs**)* **do**
    if *(gx.precondition==comb)* **then**
      BRB-list.Add(gx.post-condition)
if *(**BRB-list** is not empty)* **then**
  if *(**BRB-list.Length**==1)* **then**
    - Apply **BRB-list[0]** to environment and observe the **result**.
  else
    - Select a recovery behavior in **BRB-list** with highest weight.
if *(**result**==failure)* ***OR*** *(**BRB-list** is empty)* **then**
  - Use BAF unit for second guess generation and add these guesses to **BRB-list** by using Algorithm 2.3
  - Update the **BAF** unit using Algorithm 2.2.
  - Generate Hypothesis from the updated **BAF** using Algorithm 2.4.
  - Update the AF unit using the generated hypotheses using Algorithm 2.5.
if *(**result**==success)* **then**
  - Update the **BAF** unit using Algorithm 2.2.
return BRB-list

form of a union of singleton complete subgraphs.

We now use an illustrative example to explain the proposed method in more detail.

2.3.7 Example

Table 2.1 shows the best recovery behavior when the robot encounters an obstacle with different colors and types. Notice that this table is only used for this example and a randomly generated table is utilized for each of the 1000 independent runs for the experiments. Figures 2.6 to 2.8 show the updating

Order	Color	Type	Best Recovery Behavior
1	Red	Ball	Push
2	Red	Box	Alternative Route
3	Red	Person	Ask
4	Green	Ball	Push
5	Green	Box	Alternative Route
6	Green	Person	Ask
7	Blue	Ball	Push
8	Blue	Box	Alternative Route
9	Blue	Person	Alternative Route
10	Yellow	Ball	Push
11	Yellow	Box	Alternative Route
12	Yellow	Person	Ask
13	None	None	Continue

Table 2.1: Possible combinations of color-type with the best recovery behaviors.

procedure of the model step by step. In the hypotheses generation unit (BAF), an arrow $\rightarrow$ shows a support relation between arguments and $\nrightarrow$ shows an attack relation between them. However, in AF, $\rightarrow$ shows an attack relationship between the arguments.

Referring to Table 2.1, at the beginning of the learning procedure, the robot encounters a Red-Ball (R-Ba). It tests all the recovery behaviors in order of their run-times and finds the *Push* recovery behavior as a success (Table 2.1). Subsequently, the Bipolar Argumentation Framework is getting updated as in Fig. 2.6. In order to update the BAF, first, the best recovery node is added which is *Push* in this case. Then all the possible combinations of the feature-values of the current state are added as supporting nodes. The supporting nodes for *Push* are *R*, *Ba* and *R-Ba*. If there previously exists the same supporting node, its supporting weight will be increased. For instance in Fig. 2.7, where *8:B-Bo* enters the BAF, since *B* and *B-Bo* are new supporting nodes for the *Alt (Alternative Route)* recovery behavior, they are added to the model with a supporting weight equal to 1. On the other hand, *Bo* already exists in the set of supporting nodes for *Alt* and its weight is increased. After updating the supporting weights, a set of hypotheses is generated based on the number of occurrences of each supporting node. For instance, after observing 1:Red-Ball (R-Ba), $R \rightarrow Push$ and $Ba \rightarrow Push$ are added to the AF unit.

Encountering *2:R-Bo* and using the previously generated hypotheses (specifically $R \rightarrow Push$), the robot would infer that the best possible recovery be-

Algorithm 2.2: Updating Hypotheses Generation Unit

input: Current BAF Graph, New Data Instance (**DI**) and the **B**est **R**ecovery **B**ehavior **BRB**
output: BAF Graph

```
- Extract all feature-value combinations of DI and add them to a list
  called Combs.
if (BRB is not in BAF) then
    - Add BRB to the BAF graph;
    - Add bidirectional attack edges between BRB node and all other
      Recovery Behavior Nodes (RBN) in BAF;
for (any item in Combs) do
    - Boolean isNewCombination = true
    for (any sup in BRB.supporting-nodes) do
        if (item == sup) then
            sup.weight += 1;
            isNewCombination = false;
    if (isNewCombination) then
        BRB.supporting-nodes.Add (item);
```


havior is *Push*, which is a wrong choice in this case (Table 2.1). Therefore, the robot tries other recovery behaviors and finds *Alt* as success and updates the model accordingly. Moreover, a bidirectional attack will be added among all the recovery nodes in the *BAF* (in this case, *Alt* and *Push*). Subsequently, the new set of hypotheses is generated to update the hypotheses argumentation unit. Finally, an abstract argumentation framework is updated to model the attack relations between the set of generated hypotheses (arguments). This *BAF-AF* update cycle goes on and on during the learning procedure.

In this small example, seven out of thirteen predictions of the model are correct, and only two are wrongly classified using the proposed argumentation-based learning. In other cases, our system can provide multiple probable guesses. For instance, when *12:Y-P* enters the system in Fig. 2.8, the *AF* cannot provide any suggestion but the *BAF* will suggest both *Ask* and *Alt* as the candidate recovery behaviors. However, the mapping of the states to the best recovery behavior is randomly generated in all the experiments.

2.3.8 Hypotheses Generation Unit (BAF Unit)

This unit has two roles. Firstly, it generates a new set of hypotheses whenever the *AF* unit could not classify the new data instance correctly (1). The second role of this unit is to produce a second guess for the best recovery behavior (2):

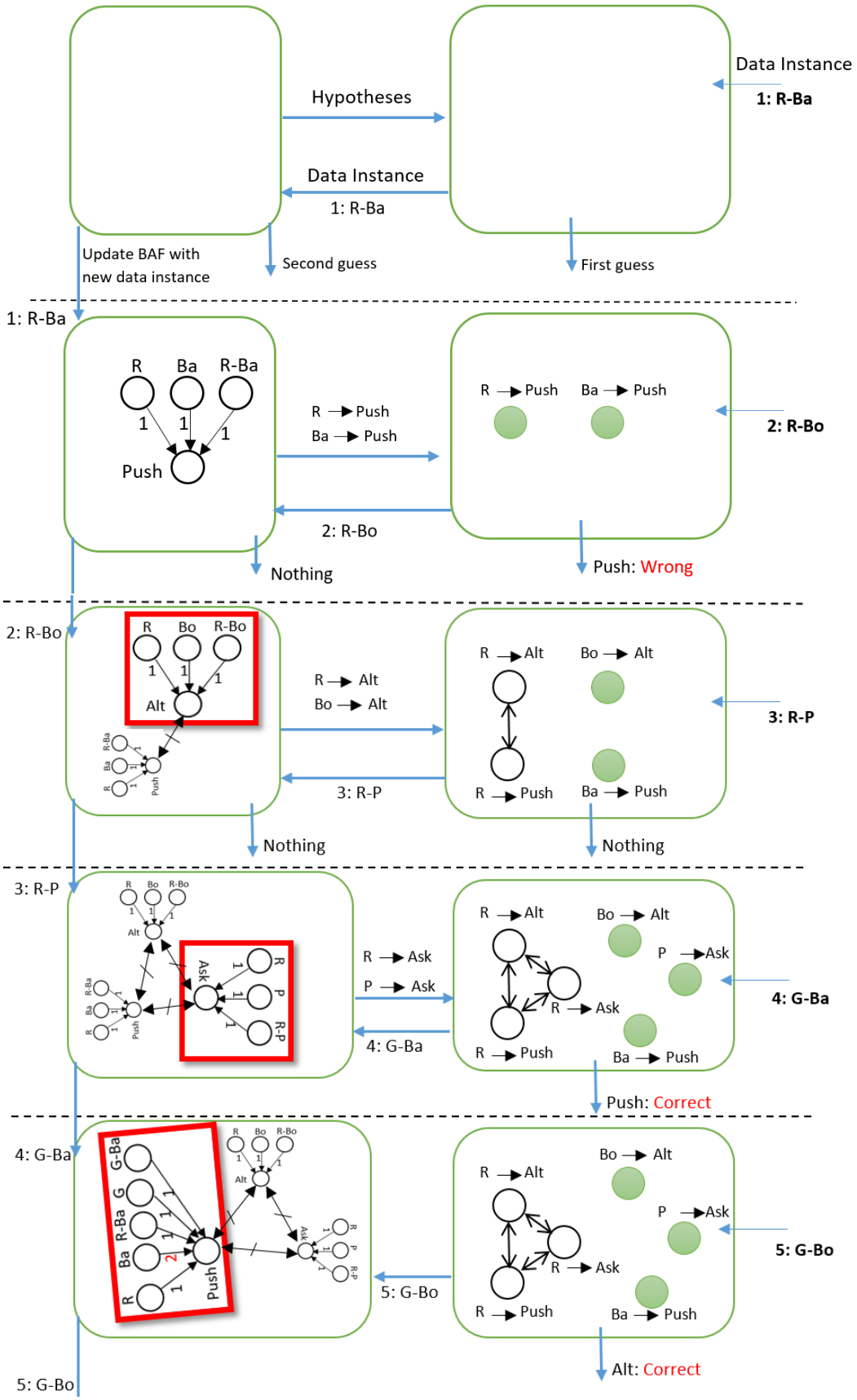
1) In order to generate a new set of hypotheses from the constructed *BAF*, only one recovery behavior is considered which is highlighted with a red box in Fig. 2.6 to 2.8. The pseudocode shown in Algorithm 2.4 shows the procedure of hypotheses generation.

The pseudocode of updating the current hypotheses generation graph (*BAF* unit) using a new data instance is shown in Algorithm 2.2. The only nodes which are getting updated during this process are the best recovery behavior for the current data instance and its supporting nodes. Autonomously identifying the best recovery behavior through trial and error, the update procedure for hypotheses generation takes place. The updating procedure searches for a node in the *BAF* graph with the best recovery behavior and appends all the possible combinations of the feature-values of the current state to the support nodes of the best recovery behavior node. In case that a supporting node already exists in the best recovery behavior node, its supporting weight is incremented.

2) In order to generate a second guess, a new *BAF* should be constructed. For an unforeseen failure state, the set of all possible combinations of feature-values is compared with the supporting nodes of each recovery behavior node. According to the sum of the matching supporting weights, the attack relations are adapted among the recovery behaviors. Therefore, only recovery behaviors with a higher sum of the matching supporting weights can attack the other recovery behavior. For instance, in the example, when *12: Y-P* enters the model for prediction, the *AF* is not be able to guess the best recovery behavior. Constructing a new *BAF* for a second guess, shown in Fig. 2.9, the calculated weighted sum for the *Alternative Route (Alt)* node is the same as *Ask* and higher than *Push*. Accordingly, the attack relations get updated. Using preferred extension semantics and its intersection with recovery behavior nodes, both *Alternative Route (Alt)* and *Ask* are chosen as the second guesses.

2.3.9 Formal Representation of Updating Procedure of the Hypothesis Generation Unit (Algorithm 2.2)

In Section II-C, the formal definition of online incremental learning is represented. The sequence of labeled data instances $d_1, ..., d_t$ is entering the model and the *BAF* unit gets updated. The hypotheses generation unit is represented


Hypotheses
Data Instance
1: R-Ba
Data Instance
1: R-Ba
Update BAF with
new data instance
Second guess
First guess
1: R-Ba
R
Ba
R-Ba
1
1
1
Push
R → Push
Ba → Push
R → Push
Ba → Push
2: R-Bo
2: R-Bo
Nothing
Push: Wrong
2: R-Bo
R
Bo
R-Bo
Alt
R → Alt
Bo → Alt
R → Alt
Bo → Alt
R → Push
Ba → Push
3: R-P
3: R-P
Nothing
Nothing
3: R-P
R → Ask
P → Ask
4: G-Ba
R → Alt
Bo → Alt
P → Ask
R → Ask
R → Push
Ba → Push
4: G-Ba
Push: Correct
4: G-Ba
G-Ba
R-Ba
Ba
R
Push
5: G-Bo
5: G-Bo
Bo → Alt
R → Alt
P → Ask
R → Ask
R → Push
Ba → Push
5: G-Bo
Alt: Correct

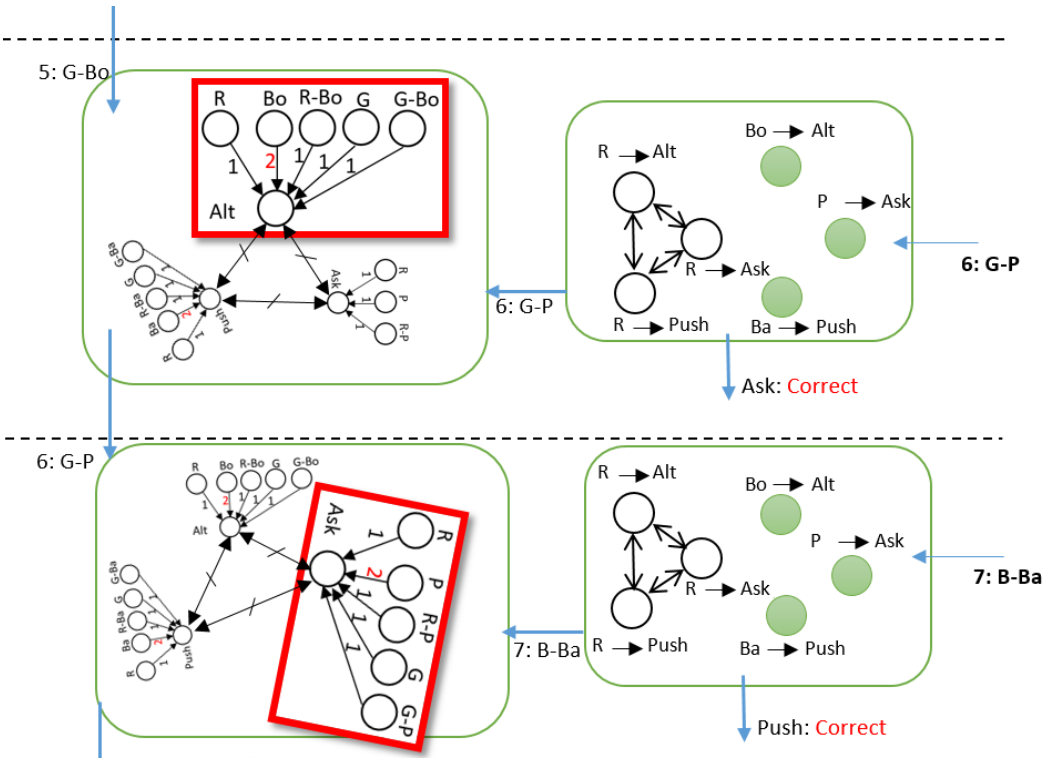


Figure 2.6: Example of argumentation-based learning for the illustrative example. (Part 1/3)

by BAF_{t+1} when a data instance d_t is entering this unit.

$$BAF_0 = \langle AR_0, R_{att_0}, R_{sup_0} \rangle = \langle \varnothing, \varnothing, \varnothing \rangle$$

$$BAF_{t+1} = update(BAF_t, d_t) \qquad \forall t \geq 0 \tag{2.1}$$

In the following lines, the update procedure for the BAF model is described. The BAF model at time $t+1$ is in the following form:

$$BAF_{t+1} = \langle AR_{t+1}, R_{att_{t+1}}, R_{sup_{t+1}} > \tag{2.2}$$

Using the best recovery behavior at time t called BRB_t (This is determined by trial and error in the environment) and the set of all the subsets of feature values in the n-dimensional data instance $d_t = (f_{1_t}, f_{2_t}, ..., f_{n_t})$ (f_{i_t} shows the ith feature value of the n-dimensional d_t vector), called $Combs(d_t)$, the arguments set of the BAF gets updated in the following form:

$$AR_{t+1} = AR_t \cup BRB_t \cup Combs_t \tag{2.3}$$

where

$$Combs_t = \mathbb{P}(d_t) \tag{2.4}$$

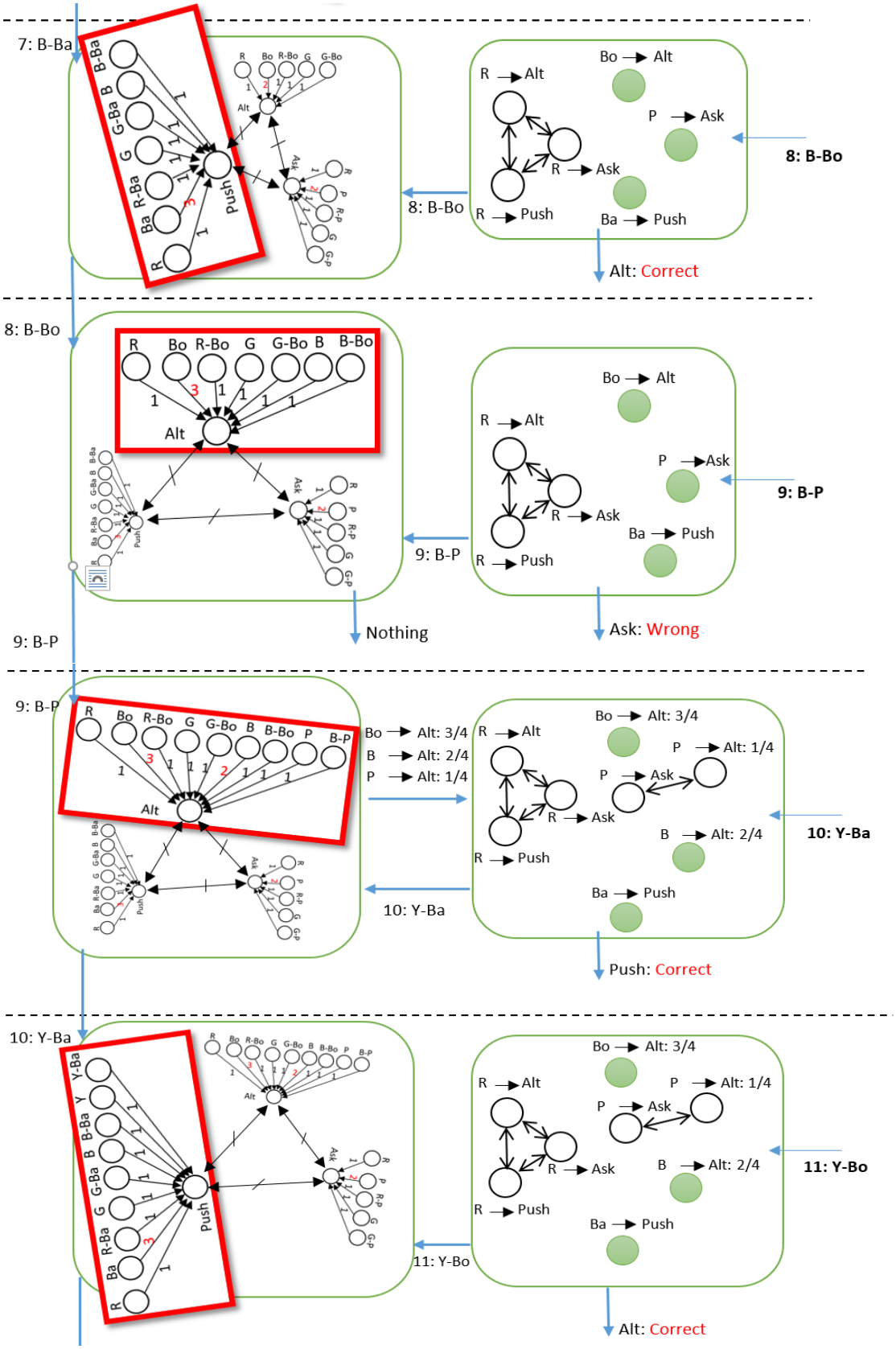


Figure 2.7: Example of argumentation-based learning for the illustrative example. (Part 2/3)

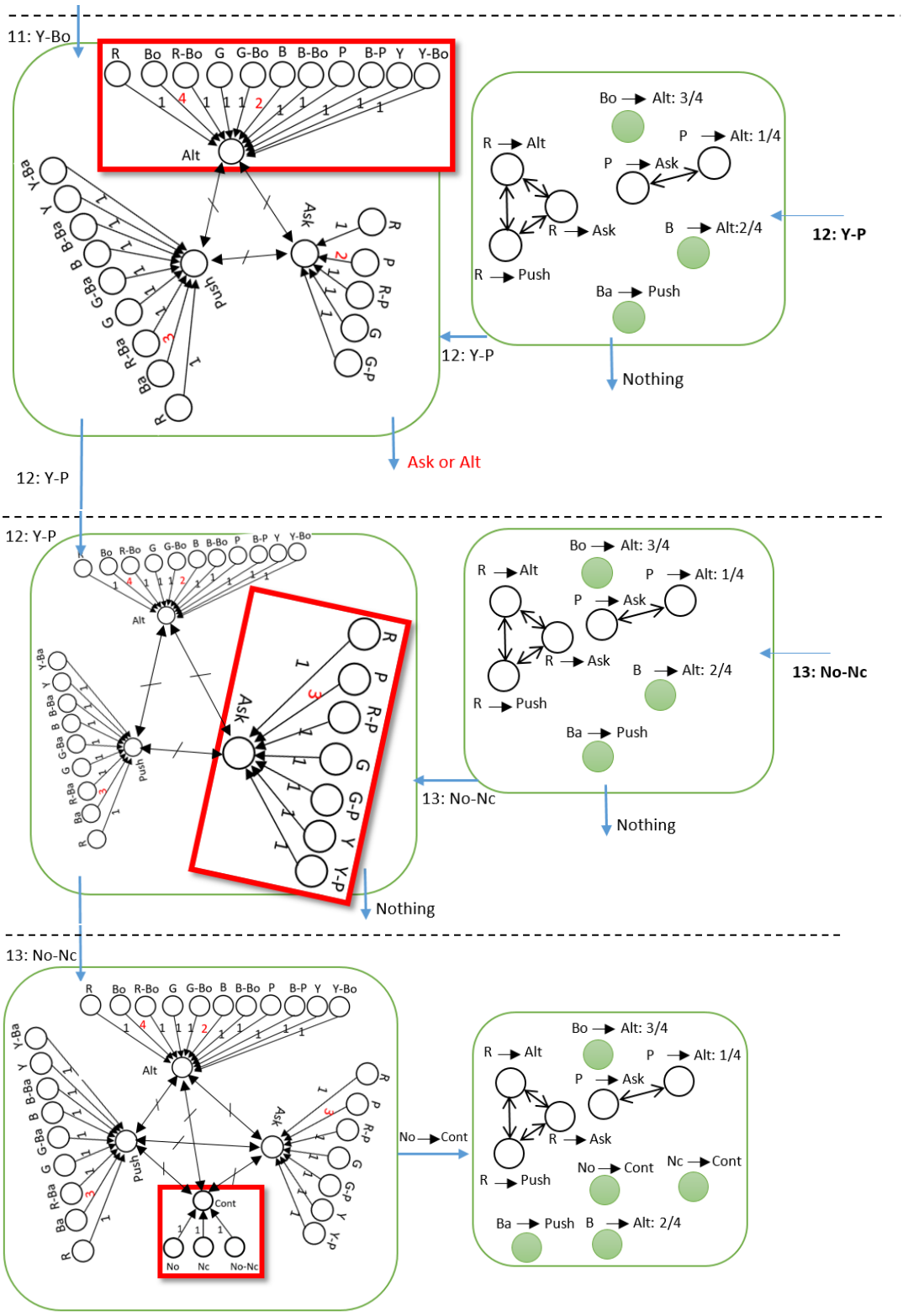


Figure 2.8: Example of argumentation-based learning for the illustrative example. (Part 3/3)

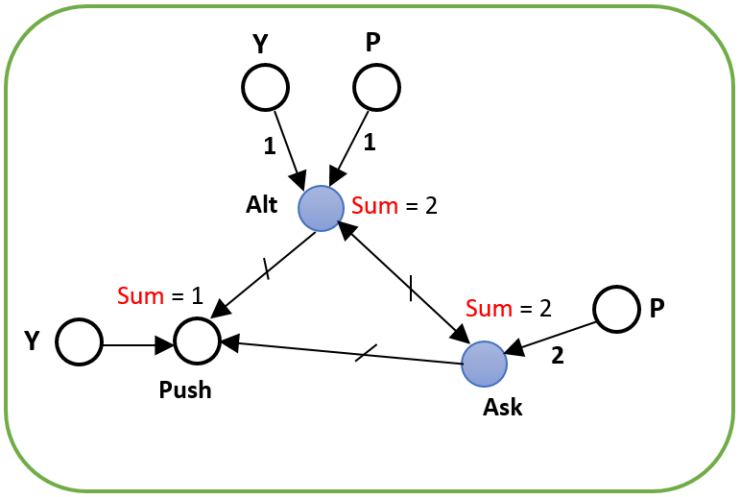


Figure 2.9: The generated *BAF* when Yellow-Person (*12:Y-P*) enters the model. Blue nodes show the intersection of preferred extensions and recovery behavior nodes.

In addition to the set of all the arguments AR, we need to keep track of the set of the Recovery Behavior Nodes (RBN) among the arguments in the following way:

$$RBN_t = \{BRB_0, ..., BRB_{t-1}\} \tag{2.5}$$

The attack relation $R_{att_{t+1}}$ is getting updated using the current set of the Recovery Behavior Nodes RBN_t and the best recovery behavior BRB_t.

$$\begin{aligned} R_{att_{t+1}} = R_{att_t} \cup \{att(BRB_t, b) | b \in RBN_t\} \\ \cup \{att(b, BRB_t) | b \in RBN_t\} \end{aligned} \tag{2.6}$$

The support relations between the arguments are getting updated as follows.

$$R_{sup_{t+1}} = R_{sup_t} \cup \{sup(c, BRB_t) | c \in Combs_t\} \tag{2.7}$$

For instance in the example, when the *1:R-Ba* enters the *BAF* unit (Fig. 2.6), all the combinations of this data instance {*R*, *Ba*, *R-Ba*} are added as support nodes to the current best recovery behavior node, which is *Push*.

There is also a weight function $W_t : R_{sup} \to \mathbb{N}^+$ which specifies the weights of the support relations in R_{sup} at time t. Whenever R_{sup} gets updated, the

Algorithm 2.3: Second Guess Generation Pseudocode

input: Current **BAF** Graph, Set of Recovery Behavior Nodes (**RBN**)
output: Set of recovery behaviors

```
- Generate a new graph G from BAF with the same set of nodes.
- G.sup = BAF.sup
for Node a in RBN do
    for Node b in RBN do
        if a != b then
            if (a.weight > b.weight) then
                G.attacks.add(attack(a,b));
            else if (a.weight == b.weight) then
                G.attacks.add(attack(a,b));
- return the set of nodes in the preferred extension of G.
```

corresponding weights for the support relations update in the following way:

$$\forall c \in Combs_t : W_{sup(c,BRB_t)_{t+1}} = \begin{cases} W_{sup(c,BRB_t)_t} + 1 & \text{if } sup(c, BRB_t) \in R_{sup_t} \\ 1 & \text{otherwise} \end{cases} \tag{2.8}$$

Here, $W_{sup(c,BRB_t)_t}$ is the weight of the support relation $sup(c, BRB_t)$ at time t. Eq. 8 means that if the supporting node c has been already existed in the BAF unit, then its weight is incremented. Otherwise, its supporting weight is set to 1.

2.3.10 Formal Representation of Generating the Second Guesses using *BAF* (Algorithm 2.3)

For generating the second guess using the incoming data instance d_t, another BAF should be constructed. Fig. 2.9 shows the new extracted BAF when the *12:Y-P* enters model. It is almost the same as the main hypotheses generation unit. However, only the attack relations R_{att} should be adapted as follows.

$$R_{att_t} = \{att(a,b) \mid a,b \in RBN_t,\ x,y \in Combs_t, \left(\sum_{(x,a)\in R_{sup_t}} W_{sup(x,a)_t} \geq \sum_{(y,b)\in R_{sup_t}} W_{sup(y,b)_t}\right)\} \tag{2.9}$$

Algorithm 2.4: Hypotheses Generation Pseudocode.

input: Current BAF Graph, Threshold, the best recovery behavior and the latest hypothesis with wrong recovery behavior called **WrongRule**

output: The set of generated hypotheses

- Choose the Best Recovery Behavior node called **BRB**.
- Normalize the supporting weights of BRB to [0, 1].
- Sort **BRB**.supporting-nodes according to their **weight** values from high to low.
- **Sum** = 0;
- **Hypotheses-List** = Empty;

```
for (any sup in BRB.supporting-nodes) do
    if (sup.weight > Threshold) then
        Add sup → BRB to the Hypotheses-List;
for any (A → BRB) in Hypotheses-List do
    for any B → BRB in Hypotheses-List do
        if (A ⊃ B) then
            Remove (A → BRB) from Hypotheses-List;
Add WrongRule.Precondition → BRB to Hypotheses-List;
return Hypotheses-List;
```

Only the recovery behavior node with the higher aggregated supporting weights can attack the other recovery behavior node in the generated BAF. For generating a second guess, the *preferred extensions* semantics is used to choose the best recovery behavior nodes as the second guess. Therefore, the elements in the intersection of the *preferred extensions* set and the set of recovery behavior nodes RBN_t are selected.

2.3.11 Formal Representation of Hypotheses Generation (Algorithm 2.4)

Using the updated Bipolar Argumentation Framework (BAF) from the previous subsection, the set of hypotheses can be generated. Therefore, we can inductively define the hypothesis set as follows:

$$HS_0 = \varnothing;$$

$$HS_{t+1} = \ GenerateHypothesis(BAF_{t+1}, BRB_{t+1}, NC_t); \tag{2.10}$$

Here, NC_t is the hypothesis used in the AF unit and was Not Correctly (NC) determined the action (recovery behavior). We also count the number of times the recovery behavior BRB_{t+1} was the best recovery behavior until now and call it $CBRB_{t+1}$. Each hypothesis in the hypotheses has the form of *precondition (pre)* $\rightarrow$ *post-condition (post): weight* where weight is the hypothesis weight. Whenever a hypothesis is shown in the form *pre* $\rightarrow$ *post* instead of the previous form, it means that the hypothesis weight is equal to 1. The formalization of generating the hypotheses set is as follows:

$$\begin{aligned} HS_{t+1} = \big\{ & (A \rightarrow BRB_{t+1}) : \ weight \mid A \in AR \setminus RBN_{t+1} \\ , \ weight = & \frac{W_{sup(A,BRB_{t+1})_{t+1}}}{CBRB_{t+1}}, \ sup(A, BRB_{t+1}) \in R_{sup_{t+1}} \\ & , \ Normalized(W_{sup(A,BRB_{t+1})_{t+1}}) \geq \ threshold, \\ & \forall a \in A \nexists b \in A : a \subset b \big\} \cup \{(NC_t.pre \rightarrow BRB_{t+1})\} \end{aligned} \tag{2.11}$$

Here, the $threshold \in [0,1]$ and *Normalize* is the linear normalization function for $W_{sup(A,BRB_{t+1})}$. This equation means that when the best recovery behavior is determined, it is used as the post-condition of the hypothesis and its supporting nodes with a weight higher than a specific threshold are chosen as the pre-condition. The hypothesis weight is also computed based on the supporting weight of the supporting node in the pre-condition and the number of times the current recovery behavior was the best recovery behavior so far. Choosing a low *threshold* value means generating more hypotheses. We have tried different values for the *threshold* ranging from 0 to 1 with step-size of 0.1 in the experiments. We found out that $threshold = 0.4$ was a good value in all the experiments. For other datasets, one might need to fine-tune this parameter.

2.3.12 Hypotheses Argumentation Unit using *AF*

As stated in the previous sections, this unit tries to justify what has been learned so far by updating the attack relations between the arguments (hypotheses). The arguments in this framework can only bidirectionally attack each other when they have the same preconditions but different post-conditions.

When a new data instance enters the model, there are three possible cases for the set of hypotheses in the grounded extension of the AF. When the grounded extension of the AF is the empty set, the second guess is generated by the BAF unit. If one argument with the same post-condition exits in the grounded extension of the AF, then this post-condition will be the AF's first guess. If more than one argument with different recovery behaviors in

Algorithm 2.5: Updating Hypotheses Argumentation Unit

input: Current **AF** Graph, the new set of generated hypotheses **HS** from *BAF* unit
output: AF Graph

```
for (all item in HS) do
    - Add item to set of AF.arguments
    - Update the attack relations between arguments as follows
    for (all arg in AF.arguments) do
        if (arg.pre == item.pre) && (arg.post != item.post) then
            AF.attacks.Add(attack(item, arg))
            AF.attacks.Add(attack(arg, item))
return AF
```


their post-condition was chosen, the weights of arguments determine which argument has more power to be selected. For instance in the example, if blue-ball enters the model after it has been trained using the complete set of data in Table-2.1, both $B \rightarrow$ *Alt: 2/4* and $Ba \rightarrow$ *Push:1* can be used for prediction. Since the $Ba \rightarrow$ *Push:1* has higher weight, the *Push* recovery behavior will be chosen, which is the correct choice for this failure state. Notice that in the proposed argumentation-based learning method, it can be proved that the grounded extension is a set of the singletons in the *AF*.

Algorithm 2.5 shows the updating process of the hypotheses generation unit.

2.3.13 Formal Representation of Updating Procedure of Hypotheses Argumentation Unit (Algorithm 2.5)

The hypotheses generation unit is represented by AF_t when data instance d_{t-1} is entering this unit for updating.

$$AF_0 = \langle AR, R_{att} \rangle = \langle \varnothing, \varnothing \rangle$$

$$AF_t = update(AF_{t-1}, HS_t) \quad \forall t \geq 1 \tag{2.12}$$

In time t the Abstract Argumentation Framework (AF) is:

$$AF_t = \langle AR_t, R_{att_t} \rangle \tag{2.13}$$

Here the argument set AR_t is updated at time t using all elements in the recently generated hypotheses set HS_t and the previous arguments set AR_{t-1}

as follows:

$$AR_t = AR_{t-1} \cup HS_t \tag{2.14}$$

The attack relationship R_{att_t} is also get updated whenever two arguments have the same preconditions but different post-conditions:

$$\begin{aligned} R_{att_t} = R_{att_{t-1}} \cup \big\{ att(x,y) | x \in AR_t, y \in AR_t, \\ x.pre = y.pre \wedge x.post \neq y.post \big\} \end{aligned} \tag{2.15}$$

Here, the *att(x,y)* is the abbreviation for $x\ R_{att}\ y$. Figures 2.6, 2.7 and 2.8 show this process. Whenever the hypothesis $R \rightarrow Push$ enters the AF unit, since it has the same precondition but different post-condition with respect to the existing hypothesis $R \rightarrow Alt$ in AF, they will bidirectionally attack one another.

Each time a new data d_t enters the AF unit for the first guess generation, the *grounded extension* called GE_{t+1} is computed. Using $Combs_{t+1}$, the Best matching Hypothesis BH_{t+1} is chosen to generate the first guess in the following way.

$$BH_{t+1} = \big\{ A \in H_{t+1} | B \in H_{t+1}, A.weight \geq B.weight \big\} \tag{2.16}$$

where

$$H_{t+1} = \big\{ h \in GE_{AF_{t+1}} \mid h.pre \in Combs_{t+1} \big\} \tag{2.17}$$

This means that only the hypothesis with the highest weight can be selected as the best matching hypothesis. Subsequently, the first guess is the post-condition of the current best hypothesis:

$$FG_{t+1} = BH_{t+1}.post \tag{2.18}$$

2.3.14 Why not Reinforcement Learning?

Reinforcement Learning (RL) techniques learn by interacting with the environment. Like our proposed method, these methods effectively learn with trial and errors, by performing actions and remembering their consequences [154]. Traditional tabular reinforcement learning methods are inefficient for large state spaces [23]. Moreover, most traditional tabular reinforcement learning techniques do not take the similarity of the features of each state into account, which is needed for the robotic scenarios in this chapter. However, there are

a few exceptions. Some more recent tabular RL techniques have the generalization capability and take the similarity of features into account [54, 190]. In order to include the generalization capability into traditional tabular RL techniques, a non-linear function approximation technique like artificial neural networks is incorporated to handle the large state space and account for the similarity of the features among the states. However, the robotics scenarios in this research have the following properties which make these RL methods behave similarly to a neural network:

- In these scenarios, the next state is not dependent on the current state and the current action because the simulated failure states are generated randomly in the experiments.
- As a consequence of independence between two consecutive states, there is no delayed reward in the corresponding robotic scenarios. Therefore, only the instant rewards, that are dependent on the success of choosing the best recovery behavior at each state, are enough for the formulation.

Considering these properties, the function approximation of the reinforcement learning approach is like a neural network which takes the current state and the current action and outputs the instant rewards. Using such a neural network, the next step is to find the action with the highest instant reward for that state to be selected as the best recovery behavior. This is similar to having a neural network which takes the current state as the input and outputs the best recovery behavior in that state. This network has been implemented as an MLP neural network in the results section. Moreover, we have compared our method with contextual bandit algorithms.

2.3.15 Contextual Bandits

Contextual bandits or associative reinforcement learning techniques have been used for scenarios similar to those studied in this thesis. Therefore, we compare the performance of the proposed ABL technique with various online contextual bandit algorithms.

Contextual bandit is defined as follows. There is an agent who can choose between a number of choices (known as "arms"), which can lead to stochastic rewards. In each round, the current state is generated, which is a set of features of a fixed dimensionality that is known as "context". The agent chooses an arm at each round and the corresponding reward for that action in that specific context is returned as a feedback to the agent. The ultimate goal of the agent is to find a policy that maximizes the long-term rewards using the history of previous actions.

Most research on finding an efficient algorithm for contextual bandit problems in the last decade can be divided into two categories, namely Upper Confidence Bounds based algorithms (UCB) [52, 192, 74, 87] and Thompson Sampling algorithms (TS) [52, 106, 4, 1]. Zhou el al. [191] proposed an offline multi-action learning approach which can take constraints on the learning policy into account, for instance budget constraints. In Section V, we will compare our method with both UCB and TS approaches.

2.3.16 Generalizing *ABL* to Other Real-Word Scenarios

So far, we have assumed that at each failure state only one recovery behavior is successful and the others fail. However, this assumption might not be the case in all the real-word scenarios. Therefore, in the following paragraphs, we explain how we can generalize the *ABL* method to handle multiple successful behaviors.

Like Reinforcement Learning (RL) techniques, each action (i.e. a recovery behavior in our case) must have a reward reflecting how good it is. For example this reward can be a function of the run-time of that recovery behavior where the lower run-time leads to higher reward. This reward function for each recovery behavior can be formulated as follows.

$$R = \begin{cases} 0 & \text{Failure} \\ \frac{1}{\text{run-time}} & \text{Successful Recovery Behavior} \end{cases} \tag{2.19}$$

Using the epsilon-greedy algorithm [154] for choosing the different recovery behaviors at each failure state, we are able to have a trade-off between the exploration of the new recovery behaviors and the exploitation the previously successful recovery behaviors. When a new recovery behavior is explored and it is successful, then the BAF unit in *ABL* should generate a new set of hypotheses based on that recovery behavior and its run-time.

Furthermore, the hypothesis format in the AF unit of the *ABL* method should be changed. The new hypotheses have the form *pre* $\rightarrow$ *post : weight : reward*. Subsequently, the set of hypotheses with the highest rewards in the grounded extension of the AF unit is found for choosing the best recovery behavior. If the hypotheses used in the previous step have the same rewards, the *weight* is used to choose the best recovery behavior as before.

The required changes in the *ABL* algorithm are listed below:

- The line "choose the Best Recovery Behavior node called **BRB**" in the Algorithm 2.4 should change to "choose the Best Recovery Behavior node called **BRB** based on the epsilon-greedy algorithm ($\epsilon = 0.05$)"

- The Reward is added to the format of each hypothesis.

$$pre \rightarrow post : weight : reward$$

- The Best Recovery Behavior from the AF unit is chosen based on the grounded extension, the reward and the weight of each hypothesis.

This methodology is only needed when we don't know the rewards of each recovery behavior (action) in advance. If we previously know the rewards for each of the recovery behaviors (as in our experiments), with the following modification, we can use the same *ABL* method as before. At each point a trial and error procedure takes place based on the ordering of the recovery behaviors from the one with the highest reward (lowest run-time) to the one with the lowest reward (highest run-time). This guarantees that the first successful recovery behavior is always a best choice.

2.4 Experiments

In this section, we compare the performance of our proposed *ABL* method with other incremental learning techniques and contextual bandits algorithms. We have utilized a device with Intel(R) Core(TM) i7-7700HQ 2.81 GHz processor and 16 GBs RAM using windows 10 OS for all the experiments. The survey by V. Losing et al. compared a broad range of incremental online machine learning techniques [112]. Using the key methods in their survey, we are also comparing the proposed method with Incremental Support Vector Machine (*ISVM*) [38, 72, 73], incremental decision tree based on *C4.5* [133] and ID3, incremental Bayesian classifier [3], Online Random Forest (*ORF*)[145] and Multi-Layer Neural Networks for classification with localist models like Radial Basis Functions (*RBF*) which work reliably in incremental settings [135, 113].

Cortes has recently compared the performance of different contextual bandit algorithms in his paper [47]. He adapted Multi-Arm Bandits (MAB) policies to contextual bandit algorithms. We have also compared our proposed ABL technique to various online contextual bandit approaches.

2.4.1 Performance Measure

The mean performance of each method is calculated over 1000 independent runs. Each run for the robotic scenarios consists of 200 failure recovery attempts. Since the order of the failures has a direct effect on all the open-ended online incremental learning methods like ours, the order of failures is randomized for each run in which there is an equal uniform probability for

each solution to be a success. We measure the accuracy using the following performance metric:

$$accuracy = \frac{\text{number of successful attempts}}{\text{number of all the attempts that has been tried so far}}$$

We are interested in knowing whether the method picked the best recovery behavior or not for a given failure state.

For the third test scenario, we randomly choose 40 data instances for online train and test procedure.

Notice that all the methods use the same randomly generated data set compatible with the conditions mentioned in the test scenarios.

Furthermore, the mapping of each state to the best recovery behavior (a table like Table 2.1), which is used for testing the performance of the model, is randomly generated at each of the 1000 independent experiments.

2.4.2 Comparison criteria

In the robotic scenarios, we need a learning approach which can quickly learn to recover from failure states in a low number of attempts. Moreover, for other test scenarios, the goal is to incrementally learn from a lower number of training instances. Therefore, the increase in learning accuracy in a lower number of attempts is one important criterion (which we call *learning speed*) to evaluate the efficiency of the method [21]. Therefore, learning curves with the highest steepness in a smaller number of attempts are desirable. Furthermore, the *final learning accuracy* is also an important criterion.

2.4.3 Comparison Methods

The first method utilized for comparison is a incremental naive Bayesian classifier [137]. We use exactly the same suggested parameters as [137] in the experiments. The second categories of methods that are used for comparison are rule extraction and decision-tree based methods. The *PART* algorithm is based on the *C4.5* decision tree classification method [175]. *PRISM* is an algorithm for inducing modular rules [39, 174]. The *ID3* algorithm constructs an unpruned decision tree for classification [132]. The *J48* algorithm is also based on a pruned or unpruned *C4.5* decision tree [133]. The incremental version of decision tree algorithm is discussed in [177]. We used the standard Weka[1] machine learning software for the implementation of these methods with suggested parameter values.

[1] https://www.cs.waikato.ac.nz/ml/weka/

The incremental version of the random forest algorithm is called Online Random Forest (*ORF*) [145]. We have used the same suggested parameters in the implementation of the online random forest method. The Multi-Layer Perceptron (*MLP*) neural network is also used for the comparison. An experiment has been conducted to find the best number of hidden layers and the best number of hidden neurons at each layer. Specifically, we have tested MLPs with the range of 1 to 10 number of hidden layers and for each hidden layer, we have tried 1 to 100 hidden neurons with ReLU activation function. The architecture with 2 number of hidden layers and 30 hidden neurons in the hidden layers turned out to perform best for all the scenarios. Notice that a high number of hidden units and nodes leads to over-fitting of the model in the initial iterations and a low number of hidden units leads to under-fitting of the model and low learning capacity of the model in the final iterations. Therefore, we chose four hidden layers with 10 nodes in each layer which had good results in our experiments. The final algorithm for the comparison is Incremental Support Vector Machine (*ISVM*). We tried different non-linear kernel types for the *ISVM* method, namely, the polynomial kernel functions, sigmoid kernel function and the radial basis kernel function. Consequently, we have chosen Radial Basis Function (RBF) for conducting all the experiments.

We have utilized several online contextual bandit approaches for our comparisons including bootstrapped upper confidence bound [47, 75], bootstrapped Thompson sampling [47, 58] and some other methods from [47], including epsilon greedy, adaptive greedy, explore-then-exploit, exploration based on active learning, softmax explorer and exploration based on active learning approaches. For all these approaches, we have utilized the implicit parameter fine-tuning method in the original implementation.

Notice that to fairly compare the ABL approach with all the contextual bandit approaches, we have used the same procedure as ABL for training the contextual bandit models. This means that we have also used the best choice of action at each state to update a contextual bandit model if it fails to predict the correct action at that state.

2.4.4 Results

As one can see in Fig. 2.10, Fig. 2.11 and Fig. 2.12, the proposed Argumentation-Based Learning (*ABL*) method outperforms all the other methods in both the comparison criteria used for this research, namely, the final learning accuracy and the learning speed. The steepness of the learning curve shows that the *ABL* learns faster in a lower number of iterations.

For the first test scenario, after observing 30 failure states, *ABL* achieves 74% accuracy, while the best method among others has 60% accuracy. The

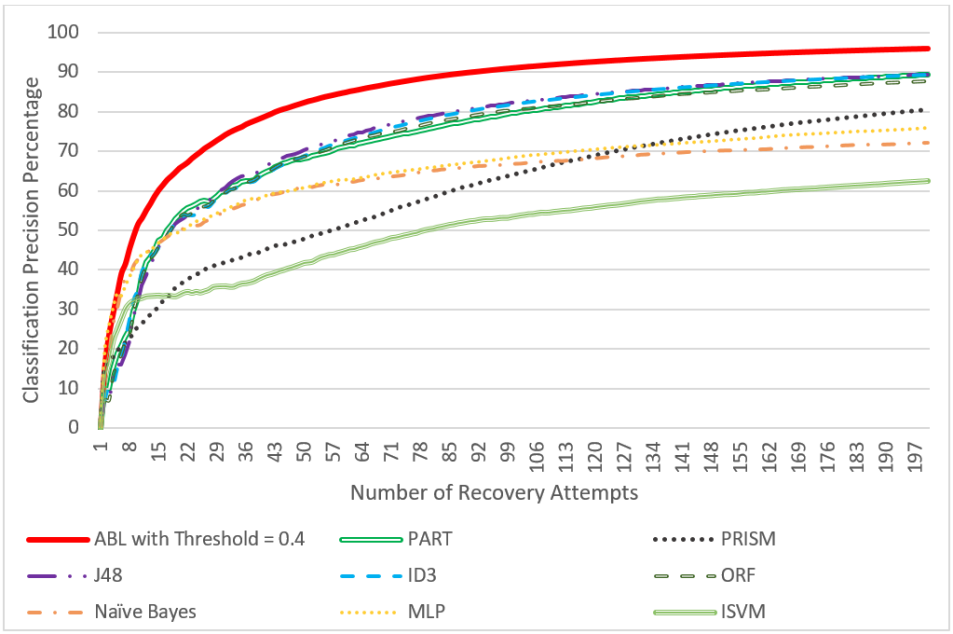


Figure 2.10: Comparison of Argumentation-Based Learning (*ABL*) with key methods for incremental online learning [112] using the first test scenario.

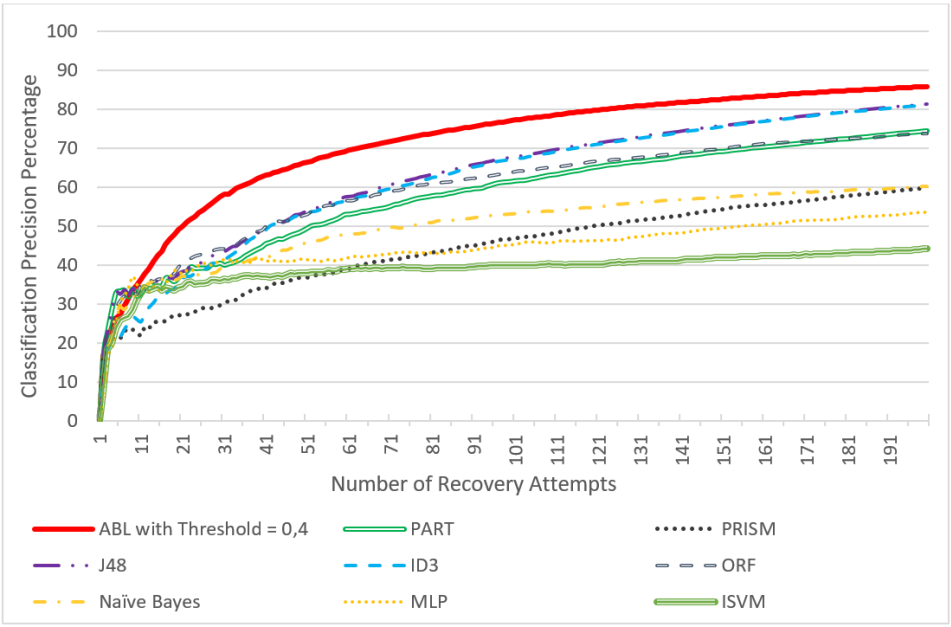


Figure 2.11: Comparison of Argumentation-Based Learning (*ABL*) with key methods for incremental online learning [112] using the second scenario.

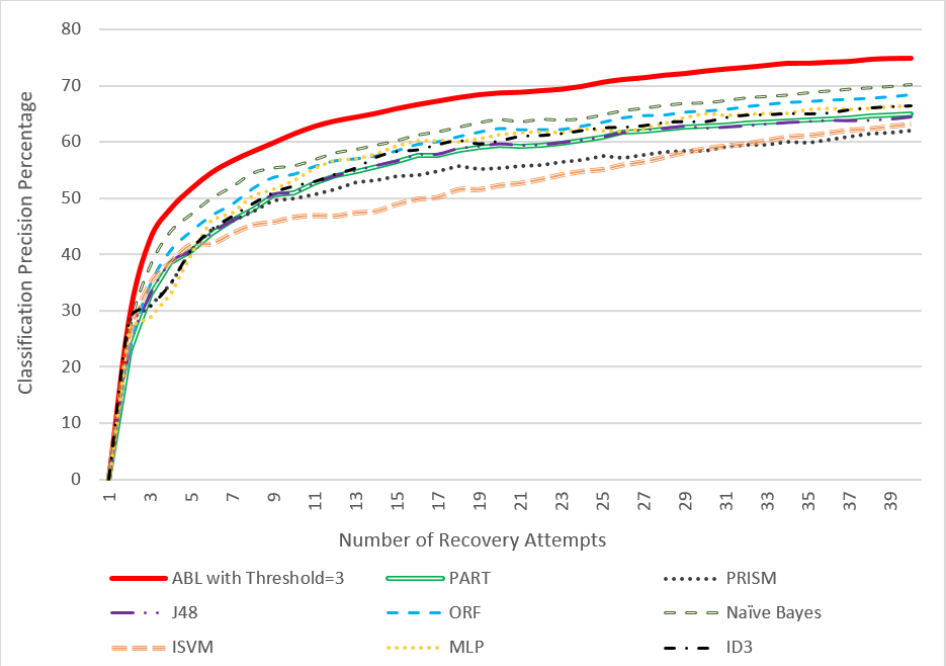


Figure 2.12: Comparison of Argumentation-Based Learning (*ABL*) with key methods for incremental online learning [112] using the third scenario. This non-robotic scenario emphasizes that the proposed ABL method is generalizable to other online incremental learning scenarios.

final accuracy of *ABL* is 95%, while the best final accuracy among other methods is 90%. For the second test scenario, after 30 observations, the *ABL* has 58% accuracy while J48 as the best performer among all other methods has 42% accuracy. Moreover, the final accuracy of *ABL* for the second test scenario is 85% while J48 and *ID3*, the best among all others, achieve almost 80% final accuracy.
In the third scenario, which differs from the two prior scenarios in context, *ABL* repeatedly outperforms all the other methods in both of the comparison criteria. Among other methods, incremental naive Bayes and incremental random forest (*ORF*) have better results. The final learning accuracy of *ABL* in this scenario is 75% while it is 70% for the incremental naive Bayes method. The slope of the learning curve also shows the faster learning speed of *ABL* with respect to all of the other methods.

Figure 2.13, 2.14 and 2.15 show the comparison of ABL with contextual bandits using the first and second scenario as well as the mushroom dataset. In all experiments, ABL outperforms the other approaches considerably, both in terms of learning speed and in terms of final learning accuracy. The explorative nature of contextual bandit algorithms may lead to this difference in

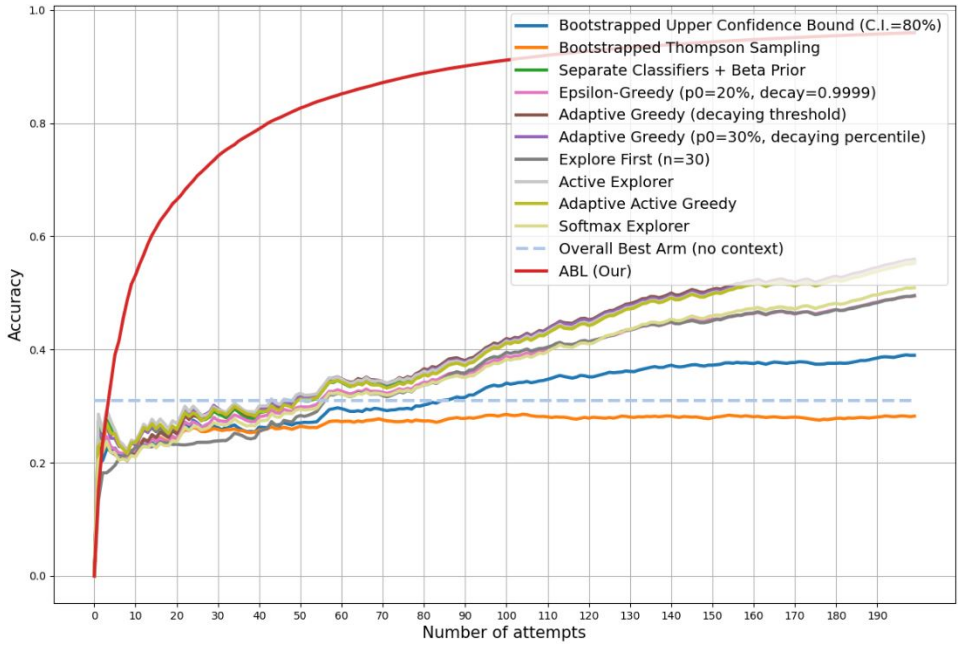


Figure 2.13: Comparison of Argumentation-Based Learning (*ABL*) and some contextual bandit methods [47] for the first scenario. The red curve shows the accuracy of ABL.

performance.

2.5 Discussion

A key reason that the proposed method works better than Naive Bayes originates from the independence assumption between all features in the Naive Bayesian formulation. In the case of neural networks, considering that there is only a small number of training data instances, a complex neural network tends to over-fit and a small neural network leads to under-fitting. Choosing the best neural network architecture dynamically according to the number of visited data is also a challenging task. On the other hand, decision-tree based techniques fail at the initial recovery attempts and then gradually learn the best recovery behavior. This is because of the change in entropy or information gain when new unforeseen data updates the decision tree. This is also the case with the Online Random Forest (*ORF*) method. Furthermore, *ISVM* does not perform well in circumstances where only a few features are associated with predicting the class label. In all the above cases, the suggested *ABL* approach performed better as it considers any possible dependence between features and

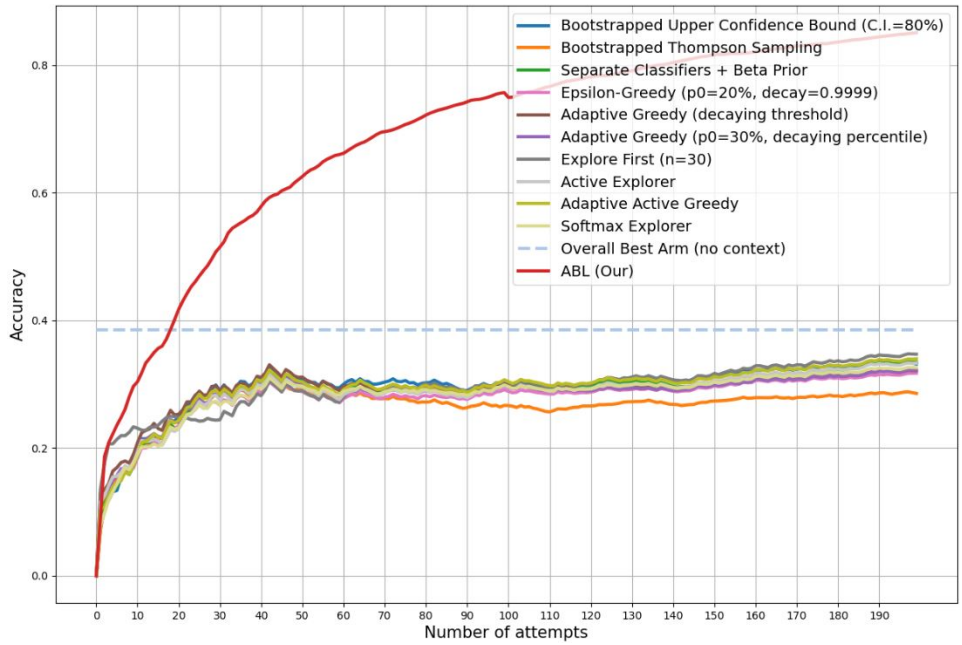


Figure 2.14: Comparison of Argumentation-Based Learning (*ABL*) and some contextual bandit methods [47] for the second scenario.

it can immediately focus on features which are most relevant for the optimal decision.

Moreover, *ABL* leads to an explicit representation of the learning process understandable for humans, as is also the case with decision-tree based techniques. In contrast, neural networks, support vector machines and Bayesian techniques are all black boxes [188] (this means that the trained models are not easily interpretable and explainable) for the humans. This explicit representation of the learning process can be utilized in combination with human-robot interaction. Employing this property, *ABL* can be used in multi-agent scenarios where agents can transfer their knowledge to each other.

The proposed *ABL* method has a limitation. It handles data sets with discrete feature values. This limitation can be addressed in future works. The complexity of ABL is exponential and we have analyzed its complexity in Chapter 3. AABL addresses the complexity issue.

Consequently, the proposed argumentation-based incremental learning algorithm could learn in fewer attempts with higher accuracy than other algorithms used for comparison. The results have also shown that ABL outperforms contextual bandit algorithms in terms of learning accuracy. There is a big gap between the performance of ABL and contextual bandits approaches for the robotics scenarios. The reason that we have such a big gap is that

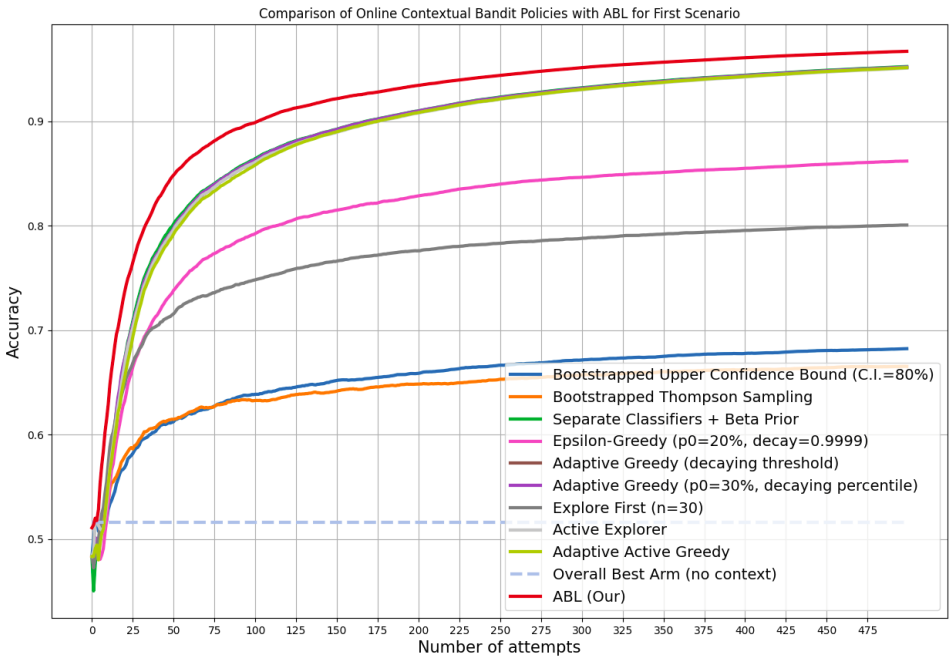


Figure 2.15: Comparison of Argumentation-Based Learning (*ABL*) and some contextual bandit methods [47] for the mushroom dataset.

the robotic experiments are randomly generated. However, contextual bandits approaches intrinsically deal with the trade-off between exploration and exploitation. This means that for such a dynamic dataset with high randomness the exploration policy make them not suitable. However, ABL can handle these type of datasets since it always prioritize exploitation to exploration. For a more standard dataset (in the contextual bandits setting) like the mushroom dataset, the performance of ABL is more close to the contextual bandits algorithms. Moreover, ABL extracts an explicit set of rules that explain the knowledge acquired by the agent over the interaction with the environment. This feature makes the method more explainable and easy to debug by an expert.

Therefore, this method can be a good alternative when the feature values are discrete. Although we have shown that the current ABL approach is working well for the aforementioned scenarios in this thesis, these results are limited to datasets with discrete feature values that are not high-dimensional. To make ABL more efficient for higher dimensional problems, we have introduced Accelerated Argumentation-Based Learning (AABL) [17] to improve the space and computational complexity of the method.

2.6 Conclusion

General purpose service robots should be able to recover from unexpected failure states caused by environmental changes. In this article, an argumentation-based learning (*ABL*) approach is proposed which is capable of generating relevant hypotheses for online incremental learning scenarios. This set of hypotheses is updated incrementally when unforeseen data enters the model. The conflicts among these hypotheses are modeled by Abstract Argumentation Frameworks.

The performance of *ABL* has been evaluated using both the robotics and the non-robotics incremental learning scenarios. The third scenario, which has a non-robotic context, is a publicly accessible dataset from the UCI machine learning repository. This scenario shows the fact that the proposed *ABL* method can be used in any online incremental learning application with discrete feature values. Moreover, we have also compared the performance of different contextual bandit algorithms with ABL. According to these experiments, the proposed method learns faster and with higher ultimate classification accuracy than various state-of-the-art online incremental learning methods.

CHAPTER 3

Argue to Learn

Accelerated Argumentation-Based Learning

This chapter is based on [17].This chapter addresses the issue of the high computational complexity of the argumentation-based learning technique proposed in Chapter 2. It uses a simplified model based only on bipolar argumentation frameworks. Moreover, the proposed model uses two strategies to reduce the complexity of the model. The resulting model has polynomial complexity with respect to the number of features, while the original argumentation-based learning technique has exponential complexity with respect to the number of features in the dataset. This makes the resulting model more suitable for higher dimensional datasets. Moreover, the accelerated argumentation-based learning method achieves higher learning speed and learning accuracy compared to the original model.

Argue to Learn: Accelerated Argumentation-Based Learning

Abstract

Human agents can acquire knowledge and learn through argumentation. Inspired by this fact, we propose a novel argumentation-based machine learning technique that can be used for online incremental learning scenarios. Existing methods for online incremental learning problems typically do not generalize well from just a few learning instances. Our previous argumentation-based online incremental learning method outperformed state-of-the-art methods in terms of accuracy and learning speed. However, it was neither memory-efficient nor computationally efficient since the algorithm used the power set of the feature values for updating the model. In this chapter, we propose an accelerated version of the algorithm, with polynomial instead of exponential complexity, while achieving higher learning accuracy. The proposed method is at least 200× faster than the original argumentation-based learning method and is more memory-efficient.

3.1 Introduction

Argumentation-Based Learning (ABL) [16, 18] outperformed other online incremental learning approaches and was shown to be successful for handling unforeseen failures. However, ABL is not efficient in terms of space and computational complexity. Therefore, the current ABL approach is not usable for high dimensional datasets. In this chapter, we propose a novel Accelerated Argumentation-Based incremental online Learning (AABL) method that has a lower space and computational complexity and higher learning accuracy. This entails lower run-time and memory consumption. Moreover, like the original ABL, AABL can generate a set of explainable hypotheses (arguments) for predicting the best recovery behavior (class label).

3.1.1 Argumentation in Machine Learning

Argumentation is a reasoning model based on interaction between arguments [164]. Argumentation has been used in various applications such as non-monotonic reasoning [140], inconsistency handling in knowledge bases [165], and decision making [13]. In [57], Dung has defined an Abstract Argumentation Framework (AF) as a pair of the arguments and a attack binary relation among the arguments. Extending Dung's idea, some arguments can support a conclusion and others might be against (attacking) that conclusion in the Bipolar Argumentation Framework (BAF) [9]. According to a survey by Cocarascu et al. [46], the works using argumentation in supervised learning are listed as follows. Argumentation-Based Machine Learning (ABML) [120] uses the CN2 classification approach [44]. This method uses experts' arguments to improve the classification results. The paper by Amgoud et al. [11, 45] explicitly uses argumentation. There are other approaches for improving classification using argumentation in the literature [36].

3.1.2 The Expansions

This research is an expansion of our previous paper [16, 18]; see Chapter 2 of this thesis. The specific expansions are listed as follows.

- Proposing a simpler architecture of the model using only a BAF rather than using both the AF and a BAF.
- Accelerating the prediction and update procedure of the model by introducing an algorithm with lower space and computational complexity by going from exponential to polynomial complexity.

- Including more evaluation scenarios with different levels of complexities.
- Adding run-time and memory usage analysis for both the proposed and previous ABL method.
- Specifying the algorithms in the proposed method by adding pseudocodes to explain argumentation-based learning in more detail.

3.2 Background

The bipolar argumentation framework [9] is the main building block of our proposed accelerated argumentation-based learning approach. Argumentation-based learning and BAF are formally defined in this section.

3.2.1 Argumentation-Based Learning

Using the combination of AF and BAF, argumentation-based learning has been proven to outperform state-of-the-art online incremental learning methods [16, 18]. ABL extracts a set of relevant hypotheses from the learning instances in an online manner and explicitly represents the knowledge acquired from the learning instances as an explainable set of rules as arguments and defeasibilty relations among them. ABL can learn with fewer learning instances. However, it lacks the ability to work with higher dimensional data since it uses all the subsets of the feature values as the supporting nodes and this makes the model slow and not memory-efficient. In this chapter, we will propose a new argumentation-based learning approach which resolves these issues.

3.2.2 Formal Definition of an Abstract Bipolar Argumentation Framework

An Abstract Bipolar Argumentation Framework (BAF) [9] is a triple of the form $\langle AR, R_{att}, R_{sup} \rangle$ where AR is the finite set of arguments, $R_{att} \subseteq AR \times AR$ is the *attack* set and $R_{sup} \subseteq AR \times AR$ is the *support* set. Considering A_i and $A_j \in AR$, then $A_i \ R_{att} \ A_j$ means that A_i attacks A_j and $A_i \ R_{sup} \ A_j$ means that A_i supports the argument A_j.

The semantics of BAF are as follows:

(**Conflict-Free**) Let $S \subseteq AR$. S is conflict-free iff there is no $B, C \in S$ such that B attacks C.

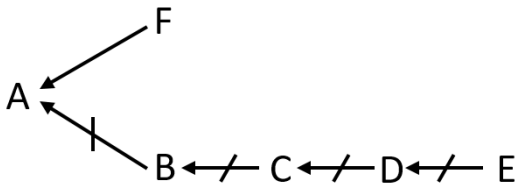


Figure 3.1: A Bipolar Argumentation Framework (BAF).

(**Admissible set**) Let $S \subseteq AR$. S is admissible iff S is conflict-free, closed for R_{sup} (if $B \in S$ and $B\ R_{sup}\ C \Rightarrow C \in S$) and S defends all its elements.

(**Preferred extension**) The set $E \subseteq AR$ is a preferred extension iff E is inclusion-maximal among the admissible sets. An inclusion-maximal set among a collection of sets is a set that is not a subset of any other set in that collection.

Figure 3.1 shows a bipolar argumentation framework. The admissible sets are {}, {E}, {A, C, E}, {A, C, E, F}. The preferred extension in this BAF is {A, C, E, F}.

3.2.3 Online Incremental Machine Learning Algorithms

A recent study on the comparison of the state-of-the-art methods for incremental online machine learning [112] shows that Incremental Support Vector Machines (*ISVM*) [152, 97] together with LASVM [34], which is an online approximate SVM solver, and Online Random Forest (*ORF*) [27] outperform the other methods. The original ABL approach outperformed all these methods in terms of accuracy and learning speed [16, 18]. Therefore, we only compare the proposed AABL method with the original ABL approach. Both AABL and ABL can be utilized in open-ended learning scenarios [20].

3.3 Scenarios

The performance of the different methods is tested using three test scenarios. The aim of these test scenarios is to model a situation where a programmer has provided an initial solution (e.g., a top level behavior such as entering the room), while he has not accounted for all possible failures (e.g., objects and persons blocking the entrance), allowing, however, the robot to find new solutions whenever a (previously unseen) failure occurs.

The basic setup of the test scenarios is illustrated in Fig. 3.2. The high-level behavior of the robot aims to proceed from the initial location to the

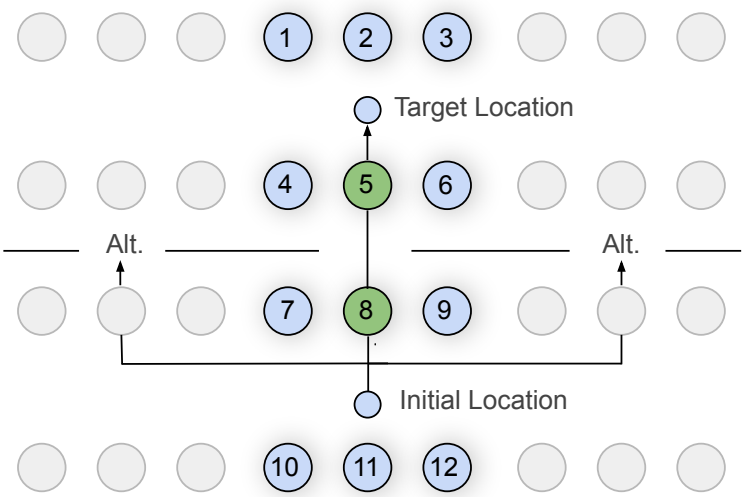


Figure 3.2: Schematic overview of the possible failure state scenarios. Only the green location is relevant for finding the best recovery behavior. Alt. stands for the Alternative Route recovery behavior.

target location using three entrances. Different obstacles might be on its way to the target location. Looking at all the obstacle locations at once, the robot can reach the goal by choosing the best recovery behavior.

3.3.1 Recovery Behaviors

Whenever the robot is confronted with an obstacle, it may use any of the following recovery behaviors to resolve the issue:

- Continue: This solution is only useful if the failure has resolved itself (e.g., the obstacle moved away).
- Push: The robot can try pushing any obstacle.
- Ask: The robot can try to ask any obstacle to move.
- Alternative Route (Alt): The robot can move to another entrance to reach the target location.

It is important to note that choosing Alternative Route as the best recovery behavior may not always lead to success, because the robot may again encounter new obstacles (Fig. 3.2). Moreover, the best recovery behavior depends on the type, color and location of the obstacle.

3.3.2 Test Scenario 1

In this scenario, three concepts (ball, box or person) with four colors (red, blue, green or yellow) can be presented in one of the locations 1 to 6 (Fig. 3.2). Locations 7 to 12 play no role in this scenario. There can be either zero or one combination of color-concept in each location. Only location number 5, marked in green (Fig. 3.2), is relevant for choosing the best recovery behavior. It is important to notice that the robot does not know this fact and it should infer it by itself.The number of possible combinations of the color-concept in each location is 13 (3 types $\times$ 4 colors + "no obstacle" = 13). Since there are 6 locations in this scenario, the number of all possible states in this scenario is $13^6 = 4,826,809$.

Notice that colors can have meaningful interpretations. For instance, the red object might be heavy and cannot be pushed, while green ones are light. Using the colors instead of these realistic features simplifies the scenarios with fewer features.

3.3.3 Test Scenario 2

This scenario is more complex than the first scenario, since each color-concept combination can be presented in one of the locations 1 to 9 (Fig. 3.2). Here, only the green locations 5 and 8 are relevant for determining the best recovery behavior. The number of all possible states in this scenario is $13^9 = 10,604,499,373$.

3.3.4 Test Scenario 3

The third scenario is the most complex scenario in all the scenarios. Each color-type combination can be presented in any of the twelve different locations (Fig. 3.2). Like the previous scenario, only locations number 5 and 8 play a role for determining the best recovery behavior. In this Scenario, the number of all the possible states is $13^{13} = 302,875,106,592,253$.

3.4 Method

In this section, with an illustrating example, we first explain the Bipolar Argumentation Framework (BAF) unit which is the main building block of the proposed approach. Subsequently, we define AABL, and its updating procedure.

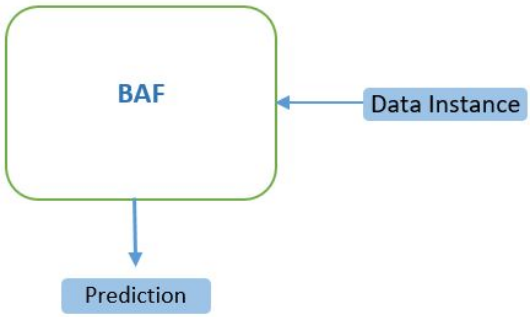


Figure 3.3: Architecture of the proposed Argumentation-based learning approach.

Order	Color	Concept	Best Recovery Behavior
1	Red	Ball	Push
2	Red	Box	Alternative Route
3	Red	Person	Ask
4	Green	Ball	Push
5	Green	Box	Alternative Route
6	Green	Person	Ask
7	Blue	Ball	Push
8	Blue	Box	Alternative Route
9	Blue	Person	Alternative Route
10	Yellow	Ball	Push
11	Yellow	Box	Alternative Route
12	Yellow	Person	Ask
13	None	None	Continue

Table 3.1: Possible combinations of color-type with the best recovery behaviors

3.4.1 Explanation of the Method with an Illustrating Example

We first use the simplified version of the test scenarios with only one location ahead of the agent (instead of 6, 9 or 12 locations). Figure 3.3 shows the architecture of the proposed argumentation-based learning approach.

Using the randomly generated Table 3.1, the robot is initially confronted with a Red-Ball (R-Ba) and tries different recovery behaviors to find out that the best choice is *Push*. The model initially gets updated by the subsets of feature values with size 1 ($L := 1$). This means that the supporting nodes R and Ba are added to the *Push* recovery behavior (Fig. 3.4). Subsequently, the agent encounters a Red-Box (R-Bo) for which the subsets of feature values

with size $L = 1$ consist of R and Bo. Looking at the current state of the BAF, R supports the *Push* recovery behavior and it is chosen as the model's prediction. Since, this is a wrong choice, the agent try other recovery behaviors and find "Alternative route" (*Alt*) as the best recovery behavior. Therefore, the *Alt* node gets updated with its supporting nodes R and Bo and also a bidirectional attack among *Alt* and *Push* nodes. Since R supports both *Push* and *Alt* recovery behaviors, it is not a unique supporter for each of them and it will be pruned from both the recovery behaviors and will be marked as a node which can no longer support any recovery behavior nodes in the future.

For the third learning instance the robot is confronted with a Red-Person (R-P) and the models does not have any prediction since no current recovery behavior node in the BAF has either P or R in its supporting nodes. The BAF unit gets updated with only P as a supporting node for the Ask since R has been previously marked as a non-supporting node and bidirectional attack relations are added among all pairs of the recovery behaviors. Subsequently, the agent encounters with a Green-Ball (G-Ba) obstacle and since Ba supports the *Push* in the BAF unit, *Push* is chosen as a prediction for the best recovery behavior. The BAF gets updated using G supporting node for *Push* recovery behavior. This process will continue and the model predicts the best recovery behavior correctly in the subsequent obstacle confrontations until the agent encounters a blue person.

When the agent encounters a Blue-Person (B-P), it chooses Ask as the best recovery behavior. This is a reasonable choice because Ask was the best recovery behavior for all the previous cases where a *Person* was the obstacle, namely, in R-P and G-P. However, it turns out that the best choice for B-P is *Alt*. Since, the subsets of feature values with size 1 were not adequate for choosing the best recovery behavior for both the Ask and the *Alt* recovery behaviors, the size of the subsets of the feature values is incremented i.e. $L := L+1$ in this case $L = 2$. Therefore, the supporting node B-P is added to the *Alt* recovery behavior and the supporting nodes R-P and G-P are added to the Ask recovery behavior while P is pruned and marked as a non-supportable node. The model predicts the correct categories for the rest of the learning instances until it is confronted with a Yellow-Person (Y-P). In this case the model does not have any guess for the best recovery behavior and gets updated with the Y-P supporting node. The last learning instance None-None is a new case based on the previous agent's experiences and it does not have any prediction for that. Finally the model gets updated and a new recovery behavior node *Continue* (*Cont*) is added to the BAF model. In this illustrating example, the proposed approach has seven correct predictions and two wrong predictions out of all the thirteen instances while having no other predictions for the other four cases. Notice that fixing the upper bound of L to 2 leads to good results

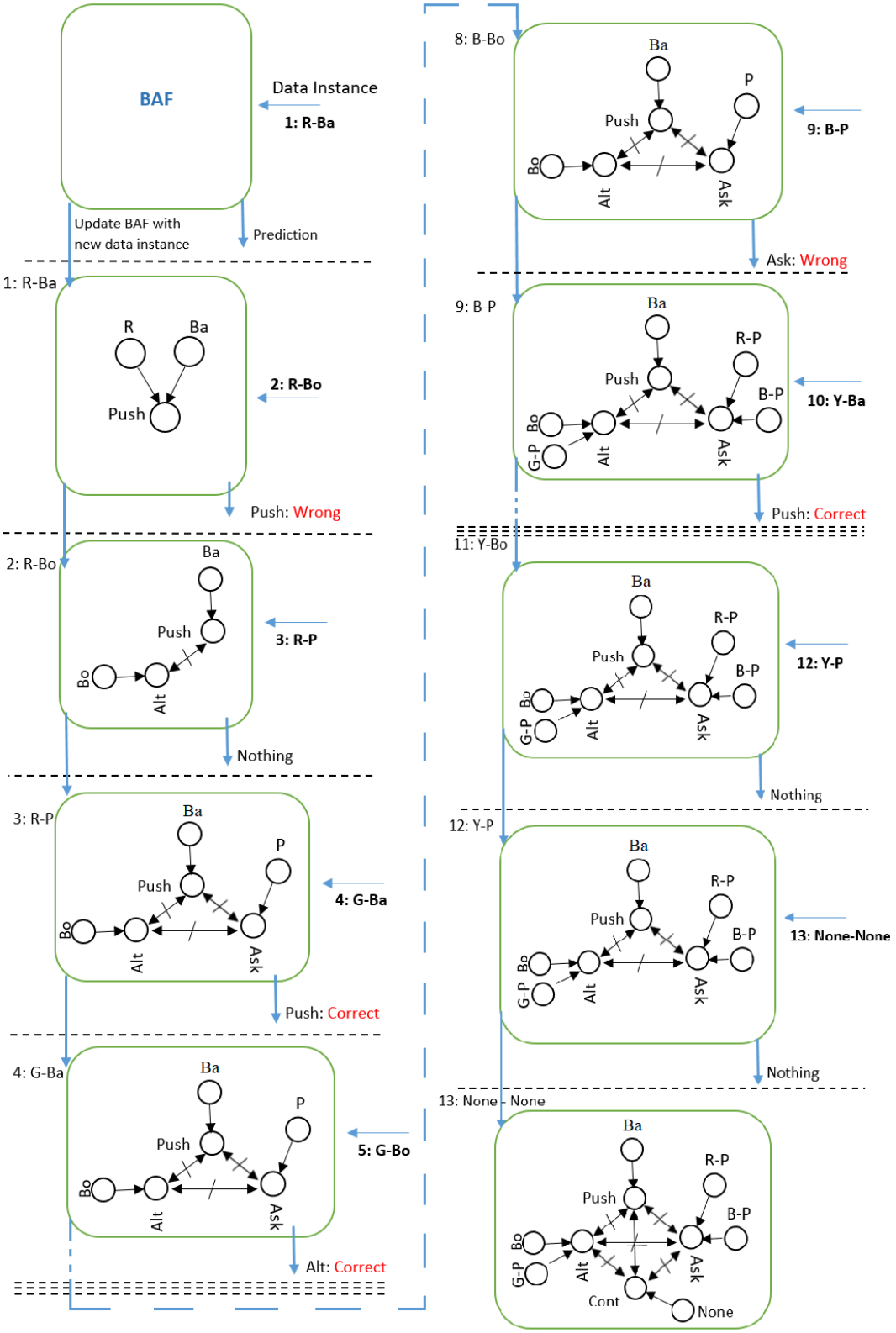
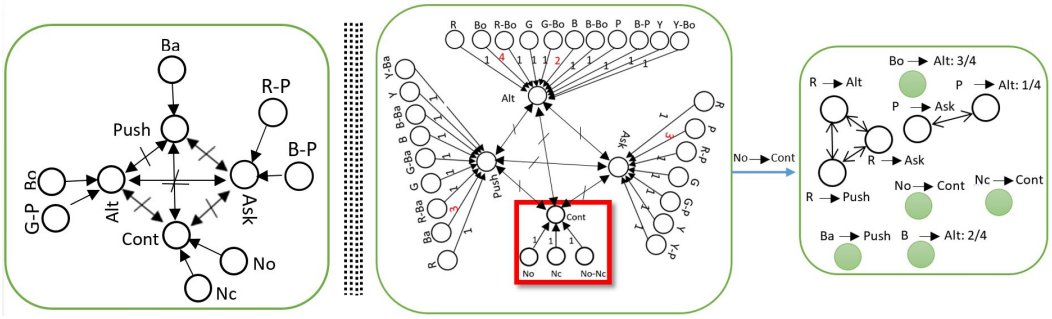


Figure 3.4: Example of Argumentation-Based Learning for the illustrating example. First part



a) Our Approach b) Original ABL approach [16, 18]

Figure 3.5: Comparison of the ABL models after training on the illustrating example

in practice. This ensures the polynomial complexity of the resulting approach.

Comparing the number of nodes between the proposed model and the previous argumentation-based learning model [16, 18], the number of saved nodes in memory for our proposed approach decreases from 44 to 11 and the number of attack or support relations decreases from 40 to 13 (Fig. 3.5). Since each attack or support relation is between two nodes, the memory usage for these relations is twice the memory usage of the saving nodes. Moreover, the supporting weights and argument weights are reduced from 40 to 0. Therefore, the total memory usage decreases from $44 + (40 \times 2) + 40 = 164$ to $11 + (13 \times 2) + 0 = 37$ which is more than four times ($4\times$) lower in this small illustrating example. Moreover, the lower number of saved nodes in the memory results in the lower number of comparisons between the feature values and lower ultimate run-time.

3.4.2 Accelerated Argumentation-Based Learning

As explained in the previous sub-section, the main difference among the proposed Accelerated Argumentation-Based Learning (AABL) approach and the ABL is the architecture of the model. Here, only a the BAF unit is utilized. Algorithm 3.1 shows the pseudocode of the proposed approach for AABL. Instead of initiating the model with all the subsets of feature values, the model begins with the subsets of size 1 and increases the size of the subsets when needed (ensuring that the maximum size of the subsets can be 2). This way the number of required computations in the algorithm is significantly reduced. Using the extracted subsets and the set of supporting nodes in the model, the best recovery behavior (action) is predicted. If the model could predict a recovery behavior in the previous step, it will be applied to the environment.

Algorithm 3.1: Argumentation-Based Learning

input: Current **BAF** graph, Data Instance **X** entering the argumentation-based learning model, feature values subsets' size **L**, The class label or **B**est **R**ecovery **B**ehavior (**BRB**) for **X**
output: The predicted label for **X** called **Y**

Start:
- Extract all feature value combinations in **X** with length **L** and add them to a list called **Combs**.
- Let **SNs** be the set of supporting nodes (in form of "supporting-node → supported-node") in the BAF.

for *(all **sn** in **SNs**)* **do**
  for *(all **comb** in **Combs**)* **do**
    if *(sn.supporting-node==comb)* **then**
      Y.Add(sn.supported-node)

if *(**Y** is not empty)* **then**
  if *(**Length(Y)**==1)* **then**
    - Apply **Y** to environment and observe the **result**.
  else
    - Select a prediction in **Y** at random (Y := Y[random_index]) and observe the **result**.

else
  - Randomly choose a prediction from the available class labels (observed recovery behaviors).

- **should_Increment_L** := Update the **BAF** unit (using Algorithm 2 with input parameters: current **BAF** graph, **BRB**, **Combs**, **SNs**).

while *(**should_Increment_L** == True) and (L≤2)* **do**
  - L := L+1
  - Compute the combinations of the feature values again as **Combs**.
  - **should_Increment_L** := Update the model with Algorithm 2.

return Y

If there exits more than one predictions, one of them is chosen randomly. Otherwise, the random choice is among all the possible recovery behaviors (actions). Subsequently, the BAF unit is updated using the algorithm 3.2. The updating process has two steps, namely, updating the attack relations and updating the support relations. When a new recovery behavior (action) is added to the model, bidirectional attacks will be added between the newly added recovery behavior (action) and all the other previous recovery behaviors (actions) in the model. The supporting nodes are then added or pruned based on their uniqueness in supporting a recovery behavior (action). When all the supporting nodes are pruned, the size of the subsets of the feature values will increase.

Algorithm 3.2: Updating the BAF Unit

input: Current **BAF** graph, class label (**B**est **R**ecovery **B**ehavior) **BRB**, Combinations of feature values for **X** called **Combs**, Set of **S**upporting **N**odes in the BAF graph **SNs**

output: A Boolean variable "**should_Increment_L**" that tells whether **L** needs to be incremented or not.

Start:

- Let **RNs** be the set of all the class labels (**R**ecovery behavior **N**ode**s**) in the BAF.
- Let **attacks** be the set all the attack relations (for **a** ∈ **RNs** and **b** ∈ **RNs** the attack relations are in form of "**a** → **b**") among the class labels (recovery behavior nodes) in the BAF.

Step 1: (Updating attack relations and class labels)

if *(**BRB** is not in BAF)* **then**
- add **BRB** to the BAF graph;
- add bidirectional attacks between **BRB** and all the other class labels (recovery behavior nodes) as follows:
- **for** *all **rn** in **RNs*** **do**
  - **attacks.add(BRB → rn)**
  - **attacks.add(rn → BRB)**

Step 2: (Updating support relations)

- Let **should_Increment_L := True**

for *(all **comb** in **Combs**)* **do**
- Let **Add_Support :=True**
- **for** *(all **sn** in **SNs**)* **do**
  - **if** *(**sn**.supporting-node == **combs**)* **then**
    - **Add_Support :=False**
    - **if** *(**sn**.supported-node ≠ **BRB**)* **then**
      - Mark **comb** as a non-unique node that can not support any node in future.
      - Remove **sn** from the set supporting nodes in BAF **SNs**.
- **if** *(**comb** is not Marked)* **then**
  - **should_Increment_L := False**
  - **if** *(**Add_Support** == **True**)* **then**
    - **SNs**.add(**comb** → **BRB**)

return should_Increment_L

3.4.3 Complexity Analysis

In this subsection, he computational complexity and the space complexity of AABL and the original ABL have been discussed.

3.4.3.1 Computational Time Complexity

In the previous subsections, we have explained that the proposed ABL method begins with extracting the subsets of feature values with size 1 and then increases the size of subsets if needed. Moreover, pruning the unnecessary supporting nodes reduces the required number of computations of the approach. Assuming that n is the number of features in a dataset, the worst-case complexity of the proposed algorithm is $O(n^2)$, while the original ABL method has $O(2^n)$ worst-case complexity.

3.4.3.2 Space Complexity

Space complexity of the proposed AABL method is directly related to the computational time complexity of the approach since we need to keep all the supporting nodes in the memory. Therefore, the worst-case space complexity of AABL is $O(n^2)$. The space complexity of the original ABL approach is $O(2^n)$.

3.5 Experiments

3.5.1 Experimental Setup

In all the experiments, a table like Table 3.1 is randomly generated. Using this randomly generated table, we then randomly generate the three scenarios as explained in Section III. Each experiment has been conducted ten times (iterations) and the average result is reported. In order to compare the accuracy of both the proposed AABL and the original version of ABL [16, 18], we have set the limit of 200 recovery attempts at each iteration. The run-times are reported in seconds.

3.5.2 Experimental Results

In this section, three sets of experiments have been conducted. First, the runtime of both approaches for all the scenarios have been compared. Second, the memory consumption of both methods has been compared. Third, the learning accuracy of both approaches has been compared.

	Proposed Approach (secs)	Original ABL (secs)
First Scenario	0.42	84.76
Second Scenario	3.16	3318.68
Third Scenario	13.56	87088.60

Table 3.2: Comparison of run-times in seconds for different scenarios.

	Proposed Approach (MBs)	Original ABL (MBs)
First Scenario	0.9	20.4
Second Scenario	1.3	161.1
Third Scenario	1.7	392.73

Table 3.3: Comparison of memory usage for different scenarios.

3.5.2.1 Comparison of Run-times

Table 3.2 shows the comparison of the run-times of the proposed AABL and the original ABL over these scenarios. As you can see, the newly proposed approach outperforms the original version of ABL by a large margin. For the first scenario, the run-time of the original ABL is 84.76s, while the newly proposed method has the run-time of 0.42s. This means that AABL runs almost 200 *times* faster than ABL. The run-time of our proposed approach for the second scenario is 3.16s, while it is 3318.68s for the original ABL approach. This means that the proposed approach is more than 10^3 *times* faster than the previous approach for the second scenario. For the third scenario, the proposed ABL algorithm runs 6×10^3 *times* faster than the original ABL algorithm.

Figure 3.6 shows the relation between the number of locations in a scenario like the first scenario and the run-time of different approaches. The run-time of the original ABL approach exponentially increases while the proposed accelerated ABL approach maintains much lower run-time that is linearly dependent on the number of locations. Since the first scenario is only dependent on the location number 5, only the subsets with the size 1 from the feature values are extracted.

3.5.2.2 Comparison of the Memory Usage

In order to compare the memory usage of both the previous ABL approach and the new proposed ABL approach, we have made a comparison. Table 3.3 shows the comparison of the memory usage of both approaches in MBs for all the scenarios. For the first scenario, our current method has more than

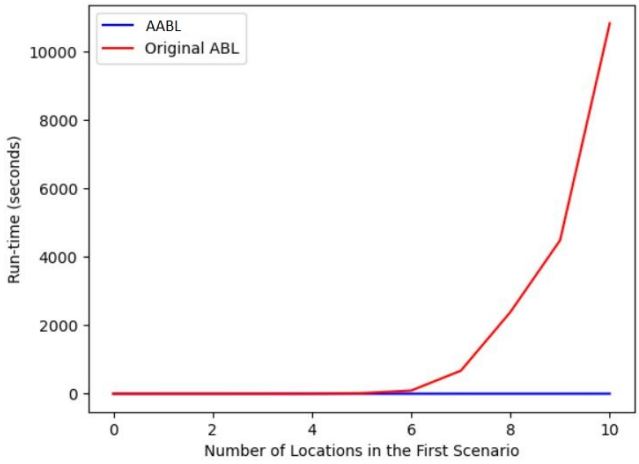


Figure 3.6: The relation between the run-time and the number of locations in a scenario like the first scenario.

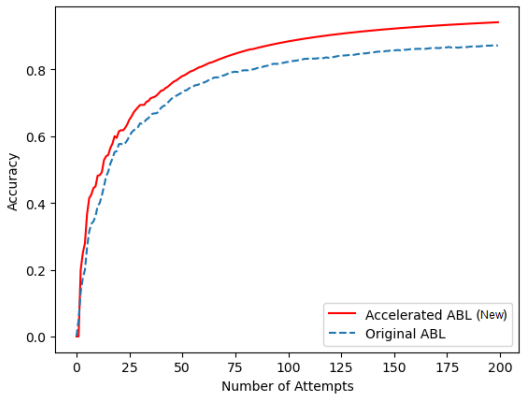


Figure 3.7: The comparison of learning accuracy vs number of recovery attempts between the proposed AABL and the original ABL for the first scenario.

20 *times* lower memory consumption. Moreover, for the second scenario, the memory consumption is more than 70 *times* reduced in the newly proposed approach. The proposed approach uses 200 *times* lower memory for the third scenario.

3.5.2.3 Evaluating the Accuracy

In order to evaluate the performance of both methods based on the learning accuracy, we have conducted two experiments. Figure 3.7 compares the accuracy of both methods for the first scenario. The comparison of both methods

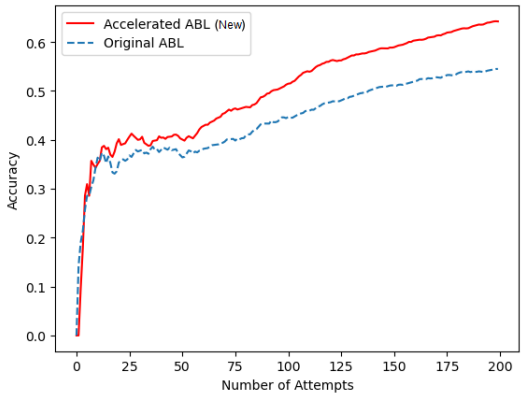


Figure 3.8: The comparison of learning accuracy vs number of recovery attempts between the proposed AABL and the original ABL for the second scenario.

for the second scenario is illustrated in Figure 3.8. In both cases, the proposed ABL approach has higher accuracy.

3.6 Conclusion

Argumentation-Based online incremental Learning (ABL) has been introduced recently in [16, 18]. ABL outperformed other state-of-the-art online incremental learning algorithms. Although ABL has higher learning accuracy than other approaches, the it is not suitable for high dimensional problems. The reason lies in the high computational complexity of the approach. In this chapter, we have proposed Accelerated Argumentation-Based Learning (AABL), which has lower computational complexity, and memory usage. The resulting approach can be used for higher dimensional problems while having a better learning accuracy than the original version of the ABL algorithm. We have conducted three sets of experiments with more complex scenarios and analyzed the run-time and memory usage of the methods. The proposed approach outperforms the original version of ABL algorithm in terms of run-time and memory usage by a large margin while slightly outperforming the original ABL in terms of accuracy. The lower computational complexity of the proposed approach makes it applicable for wider range of machine learning problems.

CHAPTER 4

Local Hierarchical Dirichlet Process

Open-Ended 3D Object Category Recognition

This chapter is based on [20] and it introduces a hierarchical Bayesian model for the 3D object category recognition task. This model learns in real-time from high dimensional datasets in an online incremental manner. It achieves high learning accuracy using only a small number of learning instances. This method is suitable for both offline static datasets and online open-ended scenarios where the number of class labels is not fixed and grows over time. The model is evaluated in two robotic applications both in real-word and simulated environments. It is a non-parametric probabilistic model that can autonomously determine the number-of-topics parameter from the dataset.

Chapter 4

Local-HDP: Interactive Open-ended 3D Object Category Recognition in Real-Time Robotic Scenarios

Abstract

We introduce a non-parametric hierarchical Bayesian approach for open-ended 3D object categorization, named the Local Hierarchical Dirichlet Process (Local-HDP). This method allows an agent to learn independent topics for each category incrementally and to adapt to the environment in time. Each topic is a distribution of the visual words over a predefined dictionary. Using an inference algorithm, these latent variables are inferred from the dataset. Subsequently, the category of an object is determined based on the likelihood of generating a 3D object from the model. Hierarchical Bayesian approaches like Latent Dirichlet Allocation (LDA) can transform low-level features to high-level conceptual topics for 3D object categorization. However, the efficiency and accuracy of LDA-based approaches depend on the number of topics that is chosen manually. Moreover, fixing the number of topics for all categories can lead to overfitting or underfitting of the model. In contrast, the proposed Local-HDP can autonomously determine the number of topics for each category. Furthermore, the online variational inference method has been adapted for fast posterior approximation in the Local-HDP model. Experiments show that the proposed Local-HDP method outperforms other state-of-the-art approaches in terms of accuracy, scalability, and memory efficiency by a large margin. Moreover, two robotic experiments have been conducted to show the applicability of the proposed approach in real-time applications.



Figure 4.1: An illustrative example of inter-category variation of the mug category in the Washington RGB-D dataset (*top*), and different object views of a mug object (*bottom*).

4.1 Introduction

Most recent object recognition/detection techniques are based on deep neural networks [79, 80, 88, 109, 136, 146, 111]. These methods typically need a large labeled dataset for a long training process. The number of object categories (class labels) should be predefined in advance for such methods. However, in real-life robotic scenarios, a robot can always face new object categories while operating in its environment and it requires learning from a small number of observations. Therefore, the model should be updated in an open-ended manner without completely retraining the model [18]. In this chapter, *open-ended learning* means that the number of categories (class labels) is not fixed and predefined for the model, and that it can grow during runtime. Furthermore, object category recognition is not a well-defined problem because of the large inter-category variation (Figure 4.1 (*top*)), multiple object views for each object (Figure 4.1 (*bottom*)), and concept drift in dynamic environments [91].

Object recognition in humans is a complex hierarchical multi-stage process of streaming visual data in the cortical regions [43]. The hierarchical structure of the brain for the object recognition task has motivated us to choose hierarchical Bayesian models like Latent Dirichlet Allocation (LDA) [30] and Hierarchical Dirichlet Process (HDP) [160] for object category recognition.

In this chapter, we suggest that 3D visual streaming data should be processed continuously, and object category learning and recognition should be performed simultaneously in an open-ended manner. We propose the Local Hierarchical Dirichlet Process (Local-HDP), an extension of the Hierarchical Dirichlet Process [160] method, which can incrementally learn new topics for each category of objects independently. In contrast to notable recent works [91, 90, 150] using a predefined number of topics, Local-HDP is more flexible

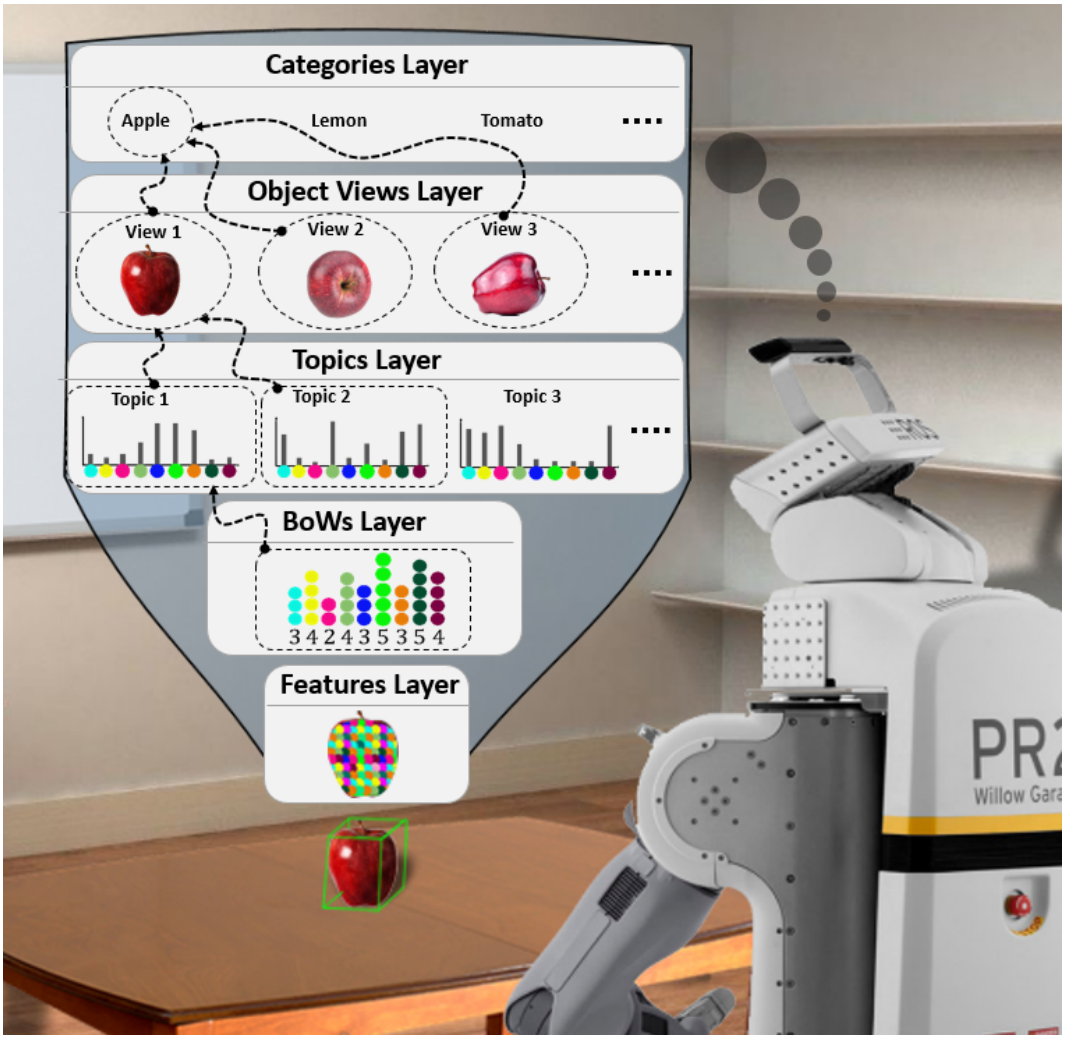


Figure 4.2: The architecture of the proposed method.

since it is a non-parametric Bayesian model that can autonomously determine the number of topics for each category at run-time.

Figure 4.2 shows the processing layers of the proposed Local-HDP. The tabletop objects are detected in the initial phase (green bounding box around apple on the table in Figure 4.2). Subsequently, the hierarchy of the five processing layers is utilized. The features layer extracts a set of local shape features using the spin-image descriptor [85].

The computed features are represented as Bag of visual Words (BoWs). The obtained representation is then sent to the topics layer, where a set of topics is inferred autonomously for the given object using the proposed Local-HDP method. Each topic is a distribution of visual words over a dictionary. In

other words, the topic layer provides an unsupervised mapping of the BoW representation to the topics space, which can fill the conceptual gap between low-level features and high-level concepts. As shown in the object views layer, the appearance of an object may vary from different perspectives (Figure 4.1 (*bottom*)). Therefore, it is necessary to infer topics using different object views. There might be different instances in an object category as well (see Figure 4.1 (*top*)). This point is addressed in the categories layer. Moreover, a simulated teacher has been developed to interact with the model and evaluate its performance in an open-ended manner.

This work extends two approaches, namely Local-LDA [91] and HDP [160], in four aspects. First, our approach can autonomously detect the number of required topics to independently represent the objects in each category, avoiding the limitation of Local-LDA for determining the number of topics in advance. This feature prevents underfitting or overfitting of the model. Second, our research adapts the online variational inference technique [169], which significantly reduces inference time. Third, the proposed local online variational inference method leads to memory optimization since it needs to store a smaller average number of instances per object category in memory. Fourth, our work extends the hierarchical Dirichlet process [160] by learning and updating local topics for each object category independently in an incremental and open-ended fashion.

4.2 Related Work

Object representation is one of the main building blocks of object recognition approaches. The underlying reason is that the output of the object representation module is used in both learning and recognition. Object representation techniques can be categorized into three groups, namely, global and local object descriptors and machine learning approaches [104]. Notable global object descriptors are Global Orthographic Object Descriptor (GOOD) [95, 92], Ensemble of Shape Functions (ESF) [176] and Viewpoint Feature Histogram (VFH) [143]. Examples of local 3D shape descriptors include Spin-Images (SI) [85], Intrinsic Shape Signature (ISS) [187], and Fast Point Feature Histogram (FPFH) [144]. Local descriptors are more robust to occlusions and clutter. However, comparing pure local descriptors is a computationally expensive task [7]. To alleviate this problem, machine learning techniques like Bag of Words (BoW) [93], Latent Dirichlet Allocation (LDA) [30, 96] and deep learning [108, 178] methods can be used for representing objects in a compact and uniform format.

Kasaei et al. [91] extended Latent Dirichlet Allocation (LDA) and proposed Local-LDA. They showed the application of Local-LDA in the context of open-ended 3D object category learning and recognition. Similar to our approach, Local-LDA learns a set of topics for each object category incrementally and independently. Unlike our approach, in Local-LDA, the same number of topics is chosen in advance based on trials and errors for all of the object categories. A good choice for the number of topics for each object category is correlated to the intra-category variation of each 3D object category. Therefore, choosing the same number of topics for all the object categories with different intra-category variation might be not reasonable. Moreover, in open-ended scenarios, it is not feasible to anticipate the inter-category variation of 3D objects that the model might see in the future and choose a fixed number of topics in advance for all the categories. To solve these issues, our approach can autonomously choose the number of topics for each object category on the fly without a need for in advanced trails and errors. This makes our approach more robust for recognizing object categories with various inter-category and intra-category variation and applicable in real-world open-ended scenarios. Local-LDA uses collapsed Gibbs sampling for approximating the posterior probability. However, we adapt the online variational inference technique [169] for Local-HDP.

Our approach builds on the Hierarchical Dirichlet Process (HDP) [160], that is based on Dirichlet process (DP) [64] and mixture of DPs [12]. Posterior inference is intractable for HDP, and much research has been done to find a proper approximate inference algorithm [160, 159, 110]. The Markov Chain Monte Carlo (MCMC) sampling method for DP mixture models has been proposed for approximate inference in HDPs [121]. David Blei et al. proposed the variational inference for DP mixtures [29]. Teh et al. [160] proposed the Chinese Restaurant Franchise metaphor for HDP and used Gibbs sampling method for the inference. The online variational inference approach is proposed by Wang et al. [169] for HDP, which can be used in online incremental learning scenarios and for large corpora. Our method is different from HDP, since the proposed Local-HDP only shares the topics in the local models for each category and not across different categories. This is especially needed in the case of 3D object categorization for open-ended scenarios [91]. The use of local topics avoids underfitting of the model by considering intra-category variations. HDP has further extensions to construct tree-structured representations for text data which have nested structure [123]. Similar to supervised hierarchical Dirichlet Process (sHDP) [49], we use the category label of each object. Unlike sHDP, we learn object categories in an open-ended fashion, while in sHDP, the number of object categories to be learned should be defined in advance.

Deep learning-based approaches [189, 184, 63] try to learn a sparse repre-

sentation for 3D objects. Unlike our approach, such methods typically need a large labeled dataset and require long training time. In particular, our proposed approach does not require a large labeled dataset and can incrementally update the model facing an unforeseen object category in an open-ended manner. Moreover, the number of categories is not fixed in open-ended approaches like ours.

4.3 Method

We assume that an object has already been segmented from the point cloud of the scene, and we hence mainly focus on detailing the Local Hierarchical Dirichlet Process (Local-HDP) approach.

4.3.1 Pre-Processing Layers

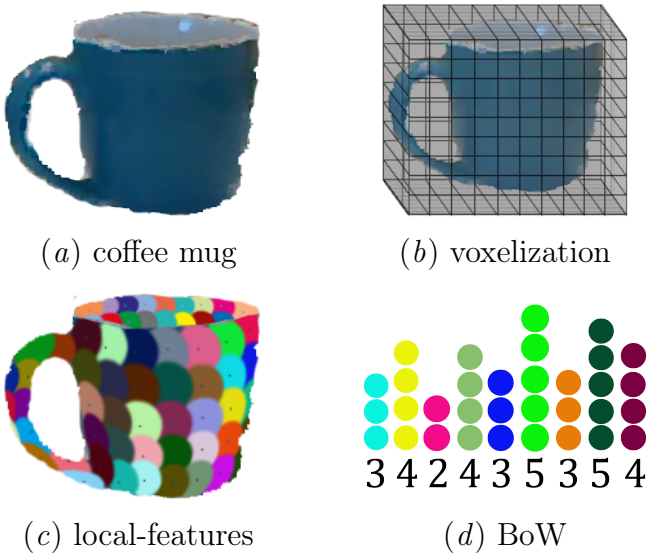
In Figure 4.2, the first two layers—the feature layer and BoWs layer— are the pre-processing layers. In the feature layer, we first select key-points for the given object and then compute a local shape feature for each key-point. Towards this goal, we first voxelized[1] the object (Figure 6.5) (b), and then, the nearest point to each voxel center is selected as a key-point. Afterwards, the spin-image descriptor[2] [85] is used to encode the surrounding shape in each key-point using the original point cloud (Figure 6.5 (c)). This way, each object view is described by a set of spin-images in the first layer, $\mathbf{O_s} = \{\mathbf{s}_1, \ldots, \mathbf{s}_N\}$ where N is the number of key-points. The obtained representation is then sent to the BoWs layer. Since HDP-based models have the bag-of-words assumption - that the order of words (visual words) in the document (3D object view) can be neglected - the BoWs layer transforms the computed spin-images to a BoW format (Figure 6.5 (d)). Towards this end, the BoWs layer requires a dictionary with V visual words (spin-images). In this work, we have created a dictionary of visual words using the same methodology as Local-LDA [91]. The obtained BoW representation is fed to the topic layer.

4.3.2 Local Hierarchical Dirichlet Process

After synthesizing the point cloud of the 3D objects to a set of visual words in BoW format, the data is ready to be inserted into the topic layer where the proposed Local-HDP method is employed. In this layer, the model transforms

[1] www.pointclouds.org/documentation/classpcl_1_1_voxel_grid.html

[2] www.pointclouds.org/documentation/classpcl_1_1_spin_image_estimation.html



(*a*) coffee mug (*b*) voxelization

(*c*) local-features (*d*) BoW

Figure 4.3: (*a*) The RGB-D image of a coffee mug. (*b*) Key-points selection using voxelizing [91]. (*c*) Key-points neighborhoods are represented by different colors. (*d*) The BoW representation for the given object.

the low-level features in BoW format to conceptual high-level topics. In other words, each object is represented as a distribution over topics, where each topic is a distribution of visual words over a dictionary. To this end, we use an incremental inference approach where the number of categories is not known beforehand and the agent does not know which additional object categories will be available at run-time.

The plate notation of Local-HDP is shown in Figure 6.4. In this graph, C is the number of categories, $|c|$ is the number of objects in each category. Each object, j, is represented by a set of N visual words, $W_{j,n}$ where j, n show the n'th visual word from the j'th object. Each visual word is an element from the vocabulary of visual words with predefined V words, that is $W_{j,n} \in \{1...V\}$. Using a *Coffee Mug* as an example, a distribution over the topics of the *Coffee Mug* should be used to generate the visual words of the object. Accordingly, a particular topic is selected out of the mixture of possible topics of the *Coffee Mug* category to generate the visual words. For instance, coffee mugs typically have a "handle", which is represented as a distribution of visual words that repeatedly occur together. This can be interpreted as the "handle" topic, which is inferred from the co-occurrence of the visual words in several objects of the same category.

The process of choosing a topic and then drawing the visual words from that topic is repeated several times to generate all the visual words of the *Coffee Mug*. It is worth mentioning that the generative process is not used in

the experiments. However, the local online variational inference technique is used to do a reverse procedure of inferring the topics, corresponding to latent variables from the 3D object views. Using the inferred topics, we then compare the log-likelihood of generating the visual words in a 3D object for each local model. The category of the local model with the highest log-likelihood is then selected as the predicted category of the 3D object.

4.3.3 Dictionary of Visual Words

In this chapter, we have used the method of [91] to construct the dictionary of the visual words. This means that a dictionary with V visual words is constructed by clustering a random subset of 50% of the training data. We have utilized the k-means method for clustering the local shape descriptors (spin-images) of the randomly selected objects into V visual words. Consequently, the nearest spin-images to the cluster centers are selected as the dictionary's visual words.

4.3.4 Local Online Variational Inference

The inference method is responsible for inferring the latent variables in the model using a dataset [21]. In this section, we adapt the online variational inference method [169] for Local-HDP. This method can be used in open-ended applications since it can handle streaming data in an online and incremental manner. Moreover, it is faster than traditional approximate inference techniques, e.g., Chinese restaurant franchise [160] and variational inference [29], and it can be used to infer the latent variables of differently scaled datasets [169].

Online variational inference for HDP is inspired by the online variational Bayes [77] method for LDA. This method tries to optimize a variational objective function [86] exploiting stochastic optimization [141]. HDP is a collection of DPs G_j that share the same base-distribution G_0 (which is also drawn from a DP). These DPs share the same set of atoms and only the atom weights are different. Mathematically, a two-level HDP is defined as follows:

$$\begin{aligned} G_0 &\sim DP(\gamma H) \\ G_j &\sim DP(\alpha_0 G_0), \text{for each j} \end{aligned} \tag{4.1}$$

where $\alpha_0 > 0$ is the scaling parameter and γ is the concentration parameter of a DP. Sethuraman's stick-breaking construction technique [65] is responsible for determining the number of topics in the model. Using the same approach as

[169] for HDP, the variational distribution for local online variational inference is in the following form:

$$q(\beta', \pi', c, z, \phi) = q(\beta')q(\pi')q(c)q(z)q(\phi) \tag{4.2}$$

In the terminology of variational inference techniques, q is called the variational approximation to the posterior p. Variational techniques try to solve an optimization problem over a class of tractable distributions Q in order to find a $q \in Q$ that is most similar to p and can be used as its approximation. Moreover, $\beta' = (\beta'_k)_{k=1}^{\infty}$ is the top-level stick proportion, $\pi' = (\phi'_{jt})_{t=1}^{\infty}$ is the bottom-level stick proportion and $c_j = (c'_{jt})_{t=1}^{\infty}$ is the vector of indicators for each G_j. Moreover, $\phi = (\phi_k)_{k=1}^{\infty}$ is the inferred topic distribution, and z_{jd} is the topic index for the nth visual word in the jth 3D object. The infinity notion (∞) shows the open-ended nature of the number of parameters.

The factorized form of $q(c)$, $q(z)$, $q(\phi)$, $q(\beta')$ and $q(\pi')$ is the same as the online variational inference for HDP [169]. Assuming that we have $|c|$ objects in each category for Local-HDP, the variational lower bound for object j in category C is calculated as follows:

$$\begin{aligned} L_j^{(C)} = & \\ & \mathbb{E}_q[log(p(w_j|c_j, z_j, \phi)p(c_j|\beta')p(z_j|\pi')p(\pi'_j|\alpha_0))] \\ & + H(q(c_j)) + H(q(z_j)) + H(q(\phi')) \\ & + \frac{1}{|c|}[E_q[logp(\beta')p(\phi)] + H(q(\beta')) + H(q(\phi)] \end{aligned} \tag{4.3}$$

Where $H(.)$ is the entropy term for the variational distribution. Therefore, the lower bound term for each category is calculated in the following way:

$$L^{(C)} = \sum_j L_j^{(C)} = \mathbb{E}_j[|c|L_j^{(C)}] \tag{4.4}$$

Using coordinate ascent equations in the same way as online variation inference, the object-level parameters $(a_j, b_j, \varphi_j, \zeta_j)$ are estimated. To be more specific, a_j and b_j are the parameters of the beta distributions for the bottom-level stick proportions π_j, φ_j is the variational parameter for the vector of indicators c_j, and ζ_j is the variational parameter for the topic z_j. These variables are defined in the same way as in [169]. Then, for the category-level parameters $(\lambda^{(C)}, u^{(C)}, v^{(C)})$, we do gradient descent with respect to a learning rate:

$$\partial\lambda_{kw}^{(C)}(j) = -\lambda_{kw} + \eta + |c| \sum_{t=1}^{T} \varphi_{jtk}(\sum_n \zeta_{jnt} I[w_{jn} = w]) \tag{4.5}$$

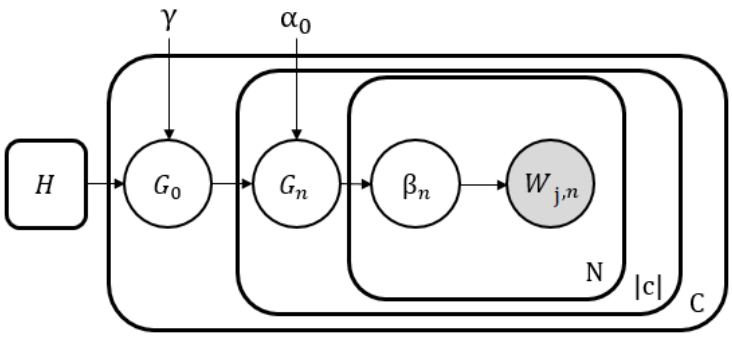


Figure 4.4: The plate notation of Local-HDP.

$$\partial u_k^{(C)}(j) = -u_k + 1 + |c| \sum_{t=1}^{T} \varphi_{jtk} \tag{4.6}$$

$$\partial v_k^{(C)}(j) = -v_k + \lambda + |c| \sum_{t=1}^{T} \sum_{l=k+1}^{K} \varphi_{jtl} \tag{4.7}$$

Here, K and T are the document (3D object view) and corpus (category) level truncates. Moreover, φ (multinomial), ζ (multinomial) and λ (Dirichlet) are the variational parameters, which are the same for all the categories. Using an appropriate learning rate p_{t_0} for online inference, the updates for $\lambda^{(C)}, u^{(C)}$ and $v^{(C)}$ become:

$$\boldsymbol{\lambda}^{(C)} \leftarrow \boldsymbol{\lambda}^{(C)} + p_{t_0} \partial \boldsymbol{\lambda}^{(C)}(j) \tag{4.8}$$

$$\boldsymbol{u}^{(C)} \leftarrow \boldsymbol{u}^{(C)} + p_{t_0} \partial \boldsymbol{u}^{(C)}(j) \tag{4.9}$$

$$\boldsymbol{v}^{(C)} \leftarrow \boldsymbol{v}^{(C)} + p_{t_0} \partial \boldsymbol{v}^{(C)}(j) \tag{4.10}$$

Algorithm 6.1 shows the pseudo-code of the proposed inference technique for the Local-HDP approach.

4.3.5 Object Category Learning and Recognition

In this subsection, the mechanism of interactive open-ended learning has been explained in more detail. Classical object recognition methods do not support open-ended learning. In contrast, our method is open-ended, and the number of categories can be incrementally extended through time. The system can

Algorithm 4.1: Local Online Variational Inference

initialization:
Randomly initialize $\lambda^{(C)} = (\lambda_k^{(C)})_{k=1}^{K}$, $u^{(C)} = (u_k^{(C)})_{k=1}^{K-1}$ and $v^{(C)} = (v_k^{(C)})_{k=1}^{K-1}$ for all the learned categories. Set $t_0 = 1$
for ***each*** *Category C* **do**
  while *Stopping criterion is not met* **do**
    - Use the object view j for updating the parameters.
    - Compute the document-level parameters $a_j, b_j, \Phi_j, \zeta_j$ using the same methodology as [169].
    - Using Eq. 5-7, compute the natural gradients $\partial\lambda^{(C)}(j)$, $\partial u^{(C)}(j)$ and $\partial v^{(C)}(j)$.
    - Set $p_{t_0} = (\tau_0 + t_0)^{-K}$, $t_0 = t_0 + 1$.
    - Update the $\lambda^{(C)}$, $u^{(C)}$, $v^{(C)}$ parameters using Eq. 8-10.
  end
end

interact with a human user to learn about new categories or to update existing category models by receiving corrective feedback when misclassification occurred. We follow the same methodology as [94] for this purpose. The user can interact with the system with one of the following actions:

- **Teach**: introducing the category of target object to the agent.
- **Ask**: inquiring the agent about the category of a target object.
- **Correct**: sending corrective feedback to the agent in case of wrong categorization.

Whenever the agent receives a teach command, it incrementally updates the local model corresponding to the category of the target object using the aforementioned online variational inference technique. In case of the ask command, the log-likelihood is used to determine the category of an object. The log-likelihood is computed in the same way as in [169]. The local model with highest likelihood is then selected as the predicted category for an object.

4.4 Experimental Results

Following the same protocol as Local-LDA [91] for interacting with a simulated teacher, two sets of experiments, namely, offline experiments and open-ended experiments, have been conducted to evaluate the performance of the

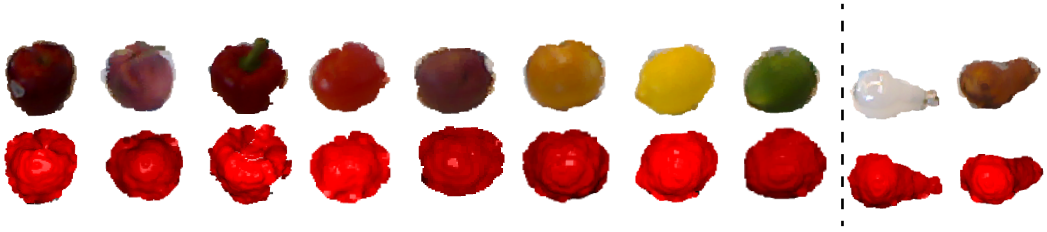


Figure 4.5: Similar object categories to those from the Washington RGBD dataset. The second row shows the same objects as the first row, but without color (in red), to emphasize the similarity of objects based on the depth data.

proposed method. The offline experiments use the k-fold cross-validation technique for evaluating the performance of the model in offline scenarios with a small number of training instances. The open-ended experiments are focused on evaluating the proposed approach for the scenarios in which the number of object categories (class labels) is not fixed and can grow over time. In open-ended scenarios, the model is updated in an incremental manner. However, in the offline evaluations, the model is trained once with a training set and then evaluated using a testing set from the dataset. For Local-HDP in all the experiments, we set $p_{t_0} = (\tau_0 + t_0)^{-K}$ where $K \in (0.5, 1]$ and $\tau_0 > 0$ as suggested by [169].

4.4.1 Datasets and Baselines for Comparison

For offline evaluation of the proposed Local-HDP and the other state-of-the-art approaches, we have used the restaurant RGB-D object dataset [94]. This dataset has 10 categories of objects and each category has a significant intra-category variation. It consists of 306 different object views for 10 household objects. Therefore, it is a suitable dataset to perform extensive sets of experiments.

The Washington RGB-D dataset [105] is used for online open-ended evaluation of the method since it is one of the largest 3D object datasets. It has 250,000 views of 300 common household objects, categorized in 51 categories. Figure 4.5 shows some of the categories of objects presented in the Washington RGBD Dataset. In all experiments, only the depth data has been used for determining the category of 3D objects. Therefore, as one can see in Figure 4.5, detecting the category of an object based solely on the depth data is a hard task even for humans.

We have compared the proposed Local-HDP using local online variational inference with Local-LDA [91], LDA with shared topics [30], BoW [93], RACE [122], and HDP with shared topics and online variational inference [169].

Table 4.1: Average accuracy of Local-HDP and Local-LDA based on 440 experiments with different parameter values.

Parameters		IW		VS				SL				
Value		4	8	0.01	0.02	0.03	0.04	0.03	0.04	0.05	0.1	0.15
Acc (%)	Local-LDA	**84**	83	81	82	**86**	83	81	83	84	**85**	83
	Local-HDP	**94**	92	91	93	**95**	93	91	92	92	**94**	91

Parameters		Dictionary Size										
Value		40	50	60	70	80	90	100	200	500	2000	2500
Acc (%)	Local-LDA	82	82	82	83	85	85	86	87	88	**90**	87
	Local-HDP	91	92	92	92	92	93	93	94	95	**96**	94

4.4.2 Offline Evaluation

Similar to Local-LDA, our approach has several parameters that should be well selected to provide an appropriate balance between recognition performance, memory usage and computation time. In order to fine-tune the parameters of our proposed method for offline evaluation, 440 experiments have been conducted with different parameter values. The voxel grid approach has been used for down-sampling and finding the keypoints for the local descriptor. Voxel grid has Voxel Size (VS) parameter which determines the size of each voxel. Moreover, the spin-image local descriptor has two parameters, namely Image Width (IW) and Support Length (SL).

In all experiments, the first level and second level concentration parameters are set to 1, chunk size for offline evaluation is set to 1, and the maximum number of topics is set to 100. All the other parameters are set to the default values as proposed in [169] . Moreover, in all the experiments the LDA parameters are set to be the same values as described in [91]. Since online variational inference is a stochastic inference technique, for each experiment the order of the data instances has been permuted 10 times and for each permutation 10-fold cross-validation has been used. Accordingly, the results have been averaged.

Table 4.1 shows the comparison of Local-HDP and Local-LDA with different parameter values. As one can see in this table, the proposed Local-HDP method outperforms Local-LDA which is the best among the other methods (see [91]). Using the best parameter values based on Table 4.1 and the corresponding tables in [91], the accuracy of all the approaches is shown in Table 4.2.

Table 4.2 shows that Local-HDP outperforms the other state-of-the-art methods in terms of accuracy with a large margin. In particular, the accuracy

Table 4.2: The comparison of different approaches using the best parameter values. The average run-time of each experiment is reported for all the approaches.

Approach	Accuracy (%)	Run-time (s)
RACE [122]	87.09	1757
BoW [93]	89.00	**195**
LDA (shared topics) [30]	88.32	227
Local-LDA [91]	91.30	348
HDP (shared topics) [169]	90.33	233
Local-HDP (our approach)	**97.11**	352

Table 4.3: The comparison of the proposed approach with some deep learning approaches for 3D object classification.

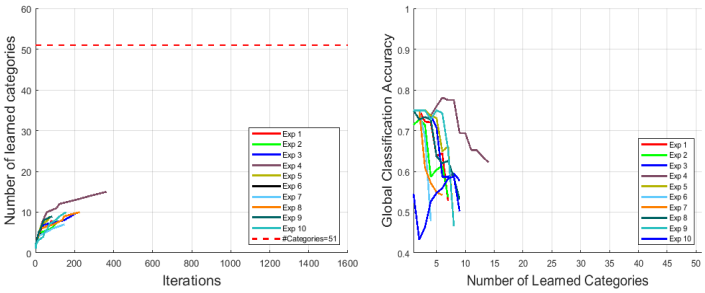
Approach	Accuracy (%) (original dataset)	Accuracy (%) (augmented dataset)
PointNet [129]	0.11	85.13
PointNet++ [130]	0.12	87.45
PointCNN [107]	0.12	88.02
Local-HDP	**97.11**	**98.64**

of Local-HDP was 97.11%, which is around 6.11 percentage point (p.p.) better than Local-LDA, and 6.78, 9.11, 8.11, 10.11 p.p better than HDP, LDA, BoW and RACE approaches respectively. Moreover, Local-HDP has almost the same run-time as Local-LDA.

Table 4.3 shows the comparison of the proposed Local-HDP approach with some deep learning architectures, namely, PointNet [129], PointNet++ [130], and PointCNN [107] for offline evaluation. Since the number of training instances for each category is limited in the restaurant RGB-D object dataset [94] (the number of training instances for offline 10-fold cross-validation for the fork category is 8 and the average number of training instances per category is 27), the deep learning approaches tend to overfit and could not generalize well. To resolve this issue for deep learning approaches, the dataset is augmented 20 times by randomly rotating the point clouds around different axes. Table 4.3 also compares the accuracy of deep learning approaches with the proposed Local-HDP after augmentation.

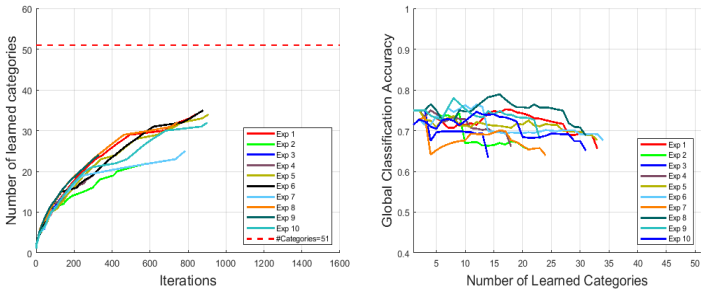
To uniformly sample 2048 points from a point cloud for the deep learning approaches, a mesh is constructed using the ball-pivoting algorithm for surface reconstruction [25]. Subsequently, the point clouds are normalized to a unit sphere (the same approach is used in PointNet [129]) to uniformly sample 2048 points from the constructed meshes.

Exp#	#QCI	#LC	AIC	GCA(%)
1	201	8	14.88	52.74
2	231	8	16.38	53.68
3	336	10	17.2	57.74
4	495	**15**	15.47	**62.22**
5	193	9	14.44	46.63
6	138	5	17.4	47.83
7	264	7	20.29	54.17
8	348	10	19.3	53.16
9	206	9	15.22	46.60
10	279	10	16.9	50.18
Avg.	269	**9.1**	**16.74**	**51**



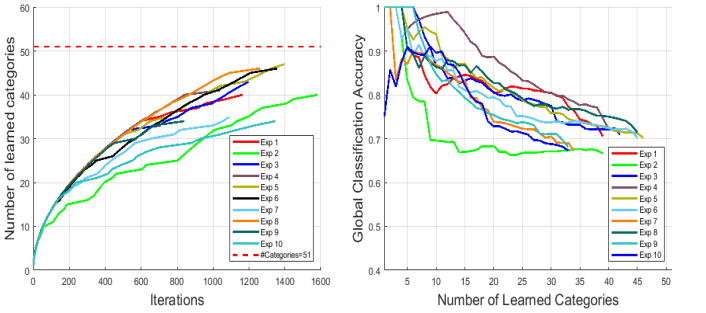
(a) Summary of experiments for LDA

Exp#	#QCI	#LC	AIC	GCA(%)
1	1011	34	13.24	65.58
2	737	22	14.59	65.40
3	306	15	10.47	63.40
4	439	19	10.84	66.06
5	1079	34	13.26	67.66
6	1052	**35**	12.74	67.59
7	937	25	16.52	63.93
8	909	32	11.88	**68.76**
9	480	24	9.417	67.92
10	1069	32	14.66	65.11
Avg.	753	**27.2**	**12.76**	**66.14**



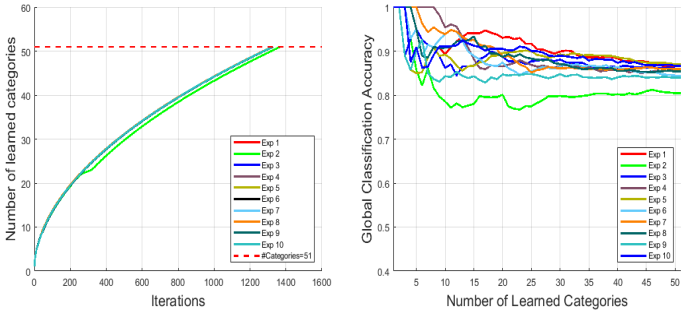
(b) Summary of experiments for HDP

Exp#	#QCI	#LC	AIC	GCA(%)
1	1346	40	12.93	70.51
2	1764	40	17.73	66.61
3	1385	43	12.4	70.83
4	1224	41	11.29	**72.22**
5	1594	**47**	13.11	70.20
6	1551	46	13.04	70.21
7	1263	35	14.83	67.22
8	1455	46	12.04	71.41
9	1012	34	12.53	67.98
10	1518	34	17.62	67.26
Avg.	1411	**40.6**	**13.75**	**69.44**



(c) Summary of experiments for Local-LDA (Online Variational Inference)

Exp#	#QCI	#LC	AIC	GCA(%)
1	1325	**51**	6.45	86.72
2	1370	**51**	8.25	80.44
3	1325	**51**	6.62	86.04
4	1325	**51**	6.70	85.74
5	1325	**51**	6.37	**87.02**
6	1325	**51**	7.03	84.45
7	1325	**51**	6.64	85.96
8	1325	**51**	6.80	85.36
9	1330	**51**	7.17	83.98
10	1327	**51**	6.47	86.66
Avg.	1330	**51**	**6.85**	**85.23**



(d) Summary of experiments for Local-HDP (our approach)

Figure 4.6: Summary of 10 experiments for open-ended evaluation LDA, HDP, Local-LDA and our proposed Local-HDP approach. The learning capacity and the global accuracy of different models is compared with the corresponding plots.

Table 4.4: The average result of 10 open-ended experiments for all the methods.

Approach	#QCI	#LC	AIC	GCA(%)
LDA	269	9.1	16.74	51.00%
HDP	753	27.2	12.76	66.14%
Local-LDA	**1411**	40.6	13.75	69.44%
Local-HDP	1330	**51.0**	**6.85**	**85.23**%

4.4.3 Open-Ended Evaluation

In order to evaluate our model in an open-ended learning scenario, we used the Washington RGBD dataset [105], and we have followed the same methodology as discussed in [91]. In particular, we have developed a simulated teacher which can interact with the model by either *teaching* a new category to it or *asking* the model to categorize the unforeseen object view. In case of wrong categorization of an object by the model, *correcting feedback* is sent to the model by the simulated teacher. In order to teach a new category, the simulated teacher presents three randomly selected object views of the corresponding category to the model. After teaching a new category, all of the previously learned categories are tested using a set of randomly selected unforeseen object views. Subsequently, the accuracy of category prediction is computed. In open-ended evaluation, the model observes the 3D objects one by one and the history of the latest $3n$ predictions of the model is considered for calculating the accuracy, where n is the number of the learned categories. If the corresponding accuracy is higher than a certain threshold $\tau = 0.66$ (which means that the number of true-positives is at least twice the number of wrong predictions), the simulated teacher will teach a new category to the model. If the learning accuracy does not exceed the threshold τ after a certain number of iterations (100 for our experiments), the teacher infers that the agent is not able to learn more categories and the experiment stops. More details on the online evaluation protocol that has been used in our experiments can be found in [90].

Since the performance of open-ended evaluation may depend on the order of introducing categories and object views (randomly selected at the beginning of each experiment), 10 independent experiments have been carried out for each approach. Several performance measures have been used to evaluate the open-ended learning capabilities of the methods, namely: (*i*) the number of Learned Categories (#LC); (*ii*) the number of Question/Correction Iterations (#QCI) by the simulated user; (*iii*) the Average number of stored Instances per Category (AIC) ; (*iv*) Global Categorization Accuracy (GCA), which represents the overall accuracy in each experiment. These performance measures have the following interpretations. #LC shows the open-ended learning capability of the model, which answers the following question: How capable is the

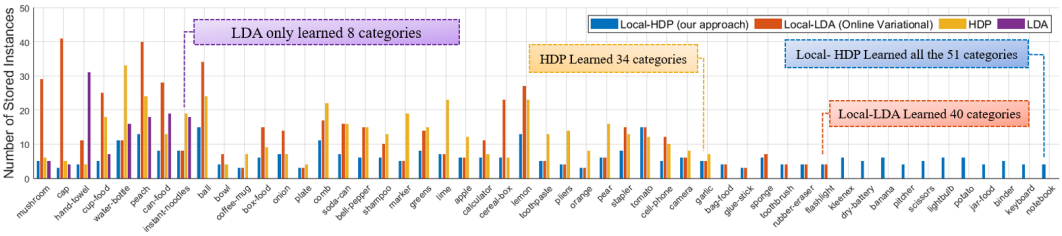


Figure 4.7: The absolute number of stored instances per category (for one out of ten open-ended experiments): the lower stored instances mean that the method is more memory efficient. The horizontal axis from left to right shows the order of introducing categories to all methods.

model in learning new categories? #QCI shows the length of the experiment (iterations). AIC represents the memory efficiency of the method. A lower average number of stored instances per category means a higher memory efficiency of the method. AIC is also related to the learning speed. A smaller AIC means that the method requires fewer observations to correctly recognize each category. #GCA shows the accuracy of the model in predicting the right category for each object.

In order to compare methods fairly, the simulated teacher shuffles data at the beginning of each round of experiments and uses the same order of object categories and instances for training and testing all the methods. Figure 4.6 (*left*) shows the detailed summary of 10 experiments for LDA, HDP, Local-LDA, and Local-HDP methods. It shows that Local-HDP could learn all 51 categories in all experiments, while Local-LDA, HDP, and LDA, on average learned 40.6, 27.2, and 9.1 categories, respectively (Table 4.4). The average AIC for Local-HDP is 6.85, while it is 13.75, 12.76, and 16.74 for Local-LDA, HDP, and LDA, respectively. This means that the proposed approach could achieve higher learning accuracy (85.23 compared to 69.44, 66.14, and 51), while observing a smaller number of 3D objects (around 50% fewer examples). This result shows the descriptive power of Local-HDP.

Figure 4.6 (*center*) shows the learning capability of the new categories as a function of the number of learned categories versus the question/correction iterations. Local-HDP achieved best performance by learning all the 51 categories in 1330.20 ± 13.95 iterations (Table 4.4). One important observation is that shuffling the order of introducing categories by the simulated user does not have a serious effect on the performance of Local-HDP, while it affects the performance of other methods significantly. The longest experiment, on average was continued for 1411.20 ± 212.75 iterations with Local-LDA and the agent was able to learn 40.60 ± 4.98.

Figure 4.6 (*right*) plots the global categorization accuracy versus the num-

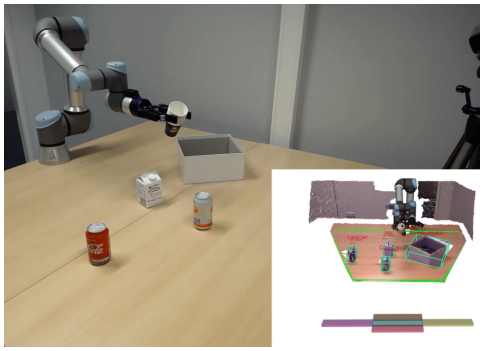
ber of learned categories. It was observed that the agent with Local-HDP not only achieved higher accuracy than other methods in all experiments but also learned all the categories. It is worth mentioning that Local-HDP concluded prematurely due to the "*lack of data*" condition, i.e., no more categories available in the dataset. This means that the agent with Local-HDP has the potential of learning more categories in an open-ended fashion. According to Table 4.4, the average GCA for Local-HDP is 85.23% and it is 69.44%, 66.14% and 51.00% for Local-LDA, HDP and LDA, respectively.

Figure 4.7 represents the absolute number of stored instances per category in one round of the open-ended experiments. It shows that the agent with Local-HDP stored a lower or equal number of instances for all of the categories. On closer review using Figure 4.6 (*left*), one can see that the Local-HDP on average stored 6.85 instances per category to learn 51 categories, while Local-LDA stored 13.75 to learn 40.6 categories. HDP achieved the third place by storing 12.76 instances to learn 27.20 categories and LDA was the worst among the evaluated approaches, i.e., on average it stored 16.74 instances to learn 9.10 categories. According to this evaluation, Local-HDP is competent for robotic applications with strict limits on the computation time and memory requirements.

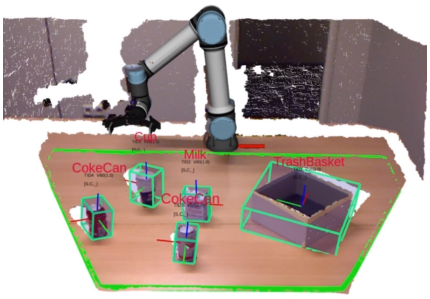
4.5 Real-time Robotic Application

To demonstrate the applicability of the proposed 3D object categorization method in real-time robotic applications, we have performed two object-manipulation experiments, as shown in Figure 4.8. In both robotic applications, the model is trained in an open-ended manner from scratch and the models are not pre-trained.

In both demonstrations, a UR5e robotic arm is used to manipulate the objects located on a table. Moreover, a Kinect camera is fixed in front of the table to acquire the visual data for further perceptual analysis. The system detects table-top objects, draws a bounding box around them and assigns a tracking ID (TID) to each object (Figures 4.8.b - 4.8.d). To compute the orientation of the bounding boxes, the Principle Component Analysis (PCA) WOLD198737 algorithm has been used. First, a local reference frame is constructed by applying PCA on the normalized covariance matrix, Σ, of the point cloud, i.e., $\Sigma V = EV$, where $E = [e_1, e_2, e_3]$ contains eigenvalues in the descending order, and $V = [v_1, v_2, v_3]$ represents the eigenvectors. Therefore, v_1 is the eigenvector with the largest variance of the points of the object. We consider v_1 and v_2 as X and Y axes, respectively. We define the Z axis as the cross product of



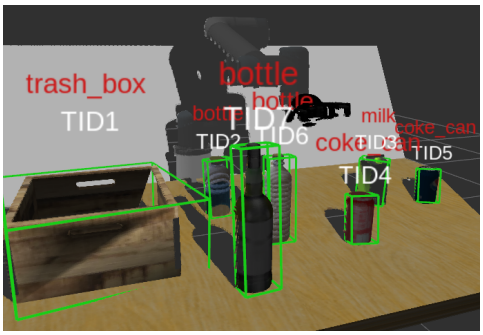
a) The robotic setup for the first demonstration.



b) Point cloud and object category visualization in RViz for the first robotic demonstration.



c) Clearing coke cans from the table for the second robotic demonstration.



d) The RViz visualization of the recognized categories for the second robotic demonstration.

Figure 4.8: The real-time application of the proposed Local-HDP 3D object category recognition method in a robotic scenario

$v_1 \times v_2$. The minimum and maximum values in each axis are then considered for computing the oriented bounding boxes.

The model does not initially have any knowledge about the category of the objects located on the table. In both scenarios, we involved a human user in the learning loop as it is necessary for a human-robot interaction. In the first scenario, a user can interact with the system through the RViz[3] [131] 3D visualization environment and assign a category label to each of the detected objects on the table. After introducing the object category labels to the model, it can detect the category of the objects even if they have been placed in a different location on the table, which might change the object view partially due to the perspective or occlusion by the other objects. Finally, the clearing task is initiated in which for each individual object, the end-effector of the robotic arm moves to the pre-grasp position of a target object, and then grasps the object and puts it into a trash box located on the table (Figure 4.8.a). This demonstration showed that the system was able to detect different object

[3] ROS Visualization: http://wiki.ros.org/rviz

categories and learned about new object categories using very few examples on-site. Furthermore, it was observed that the proposed approach was able to distinguish geometrically very similar objects from each other (e.g., *Cup* vs *CokeCan*). The video of this robotic demonstration is available at: https://youtu.be/otxd8D8yYLc

The second robotic demonstration has more emphasis on category recognition of unforeseen objects and performing a category-specific robotic task. In this demonstration, a user interacts with the system through voice commands and introduces the initially located objects on the table to the model. The model uses the segmented point cloud of these table-top objects to train the model. Subsequently, three new objects will be spawned on the table in the Gazebo simulator [99]. After the detection of each of the new objects, the system tells the predicted category to the user and asks for corrective feedback in case of a wrong prediction. This way the system learns about new object categories incrementally and updates the category models once a misclassification happens.

After recognizing all object categories, the user commands the robot to clear all the coke cans from the table and put them into the trash box located on the table. To accomplish this task, the robot should detect the pose as well as the label of all objects. Then, the robot grasps and manipulates all the coke cans from the table while keeping the rest of the objects from different categories on the table (Figure 4.8.c). A video for this robotic demonstration is available at: https://youtu.be/YPsrBpqXWU4

4.6 Conclusion

We propose a non-parametric hierarchical Bayesian model called Local Hierarchical Dirichlet Process (Local-HDP) for interactive open-ended 3D object category learning and recognition. Each object is initially represented as a bag of visual words and then transformed into a high-level conceptual topics representation.

We have conducted an extensive set of experiments in both offline and open-ended scenarios to validate our approach and compare its performance with state-of-the-art methods. For the offline evaluations, we mainly used 10-fold cross-validation (train-then-test). Local-HDP outperformed the selected state-of-the-art (i.e., RACE, BoW, LDA, Local-LDA, and HDP) by a large margin, achieving appropriate computation time and object recognition accuracy. In the case of open-ended evaluation, we have developed a simulated teacher to assess the performance of all approaches using a recently proposed test-then-train protocol. Results show that the overall performance of Local-

HDP is better than the best results obtained with the other state-of-the-art approaches.

Local-HDP can autonomously determine the number of topics, even though finding a good choice for the number of topics is not a trivial task in LDA-based approaches. Moreover, the number of topics in Local-LDA should be defined in advance and is the same for all object categories, which may lead to overfitting or underfitting of the model. Local-HDP has resolved this issue by finding the number of topics for each category based on the intra-category variation of objects. Adapting online variational inference to the proposed approach enables Local-HDP to approximate the posterior for large datasets rapidly.

In order to demonstrate the applicability of the proposed approach in real-time robotic applications, two robotic demonstrations have been conducted using a UR5e robotic arm. These experiments showed that the robot was able to learn new object categories using very few examples over time by interacting with non-expert human users.

In the continuation of this work, we would like to investigate the possibility of using the proposed method for graspable part segmentation of 3D objects. This way, we can address the problem of 3D object recognition and affordance detection (i.e., detecting graspable parts) simultaneously.

CHAPTER 5

Swift Distance Transformed Loopy Belief Propagation

using a Novel Dynamic Label Pruning Method

This chapter is based on my master program in artificial intelligence at Yazd University. This research topic has been extended during my PhD. This chapter proposes a fast inference algorithm for the loopy belief propagation technique in Markov Random Fields (MRF) probabilistic graphical models [21]. This method enables an MRF model to infer the parameters in real-time and decreases the computational complexity of the previous methods. It has been utilized for the image completion (inpainting) task. Image completion is the task of filling in a relatively large missing part of an image. This is a challenging problem since in this chapter, the model only utilizes the local information of the same image to inpaint the missing part and there is no training set like in most image completion methods that are based on deep neural networks.

Swift Distance Transformed Belief Propagation using a Novel Dynamic Label Pruning Method

Abstract

Loopy belief propagation (LBP) suffers from high computational time, specifically when each node in the Markov random field (MRF) model has lots of labels. In this study, a swift distance transformed belief propagation (SDT-BP) method is proposed. SDT-BP employs an efficient dynamic label pruning approach together with distance transformation to boost the running time of the LBP. The proposed dynamic label pruning approach is independent of any specific message scheduling. The resultant solution's energy is less than Priority-BP. Furthermore, SDT-BP guarantees convergence in fewer numbers of iterations. The direct combination of distance transformed belief propagation (DT-BP) with the dynamic label pruning in Priority-BP has O(KTNlog N) computational complexity. However, the proposed method results in O(KTN) complexity. Where N is the number of nodes, K is the number of labels for each node, and T is the number of iterations. The authors conduct several experiments on image inpainting case studies, to evaluate this method. According to this analysis, DT-BP faces nearly 90% speedup by preserving the energy of the solution at almost the same level. Furthermore, this method can be utilized in any MRF model where its distance function is transformable, i.e. in various image processing and computer vision problems.

5.1 Introduction

Nowadays, Markov Random Field (MRF) is a commonplace means for solving diverse problems in computer vision, namely, image restoration, image segmentation, stereo, 3D vision, object labeling, optical flow, etc. This is specially true for cases where training based on a large dataset of images is not possible. The MRF formulation of these problems turns into energy minimization which is NP-hard. Belief Propagation (BP), introduced by Pearl. J [127], is one of the approximation algorithms proposed to deal with this issue. This algorithm can efficiently approximate the solution. It can also find the exact solution if certain constraints hold for the problem [127].

Although there were remarkable advances in the way of finding an efficient algorithm, the standard Loopy Belief Propagation (LBP) still suffers chiefly from high run-time and computational complexity. Thus, reference [61] proposed a new phenomenal approach for reducing the computational complexity of belief propagation via the notion of Distance Transformation (DT) [35]. We call this method Distance Transformed Belief Propagation (DT-BP). Komodakis et al. [101] introduced another effective method for accelerating the belief propagation. This method dynamically schedules the message passing and tries to prune the proper labels for each random variable of a MRF model during the inference phase. In this chapter, we propose a combinatorial approach by combining distance transformation with a new dynamic label pruning approach to accelerate LBP.

The general framework for many computer vision problems is defined in the following way. Let P be the set of image pixels and let L be the label state space e.g. intensity values in pixel-wise inference or similar patch offsets in patch-wise inference [76]. We use intensity values of image pixels as labels in all the experiments. A labeling f is the assignment of $f_p \in L$ to each pixel $p \in P$. The formulation is based on two assumptions: first, the output image should be similar to our input image (for inpainting application, this assumption is true for all the pixels of the image except the ones in the missing regions). Second, labels of neighboring pixels should vary smoothly all over the image except on objects' boundaries. The quality of each labeling is measured by the following energy function:

$$E(f) = \sum_{p \in P} D_p(f_p) + \sum_{(p,q) \in \mathcal{N}} V(f_p, f_q). \tag{5.1}$$

Here, $\mathcal{N}$ is the set of edges in a four-connected image grid graph. $D_p(f_p)$ is the data cost according to our first assumption. $V(f_p, f_q)$ is the smoothing or discontinuity cost corresponding to our second formulation assumption. f_p

and f_q are the labels of two adjacent pixels. Furthermore, finding a labeling with minimum energy, which is equivalent to MAP estimation problem for the corresponding MRF, is the ultimate goal. Notice that the best labeling f^* is the one with the lowest energy level:

$$f^* = \underset{f \in L}{\operatorname{argmin}}(E(f))$$

Although [101] has notable pros, its dynamic label pruning method cannot be used directly with DT-BP without increasing the computational complexity of the resulting algorithm. Komodakis et al. employed the notion of priority in their message scheduling approach. Each data structure managing to deal with priority has at least $O(logN)$ computational complexity for either insert or delete operation. Therefore, direct combination of DT-BP [61] and Priority-BP [101] has $O(KTNlogN)$ computational complexity (notice that the computational complexity of Priority-BP method is $O(K^2TNlogN)$). In contrast, our proposed approach uses a new dynamic label pruning method to resolve this issue and it results in $O(KTN)$ worst-case computational complexity, where N is the number of nodes in a MRF, T is the number of iterations of the algorithm, and K is the number of labels for each node. Due to the label pruning procedure, our proposed method accelerates DT-BP up to almost 90%.

Moreover, Komodakis et al. mentioned that their label pruning approach [101] wouldn't work properly without their priority-based message passing. However, our method can be utilized in combination with any message scheduling algorithm.

Another foremost characteristic of our approach is the guarantee of convergence. According to the experimental results, the convergence speed of the proposed method is higher than Priority-BP. This means that our approach converges in fewer numbers of iterations.

The dynamic label pruning approach in Priority-BP yields solutions with higher energy level than DT-BP. However, the solutions of our approach have quite lower energy than Priority-BP. In the case of image inpainting and denoising, we obtain results with nearly the same degree of accuracy (energy level) as DT-BP. We call our approach Swift Distance Transformed Belief Propagation (SDT-BP) method.

This chapter is organized as follows: Section 2 describes the background and related works. The proposed method is explained in Section 3. Section 4 demonstrates the experimental results and Implementation details. Possible future works are explained in Section 5. The chapter is concluded with Section 6.

5.2 Background and Related Works

In this section, we first review the Loopy Belief Propagation (LBP) inference method and the transformation of the linear distance function. Subsequently, Priority-BP is reviewed. We explain the criteria for measuring the efficiency of any energy minimization problems afterward. Finally, we will address some related works.

5.2.1 Loopy belief propagation

Belief propagation algorithm has various applications in different scientific researches [102, 151, 84, 186, 51] including computer vision and robotic fields.

The max-product algorithm can be used to find the labeling with the minimum energy level for equation (1). Conventionally, this algorithm is defined by probability distributions, but an equivalent approach is to convert them to negative log probabilities. In this way, the min-sum algorithm can be used to find the optimal labeling [61]. The min-sum algorithm is less sensitive to numerical artifacts, and it directly incorporates the energy function definition [61].

The notion of message passing has a fundamental role for understanding how original BP has excessive computational time. The max-product algorithm works by passing local messages all over the nodes of a MRF model. In our case, this model is a four-connected neighborhood system. At each iteration, every node sends messages to all of its neighboring nodes while receiving incoming messages from those nodes. This process repeats until all of these messages converge, i.e., they do not face discernible change anymore. Sometimes, the messages do not converge, and the algorithm continues until the maximum number of iterations. For instance, pixel-wise image inpainting problems usually encounter this issue.

Each message is a vector whose size is defined by the number of possible associated labels. Let m^t_{pq} be the message from node p to its adjacent node q in iteration t. Because we use the negative log in the min-sum algorithm, the messages m^0_{pq} between every pair of adjacent nodes should be initialized with zero. In each iteration, every single message is computed in the following way:

$$m^t_{pq}(f_q) = \min_{f_p} \left(D_p(f_p) + V(f_p, f_q) + \sum_{s \in \mathcal{N}(p) \backslash q} m^{t-1}_{sp}(f_p) \right). \quad (5.2)$$

Here, $\mathcal{N}(p) \backslash q$ is all the adjacent nodes with p except q.

5.2.2 Distance transform

The standard implementation of LBP algorithm without any acceleration method has $O(K^2TN)$ computational cost. We supposed that all MRF nodes have the same number of labels, which are intensity values for each pixel of the image. The complexity of computing each message using equation (2) is $O(K^2)$. Felzenszwalb et al. proposed the following distance transform approach for reducing the cost of each message computation to $O(K)$ [61]. In this approach, they used the distance transformation method [35].

Defining the smoothness cost function based on the degree of difference between labels,

$$V(f_p, f_q) = s \parallel f_p - f_q \parallel . \tag{5.3}$$

We first rewrite equation (2) as:

$$m_{pq}^t(f_q) = \min_{f_p} \Big(h(f_p) + V(f_p, f_q) \Big). \tag{5.4}$$

Here, $h(f_p) = D_p(f_p) + \sum_{s \in \mathcal{N}(p) \setminus q} m_{sp}^{t-1}(f_p)$. We initialize the message vector m with $h(f_p)$ values and then update each vector entry sequentially. Eventually, the proper distance transformed message computation method is computed in the forward and backward pass like below:

Algorithm 5.1: linear distance transformation

```
Forward Pass:
for f_q ← 1 to k − 1 do
    m(f_q) ← min(m(f_q), m(f_q − 1) + s);
end for
Backward Pass:
for f_q ← k − 2 to 0 do
    m(f_q) ← min(m(f_q), m(f_q + 1) + s);
end for
```

This distance transformation algorithm only works for a linear smoothing cost function. We call this method DT-BP. You can find more details on the distance transformation algorithms in [61] and [183].

After T iterations, belief is computed for each node:

$$b_q(f_q) = D_q(f_q) + \sum_{p \in \mathcal{N}(q)} m_{pq}^T(f_q). \tag{5.5}$$

Finally, the label f_q^* which minimizes the b_q is allocated to each node q.

5.2.3 Priority-BP

Another influential method for dealing with large label state spaces in BP was proposed in [101]. This method is known as Priority-BP. Their approach has two main extensions to standard BP: dynamic label pruning which occurs while BP is running. It is based on the principal idea of dramatically reducing the number of labels to enhance the BP algorithm. The second extension is named priority-based message passing using label pruning and allowing the algorithm to send cheap messages between the nodes of a MRF model [101]. Consequently, it increases BP's convergence speed, thus accelerating this algorithm. In spite of these advantages, our experiments revealed that this algorithm leads to considerable energy level increase comparing to DT-BP in pixel-wise inference. Notice that the direct combination of DT-BP and Priority-BP is impossible without increasing the computational complexity of DT-BP. The reason behind this fact is explained in the introduction section.

5.2.4 Measuring efficiency

There are two criteria for measuring the efficiency of any algorithm in this domain: energy level and running time. The energy level (equation (1)) pertains to the quality of our solution. As we defined in the previous section, the solution with less energy is more desirable. After T iterations of our algorithm, we should first find the f_q^*:

$$f_q^* = \underset{f_q \in L}{\operatorname{argmin}}\, b_q(f_q). \tag{5.6}$$

For each node, this obtained f_q^* is assigned to node q. Subsequently, the energy level is computed with equation (1).

5.2.5 Image inpainting and Image completion

Image inpainting is the process of plausibly filling in small missing regions within an image. Image completion is another similar computer vision problem which covers a broader range of applications. However, these terms are often used interchangeably. There are many researchers focusing on finding efficient (in terms of accuracy and run-time) solutions for image inpainting problems. Achanta et al. [2] proposed a solution for the challenging problem of completing an image whose 99% of pixels are randomly missing. Although the computational complexity of their method is linear in the number of pixels of the full image, it fails to fill big missing regions and holes. Recently, image completion techniques based on deep neural networks have been widely

used. The methods in [81] and [125] use deep neural network to complete images of arbitrary resolutions. Gao et al. [70] proposed another data-driven image completion method. Moreover, Yu et al. [183] proposed a generative model-based method which not only can reconstruct the missing regions but it can also explicitly use the surrounding image features while it is training the network. Using deep neural networks has also been used for video completion [180]. Although deep learning-based approaches can outperform pixel-wise and patch-wise image completion in terms of the quality of results, they are highly dependent on the training set which may not be available for specific applications. If the picture to be inpainted is nothing like any of the samples in the training set, their output will not be plausible. The other consequence of using this kind of methods is their long training times.

The result of the proposed method has not been compared with deep neural network based image completion approaches because of two reasons. 1) The purpose of using the proposed method in image completion problems is to fill in the missing regions by just using the local information available in the single image itself (the neighborhood of the filling region). Therefore, the proposed method does not use a separate training and testing phase using a large set of images for tuning the parameters of the model as most of the deep learning approaches do. 2) Since our focus is on accelerating the inference phase for real-time applications, it is not possible to use deep neural network based models which usually need long training time.

We conducted experiments on three image inpainting and denoising case studies to elucidate the effectiveness of our proposed approach.

5.2.6 Other related works

There are other attempts to enhance BP's computational cost and running time by properly scheduling message passing and even altering the message computation formula. The so-called tree-reweighted message passing algorithm ([167], [168], and [100]) slightly differs from the standard BP in message computation equation. Moreover, reference [60] used a simple idea for message scheduling in which the message with the largest abrupt change in two successive iterations has the most priority. Practically, this approach leads to faster convergence, and it decreases the likelihood of getting stuck in local minima. There is no consideration for handling large label space in these approaches. Besides, Sudderth et al. [15] and Isard [82] have independently proposed a non-parametric BP algorithm. Their technique is based on an efficient sampling procedure, but it can be used mainly for extending BP to non-discrete distributions.

Generalized Belief Propagation (GBP) [181] is a region-based BP algorithm leading to faster convergence in Markov random fields. However, its computational time is high for practical applications. Therefore, [41] and [40] independently tried to accelerate GBP with diverse methods.

The problem of image inpainting and denoising with various inference methods is considered comprehensively by [155] and [89]. Although stating the image inpainting problem is simple, finding a plausible solution for this problem is far from a trivial task, specifically in pixel-wise methods.

The computational complexity of distance transformed belief propagation is $O(KTN)$. Moreover, the time complexity of the Priority-BP method in [101] is $O(K^2TNlogN)$. Therefore, its running time is higher than DT-BP. According to our experimental results, the quality of the Priority-BP's solutions is almost 5% lower (5% higher energy level) than DT-BP. In this chapter, we propose a new belief propagation algorithm which is swifter than DT-BP and has quite comparable results with DT-BP.

5.3 Swift Distance Transformed Belief Propagation

This section covers our proposed method, which reduces the message passing computational complexity and run-time by using distance transformation and a new dynamic label pruning approach. We have covered distance transformation for the linear distance function in the previous section. We explain our dynamic label pruning method in this section. It has been proven that belief propagation is impractical for problems with an extensive number of labels [101]. In order to overcome this problem, we prune the labels which correspond to improbable label assignments. Furthermore, we talk about the incorporation of the proposed dynamic label pruning method in the distance transformed belief propagation approach. Besides, when the algorithm does not converge, we propose to resolve the issue with an early stopping criterion.

5.3.1 Dynamic label pruning

As we mentioned in the introduction section, the direct combination of DT-BP [62] and Priority-BP [101] leads to an increase in computational complexity. Therefore, we propose a new dynamic label pruning method which is not dependent on any specific message scheduling and keeps the worst-case computational complexity as low as DT-BP (i.e. $O(KTN)$). It is important to note that the dynamic label pruning capability of the proposed approach leads to

much lower run-time than DT-BP. This is due to the fact that K (the number of labels for each node) is continuously decreasing in the successive iterations of our approach.

The label pruning method is the main part of our algorithm. Therefore, the following preliminary steps are essential. First, a belief vector is computed using equation (5) for each node. We normalize beliefs by subtracting the least value in each belief vector from all belief values in that vector. Like [101], we also call these normalized beliefs the *relative beliefs* ($b_q^{rel}(f_q)$) . As mentioned in equation (6), the labels with lower belief values are more desirable to choose. At this point, the pruning procedure is ready to take a roll. However, we need a proper method for assessing the possibility of pruning of each node.

Finding eligible labels for pruning is another fundamental step in the label pruning procedure. If a node is quite uncertain about choosing the proper label assignment, no pruning should take place. For instance, the nodes in the missing region are not certain enough about appropriate labels at early iterations. Therefore, we need to measure each node's confidence level. Thanks to [101], we have the idea for finding this confidence level. The confidence measurement approach originates from the following simple assumption. For each node and belief vector, we count the number of belief values that are below a specific threshold noted by B_{conf}. Lower count means the node has higher certainty about choosing the proper label.

Before going into details of this approach, we should mention the following fact. [55] and [48] have already used confidence level in their works. Reference [101] uses this concept in priority-based message passing which is inseparable from the dynamic label pruning approach. However, we used confidence level for evaluating the possibility of pruning for each node and also finding the proper labels to be pruned.

Algorithm 5.2 measures confidence level for every single node. The *count* value is in scope $[1,K_{np}]$ where K_{np} is the number of labels which have not been pruned in the current iteration for the associated node. Notice that in each iteration, we do not count the labels which have previously been pruned. When the count value is high, there are many labels among which the node should choose. However, a node with low *count* value is more confident about the most probable label assignment.

Confidence level has a strong correlation with the number of possible labels for pruning. In other words, high confidence level lets the algorithm prune more labels safely. Therefore, we consider the following three assumptions in the label pruning procedure:

- Least confidence level ($count = K_{np}$) means no pruning,

Algorithm 5.2: confidence level computation for each node

```
for f_q ← 1 to k do
    if f_q has not been pruned then
        Increase the count value if b_q^rel(f_q) < B_conf;
    end if
end for
```

- Highest confidence level ($count = 1$) means pruning of all labels except the most probable one,

- For confidence levels falling in scope $(1,K_{np})$, we should use a proper mapping.

We define a new threshold for label pruning indicated by B_{prune}. In our approach, B_{prune} is in scope $[B_{conf},B_{max}]$ where $B_{max} = \max(b_q^{rel}(f_q))$. The algorithm prunes any label whose relative belief value is higher than B_{prune}. Moreover, finding B_{prune} according to confidence level makes our label pruning algorithm independent of any specific message scheduling algorithm. In order to find B_{prune} w.r.t confidence level, we employ the three aforementioned assumptions. Fig. 5.1 shows the corresponding linear mapping from *Count* value to B_{prune}.

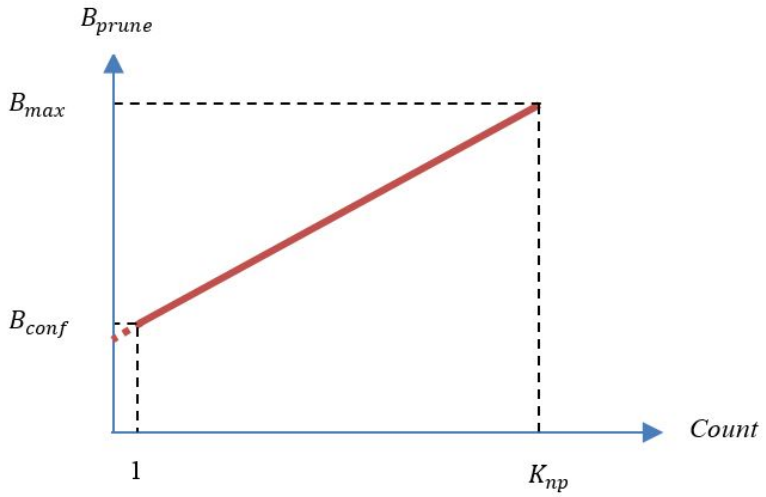


Figure 5.1: Linear mapping between confidence level (*Count*) and pruning threshold (B_{prune}). K_{np} is the number of not pruned labels in the current iteration.

After computing B_{prune}, the algorithm prunes the labels whose relative beliefs are above this threshold. Using the proposed linear mapping, nodes with a higher confidence level have a lower pruning threshold. Thus, the pruning method prunes more labels for these nodes. Notice that in each iteration, SDT-BP employs the pruning procedure after the message computation process for all nodes.

As shown in Fig. 5.1, the linear mapping is only sensitive to B_{conf}. There are two choices for determination of the appropriate B_{conf}. Either assigning a static B_{conf} for all nodes or dynamically finding the proper B_{conf} for every single node of the MRF model. The following subsection discusses the results of these choices in detail.

5.3.2 Constant versus dynamic B_{conf}

We found out that B_{conf} has a key role in the balance between the run-time and energy of the final solution. As we mentioned in the previous subsection, there are two alternatives for B_{conf} determination.

- Constant (fixed) B_{conf} for all the nodes,
- Dynamic B_{conf} for every single node.

The higher B_{conf} means larger count value. Therefore, the linear mapping leads to higher B_{prune} and less pruning. As a result, the final solution achieves a lower energy level at the cost of higher run-time than the second alternative. On the other hand, choosing lower B_{conf} has the inverse results. According to the aforementioned facts, there is a trade-off between run-time and energy level balancing by B_{conf}. Therefore, finding the proper constant B_{conf} for all nodes is important.

Furthermore, every single node can determine the B_{conf} value dynamically according to its own belief vector. We tried many alternatives and eventually chose $B_{conf} = \frac{B_{max}}{3}$. When the confidence level of a node is high, there are only a few labels with zero relative beliefs. The other labels have much higher relative beliefs. Thus, assigning $B_{conf} = \frac{B_{max}}{3}$ leads to an intense pruning rate. On the other hand, many near-zero relative beliefs exist when the confidence level is low. In preliminary iterations of the algorithm, some higher relative beliefs exist, too. So, assigning $B_{conf} = \frac{B_{max}}{3}$ preserves the near zero labels and prunes the other ones. If a node does not become confident enough after notable iterations, it normally oscillates between specific numbers of labels. These labels usually have high similarity to each other. Such cases hinder the

convergence of the algorithm. By assigning $B_{conf} = \frac{B_{max}}{3}$, the node is forced to choose the most probable label by pruning the other ones.

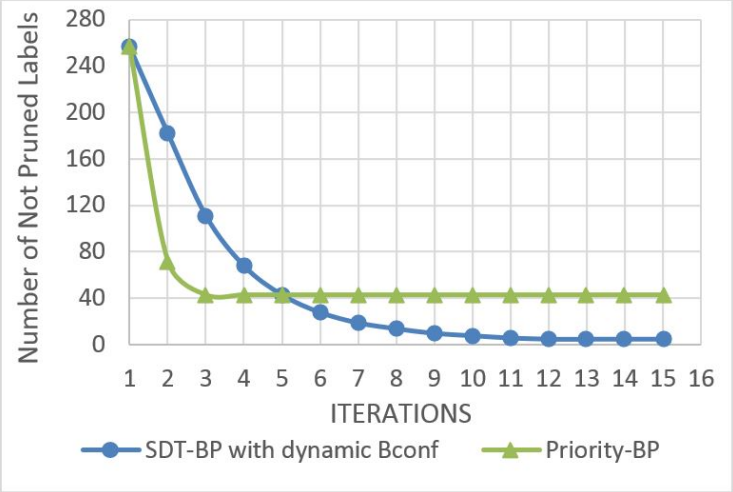


Figure 5.2: The comparison of the pruning process between SDT-BP and Priority-BP. Although SDT-BP prunes more labels after 16 iterations, it reaches lower energy level than Priority-BP according to our experimental results. Priority-BP greedily prunes many labels at early iterations. This comparison is done for the pixel with coordination (100, 10) in our first case study (the penguin case study).

This dynamic B_{conf} leads to the guarantee of convergence in a few iterations. While the algorithm goes on, most nodes are gradually forced to prune all their labels except the most probable one. The *pruning map* of Fig. 5.3 shows this fact. In this way, even if there are some nodes which are not sure about their best label assignment, they will be persuaded by other nodes to prefer one of these label assignments. The guarantee of convergence is at the cost of a small increase in energy level. We should emphasize that Priority-BP caused considerable energy level increase comparing with DT-BP. According to experimental results, our proposed method outperforms Priority-BP in terms of both the solution's quality (lower energy level) and the run-time. This discernible energy increase in [101] is due to the fact that the pruning threshold (B_{prune}) is a constant value. In this way, even the nodes with a low confidence level, which have a low priority in each iteration, prune some of their valuable labels. Moreover, constant B_{prune} leads to intense pruning in initial iterations of the algorithm as shown in Fig. 5.2. This makes the algorithm disregard a few informative labels at early iterations. Therefore, Priority-BP prunes labels more greedily than SDT-BP.

Consequently, dynamic B_{conf} determination always leads to convergence,

but static B_{conf} does not guarantee it like DT-BP. In order to reasonably compare our method using static B_{conf} with DT-BP in non-converging cases, we propose a simple early stopping criterion for BP. Notice that this early stopping criterion does not have any effect on the termination of our proposed method using dynamic B_{conf}.

5.3.3 Early stopping criterion for belief propagation

Occasionally, the BP algorithm does not converge even after a considerable number of iterations. In order to perform a more legitimate comparison for such cases, we need an early stopping criterion for belief propagation. Furthermore, as mentioned in the previous section, one of the most paramount features of dynamical B_{conf} computation is the guarantee of convergence in a few iterations.

After each iteration of the algorithm, we need to find out whether any assigned labels of nodes have been changed since the previous iteration. If there is no change in label assignment, the algorithm is converged. This is the *normal stopping criterion* for BP. In addition to such procedure, we also count the number of nodes which have different assigned labels from their previously assigned ones. We refer to such count value as $\mathcal{C}$. Consequently, the Algorithm 5.3 is the proposed early stopping criterion for belief propagation.

We used $\beta = 30$ for all our experiments. This means that after 30 successive iterations, if the value $\mathcal{C}$ does not decrease, we suppose the algorithm is oscillating between several repetitious label assignments and there is no way for convergence. In practice, this criterion worked fine for all converging and non-converging cases. In other words, if the algorithm converges, this criterion mostly does not interfere with the convergence and does not stop the algorithm earlier than the convergence time. That is why we choose such a high β value.

Notice that using energy level instead of $\mathcal{C}$ and minimum energy level instead of $min\mathcal{C}$ in Algorithm 5.3 will lead to almost the same results in all the experiments. For the sake of conciseness, we do not mention such results with energy level-based early stopping criterion.

5.3.4 Incorporating dynamic label pruning in DT-BP

After each iteration, the appropriate labels are pruned safely. So it is time to disregard the pruned labels in the procedure of distance transformed message passing. As we previously mentioned, the message vectors are initialized with $h(f_p)$ values. In order to incorporate the label pruning effect in this process, we do not compute $h(f_p)$ values for the pruned f_p labels. Instead, we initialize their corresponding message value with a large constant integer (1000 is used

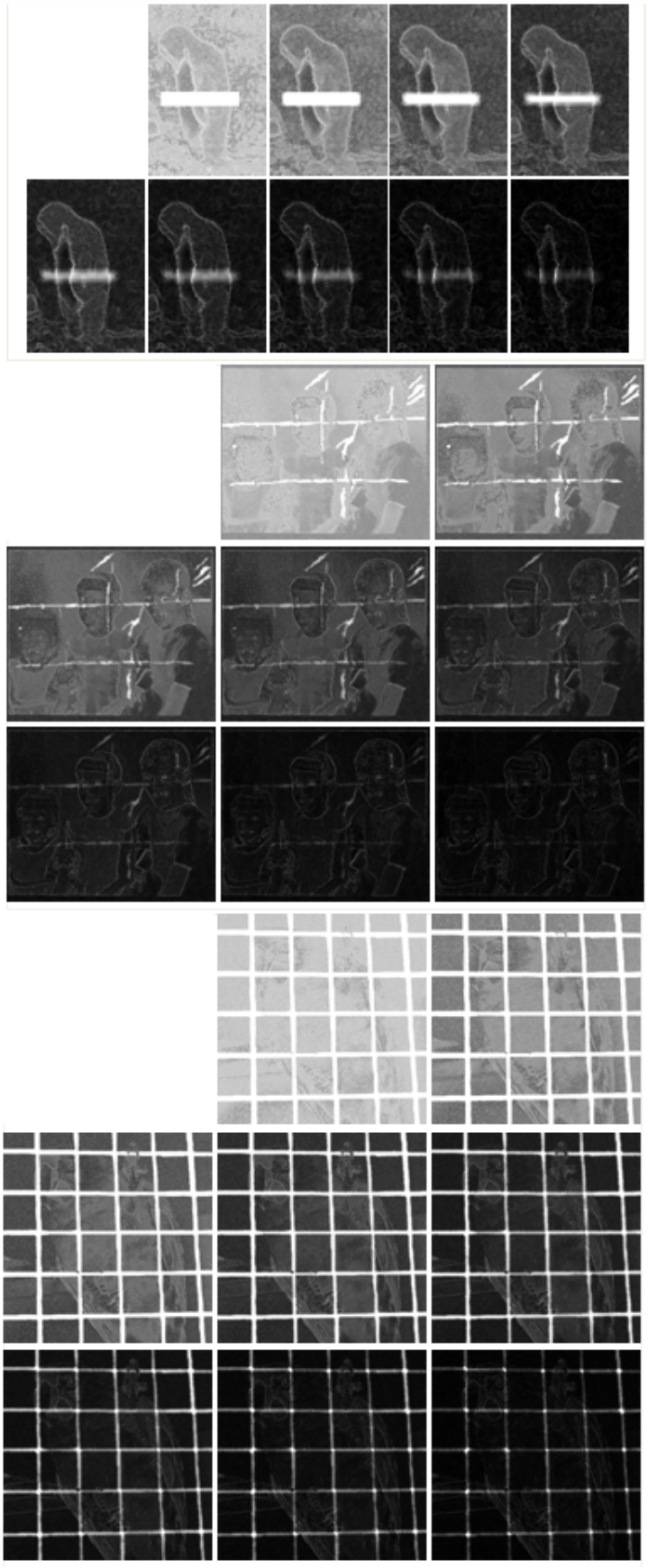


Figure 5.3: Pruning map. Darker pixels express that more labels are pruned for the corresponding node. This pruning map is for the primary iterations of SDT-BP with dynamic B_{conf}.

Algorithm 5.3: first early stopping criterion

```
1: define st = 0
2: define minC as a big integer constant
3: while st <= β do
4:    compute C
5:    st = st + 1
6:    if C < minC then
7:       minC = C
8:       st = 0
9:    end if
10: end while
```

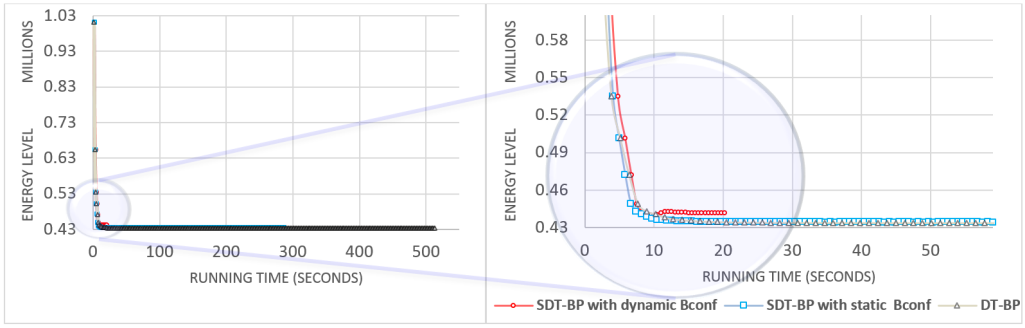

in our experiments). Subsequently, the distance transformed message passing procedure for linear smoothing cost function can disregard the pruning labels in the forward and backward pass of Algorithm 5.4 (because it uses the minimum function).

Algorithm 5.4: SDT-BP distance transformation

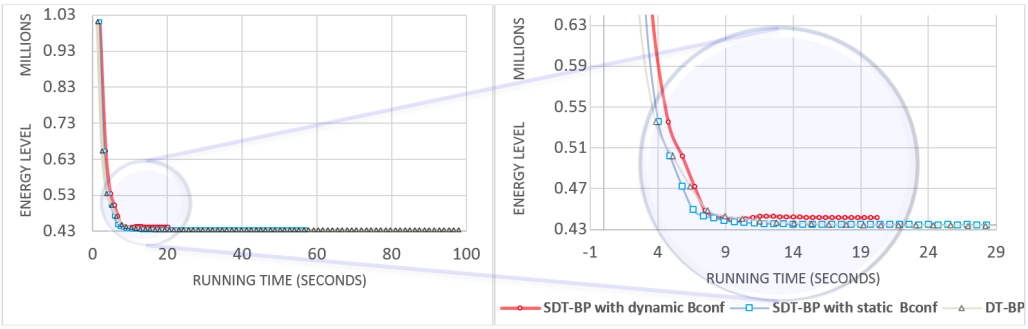
```
1: for f_q ← 1 to k − 1 do
2:    if f_q is not pruned then
3:       m(f_q) ← min(m(f_q), m(f_q − 1) + s);
4:    end if
5: end for
6: for f_q ← k − 2 to 0 do
7:    if f_q is not pruned then
8:       m(f_q) ← min(m(f_q), m(f_q + 1) + s);
9:    end if
10: end for
```

Consequently, after each iteration of the distance transformed message passing algorithm, we use the proposed dynamic label pruning method for all nodes. Notice that we also ignored pruned labels in the process of computing $\mathcal{C}$ in Algorithm 5.3.

Assuming that T is the number of iterations of the proposed method and N is the number of nodes in the MRF model and K is the number of labels for each node, the inference procedure for all the iterations of the algorithm in the worst case has $O(KTN)$ computational complexity which is lower than $O(K^2TNlogN)$ for the priority-BP. It is important to emphasize that, due to the pruning procedure, K (the number of labels for each node) is decreasing in consecutive iterations of the algorithm. This means that practically much lower than $O(KTN)$ operations are needed for the proposed algorithm.



(a)



(b)

Figure 5.4: The penguin case study running time - energy level plot with normal convergence criterion (**a**) and with early stopping criterion (**b**). **Left** is the full-size plot. **Right** is the zoomed in plot. Notice that each marker point is an iteration of the algorithm.

5.4 Experiments

In this section, we evaluate the efficiency of our proposed algorithm for accelerating distance transformed belief propagation. There are three examples of image inpainting and denoising case studies. We compare our algorithm with DT-BP in run-time and energy-level of the final solution. Fig. 5.4 to Fig. 5.9 illustrate the ultimate results of our proposed method (SDT-BP) versus distance transformed belief propagation method (DT-BP).

Before going into detailed results, notice that all the algorithms are implemented in C++ on Linux Ubuntu. The experiments were conducted on an Intel core 2 duo machine with 2.53GHz CPU and 4 GB RAM. We fixed the parameters to $\beta = 30$ and $s = 1$ in SDT-BP and DT-BP. Moreover, we used the proposed scheduling method in [61] and accelerated updating of [101] for SDT-BP and DT-BP. Notice that our implementation of DT-BP and SDT-BP only differs in dynamic label pruning procedure and its incorporation into the process of distance transformation. The other parts of these implementations are exactly the same. In order to implement Priority-BP, the parameters are

set in the following way. $L_{min} = 3$, $L_{max} = 20$, $B_{conf} = 80$, $B_{prune} = 100$.

The penguin case study is a noisy picture with a missing region to be inpainted. The size of this image is 122×179 pixels. As shown in Fig. 5.5, the visual qualities of SDT-BP and DT-BP solutions are almost the same. Using static B_{conf} determination leads to quite identical results to DT-BP. Fig. 5.4.a illustrates the running time - energy level plot for the penguin case study by utilizing the normal stopping criterion and dynamic B_{conf}. Notice that DT-BP is not converging in this case and its run-time is 513.08 second. Therefore, these algorithms are terminated by the defined maximum number of iterations, which is 400 in all the experiments. As we have proven in the previous section, SDT-BP with dynamic B_{conf} is converged in 31 iterations and 20.19 seconds. Fig. 5.4.a also demonstrates the running time - energy level plot for SDT-BP with static B_{conf} and normal stopping criterion. Both DT-BP and SDT-BP with static B_{conf} do not converge in the specified number of iterations.

In order to have a more rigorous comparison with DT-BP, we used our proposed early stopping criterion with dynamic B_{conf} in Fig. 5.4.b. This figure illustrates the running time - energy level plot for the penguin case study. DT-BP is terminated by our early stopping criterion in 76 iterations and in 97.92 seconds. The running time - energy level plot for SDT-BP with static B_{conf} alongside the early stopping criterion is also illustrated in Fig. 5.4.b which runs in 56.79 seconds. Both DT-BP and SDT-BP with fixed B_{conf} are stopped in 76 iterations by the early stopping criterion, whereas SDT-BP with dynamic B_{conf} converged in 31 iterations and 20.19 seconds.

Table 1 shows the numerical results of comparing our algorithm with DT-BP for SDT-BP with static and dynamic B_{conf}. Dynamic B_{conf} determination with normal stopping criterion has 96.06% running time reduction in the penguin case study. Using the proposed early stopping criterion with dynamic B_{conf} leads to 79.37% acceleration. On the other hand, the energy level increase for both of these specifications is 1.85%.

Using static B_{conf} with normal stopping criterion results in 44.22% speed up and 0.18% energy level increase in the penguin case study according to Table 1. Utilizing our proposed early stopping criterion with static B_{conf} yields 41.99% acceleration and 0.18% energy level increase. Notice that in all cases with static B_{conf} determination, SDT-BP and DT-BP are terminated due to the early stopping criterion.

Table 2 demonstrations the energy level increase of our proposed method with dynamic B_{conf} versus Priority-BP comparing with DT-BP. The Priority-BP has 6.16% energy level increase for the penguin case study while our proposed method has only 1.85% increase. Furthermore, our method leads to faster convergence. SDT-BP with dynamic B_{conf} converges in 31 iterations whereas Priority-BP does not converge in the specified maximum number of

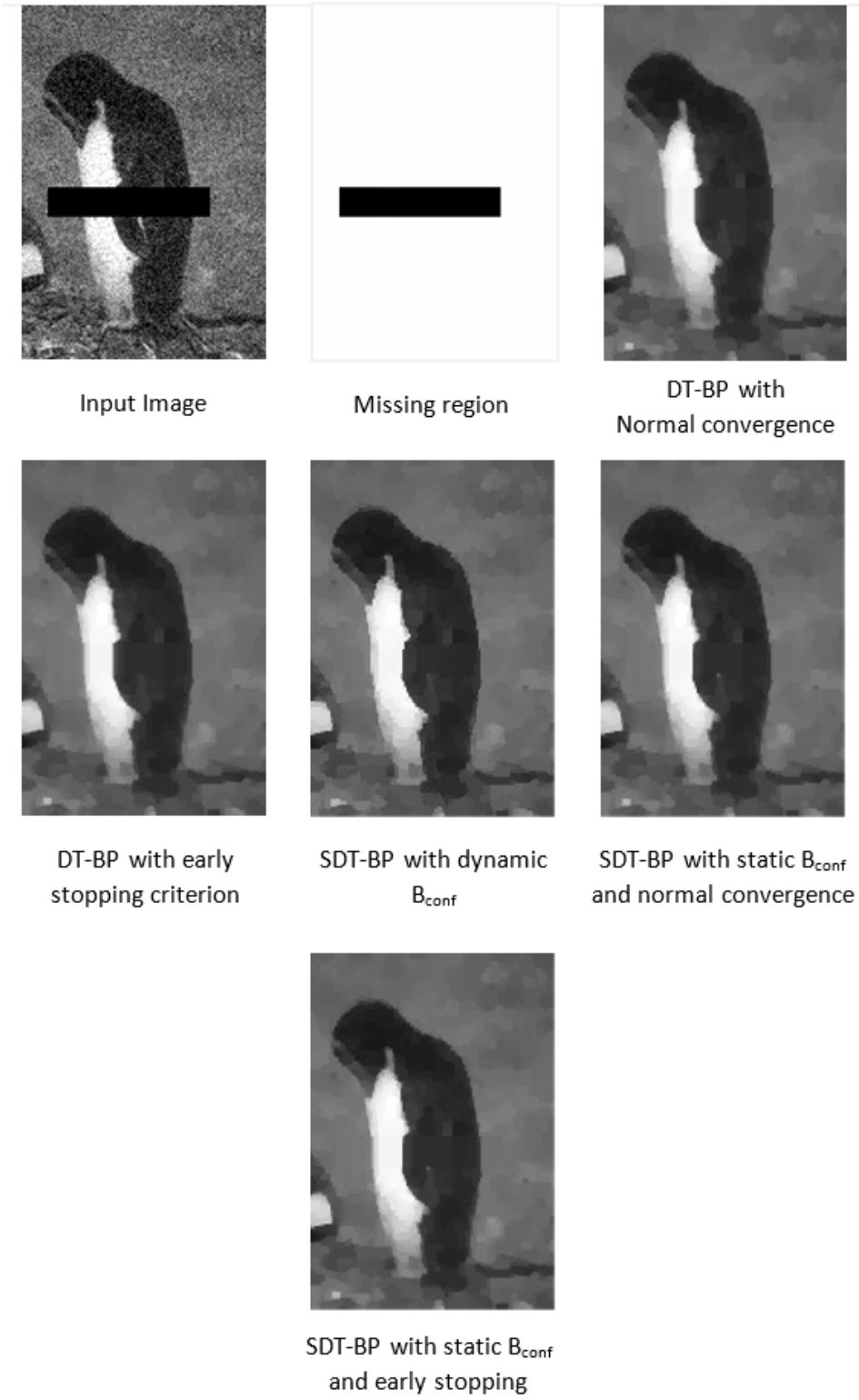


Figure 5.5: The penguin case study. Results of applying different methods with different possibilities of stopping criterion and B_{conf}. The size of all these images is 122×179.

Table 5.1: Acceleration and energy level increase percentages comparing our proposed method with DT-BP. Notice that $\beta = 30$ for the proposed early stopping criterion. SDT-BP has $B_{conf} = 80$ in all the static cases.

mapping	Stopping criterion		Penguin	Three Girls	Parrot
Dynamic B_{conf}	Normal	acceleration	96.06 %	95.68 %	93.57 %
		Energy increase	1.85 %	1.08 %	1.36 %
	Early stopping	acceleration	79.37 %	80.47 %	89.96 %
		Energy increase	1.85 %	1.08 %	1.36 %
Static B_{conf}	Normal	acceleration	44.22 %	44.83 %	45.12 %
		Energy increase	0.18 %	0.27 %	0.12 %
	Early stopping	acceleration	41.99 %	43.54 %	56.24 %
		Energy increase	0.18 %	0.27 %	0.12 %

Table 5.2: Comparing SDT-BP using dynamic B_{conf} and Priority-BP with DT-BP. The comparison is done on energy level increase and the number of iterations for convergence. The symbol $\times$ means that the corresponding algorithm does not converge in specified number of iterations.

method	Penguin		Three Girls		Parrot	
	Energy increase	iterations	Energy increase	iterations	Energy increase	iterations
SDT-BP	1.85%	31	1.08%	37	1.36%	54
Priority BP	6.16%	$\times$	5.12%	$\times$	6.24%	$\times$

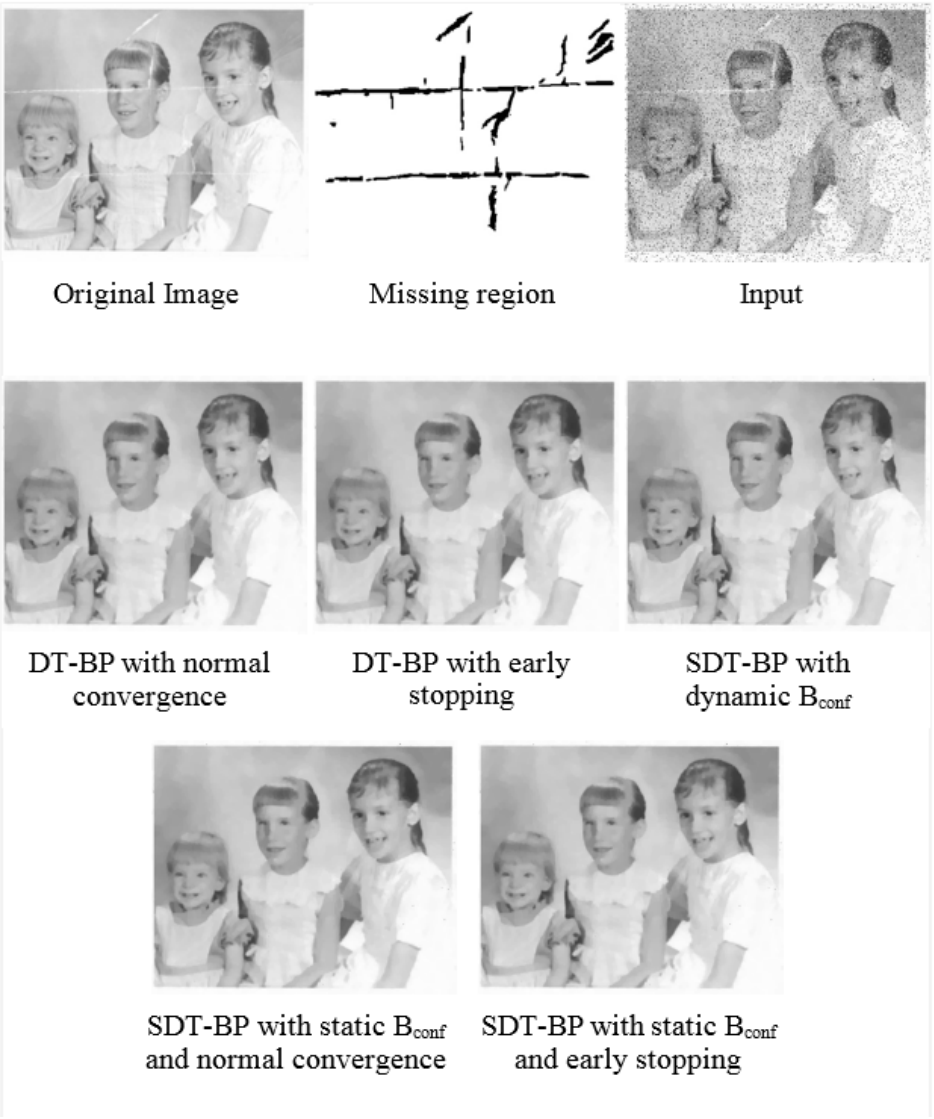
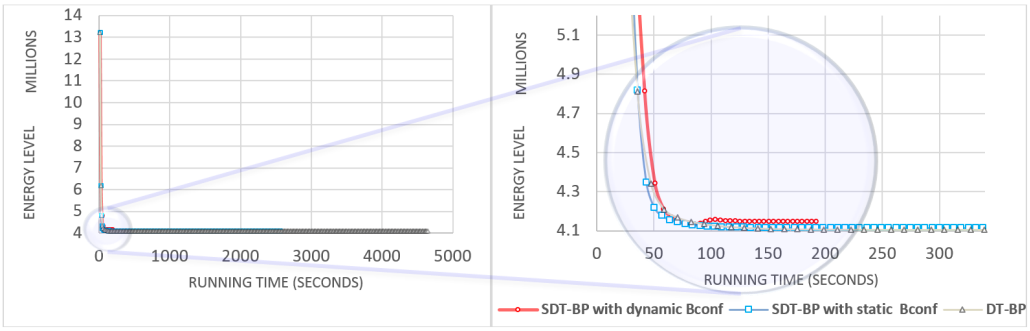


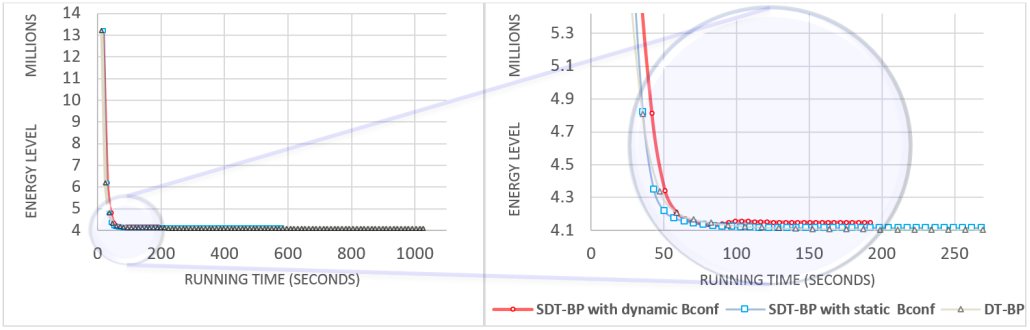
Figure 5.6: The three girls case study. Results of applying different methods with different possibilities of stopping criterion and B_{conf}. The size of all these images is 483×405. Input is the result of adding salt and pepper noise to the original image.

iterations. The running time of Priority-BP is extremely higher than DT-BP. Besides, Priority-BP has extremely higher computational complexity. Therefore, we do not compare our proposed method with Priority-BP in running time.

The three girls case study is a scratched image and our goal is to inpaint the missing region. In order to add noise to this image, we used salt and pepper



(a)

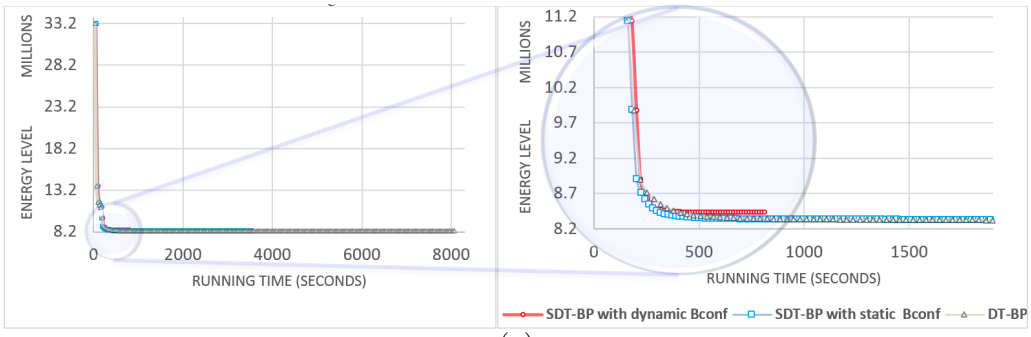


(b)

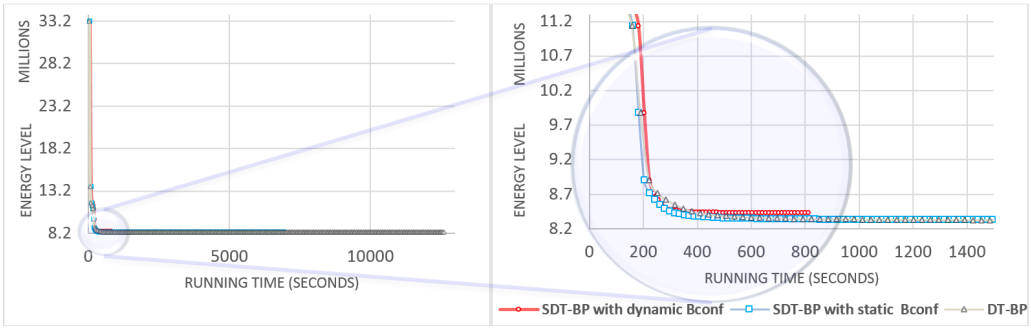
Figure 5.7: The three girls case study running time - energy level plot with normal convergence criterion (**a**) and with early stopping criterion (**b**). **Left** is the full-size plot. **Right** is the zoomed in plot. Notice that each marker point is an iteration of the algorithm.

noise. The image size is 483×405 pixels. In the case of dynamic B_{conf} in Fig. 5.6, the quality of SDT-BP and DT-BP solution varies in small extent. SDT-BP with static B_{conf} results in visually the same solutions as DT-BP. Fig. 5.7.a is the energy level - running time plot for the three girls case study by utilizing the normal stopping criterion. DT-BP is not converging in this case. However, SDT-BP with dynamic B_{conf} is converging in 37 iterations as expected. The running time - energy level plot that employs the static B_{conf} with normal stopping criterion is also illustrated in Fig. 5.7.a. SDT-BP with fixed B_{conf} is non-convergent. Fig. 5.7.b is the running time - energy level plot for the proposed method with early stopping criterion and dynamic B_{conf}. DT-BP terminated in 88 iterations by the early stopping criterion. The running time - energy level plot for SDT-BP with fixed B_{conf} and early stopping criterion is also represented in Fig. 5.7.b.

According to Table 1, SDT-BP with dynamic B_{conf} leads to 95.68% running acceleration and 1.08% energy level increase using normal stopping criterion for the three girls case study. Incorporating our proposed stopping criterion



(a)



(b)

Figure 5.8: The parrot case study running time - energy level plot with normal convergence criterion (**a**) and with early stopping criterion (**b**). **Left** plot is the full-size plot. **Right** plot is the zoomed in plot.

results in 80.47% running time reduction and 1.08% energy level increase.

It is shown in Table 1 that SDT-BP with static B_{conf} and normal stopping criterion leads to 44.83% acceleration and 0.27% increase of energy level comparing with standard DT-BP approach. Employing the proposed early stopping criterion SDT-BP yield 43.54% run-time reduction and 0.27% energy level increase.

According to Table 2, the energy level increase of Priority-BP is 5.12% for the three girls case study while SDT-BP results in 1.08% increase. Therefore, SDT-BP yields more accurate results. Moreover, SDT-BP with dynamic B_{conf} converged in 37 iterations while Priority-BP does not converge in the specified number of iterations. Consequently, the convergence speed of SDT-BP is higher than Priority-BP.

The parrot case study is a picture to be inpainted, in which the cage has to be removed. The image size is 731×736 pixels. We add salt and pepper noise to this picture. As it is shown in Fig. 5.9, the difference in the quality of our solution and DT-BP solution is hardly discernible when the dynamic B_{conf} determination is used. SDT-BP with static B_{conf} results in quite similar results. The running time - energy level plot for SDT-BP with dynamic B_{conf} and

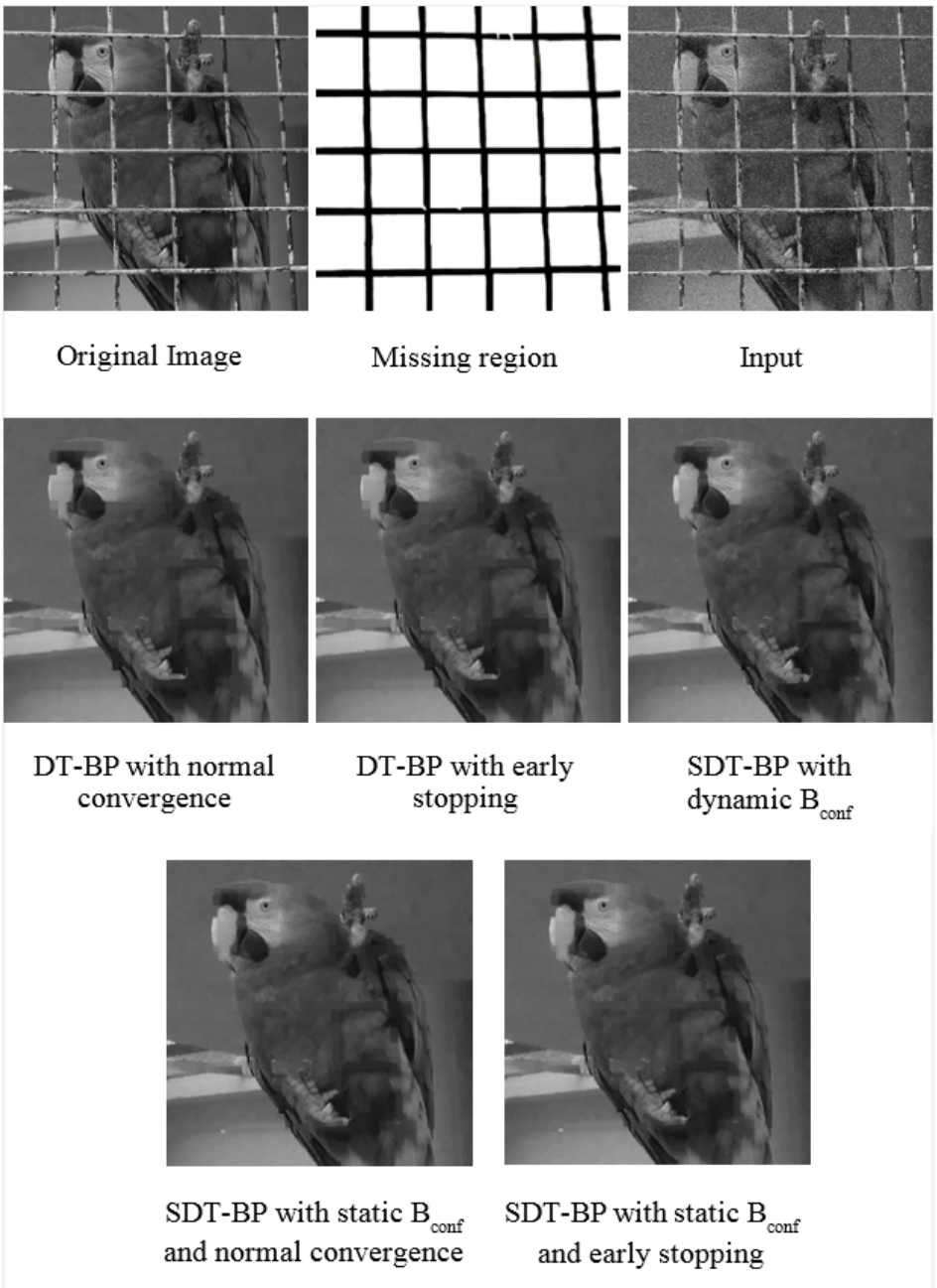


Figure 5.9: The parrot case study. Results of applying different methods with different possibilities of stopping criterion and B_{conf}. The size of all these images is 731×736. Input is the result of adding salt and pepper noise to the original image.

normal stopping criterion is shown in Fig. 5.8.a. DT-BP is not converging in this case study again. SDT-BP with dynamic B_{conf} converges in 54 iterations. The running time - energy level plot for SDT-BP using static B_{conf} and normal stopping criterion is also presented in Fig. 5.8.a. Fig. 5.8.b demonstrates the running time - energy level plot employing dynamic B_{conf} and the proposed early stopping criterion. DT-BP stopped in 256 iterations. Fig. 5.8.b is also the illustration of running time - energy level plot for SDT-BP with fixed B_{conf} and the proposed early stopping criterion. SDT-BP with static B_{conf} stopped in 196 iterations in this case.

According to the last column of Table 1, SDT-BP accelerates DT-BP 93.57% and increases the energy level 1.36% employing dynamical determination of B_{conf} and normal stopping criterion. Assimilating the early stopping criterion with dynamic B_{conf} in SDT-BP leads to 89.96% acceleration and 1.36% energy level increase comparing to DT-BP in the parrot case study.

Table 1 shows that employing static B_{conf} alongside normal stopping criterion yield 45.12% acceleration and 0.12% energy level increase w.r.t DT-BP on the parrot case study. The early stopping criterion incorporated in SDT-BP with fixed B_{conf} leads to 56.24% acceleration and 0.12% energy level increase.

According to Table 2, Priority-BP results in 6.24% increase in energy level. Moreover, SDT-BP with dynamic B_{conf} converges in 54 iterations and Priority-BP does not converge in 400 iterations. This shows higher convergence speed in the proposed approach.

Notice that in all the experiments, the well-known distance transformed BP does not converge. Our proposed algorithm with dynamic B_{conf} has converged in a few iterations in all the experiments. Moreover, Priority-BP does not converge in all cases. Therefore, SDT-BP with dynamic B_{conf} has higher convergence speed than Priority-BP. SDT-BP static B_{conf} and DT-BP are terminated due to early stopping criterion in all case studies.

The pruning map for all the case studies is shown in Fig. 5.3. The pruning map shows the number of pruned labels for each pixel. Darker pixels stand for more pruned labels for the corresponding node in the MRF model. It is conspicuous that after a few iterations, most nodes prune all their labels except the most probable one.

5.5 Future works

Our proposed method enhances the computational complexity and run-time of the belief propagation algorithm by pruning the appropriate labels securely and dynamically. In all our experiments, there are 256 labels for the nodes (corresponding to the gray-scale intensity values of pixels) in the MRF model.

We believe that our SDT-BP algorithm can engender better results in other problems with larger label state space. Consequently, this algorithm can be incorporated more successfully in other domains like patch-based image completion, etc. where there exist numerous numbers of labels. Distance transformation is not always feasible in all of these cases. Therefore, we need a new general-purpose distance transformation method usable for a broader range of distance functions.

5.6 Conclusion

Loopy belief propagation (LBP) as an approximation inference algorithm for Markov random fields is widely used in the recent researches. The main limitation of the LBP algorithm is its high computational time for message passing, especially in problems with large label state space. We have proposed a novel approach named the swift distance transformed belief propagation (SDT-BP) which substantially reduces the run-time of the distance transformed belief propagation (DT-BP) method, leading to solutions with almost the same quality. The notion of dynamic label pruning is used, which is independent of the message scheduling algorithm, unlike the other methods like priority-BP, which is limited to certain kind of message scheduling. This new label pruning approach is incorporated alongside distance transformed belief propagation. Moreover, SDT-BP outperforms Priority-BP in both the quality of the ultimate solution and convergence speed. The proposed method converges in fewer iterations than Priority-BP. Proved in previous sections, the direct combination of Priority-BP and DT-BP has at least $O(KTNlogN)$ time complexity. In contrast, the computational complexity of our proposed approach is $O(KTN)$. Furthermore, this method is faster than DT-BP because of the new dynamic label pruning approach incorporated in our method. As Komodakis et al. mentioned in Priority-BP's paper [101], their label pruning method malfunctions unless being used beside priority-based message passing algorithm. However, SDT-BP can be incorporated in combination with any message scheduling algorithm. The other foremost facet of the proposed approach is the guarantee of convergence in a low number of iterations when dynamic B_{conf} is used. We used several case studies in image completion and inpainting to evaluate the proposed method.

CHAPTER 6

Explain What You See

3D Object Recognition and Parts Segmentation using Local-HDP and Argumentation

This chapter is prepared for a submission. In this chapter, we first adapt the proposed local hierarchical Dirichlet process (Chapter 4) for the 3D object part segmentation task. The proposed technique does not require the dictionary construction step and it is more suitable for open-ended scenarios. Subsequently, argumentation-based learning is integrated with the proposed segmentation technique to recognize the category of 3D objects. Integrating these models leads to explainability of the object recognition module as well as to robustness when facing uncertainty. This enables the model to handle high degrees of occlusion in the testing phase, outperforming state-of-the-art techniques by a large margin. The resulting model produces explanations for the object recognition task in the context of the learned object parts. This makes the model more transparent and easier to debug. The results show that the proposed model has higher mean Intersection over Union (mIoU) and recognition accuracy than state-of-the-art techniques.

Explain What You See: 3D Object Recognition and Parts Segmentation using Local-HDP and Argumentation

Abstract

The local hierarchical Dirichlet process is a hierarchical Bayesian method that has recently been used for 3D object category recognition. This method has been proven to be efficient for recognizing the category of 3D objects using a lower number of learning instances in offline and open-ended scenarios than other state-of-the-art techniques. In this chapter, the local hierarchical Dirichlet process is adapted for the semantic 3D object parts segmentation task. In contrast to most deep neural models for semantic parts segmentation, the proposed technique requires a lower number of learning instances to achieve a high learning accuracy. The proposed model can also be utilized in open-ended scenarios where the number of 3D object parts is not fixed for each category and can increase over time. The model has been integrated with argumentation-based online incremental learning, recently introduced as an explainable machine learning technique. This method has outperformed other state-of-the-art online incremental learning techniques in terms of learning speed and learning accuracy. Using this method together with local hierarchical Dirichlet process for semantic 3D object parts segmentation, an explainable 3D object category recognition technique is proposed that is more robust to occlusion. The resulting technique can produce explainable justifications for the reasoning process.

6.1 Introduction

Object segmentation is one of the challenging problems in 3D shape analysis. In this domain, data-driven part-segmentation techniques outperformed traditional geometrical methods [179]. In recent years, deep learning approaches have been widely exploited among researchers in this field [182]. These neural network architectures typically extract data-driven features rather than using hand-crafted features. Although these techniques show promising results in some applications, they are not well-suited for open-ended learning scenarios where the number of object categories and part-segments are not predefined and can be extended over time.

The majority of existing models for 3D shape segmentation has five limitations to be used in open-ended dynamic environments. First, most of these models are trained with a fixed set of labels, which greatly limits their flexibility and adaptivity. For instance, a model trained to segment a table into three semantic parts cannot be used to correctly segment a table with four parts, even if they both belong to the same shape family. Second, using a fixed set of labels limits the number of object categories that the model can segment. For example, a model which previously learned how to segment a cup and a table cannot learn to segment a new object like an airplane unless the model is retrained. Third, the training time of most of the state-of-the-art techniques is high to achieve a good learning accuracy. Table 6.1 shows the average training time of two well-known approaches in the literature compared to our proposed approach. Fourth, most of the object segmentation and object category recognition methods in the literature use a large training set, while learning with a lower number of learning instances is required for the faster adaptation of the model to an open-ended dynamic environment. Fifth, 3D object category recognition techniques are typically not robust to high degrees of occlusion. Facing with occluded objects is common in real-word dynamic environments. These limitations motivated us to design an open-ended model which can learn new object categories and new object parts using an online incremental technique. The proposed method is more robust to occlusion and can learn with a lower number of training instances.

Open-ended 3D object segmentation is required in some applications, such as robotic grasping where the graspable parts should be detected for any manipulable object. Considering the application of robot applications in more complex and dynamic environments, it is evident that pre-programing all possible object categories and object parts segments for a robotic application is impossible. Instead, robots should learn autonomously from novel experiences, supported by a feedback from human teachers [91]. Therefore, the competence

Approach	Run-time
PointNet++ [130]	114912
PartNet [182]	161568
Ours	**253**

Table 6.1: Comparing the training time of two well-known deep learning approaches in the literature with our approach.

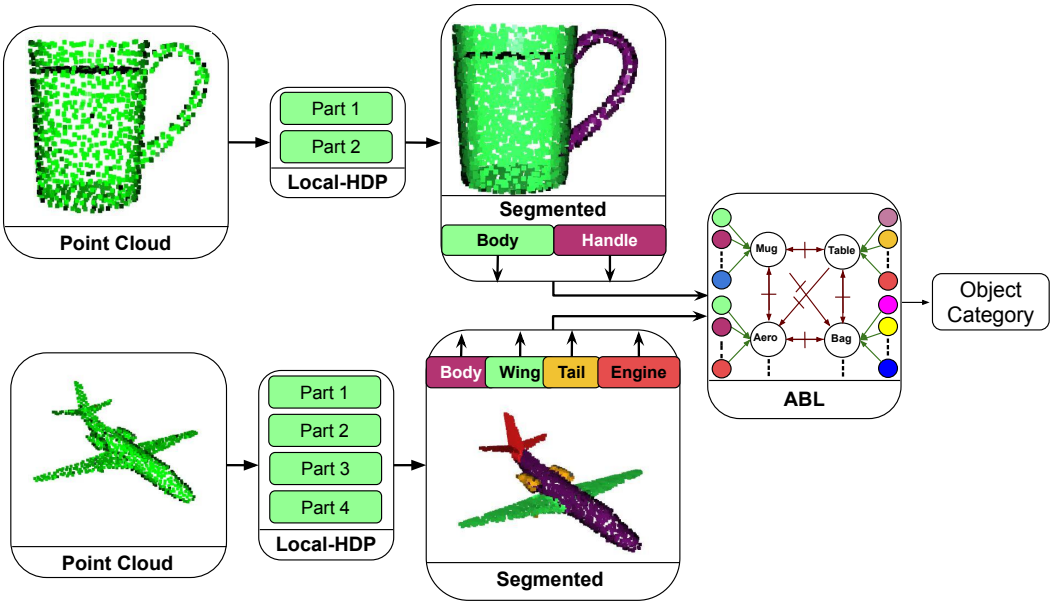


Figure 6.1: The architecture of the proposed method.

of a service robot should increase over time, i.e., robots must robustly adapt to new environments by being capable of segmenting new objects and parts. In order to incrementally adapt to new environments, an autonomous assistive robot must have the ability to process visual information and conduct learning and recognition tasks in a concurrent and open-ended fashion [96].

Using the output of the semantic object parts segmentation technique, the Argumentation-Based Learning (ABL) technique [18] is incorporated to recognize the category of the object. This method can explain the reason for classifying a point cloud to certain object category. Moreover, this partial information can lead to a more robust object category recognition technique that can perform better when there is an occluded object in the scene. Figure 6.1 shows the architecture of the model for recognizing the category of different objects.

Our approach for object parts segmentation and object category recognition extends the previous Local-HDP method for object category recognition

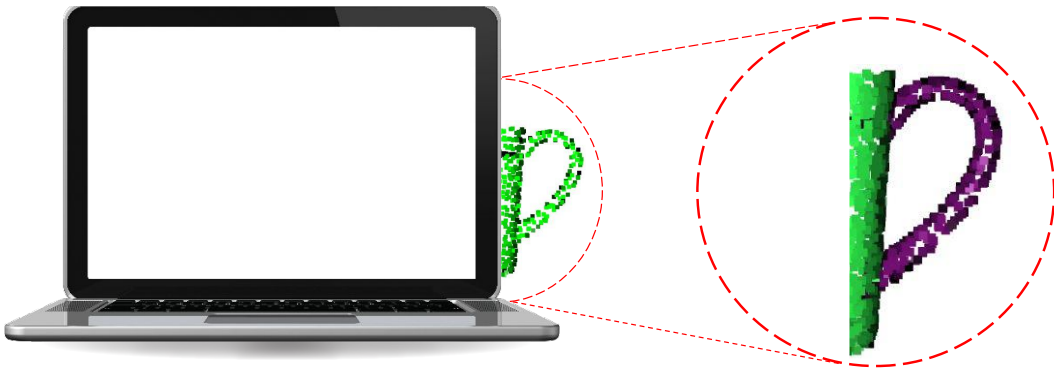


Figure 6.2: An example of an occluded 3D object in a scene.

[20] and argumentation-based learning [18]. Firstly, it uses local-to-global and global-to-local object descriptors to represent each point on the point cloud of an object. Therefore, there is no need for dictionary construction for Local-HDP. Here each bin in the histogram of the descriptors represents a word in the dictionary. This is an important feature for open-ended applications since there is no need for constructing a pool of 3D objects and performing clustering to construct a dictionary in the pre-processing phase like in Local-LDA [91] and Local-HDP for object category recognition [20]. Secondly, the integration of the argumentation-based learning and Local-HDP enables the model to handle high dimensional data and provide explanations for the reasoning process. In contrast, argumentation-based learning is limited to low-dimensional datasets [17]. Thirdly, using the argumentation-based learning technique for the object category recognition module makes the model robust to high degrees of occlusion while most state-of-the-art approaches do not perform well in these scenarios.

On the architecture level of the Local-HDP method, first the 3D point cloud of an object is down-sampled using the voxelized grid approach. For each point in the point cloud, a combination of local-to-global and global-to-local shape descriptors is extracted and they form a histogram. This histogram is then used as a Bag of Words (BoWs) representation of each point that can be fed directly to the topic layer of Local-HDP. For the part-segmentation layer, a semantic object part's label is determined by selecting a local model with the highest likelihood of generating that point.

As mentioned before, an explainable machine learning technique like ABL can lead to a 3D object category recognition method that is more robust to occlusion. As an example, the ABL model can explain that a 3D object is classified as a mug category since it has a handle like a mug and a body that looks like a mug. This explanation can help the model to detect the category of a mug object when it is occluded in the scene by other objects. Figure 6.2 shows a possible scene with an occluded mug. Even if the model has been

previously trained on the mug objects with both body and handle parts, using ABL, the model can infer that this object is a mug although it only has a handle looking like a mug. Moreover, an expert can inject knowledge into the model by interacting with the model and saying 'Although all the previous mugs that the model has seen previously consist of a body and a handle, there exist some mugs that does no have a handle'. Therefore, the model can correctly predict the category of 3D objects that has never been seen before and only have a body like mug.

The rest of this chapter is organized as follows. Section 2 discusses related work. The required background in topic modeling and hierarchical Dirichlet process is explained in Section 3. Section 4 discusses background on argumentation-based online incremental learning. The proposed methodology for open-ended 3D object category recognition is clarified in Section 4. Section 5 shows the results related to the two sets of offline and open-ended experiments. The conclusion and future possible work is discussed in Section 6.

6.2 Related Work

3D object segmentation is a challenging task [161, 142]. The methods in this field can be categorized based on their different characteristics, e.g, goal (surface type vs. part-type), geometric criteria (region-based vs. boundary-based), degree of learning (supervised vs. unsupervised), user involvement (automatic vs. interactive), number of objects used as information sources (single vs. multiple), type of features (geometric vs. structural), and finally the granularity of the produced result (hierarchical vs. non-hierarchical) [161]. In this research, we focus on non-hierarchical, part-type, supervised, interactive 3D object segmentation which can be further used for other applications like grasp points detection. For instance, a handle can be segmented as a graspable part in a mug. Our method is data-driven and can learn to segment new objects in an open-ended manner.

In this work, we have focused on supervised part-based segmentation using point clouds. State-of-the-art approaches for this purpose include, PointNet [129], PointNet++ [130], PointCNN [107], O-CNN [170], SSCN [71], PCNN [14], SPLATNet [153] and PartNet [182]. PartNet is a deep neural network shown to outperform all other approaches[182]. However, it is a hierarchical segmentation technique while our approach is non-hierarchical. Moreover, unlike all these methods, our method is not based on deep neural networks and can be used in open-ended domains where the number of objects and part-segments are not predetermined. This is not the case for deep neural networks with

a fixed number of class labels that require retraining the model when a new object category is added to the dataset. Furthermore, unlike deep neural architectures, our approach does not need a long separate training process before testing the model. This means that the model is trained incrementally and can be tested in real-time applications. The large number of parameters in deep neural architectures require a large number of training examples for optimization. However, experiments have shown that our model is capable of obtaining the same level of performance by only observing a small fraction of the training set used in other deep learning based approaches.

The main building block and basic component of any 3D shape analysis task is a mechanism for finding the similarity between different 3D models. The core component of this mechanism is the object representation method. The performance of object representation techniques depends heavily on their underlying object descriptor which can be categorized into two groups, namely global object descriptors and local object descriptors [104].

A good object descriptor should be robust to different nuisances, such as noise, occlusion, clutter, and deformation. Moreover, they should lead to 3D object recognition using different view-points of objects from different perspectives. Global 3D shape descriptors encode the entire 3D object, while local descriptors encode the small neighborhood around a set of detected keypoints. We use ensemble of both techniques for describing the object parts.

Examples of local 3D shape descriptors include Spin Images (SI) [85] (which has been used in this chapter), 3D shape context [68], Intrinsic Shape Signature (ISS) [187], Signature of Histograms of Orientations (SHOT) [162], Fast Point Feature Histogram (FPFH) [144], Hierarchical Kernel Descriptors [32]. Local descriptors are more robust to occlusions and clutter and are more suitable for 3D object recognition using different view points of objects. However, comparing local descriptors with each other for recognizing the 3D object is a computationally expensive task [7]. To alleviate this problem, other machine learning techniques like Bag of Words (BoWs) approach [93], Latent Dirichlet Allocation (LDA) [30, 96] and deep learning [108, 178] methods can be used for representing objects in a compact uniform format. Towards this goal, Local Latent Dirichlet Allocation (Local-LDA) [91] extends the LDA approach for open-ended 3D object category learning. Similar to our approach, Local-LDA shares topics among the objects in the same categories. However, the number of topics should be chosen based on trial and error in local-LDA while it is autonomously determined in our proposed method.

In the recent years, good progress has been observed in semantic 3D object segmentation methods using the deep neural architectures [129, 130, 107, 182, 78]. This techniques are typically trained with large dataset and the testing set contains the same class labels as the training set. However, our proposed

approach can extend the number of class labels in run-time and it does not need a large training dataset. Moreover, the state-of-the-art approaches are typically tested with complete 3D objects with no occlusion. However, in real-word scenarios it is typical to have occluded objects in the scene and the method should also work well with the occluded objects in the testing set. Our proposed method can perform well in the presence of high degrees of occlusion in the testing set.

6.3 Background

In this section, we will discuss the Hierarchical Dirichlet Process (HDP) [160] and the Local Hierarchical Dirichlet Process (Local-HDP) for 3D object category recognition [20] in more detail. Furthermore, we briefly explain Argumentation-Based Learning (ABL) method for online incremental learning with discrete feature values [18, 17, 16].

6.3.1 Hierarchical Dirichlet Process

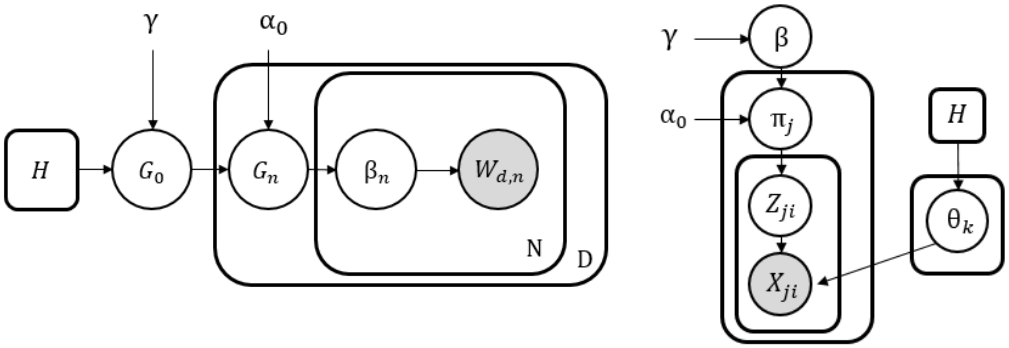
Hierarchical Dirichlet Process (HDP) [160] is a non-parametric hierarchical Bayesian model which can automatically detect the required number of topics during the inference. Let (θ, β) be a measurable space, with G_0 a probability measure on it. Let α_0 be a positive real number. A *Dirichlet Process* DP(α_0, G_0) is defined to be the distribution of a random probability measure G over (θ, β) such that, for any finite measurable partition $(A_1, A_2, ..., A_r)$ of θ, the random vector $(G(A_1), ..., G(A_r))$ is distributed as a finite dimensional Dirichlet distribution with parameters $(\alpha_0 G_0(A_1), ..., \alpha_0 G_0(A_r))$:

$$(G(A_1), ..., G(A_r)) \sim Dir(\alpha_0 G_0(A_1), ..., \alpha_0 G_0(A_r)) \tag{6.1}$$

G_0 is itself a draw from a Dirichlet process DP(H,γ). The steak-breaking construction and Chinese restaurant franchise metaphor has also been formulated for hierarchical Dirichlet process [160]. The probabilistic graphical model of HDP is shown in Fig. 6.3a. The stick-breaking construction of HDP is illustrated in Fig. 6.3b.

6.3.2 Local Hierarchical Dirichlet Process for 3D Object Category Recognition

Local-HDP is a non-parametric hierarchical Bayesian method that has been recently introduced for 3D objects category recognition in offline and open-ended scenarios. This method uses the non-parametric nature of the HDP



(a) HDP (b) Stick-breaking

Figure 6.3: The plate notations of the hierarchical Dirichlet process mixture model and its stick-breaking construction.

together with the fast inference approximation of the posterior from the online variational inference technique [169]. This method constructs a local HDP model for each of the object categories. The approximation inference technique infers the topics (each topic is distribution of the visual words over the dictionary) and the topic proportions (distribution of each BoWs format of the 3D object over the inferred topics) from the training dataset for each of the local models corresponding to each object category. Subsequently, the likelihood of generating the BoWs format of a 3D object using the local spin-image shape descriptors is calculated for each of the local models and the one with the highest likelihood is selected as the predicted category. For the open-ended scenarios the system interacts with a human teacher to learn new object categories when needed and extend the number of learned categories over time. The locality of the models gives them this flexibility. Figure 6.4 shows the plate-notations of the Local-HDP method for 3D object category recognition. Here, $|C|$ is the number of 3D objects in each local model corresponding to category C. The evaluation results show that Local-HDP learns with fewer number of learning instances compared to state-of-the-art methods for 3D object category recognition. Moreover, the experimental results shows that it has higher learning accuracy and lower memory consumption than the compared methods.

6.3.3 Argumentation-Based Learning

Argumentation-Based Learning is an online incremental learning method that is composed of different argumentation frameworks from the argumentation theory. Argumentation theory is a reasoning model that models the interaction between multiple arguments [164]. Dung defines an Abstract argumentation Framework (AF) as a pair of arguments and a set of binary relations showing

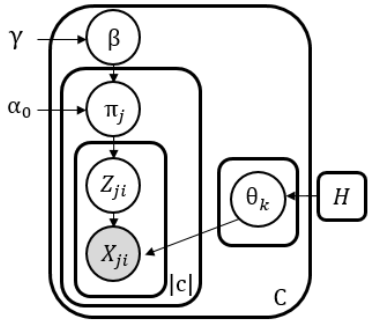
the interaction between them. This relations shows the attack relations between two arguments. ABL assumes that the inner structure of the arguments A and B is $pre \rightarrow post$. Therefore, A and B bidirectionally attack each other if and only if both A and B have the same *pre*s but have different *post*s. ABL also uses Bipolar Argumentation Framework [9] that has support relations in addition to the attack relations between the pairs of arguments. ABL assumes that A supports B if and only if $A.post$ equals $B.pre$. ABL also defines an argument that is neither attacked nor supported by any other arguments as the facts with *post* inner structure.

Argumentation-based learning uses BAF unit and AF unit for generating hypotheses from the learning instances and modeling the interaction between the generated hypotheses to predict the class lables for the testing instances, respectively. ABL outperformed state-of-the-art online incremental learning techniques, neural architectures, reinforcement learning and contextual bandit algorithms in the experimental results using different case studies. This method is capable of learning with lower number of learning instances while achieving higher learning accuracy. Moreover, ABL is limited to low dimensional datasets due to its high computation complexity. This problem is address in Accelerated Argumentation Based Learning (AABL) technique [17]. AABL simplifies the ABL's model by using solely the bipolar argumentation framework. This method uses a pruning mechanism together with a new strategy to extract the subsets of feature values for constructing the support relations in the BAF. These two mechanisms reduces the computational and space complexity of the method from exponential to polynomial. Both the run-time and memory consumption of AABL is exponentially lower than ABL. Consequently, AABL can be utilized in higher dimensional datasets. The experimental results shows that AABL also has better learning accuracy than ABL.

Both ABL and AABL are capable of generating explanations for the reasoning process. Therefore, they can be used for explaining the reason for choosing each class label as the prediction of the model. This enables the model to be more understandable for humans. Moreover, an expert can debug the model by injecting knowledge to the model in form of a set of arguments that are attacking or supporting each other.

6.4 Method

The main goal of the proposed 3D object segmentation method is to independently learn topics for each semantic segment of the object using local shape descriptors to represent the local neighborhood around each point. In this



(a) Stick-breaking

Figure 6.4: The plate notations of the local hierarchical Dirichlet process mixture model and its stick-breaking construction. In this graphical model, C is the number of learned categories and $|c|$ is the number of objects in each category.

way the input to the model will be a local shape descriptor of a keypoint in an object. The model can find a proper label for each keypoint comparing the likelihood of the topic distribution of the current local shape descriptor with each local model. In the following subsections, we first describe the preprocessing layers and then explain our local-HDP model in detail.

6.4.1 Processing Layers

The first two layers used in Local-HDP are the feature layer and Bag-of-Words (BoWs) layer. Unlike Local-HDP for object category recognition [20], the BoWs layer is not used in the same way. The input to the model is the histogram of a mixture of local-to-global and global-to-local shape descriptors for each individual point in the 3D point cloud. Therefore, the histogram of a descriptor for each point is considered as a BoWs representation. This means that there is no need for dictionary construction in this research and each bin in the histogram of a descriptor is considered as a word in a dictionary. In the feature layer, a new composition of local-to-global and global-to-local information is computed for all the points belonging to the object. Spin image descriptor [85][1] is view-independent and pose and illumination invariant; thus, it is an appropriate choice for robotic applications. In this chapter, spin images have been used for local-to-global representation of the points in the 3D objects. This means that the supporting radius of the spin-images are considered large enough to cover all the points in the object's point cloud. Figure 6.6 shows the local-to-global spin-image of a keypoint and its representation in 2D

[1] http://docs.pointclouds.org/trunk/classpcl_1_1_spin_image_estimation.html



(a) Coffee mug (b) Voxel Grid

Figure 6.5: a) The RGB-D image of a coffee mug from the Washington RGB-D dataset. b) Using voxel grid method for keypoint detection and downsampling.



(a) A Keypoint's Normal (b) Local-to-Global Spin-image (c) 2D Bins

Figure 6.6: a) The surface normal of a keypoint shown in red. b) The corresponding local-to-global spin-image of the keypoint. c) The representation of the spin-image in 2D bins.

bins. Subsequently, as shown in Fig. 6.5b, the voxel grid method[2] is used to simultaneously down-sample the point cloud and find the keypoints. The centers of the voxels that have an intersection with the object's point cloud are then used as keypoint. The computed local-to-global descriptor for the closest point to each keypoint is used as the descriptor of the global neighborhood around that keypoint (Fig. 6.6).They are incorporated in combination with a global-to-local descriptor, which is described in the the next subsection, as inputs to the Local-HDP model. The obtained representation is then inferred to a set of topics in the topic layer.

6.4.2 Oriented Local to Global 3D Object Descriptor

Our experiments showed that using solely a local-to-global descriptor for each point in the object's point cloud is not enough for the segmentation task. This representation lacks the global-to-local information required for the object parts semantic segmentation. As stated previously, each point in our model is

[2] \protect\protect\unhbox\voidb@x\hbox{http://docs.pointclouds.org/trunk/classpcl_1_1_voxel_grid.html}

described using a local-to-global shape descriptor. This representation has the key role in finding the segmentation label of that point. State-of-the-art local shape descriptors define local reference frames to construct the local descriptor. For instance, approaches like spin images use a cylindrical reference frame, which is rotation invariant. Other approaches like SHOT [162] use a local reference frame which is dependent on the structure of the surface-normals of the neighboring nodes of the keypoint. The spin-image shape descriptor loses some information due to the cylindrical structure of the reference frame in order to be robust to noise and be invariant to different transformations. Moreover, the SHOT descriptor is not robust to sparse point cloud structures.

For 3D segmentation of the object, we can use a global-to-local shape descriptor. The first descriptor shows the location of each point with respect to all other points in the point cloud. Using this descriptor one can estimate the location of each point in the point cloud. This descriptor pin-points a specific point on a global representation of the object with a repeatable global reference frame.

The global-to-local descriptor can pin-point any point p in the point cloud using a global reference frame. In order to find the global reference frame, we use the same methodology as GOOD 3D object descriptor [95]. We use Principle Components Analysis (PCA) to find the most three dominant eigenvectors that show the principal directions with regard to the distribution of the points in the point cloud. The directions of the vectors need to be disambiguated using the same approach as GOOD. Like GOOD, for each projected point $\hat{p} = (\alpha, \beta)$ from the point cloud P, where α is along the x axis and β is along the y axis, first a row $r(\hat{p})$ and a column $c(\hat{p})$ are defined as follows:

$$r(\hat{p}) = \left\lfloor \frac{\alpha + \frac{l}{2}}{\frac{l+\epsilon}{n}} \right\rfloor \tag{6.2}$$

$$c(\hat{p}) = \left\lfloor \frac{\beta + \frac{l}{2}}{\frac{l+\epsilon}{n}} \right\rfloor \tag{6.3}$$

Where l is the support length, n is the number of bins and ϵ is a small value used for robustness in the GOOD descriptor. The left panel of Figure 6.7 shows the three projected planes for a mug. Unlike the GOOD global descriptor, using the orthogonal projections of the points in the point cloud, we use a different technique to pinpoint the point p^* in the projection bins. For each bin with row i and column j, the value of each bin is calculated as follows:

$$bin_{r,c}^{i,j} = \sum_{p \in (r(i), c(j))} \left(\frac{(1-d)}{l}\right)^2 \tag{6.4}$$

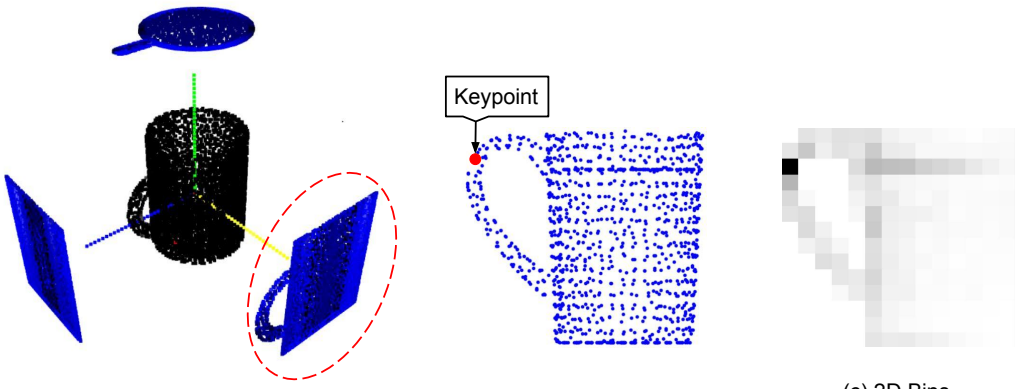


Figure 6.7: a) The global projections of a mug object. b) A keypoint on a selected projection is shown with a red dot. c) The projected point-cloud is then down sampled to a set of bins and the keypoint is pinpointed on it.

Here, $d = ||p - p^*||$ is the Euclidean distance of point p from the keypoint p^* that we want to pinpoint. This process will be repeated for all the points in the point cloud and the three projected planes. At this step, we can find the location of the point in the point cloud using the aforementioned global-to-local descriptor.

Using the combination of the two aforementioned local-to-global and global-to-local descriptors can lead to a better representation for each point in the point cloud. We convert this descriptor to a histogram of a specific size and each bin is a considered as a word in a document. This histogram is considered as a Bag of Words (BoWs) to be fed to the model. The performance of this descriptor is evaluated in the results section using the 3D object parts semantic segmentation task. Notice that only the local-to-global descriptor is used for segmenting the occluded objects.

6.4.3 3D Object Segmentation using Local-HDP

After synthesizing the point cloud of the 3D objects to a set of BoWs, the data is ready as input to the topic layer where the Local Hierarchical Dirchlet Process (Local-HDP) method is utilized. Unlike Local-LDA [91] and Local-HDP, there is no need for dictionary construction in the proposed method. This is a notable difference since in open-ended applications, dictionary construction is one of the main steps of the topic modeling approaches, which can be a hard task in practice. As stated previously, each point is represented as a set of descriptors with V bins $\{s_0, ..., s_V\}$. For each object, we use Local-HDP to construct an independent local model for all the points located at each semantic part of the 3D object. Therefore, the number of local models are

corresponding to the number of parts in an object.

Using a hierarchical Bayesian model called Local-HDP, we are able to transform the low-level spin-image feature to conceptual high-level topics features . Assuming that each point is a distribution over topics, each topic is itself a distribution over the bins of the local descriptor (s_i). The topics in our method are only locally shared among the points of the same semantic segment of the object and not across all points. This means that the point's visual word is only used to update a local model related to that part of the object. This uses an incremental inference approach where the number of object parts and object categories are open-ended and not known beforehand. The proposed method also uses unsupervised HDP topics inference. However, a human teacher or simulation-based interaction can teach the model new objects and object parts when the teacher introduces a new object that cannot be assigned to one of the previously learned categories. Here, C is the number of open-ended objects and $|c|$ is the number of the points in each cluster. The points consist of a set of N bins $W_{n,d}$ where n,d means the n-th bin from the d-th point (document). Each bin is an element from the vocabulary of bins with predefined V words, that is $W_{d,n} \in \{1...V\}$.

After the model is constructed in a generative manner, the reverse procedure for inferring the latent variables from the data should be used. The next subsection shows the proposed inference mechanism for local-HDP.

6.4.4 Local Online Variational Inference

Posterior inference is intractable for the HDP, and Local-HDP has adapted the online variational inference [169] to approximate the inference method. This method is suitable for open-ended application since it is fast and it infers the topics in an online incremental manner.

A two-level Hierarchical Dirichlet Process (HDP) is made from a set of Dirichlet Processes (DP) sharing a basic distribution G_0. This distribution (G_0) is also drawn from a DP. Mathematically, this can be explained as follows:

$$\begin{aligned} G_0 &\sim DP(\gamma H) \\ G_j &\sim DP(\alpha_0 G_0) \end{aligned} \tag{6.5}$$

G_j represents a DP for each document. All G_js share the same atoms and the only difference between the G_js is related to their corresponding weights. H is a symmetric Dirichlet over the vocabulary simplex known as topics.

In order to generate the data from HDP, we need the following two steps. First the topic association for the jth document (θ_{jn}) is generated. Second the

nth word of the jth document (w_{jn} is generated using the topics.

$$\theta_{jn} \sim G_j, \quad w_{jn} \sim Mult(\theta_{jn}) \tag{6.6}$$

An inference algorithm performs the reverse process by inferring the distributions over topics and the distribution over vocabulary words (θs) using the dataset of documents. For the segmentation task, each document is the BoWs representation of the local-to-global shape descriptor for each point from the point cloud.

Using Sethuraman's stick-breaking construction for HDP [160], the variational distribution for online variational inference is as follows.

$$q(\beta', \pi', c, z, \phi) = q(\beta')q(\pi')q(c)q(z)q(\phi). \tag{6.7}$$

Where $\beta' = (\beta_k^{'})_{k=1}^{\infty}$ is top-level stick proportions, $\pi' = (\phi_{jt}^{'})_{t=1}^{\infty}$ is the bottom-level stick proportion and $c_j = (c_{jt}^{'})_{t=1}^{\infty}$ is the vector of indicators for each G_j. Moreover, $\phi = (\phi_k)_{k=1}^{\infty}$ is the infered topic distributions, z_{jn} is the topic index for each word w_{jn}.

The factorized form of $q(c)$, $q(z)$, $q(\phi)$, $q(\beta')$ and $q(\pi')$ is the same as the online variation inference for HDP [169]. Assuming that we have $|C|$ objects in each category for Local-HDP, the variational lower bound for document j in the semantic object part C is calculated as follows:

$$\begin{aligned} L_j^{(C)} = \mathbb{E}_q[log(p(w_j|c_j, z_j, \phi)p(c_j|\beta')p(z_j|\pi')p(\pi_j^{'}|\alpha_0))] \\ + H(q(c_j)) + H(q(z_j)) + H(q(\phi')) \\ + \frac{1}{|C|}[E_q[logp(\beta')p(\phi)] + H(q(\beta')) + H(q(\phi)] \end{aligned} \tag{6.8}$$

Where H(.) is the entropy term for the variational distribution. Therefore, the lower bound term for each category is calculated in the following way.

$$L^{(C)} = \sum_{j \in C} L_j = \mathbb{E}_j[|C|L_j] \tag{6.9}$$

Using coordinate ascent equations in the same way as online variation inference for HDP, the document-level parameters $(a_j, b_j, \psi_j, \zeta_j)$ are estimated [160]. Then for the category-level parameters $(\lambda^{(C)}, u^{(C)}, v^{(C)})$, we do gradient descending with respect to a learning rate.

$$\partial\lambda_{kw}^{(C)}(j) = -\lambda_{kw} + \eta + |C| \sum_{t=1}^{T} \varphi_{jtk}(\sum_{n} \zeta_{jnt} I[w_{jn} = w], \tag{6.10}$$

Algorithm 6.1: Local Online Variational Inference

initialization:
Randomly initialize $\lambda^{(C)} = (\lambda_k^{(C)})_{k=1}^{K}$, $u^{(C)} = (u_k^{(C)})_{k=1}^{K-1}$ and $v^{(C)} = (v_k^{(C)})_{k=1}^{K-1}$ for all the learned semantic objects parts. Set $t_0 = 1$
for ***each*** *Semantic Object Part C* **do**
  while *Stopping criterion is not met* **do**
    - Use the object view j for updating the parameters.
    - Computer the document-level parameters $a_j, b_j, \Phi_j, \zeta_j$ using the same methodology as [169].
    - Using Eq. 7-9, compute the natural gradients $\partial\lambda^{(C)}(j)$, $\partial u^{(C)}(j)$ and $\partial v^{(C)}(j)$.
    - Set $p_{t_0} = (\tau_0 + t_0)^{-K}$, $t_0 = t_0 + 1$.
    - Update the $\lambda^{(C)}$, $u^{(C)}$, $v^{(C)}$ parameters using Eq. 10-12.
  end
end

$$\partial u_k^{(C)}(j) = -u_k + 1 + |C| \sum_{t=1}^{T} \varphi_{jtk}, \tag{6.11}$$

$$\partial v_k^{(C)}(j) = -v_k + \lambda + |c| \sum_{t=1}^{T} \sum_{l=k+1}^{K} \varphi_{jtl}. \tag{6.12}$$

Where φ (multinomial), ζ (multinomial) and λ (Dirichlet) are the variational parameters which are the same for all the categories. Using an appropriate learning rate p_{t_0} for online inference, the updates for $\lambda^{(C)}, u^{(C)}$ and $v^{(C)}$ become:

$$\boldsymbol{\lambda}^{(C)} \leftarrow \boldsymbol{\lambda}^{(C)} + p_{t_0} \partial \boldsymbol{\lambda}^{(C)}(j) \tag{6.13}$$

$$\boldsymbol{u}^{(C)} \leftarrow \boldsymbol{u}^{(C)} + p_{t_0} \partial \boldsymbol{u}^{(C)}(j) \tag{6.14}$$

$$\boldsymbol{v}^{(C)} \leftarrow \boldsymbol{v}^{(C)} + p_{t_0} \partial \boldsymbol{v}^{(C)}(j) \tag{6.15}$$

Here, p_{t_0} should satisfy the following condition. This condition guarantees convergence [141].

$$\sum_{t_0=1}^{\infty} p_{t_0} = \infty, \quad \sum_{t_0}^{\infty} p_{t_0}^2 < \infty \tag{6.16}$$

In all the experiments, we set $p_{t_0} = (\tau_0 + t_0)^{-K}$ where $K \in (0.5, 1]$ and $\tau_0 > 0$. Algorithm 6.1 shows the pseudo-code of the proposed inference technique for the Local-HDP approach. Notice that a local HDP model is used for each of the semantic objects parts in the dataset.

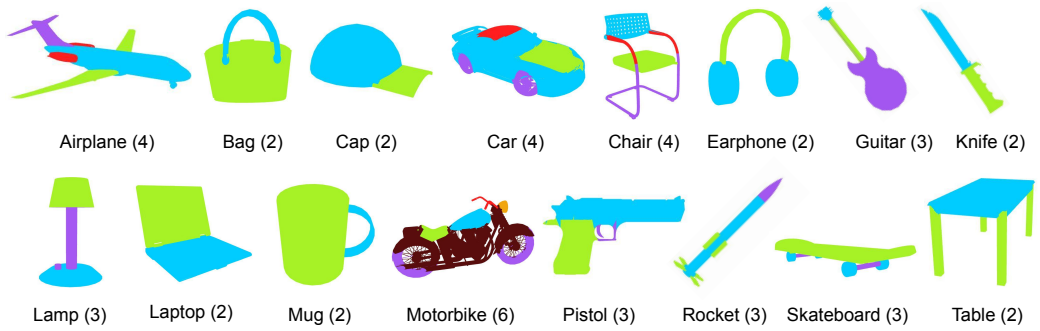


Figure 6.8: Different object categories of the ShapeNet core dataset. The number of object category parts is written in the parentheses.

6.5 Experimental Results

Using a platform with Intel(R) Core(TM) i7, 3.30GHz processor and 16GB RAM with SSD hard disk, we compared our proposed 3D semantic segmentation method using the adapted online variational inference technique with other methods. Specifically for open-ended evaluation, the comparision is done with Local-LDA [91], LDA with shared topics [30], BoWs [93], RACE [122] and HDP with shared topics and the online variational inference technique [169]. For offline evaluation of the method we have compared our method with PointNet [129], PointNet++ [130], PointCNN [107], O-CNN [170], SSCN [71], PCNN [14], SPLATNet [153] and PartNet [182].

Two sets of experiments have been done to evaluate the performance of the proposed method. Firstly, we use the ShapeNet core segmentation dataset [117] for offline evaluation of all the methods. This dataset consists of 573,585 part instances over 26,671 3D models covering 24 object categories. Like all the other approaches, we have used the average Intersection over Union (IoU)(%) metric for evaluation of the part-based segmentation quality. For the second set of experiments, the same dataset has been used. However, this time a simulated teacher introduces an object to an agent after random shuffling of all instances in the dataset.

6.5.1 Offline Evaluation of the 3D Object Parts Segmentation

For offline evaluation of the proposed method, we used 2-fold cross validation using 50% of the data instances for training and 50% for testing. Figure 6.8 shows some of the object categories of the PartNet dataset.

Table 6.2 shows the comparison of offline evaluation of our approach versus other approaches.The proposed Local-HDP method for the objects parts seg-

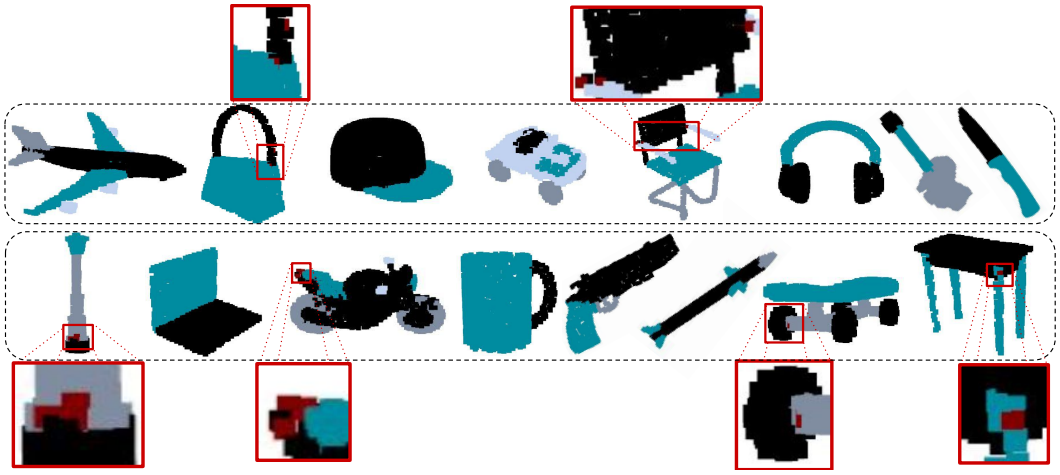


Figure 6.9: A few examples of the resulting 3D object segmentation outputs using the Local-HDP model. The red points shows the wrongly segmented points.

Method	Aero	Bag	Cap	Car	Chair	Eph.	Guitar	Knife	Lamp	Laptop	Motor	Mug	Pistol	Rocket	Skate	Table
PN	83.4	78.7	82.5	74.9	89.6	73.0	91.5	85.9	80.8	95.3	65.2	93.0	81.2	57.9	72.8	80.6
PN++	82.4	79.0	87.7	77.3	90.8	71.8	91.0	85.9	83.7	95.3	71.6	94.1	81.3	58.7	76.4	82.6
O-CNN	85.5	87.1	84.7	77.0	91.1	85.1	91.9	87.4	83.3	95.4	56.9	96.2	81.6	53.5	74.1	84.4
SSCN	84.1	83.0	84.0	80.8	91.4	78.2	91.6	89.1	85.0	95.8	73.7	95.2	84.0	58.5	76.0	82.7
PCNN	82.4	80.1	85.5	79.5	90.8	73.2	91.3	86.0	85.0	95.7	73.2	94.8	83.3	51.0	75.0	81.8
SPLAT	83.2	84.3	89.1	80.3	90.7	75.5	92.1	87.1	83.9	96.3	75.6	95.8	83.8	64.0	75.5	81.8
PtCNN	84.1	86.4	86.0	80.8	90.6	79.7	92.3	88.4	85.3	96.1	77.2	95.3	84.2	64.2	80.0	83.0
PartNet	87.8	86.7	89.7	80.5	91.9	75.7	91.8	85.9	83.6	97.0	74.6	97.3	83.6	64.6	78.4	85.8
Our	**91.9**	**88.2**	**92.4**	**82.8**	**93.4**	**89.5**	**93.7**	**91.5**	**87.6**	**97.9**	**78.0**	**98.2**	**86.3**	**68.8**	**83.7**	**88.6**

Table 6.2: Results of offline evaluation of our approach compared with other state-of-the-arts approaches. The average part-wise IoU(%) metric has been used for the evaluation of the 3D segmentation methods. In this table, the words 'PN', 'PN++', 'SPLAT', 'PtCNN' stand for 'PointNet', 'PointNet++', 'SPLATNet' and 'PointCNN', respectively. Moreover, the word 'Aero' is used instead for the 'Airplane' category.

mentation task outperformed PartNet by 4% higher average IoU. Figure 6.9 shows the segmented 3D objects with false segmented points colored in red. This representation shows that most of the misclassified points are located near a region where two or more parts are connecting to each other.

6.5.2 Open-Ended Evaluation of the 3D Object Parts Segmentation

For online evaluation of Local-HDP for 3D object segmentation, we have two set of experiments. For the first set of experiments, we follow the same method-

ology as Local-LDA [91]. We use a simulated teacher which can either teach a new category segments to the model or ask the model to do the semantic part segmentation for an unforeseen object view. In case of wrong segmentation of an object by the model, the correcting feedback is sent to the model by the simulated teacher. In order to teach a new category's parts segments, the simulated teacher presents three randomly selected object views of the corresponding category to the model. After this step, a set of randomly selected unforeseen object views of all the previously learned categories is presented to the model for category prediction. Subsequently, the average part-wise IoU (%) for the semantic segmentation is computed. If the corresponding IoU is higher than a certain threshold $T = 0.75$ (which means that the number of true-positives is at least triple the number of wrong category predictions), a new category will be taught by the simulated teacher to the model. If the learning accuracy does not exceed the threshold T after a certain number of iterations (100 for our experiments), the teacher supposes that the agent cannot learn further and stops the experiment. In order to calculate the part-wise IoU of the model at each point, a sliding window of size $3n$, where n is the number of learned categories, has been used. More details on the online evaluation protocol which has been used in our experiments can be found in [90].

For the second set of evaluations, Local-HDP is compared to state-of-the-art deep architectures for 3D object segmentation. For each object category, we have plotted the mIoU(%) metric for segmentation accuracy using the different number of training examples. This shows how well the models generalize with respect to the number of observed training instances. A model with higher accuracy and and lower number of observations would then be more preferable for open-ended applications where the agent should quickly adapt to the changes in the environment and the previously unseen observations.

For the first set of experiments, 10 independent experiments have been carried out. This way the random initialization of the models cannot abruptly change the results. Moreover, the order of presenting the objects to both models is kept the same using the simulated teacher. Several performance measures have been used to evaluate the open-ended learning capabilities of the methods, namely: (i) the number of Learned object Parts (#LP); (ii) the number of Correction Iterations (#CI) by the simulated user; (iii) the Average number of stored Instances per object Part (AIP) ; (iv) Mean part-wise IoU (mIoU), the average part-wise IoU of each open-ended experiment. The left part of Table 6.3 shows the result of the experiments for Local-LDA using the aforementioned performance measures. The right part of Table 6.3 shows the performance of our proposed method (Local-HDP).

According to Table 6.3, the average number of learned object parts for Local-HDP is all the 47 object parts from the 16 categories, while Local-LDA

Exp#	#CI	#LP	AIP	mIoU (%)
1	**891**	38	9.19	0.78
2	654	37	8.40	0.76
3	514	39	**7.87**	0.77
4	873	38	8.14	**0.79**
5	579	37	9.04	0.76
6	605	**40**	7.91	0.77
7	849	36	9.20	0.75
8	550	34	8.32	0.78
9	560	39	7.99	0.76
10	564	37	8.92	0.77
Avg	564	37.50	8.49	0.76
$\pm std$	$\pm$147	$\pm$1.71	$\pm$0.53	$\pm$0.01

(a) Local-LDA

Exp#	#CI	#LP	AIP	mIoU (%)
1	1179	**47**	3.31	0.89
2	1224	**47**	3.66	0.90
3	1241	**47**	3.54	0.89
4	1217	**47**	**3.13**	0.91
5	1123	**47**	3.23	0.87
6	1235	**47**	3.81	0.88
7	1252	**47**	3.29	**0.92**
8	1115	**47**	3.72	0.89
9	**1291**	**47**	3.73	0.90
10	1232	**47**	3.01	0.91
Avg	1210	47	3.44	0.89
$\pm std$	$\pm$55	$\pm$0	$\pm$0.28	$\pm$0.01

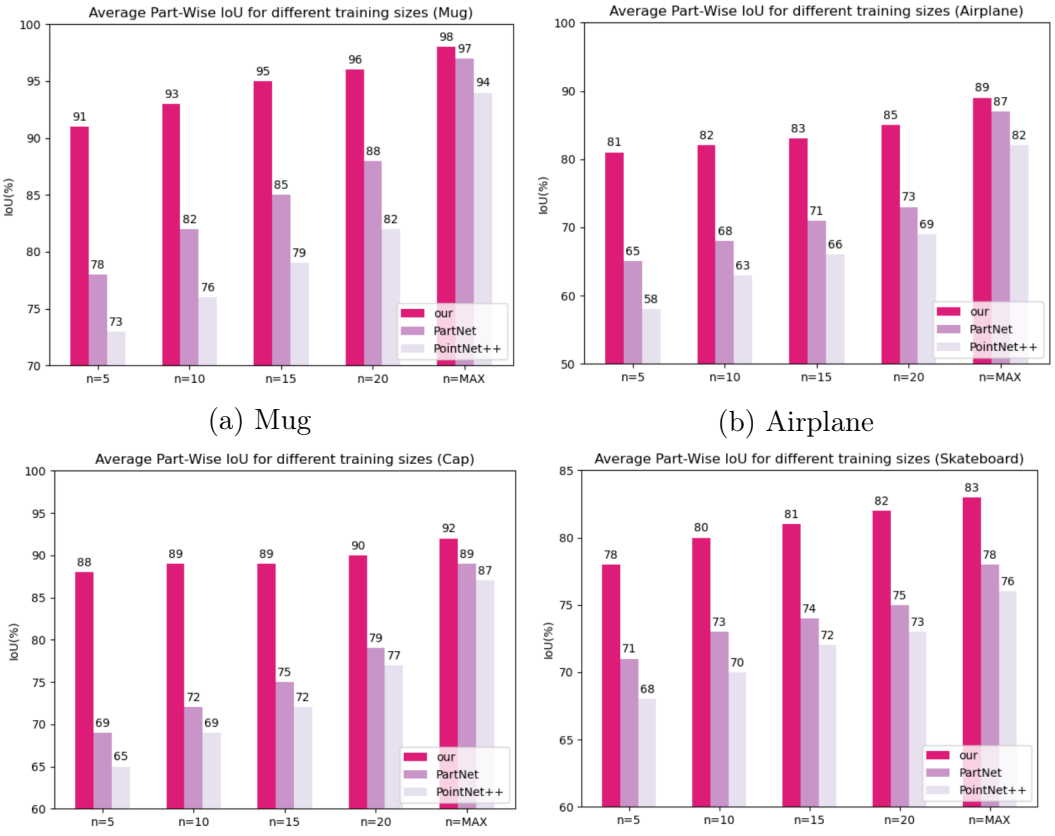
(b) Local-HDP (our)

Table 6.3: Summary of 10 experiments for the open-ended evaluation.

learned 37.50 ± 1.71 object parts on average. This means that Local-HDP can learn better than Local-LDA using the same dataset and learning protocol. Since online variational inference has been used for the inference in Local-HDP method, AIP is much lower for Local-HDP with the average number of 3.44 ± 0.28 instances per object part, which validates memory efficiency. On the other hand, Local-LDA needs to save on average 8.49 ± 0.53 instances per object part. In terms of global segmentation mIoU, Local-HDP performs much better than Local-LDA.

6.5.3 Object Category Recognition with Argumentation-Based Learning

At this step, the 3D objects are segmented to different parts and ready to be used as an input to an explainable machine learning technique to recognize its category. Argumentation-Based online incremental Learning (ABL) is used in this research for this purpose. A number of parts labels $(s_1, s_2, ..., s_n)$ are segmented for a 3D object view O. These segmented parts labels are used as



(a) Mug (b) Airplane

(c) Cap (d) Skateboard

Figure 6.10: The comparison of our proposed Local-HDP method for 3D object segmentation with PartNet and PointNet++ deep neural network architectures using four different training sizes (n). Max means that the maximum size of the training set is used for each object category.

the features for training the model.

ABL has been used for 3D object category recognition using a scenario for simulating occluded objects from the Shapenet core dataset. In this scenario, an object is occluded and a part of its point cloud is randomly removed (Figure 6.11). The parts segmentation model is trained on the complete point-clouds with no missing parts and the occluded objects are only used in the test phase. This scenario can be used to evaluate the robustness of the proposed approach to occlusion. Providing the segmentation information to ABL, the model predicts the category of the object and produce an explanation for its prediction.

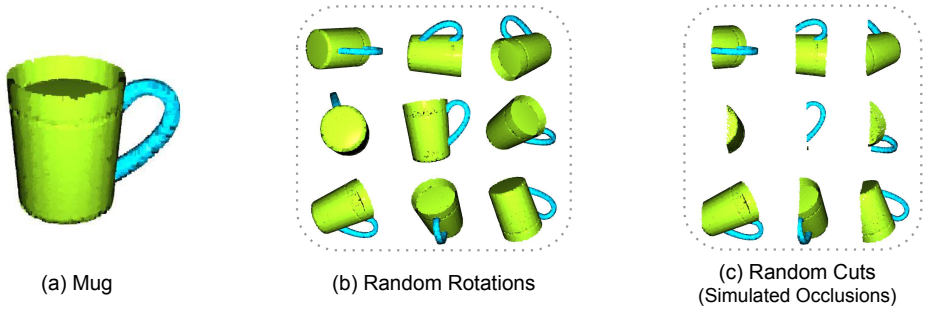


Figure 6.11: The process of simulating the occluded objects. a) A randomly selected mug. b) Random rotations of the selected mug. c) Random cuts from the random rotations.

6.5.3.1 Occlusion

In order to evaluate the robustness of the model to occlusion, the same dataset is used to simulate an occluded set of objects. Using the Shapenet core dataset [117], the objects are randomly rotated in 3D and a piece of its point-cloud is randomly removed using a the following procedure. A 2D plane is used to randomly remove a piece of the point cloud of the object. This plane is parallel to the yz plane and it passes from a random point on the x-axis between the minimum (min) and the maximum (max) of x coordinate of all the points in the point-cloud. To avoid too small or too large removing parts from the point-cloud of the rotated 3D object, the random cutting point on the x-axis is chosen in range of $[min + \frac{max-min}{4}, max - \frac{max-min}{4}]$. Then, the part of the object on the left side of the plane is omitted. Figure 6.11 shows this process for nine random rotations and random cuts. This way a simulated occluded object is constructed using the same dataset as the segmentation task. Although in general occlusion can result to more complicated shapes, the aforementioned process is chosen for simplicity.

To compare the performance of different approaches on the original Shapenet core and the occluded dataset, two set of experiments are conducted. The proposed approach is compared with the Local-HDP method for the object category recognition task [20], PointNet (PN) [129], and PointNet++ (PN++) [130]. The model is trained with the original Shapenet core dataset for in both sets of experiments. The first set of experiments also uses the original dataset for the testing phase. However, the second set of experiments uses the occluded dataset to evaluate the robustness of the model to the occlusion. This means that in the second set of experiments, the model is trained using 90% of the original Shapenet core dataset and the rest of the 10% objects are occluded, using the aforementioned procedure, to construct the testing set. Notice that the 10-fold cross validation technique is used for both sets of experiments and

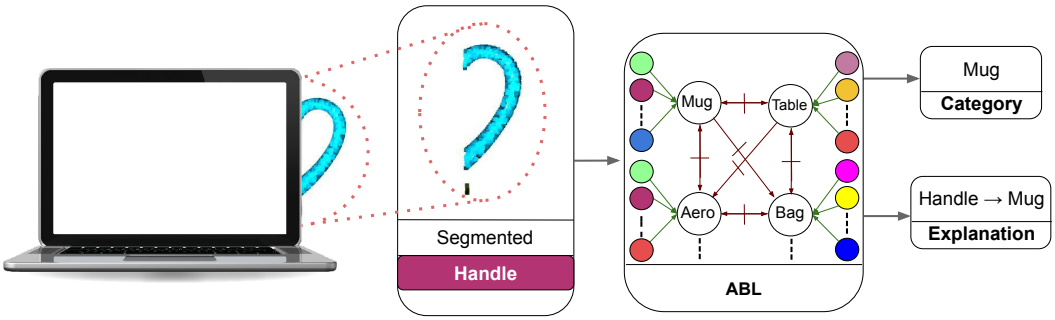


Figure 6.12: An example of the generated explanations from the argumentation-based learning model.

Ex#	Original Dataset				Occluded Dataset			
	Our	Local-HDP	PN++	PrN	Our	Local-HDP	PN++	PrN
1	99	97	97	97	83	45	21	25
2	97	96	96	96	85	41	19	22
3	98	97	96	97	82	43	23	22
4	99	98	97	98	87	38	21	24
5	98	98	97	97	84	42	19	23
6	99	98	97	98	83	47	22	24
7	98	97	97	97	88	40	21	25
8	99	98	96	98	85	46	24	23
9	99	98	97	97	84	44	17	25
10	98	97	97	97	84	46	19	24
Avg	98	97	96	97	84	43	20	23
$\pm std$	$\pm$0.69	$\pm$0.69	$\pm$0.48	$\pm$0.63	$\pm$1.84	$\pm$2.93	$\pm$2.11	$\pm$1.15

Table 6.4: The comparison of the recognition accuracy (%) the trained model on the original dataset for the object category recognition task with different testing sets from the original dataset and the occluded dataset.

the resulting accuracies are reported for each fold.

Table 6.4 compares the recognition accuracy (%) of the trained model for different testing sets namely, the original dataset and the occluded dataset. This table shows that the proposed outperforms all the other methods for both the original and the occluded dataset. The results of recognition accuracies on the occluded dataset shows that the proposed method is more robust than other methods to occlusion with the average 84% accuracy. The other approaches namely, Local-HDP, PointNet++ and PointNet achieved achieved 43%, 20% and 23% learning accuracies, respectively. Therefore, the proposed approach achieved on average 41% higher accuracy than the second best per-

forming approach i.e. Local-HDP. Figure 6.12 shows an example of an explanation generated by ABL for the object category classification task. This means that the proposed method can also generate an explanation for choosing a certain category as the prediction of the model.

6.6 Conclusion

In this research, we proposed a non-parametric hierarchical Bayesian model called Local Hierarchical Dirichlet Process (Local-HDP) for 3D object part segmentation. The proposed technique is integrated with argumentation-based online incremental learning technique for the 3D object category recognition task. All points in the point cloud of the 3D objects are represented by a local-to-global and a global-to-local descriptor in a bag of words format. The proposed approach transfers these bags of words to the hierarchical distribution over a set of topics. Each topic is a distribution over the words. State-of-the-art techniques for the 3D object parts segmentation task, already have a high learning accuracy. These, techniques typically use deep neural networks that take a long time for training. Therefore, they are not suitable for open-ended or class-incremental scenarios where the number of class labels (object parts and object categories) may increase over time. These scenarios are very likely when a general purpose service robot performs in a home like dynamic environment. This means that the proposed model should be able to get updated incrementally in run-time without a need for retraining when a new object category or an object part is observed in the run-time. Moreover, most of the current 3D object parts category recognition techniques are not robust to occlusion. This means that when the models are trained on the complete set of 3D objects with no missing parts, they can poorly segment or recognize an occluded object from a previously learned category.

The extensive set of experimental results showed that the proposed Local-HDP approach outperforms the other state-of-the-art object parts segmentation and 3D object category recognition methods, in term of both accuracy and run-time. Mean part-wise Intersection of Union (mIoU) is uses as the evaluation metric for the objects parts segmentation task. The offline evaluation of the model shows higher mIoU for the parts segmentation task compared to state-of-the-art deep neural network based approaches like PartNet, PointCNN and PointNet++. The open-ended evaluations showed that the proposed model has higher generalization capability by observing much lower number of training examples. This is very useful for open-ended domains where the agent should learn to segment new object categories and new parts with only a few training instances to be able adapt to the surrounding en-

vironment in a short time period. The experimental results for evaluating robustness of the 3D object category recognition technique to occlusion shows that the proposed technique outperforms other techniques with a large margin. Moreover, the argumentation-based learning technique can provide the reason for choosing a certain category for a 3D object. The applications of the provided explanations of the model can be investigation for future work since it seems suitable for human-robot interaction. Using the explanations of the model, a human user can understand the robots decisions, debug the model, and interact with it.

CHAPTER 7

Discussion and Conclusion

Considering a general purpose service robot that needs to operate in a home-like dynamic environment, a robot programmer cannot predict all the possible failure conditions that the robot might encounter in its lifetime. Therefore, the model should have a mechanism to adapt to the changes in the environment. Open-ended (lifelong or class-incremental) learning and online incremental learning are required for this purpose. The model should adapt to changes over time without the need for retraining the model. Moreover, the number of classes that a model can learn should not be fixed and predefined.

In this thesis, we have addressed the problem of online incremental learning in Chapter 2 and Chapter 3. Open-ended learning is addressed in Chapter 4 and Chapter 6. Chapter 2 proposes Argumentation-Based Learning (ABL) for an agent to learn incrementally based on its interaction with the surrounding environment to handle unforeseen failure states. Accelerated Argumentation-Based Learning (AABL) is proposed in Chapter 3 to address the restrictions of ABL. AABL has a simpler architecture than ABL. Chapter 5 addresses a probabilistic inference technique to infer the posterior probability of a Markov Random Fields (MRF) model. To solve real-world robotic vision problems like 3D object category recognition and semantic 3D object parts segmentation, the Local Hierarchical Dirichlet Process is proposed in Chapter 4 and Chapter 6. Local-HDP has some restrictions for 3D object category recognition when the 3D objects are highly occluded in the testing set. Finally, argumentation-based learning has been integrated with Local-HDP to handle this problem in Chapter 6.

In the following section, we address the contributions of all the approaches in this thesis and compare them with each other in terms of underlying structures, learning approaches, computational complexity, learning speeds, interoperability, and accuracy. Also there the main strengths and restrictions of the approaches are summarized. Section 7.2 explains possible future work. The final section returns to the features of a desirable model as discussed in Section 1.3.

	1	2	3	4	5	6	7	8	9
Method	AF	BAF	Online	Inc	Approach	Complexity	NFH	Speed	Accuracy
ABL	✓	✓	✓	✓	Supervised	Exponential	tens	High	High
AABL	✗	✓	✓	✓	Supervised	Polynomial	hundreds	Higher	Higher

Table 7.1: The comparison between ABL and AABL in their architectures and performances.

7.1 Contributions

In this thesis, different approaches including ABL, AABL, Local-HDP and their combination have been proposed. We now summarize the contributions of this thesis as well as the strengths and restrictions of each approach. Argumentation-based online incremental learning (ABL) shows promising results in terms of learning accuracy and learning speed. Moreover, it produces a set of explanations for the reasoning process behind the classification task. However, the original version of ABL is computationally complex and it is not memory efficient. To address this issue, accelerated argumentation-based learning (AABL) proposes two strategies to simplify the architecture of the model and the learning process.

Table 7.1 shows the comparison of the two methods in terms of their architectures and performances. The first two columns compare the two methods based on the argumentation frameworks used in their architectures. ABL uses both the Abstract argumentation Framework (AF) and Bipolar Argumentation Framework (BAF), while AABL only uses BAF. The online and incremental (Inc) learning capabilities are addressed in columns three and four. Column five shows that both of these can be used in supervised learning applications. The space and computational complexity of the two methods are addressed in column six. Column seven shows the Number of Features that both methods can roughly Handle (NFH). The last two columns compare the two methods in terms of the learning speed and learning accuracy. Both methods have higher learning speed and learning accuracy than state-of-the-art online incremental machine learning techniques. A method with a higher learning speed needs a lower number of learning instances to achieve higher learning accuracy. AABL has higher learning speed and learning accuracy than ABL. Notice that both ABL and AABL are used for supervised learning or classification tasks.

Table 7.2 compares ABL, AABL, and Local-HDP for two different applications, namely, 3D object recognition and 3D object part segmentation, and the integrated model. The integrated model explained in Chapter 6 combines Local-HDP for the 3D object part segmentation task with ABL for the ultimate task of 3D object category recognition. All the proposed methods are online incremental but they have different architectures. Local-HDP uses the

Method	Online Incremental	Open-Ended	NFH	Missing Data	Explainable
ABL	✓	✓	tens	✗	✓
AABL	✓	✓	hundreds	✗	✓
Local-HDP (Object Recognition)	✓	✓	thousands	✗	✗
Local-HDP (Segmentation)	✓	✓	thousands	✓	✗
Integrated (Object Recognition)	✓	✓	thousands	✓	✓

Table 7.2: The comparison of all the proposed machine learning methods in this thesis. The integration of ABL and Local-HDP has all the advantages of both techniques.

Hierarchical Bayesian (HB) approach while ABL and AABL are based on Argumentation Theory (AT). Moreover, Local-HDP is designed for open-ended scenarios, while ABL and AABL are not utilized in open-ended scenarios. Notice that the architectures of ABL and AABL do not make any assumption on the number of classes. Therefore, they can also be used in open-ended applications, although this has not been investigated here. Local-HDP has been used for robotic vision applications with thousands of features while ABL and AABL are utilized in scenarios with a smaller number of features. Local-HDP for 3D object category recognition cannot handle a high degree of missing data in the testing set, while the integrated technique can handle highly occluded objects. The proposed Local-HDP for 3D object parts segmentation can handle missing data. Moreover, ABL and AABL are structured machine learning techniques that are not designed to handle missing data. In terms of explainability, ABL, AABL and the integrated method all produce explanations for the reasoning process. However, Local-HDP does not produce any explanations for the reasoning process apart from the learned topics that are inferred from the dataset. The learned topics are not necessarily visually explainable.

The experimental results show that the integrated technique has the advantages of ABL, AABL and Local-HDP. This means that the integrated model has the following features (as listed in Section 1.3 on features desirable for General Purpose Service Robots):

1. The integrated model can learn in an online incremental manner.

2. The integrated model can learn with a small number of learning instances.

3. The integrated model is suitable for open-ended class-incremental learning.

4. The integrated model learns to recognize the category of 3D objects.

5. The integrated model can semantically segment a 3D object.

6. The integrated model uses an accelerated inference technique for the task of 3D object segmentation, namely, online variational inference.

7. The integrated model can handle a high degree of occlusion for recognizing the category of a 3D object.

8. The integrated model can be the basis of explanations for the predictions of the model.

9. The integrated model can interact with a human teacher to get correcting feedback and learn from the user's interactions. This can be further used for debugging the model.

The aforementioned characteristics of the proposed integrated model make it a suitable choice for robotic applications.

We now summarize the main strengths and restrictions of the methods proposed. As mentioned in Chapter 2, the original version of the argumentation-based learning is restricted to low-dimensional datasets since the computational complexity of this approach is high. This problem is addressed in Chapter 3 as accelerated argumentation-based learning technique. Both ABL and AABL achieve higher learning precision and can learn with a smaller number of learning instances than state-of-the-art online incremental learning techniques. AABL is faster, more memory efficient and it has higher learning precision than ABL. Although, the complexity of AABL is lower than ABL, it is restricted to discrete feature values. This limits the application of the approach in a setting of high-dimensional, continuous feature values. To address this restriction, one can change the architecture of the model or use this model as a high-level learner on top of a low-level feature extraction technique that extracts high-level features rather than low-level features. This has been done in Chapter 6, where the AABL is combined with the Local-HDP approach as a high-level object category recognizer using the segmentation labels produced by Local-HDP.

Local-HDP for object category recognition works well with 3D object point-clouds. It achieves higher learning precision and it can handle open-ended number of object classes. This is especially required in robotic applications. It can learn new 3D object categories with a few learning instances and achieve high learning precision. However, this technique, like most state-of-the-art techniques, is not robust against a high degree of missing parts in the point-cloud or a high degree of occlusion. Chapter 6 resolves this issue by a pipeline of a low to high-level 3D object part segmentation technique that gets a low-level

description of the points and produces high-level parts segmentation labels for an object. ABL is used as an explainable technique that can handle missing data and achieve a better classification accuracy by using the high-level descriptions of object parts in a 3D object. Such a mixture of a low-level and high-level model addresses the previously mentioned restrictions of our approaches for the 3D object recognition task. This approach assumes that the point-cloud of 3D objects is previously known or can be easily detected in the environment. Therefore, it cannot handle clutter scenes where we have a pile of objects and the objects' bounding boxes are not easily separable.

7.2 Future research

In this section, we address possible directions of research focusing on the methods developed in this thesis.

7.2.1 Argumentation-Based Learning

Argumentation-Based online incremental Learning is introduced in Chapter 2 and Chapter 3. ABL is then used in Chapter 6 to learn as a high-level machine learning technique on top of the proposed local hierarchical Dirichlet process model for the task of 3D object part segmentation that produces the low-level segmentation labels. This combination shows promising results when the objects are occluded due to explainability of the argumentation-based learning. The explainable set of rules produced in this process can be further used for debugging the model in future work. For example, when ABL predicts a wrong label for a testing example, a user's feedback can be useful to enhance the two argumentation frameworks in a way that this example can be classified correctly at later stages. Moreover, this technique can be useful for human-robot interactions. For instance, when the AF or BAF unit in ABL cannot provide a first or second guess, the model can interact with a user to learn how to classify an unforeseen instance.

Argumentation-Based Learning has already been compared to contextual bandit approaches that are typically used in recommender systems. Therefore, as a future work, we can use argumentation-based learning as a recommender system and compare it with state-of-the-art techniques.

7.2.2 Local Hierarchical Dirichlet Process

Local Hierarchical Dirichlet Process is a hierarchical Bayesian model that has been introduced in Chapter 4. Local-HDP has been used in 3D object category

recognition in robotics. This model has later been adapted in Chapter 6 for the task of 3D object segmentation. Local-HDP led to promising results in both offline and open-ended (class-incremental) scenarios. This is especially interesting for the robotics domain where the environment surrounding a robot is dynamic and prone to changes.

Local-HDP is a two-level Bayesian model that can autonomously determine the number of topics and find the topic proportions. This is suitable for the tasks like image recognition and segmentation. However, for the hierarchical segmentation of objects, having more than two levels in the model might be more useful. This can be investigated in future work. Moreover, Local-HDP can be utilized in other supervised learning domains.

7.2.3 Inference Algorithms

We have introduced two inference techniques in this thesis, namely, an online variational inference technique in Chapter 4 for the Local-HDP model and swift distance transformed loopy belief propagation for Markov random fields in Chapter 5. Both of these approaches are fast approximations of the posterior probabilities using the two different techniques for two different models. Targeting hierarchical segmentation might need more than two levels for the hierarchical Bayesian approach and the new model requires a new inference technique for future works to address this problem.

7.2.4 Argumentative Explanations for Machine Learning Models

Producing argumentative explanations using the Argumentation-Based Learning approach (Chapter 2, Chapter 3 and Chapter 6) has shown promising results compared to state-of-the-art techniques. Specifically, the argumentative explanations help the 3D object category recognition model to detect the category of occluded objects. The power of argumentative explanations motivates us to look further into the argumentative explanations for different black-box machine learning techniques like deep neural networks. For future work, we aim to produce argumentative explanations for deep neural networks, namely, feed-forward neural networks and convolutional neural networks, in order to explain the intrinsic reasoning process for each prediction using an argumentation framework.

7.2.5 Explanations for Debugging the Model and Human-Robot Interaction

In this thesis, we have introduced explainable machine learning techniques that can produce a set of reasons (explanations) for their predictions (Chapter 3, Chapter 3 and Chapter 6). However, we have not explicitly used these explanations for debugging a trained model. A comprehensive user study can be conducted in future work to incorporate the obtained explanations for the process of debugging the model. Moreover, this can be further extended to add a human user in the loop to fine-tune the model according to his/her desire. Therefore, the model could be altered by interacting with humans as well as learning from the data.

7.3 Conclusion

Aiming to propose a suitable method for explaining what a robot sees, different approaches for explainable online incremental learning in offline and open-ended domains are proposed in this dissertation. These approaches are utilized in different supervised learning scenarios as well as robotics vision. The Argumentation-Based online incremental Learning (ABL) is proposed as an explainable machine learning technique that can learn using a lower number of learning instances than most common machine learning techniques. ABL is composed of two argumentation formalisms namely, an Abstract argumentation Framework (AF) and a Bipolar Argumentation Framework (BAF). The BAF unit is responsible for generating a set of hypotheses/arguments out of the training instances by using the grounded extension semantics. Modeling the defeasibility relations between the generated arguments is the responsibility of the AF unit using the preferred extension semantics.

The resulting ABL model is then compared with state-of-the-art online incremental learning methods, deep reinforcement learning techniques and contextual bandits approaches. The experimental results show that the proposed approach can learn faster and better than other methods. This is specially important when there is a limited number of available training instances. Moreover, online incremental learning methods typically face the phenomenon of concept shift. This means that the underlying rules for classifying data as a specific class can change over time. Therefore, a machine learning technique like ABL that can learn faster and adapts more quickly to the underlying changes is more preferable. ABL also allows the generation of explainable rules for the underlying reasoning process. This enables ABL to be more user-friendly, since a user can easily understand the reason behind each decision and

debug the model if needed. Moreover, a user can inject knowledge into the model in the form of arguments. Therefore, ABL is suitable for human-robot interaction.

ABL also has some restrictions since its computational complexity is high. Consequently, it is not suitable for high-dimensional datasets. This motivated the introduction of a simpler model called Accelerated Argumentation-Based Learning (AABL). AABL simplifies ABL by including only a BAF unit. To reduce the complexity of the model, two strategies are used. First, instead of extracting all the subsets of the feature values for each training example, AABL starts with extracting length-one subsets and increments the length of the subsets if required (considering that fixing the higher bound of length of the subsets to two works well in practice). The condition for increasing the length of the subsets depends on the second strategy, that is the pruning step. In this step, the supporting nodes that are not uniquely supporting a class label are pruned. Using these strategies, the resulting AABL model has polynomial complexity rather than the exponential complexity of ABL by the use of a fixed upper bound on the length of feature value subsets. Therefore, it has much lower run-time and memory consumption and it is more suitable for higher-dimensional datasets. The experimental results also show that the learning accuracy is improved in comparison to the original ABL model. The reason for the resulting improvement in the learning accuracy is that the AABL model can focus only on the relevant supporting nodes after the pruning step, while ABL sometimes utilizes wrong supporting nodes. Consequently, AABL has better performance than ABL since it can learn with lower number of learning instances and achieve higher learning accuracy. Moreover, it is much faster and more memory-efficient than ABL.

Local Hierarchical Dirichlet Process (Local-HDP) is a hierarchical Bayesian approach that can handle very high dimensions of data and lots of training examples. Therefore, it is suitable for computer vision problems with thousands of features. Local-HDP is capable of learning in both offline and open-ended scenarios. In open-ended scenarios, the number of class labels is not fixed and can grow over time. Therefore, Local-HDP is a good choice for general purpose service robots performing in a home-like dynamic environment. It combines the advantages of the Hierarchical Dirichlet Process (HDP) and Local Latent Dirichlet Allocation (Local-LDA). It is a non-parametric model that can autonomously determine the number of required topics in the inference phase. Therefore, it does not have the limitations of LDA-based approaches. Moreover, it uses local models for each class label and can extend the number of local models over time. Local-HDP has been used for the problem of 3D object category recognition and outperformed all the state-of-the-art approaches by a large margin. It is also much faster than Local-LDA since it adapts an

online variational inference technique. Local-HDP can learn new 3D object categories with a much lower number of learning instances and incrementally adapts to the dynamic environment. It has been used in two robotic experiments, namely real-time 3D object object category recognition and object manipulation with a robotic arm. It interacts with a human teacher to learn new object categories and it is suitable for human-robot interaction.

In order to achieve higher robustness against nuisance such as occlusion of 3D object categories, ABL and Local-HDP have been integrated. Local-HDP has been adapted to be used for the 3D semantic object parts segmentation task. This enables the model to segment 3D objects to different semantic parts. This method results in a higher mean Intersection of Union (mIoU %) for different 3D object categories. Most of the 3D semantic part segmentation methods are based on deep neural networks in the literature. These methods typically need a large training dataset that is not available for general purpose service robots operating in a dynamic home-like environment. Local-HDP for semantic 3D object parts segmentation can learn to segment different object categories with a lower number of learning instances. Moreover, it is suitable for open-ended applications where the number of object categories and the semantic parts are not fixed and predefined and can grow over time. This method has been compared with state-of-the-art deep neural architectures and outperformed all in terms of run-time, learning accuracy and mLoU %. Using the output of the proposed Local-HDP method for semantic 3D object parts segmentation as an input to the ABL model, an integrated model has been proposed for 3D object recognition. In particular, the ABL model uses the extracted semantic part labels as learning features and predicts the category of the objects. Using a local-to-global feature descriptor together with the explainability of ABL, the resulting model is robust to a high level of occlusion. Most state-of-the-art approaches could not handle a high amount of occlusion for correct 3D object category recognition. The experimental results show that the resulting integrated model is much more robust to occlusion than state-of-the-art approaches.

To summarize, proposing the ABL technique led to an explainable online incremental learning that has been improved by introducing AABL. To handle higher-dimensional data in the robotic vision domain, Local-HDP has been proposed, which can operate in dynamic open-ended environments. Consequently, Local-HDP has been adapted for the segmentation task and has been integrated with ABL for the 3D object recognition task. The resulting model has all the advantages of ABL, AABL and Local-HDP and can be widely used in different robotic vision tasks.

References

[1] M. Abeille, A. Lazaric *et al.*, "Linear Thompson sampling revisited," *Electronic Journal of Statistics*, vol. 11, no. 2, pp. 5165–5197, 2017.

[2] R. Achanta, N. Arvanitopoulos, and S. Süsstrunk, "Extreme image completion," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 1333–1337.

[3] R. Agrawal and R. Bala, "Incremental Bayesian classification for multivariate normal distribution data," *Pattern Recognition Letters*, vol. 29, no. 13, pp. 1873–1876, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865508001992

[4] S. Agrawal and N. Goyal, "Thompson sampling for contextual bandits with linear payoffs," in *International Conference on Machine Learning*, 2013, pp. 127–135.

[5] Z. Akata, D. Balliet, M. de Rijke, F. Dignum, V. Dignum, G. Eiben, A. Fokkens, D. Grossi, K. Hindriks, H. Hoos, H. Hung, C. Jonker, C. Monz, M. Neerincx, F. Oliehoek, H. Prakken, S. Schlobach, L. van der Gaag, F. van Harmelen, H. van Hoof, B. van Riemsdijk, A. van Wynsberghe, R. Verbrugge, B. Verheij, P. Vossen, and M. Welling, "A research agenda for hybrid intelligence: Augmenting human intellect with collaborative, adaptive, responsible, and explainable artificial intelligence," *Computer*, vol. 53, no. 8, pp. 18–28, 2020.

[6] N. Akhtar, A. Kuestenmacher, P. G. Ploger, and G. Lakemeyer, "Simulation-based approach for avoiding external faults," in *16th International Conference on Advanced Robotics (ICAR)*. IEEE, 2013, pp. 1–8.

[7] A. Aldoma, Z.-C. Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R. B. Rusu, S. Gedikli, and M. Vincze, "Tutorial: Point cloud library: Three-dimensional object recognition and 6 DoF pose estimation," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 80–91, 2012.

[8] L. Amgoud and C. Cayrol, "Inferring from inconsistency in preference-based argumentation frameworks," *Journal of Automated Reasoning*, vol. 29, no. 2, pp. 125–169, 2002.

[9] L. Amgoud, C. Cayrol, M.-C. Lagasquie-Schiex, and P. Livet, "On bipolarity in argumentation frameworks," *International Journal of Intelligent Systems*, vol. 23, no. 10, pp. 1062–1093, 2008.

[10] L. Amgoud and H. Prade, "Explaining qualitative decision under uncertainty by argumentation." in *Proceedings of the National Conference on Artificial Intelligence*, vol. 21, no. 1. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006, p. 219.

[11] L. Amgoud and M. Serrurier, "Agents that argue and explain classifications," *Autonomous Agents and Multi-Agent Systems*, vol. 16, no. 2, pp. 187–209, 2008.

[12] C. E. Antoniak, "Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems," *The Annals of Statistics*, pp. 1152–1174, 1974.

[13] K. Atkinson, T. Bench-Capon, and D. Bollegala, "Explanation in AI and law: Past, present and future," *Artificial Intelligence*, vol. 289, p. 103387, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370220301375

[14] M. Atzmon, H. Maron, and Y. Lipman, "Point convolutional neural networks by extension operators," *ACM Trans. Graph.*, vol. 37, no. 4, Jul. 2018. [Online]. Available: https://doi.org/10.1145/3197517.3201301

[15] H. Avissar, "Non-parametric belief propagation applications," Ph.D. dissertation, Hebrew University of Jerusalem, 2009.

[16] H. Ayoobi, M. Cao, R. Verbrugge, and B. Verheij, "Handling unforeseen failures using argumentation-based learning," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, Aug 2019, pp. 1699–1704.

[17] H. Ayoobi, M. Cao, R. Verbrugge, and B. Verheij, "Argue to learn: Accelerated argumentation-based learning," in *20th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2021.

[18] H. Ayoobi, M. Cao, R. Verbrugge, and B. Verheij, "Argumentation-based online incremental learning," *IEEE Transactions on Automation Science and Engineering (TASE)*, 2021.

[19] H. Ayoobi, H. Kasaei, M. Cao, R. Verbrugge, and B. Verheij, "Explain What You See: 3D Object Recognition and Parts Segmentation," 2022, (Under Preparation).

[20] H. Ayoobi, H. Kasaei, M. Cao, R. Verbrugge, and B. Verheij, "Local-HDP: Interactive open-ended 3D object category recognition in real-time robotic scenarios," *Robotics and Autonomous Systems (RAS)*, vol. 147, p. 103911, 2022.

[21] H. Ayoobi and M. Rezaeian, "Swift distance transformed belief propagation using a novel dynamic label pruning method," *IET Image Processing*, vol. 14, no. 9, pp. 1822–1831, 2020.

[22] P. Baroni, F. Toni, and B. Verheij, "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games: 25 years later." *Argument & Computation*, vol. 11, no. 1-2, pp. 1–14, 2020.

[23] R. Bellman, "A Markovian decision process," *Journal of Mathematics and Mechanics*, pp. 679–684, 1957.

[24] E. Belouadah, A. Popescu, and I. Kanellos, "A comprehensive study of class incremental learning algorithms for visual tasks," *Neural Networks*, vol. 135, pp. 38–54, 2021.

[25] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 4, pp. 349–359, 1999.

[26] P. Besnard and A. Hunter, "A logic-based theory of deductive arguments," *Artificial Intelligence*, vol. 128, no. 1-2, pp. 203–235, 2001.

[27] G. Biau and E. Scornet, "A random forest guided tour," *Test*, vol. 25, no. 2, pp. 197–227, 2016.

[28] M. Bishop, C. Gates, and K. Levitt, "Augmenting machine learning with argumentation," in *Proceedings of the New Security Paradigms Workshop*, ser. NSPW '18. New York, NY, USA: ACM, 2018, pp. 1–11. [Online]. Available: http://doi.acm.org/10.1145/3285002.3285005

[29] D. M. Blei and M. I. Jordan, "Variational inference for Dirichlet process mixtures," *Bayesian Analysis*, vol. 1, no. 1, pp. 121–143, 03 2006. [Online]. Available: https://doi.org/10.1214/06-BA104

[30] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[31] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *International Conference on Machine Learning*. PMLR, 2015, pp. 1613–1622.

[32] L. Bo, K. Lai, X. Ren, and D. Fox, "Object recognition with hierarchical kernel descriptors," in *CVPR 2011*. IEEE, 2011, pp. 1729–1736.

[33] B. Bonet and H. Geffner, "Arguing for decisions: A qualitative model of decision making," in *Proceedings of the Twelfth international conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 1996, pp. 98–105.

[34] A. Bordes, S. Ertekin, J. Weston, and L. Bottou, "Fast kernel classifiers with online and active learning," *Journal of Machine Learning Research*, vol. 6, no. Sep, pp. 1579–1619, 2005.

[35] G. Borgefors, "Distance transformations in digital images," *Computer Vision, Graphics, and Image Processing*, vol. 34, no. 3, pp. 344–371, 1986.

[36] L. Carstens and F. Toni, "Using argumentation to improve classification in natural language problems," *ACM Transactions on Internet Technology (TOIT)*, vol. 17, no. 3, p. 30, 2017.

[37] L. Carvalho and A. von Wangenheim, "3D object recognition and classification: A systematic literature review," *Pattern Analysis and Applications*, vol. 22, no. 4, pp. 1243–1292, 2019.

[38] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," in *Advances in Neural Information Processing Systems*, 2001, pp. 409–415.

[39] J. Cendrowska, "PRISM: An algorithm for inducing modular rules," *International Journal of Man-Machine Studies*, vol. 27, no. 4, pp. 349–370, 1987.

[40] S. Chen and Z. Wang, "Acceleration strategies in generalized belief propagation," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 1, pp. 41–48, 2011.

[41] S. Chen, H. Tong, Z. Wang, S. Liu, M. Li, and B. Zhang, "Improved generalized belief propagation for vision processing," *Mathematical Problems in Engineering*, vol. 2011, 2011.

[42] Z. Chen and B. Liu, "Lifelong machine learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 12, no. 3, pp. 1–207, 2018.

[43] R. M. Cichy, A. Khosla, D. Pantazis, A. Torralba, and A. Oliva, "Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence," *Scientific Reports*, vol. 6, p. 27755, 2016.

[44] P. Clark and T. Niblett, "The CN2 Induction Algorithm," *Machine Learning*, vol. 3, no. 4, pp. 261–283, 1989.

[45] O. Cocarascu, A. Stylianou, K. Čyras, and F. Toni, "Data-empowered argumentation for dialectically explainable predictions," in *ECAI 2020*. IOS Press, 2020, pp. 2449–2456.

[46] O. Cocarascu and F. Toni, "Argumentation for machine learning: A survey," in *Computational Models of Argument: Proceedings of COMMA 2016*, P. Baroni, T. F. Gordon, T. Scheffler, and M. Stede, Eds. IOS Press, pp. 219–230.

[47] D. Cortes, "Adapting multi-armed bandits policies to contextual bandits scenarios," *arXiv preprint arXiv:1811.04383*, 2019.

[48] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions on image processing*, vol. 13, no. 9, pp. 1200–1212, 2004.

[49] A. M. Dai and A. J. Storkey, "The Supervised Hierarchical Dirichlet Process," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 2, pp. 243–255, Feb 2015.

[50] M. Delange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[51] K. Desingh, S. Lu, A. Opipari, and O. C. Jenkins, "Efficient nonparametric belief propagation for pose estimation and manipulation of articulated objects," *Science Robotics*, 2019.

[52] M. Dimakopoulou, Z. Zhou, S. Athey, and G. Imbens, "Balanced linear contextual bandits," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 3445–3453.

[53] T. Do Van, H. Do Duc, and G. C. Nguyen, "Classify high dimensional datasets using discriminant positive negative association rules," in *2018 5th Asian Conference on Defense Technology (ACDT)*. IEEE, 2018, pp. 1–7.

[54] S. Dong, B. V. Roy, and Z. Zhou, "Provably efficient reinforcement learning with aggregated states," *arXiv preprint arXiv:1912.06366*, 2020.

[55] I. Drori, D. Cohen-Or, and H. Yeshurun, "Fragment-based image completion," in *ACM Transactions on Graphics*, vol. 22, no. 3. ACM, 2003, pp. 303–312.

[56] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[57] P. M. Dung, "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games," *Artificial Intelligence*, vol. 77, no. 2, pp. 321–357, 1995.

[58] D. Eckles and M. Kaptein, "Bootstrap Thompson sampling and sequential decision problems in the behavioral sciences," *SAGE Open*, vol. 9, no. 2, p. 2158244019851675, 2019. [Online]. Available: https://doi.org/10.1177/2158244019851675

[59] S. Eger, J. Daxenberger, and I. Gurevych, "Neural end-to-end learning for computational argumentation mining," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 11–22. [Online]. Available: https://www.aclweb.org/anthology/P17-1002

[60] G. Elidan, I. McGraw, and D. Koller, "Residual belief propagation: Informed scheduling for asynchronous message passing," *arXiv preprint arXiv:1206.6837*, 2012.

[61] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," *International Journal of Computer Vision*, vol. 70, no. 1, pp. 41–54, 2006.

[62] P. F. Felzenszwalb and D. P. Huttenlocher, "Distance transforms of sampled functions," *Theory of computing*, vol. 8, no. 1, pp. 415–428, 2012.

[63] Y. Feng, Z. Zhang, X. Zhao, R. Ji, and Y. Gao, "GVCNN: Group-view convolutional neural networks for 3D shape recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[64] T. S. Ferguson, "A Bayesian analysis of some nonparametric problems," *The Annals of Statistics*, pp. 209–230, 1973.

[65] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, "An HDP-HMM for systems with state persistence," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 312–319.

[66] J. Fox and S. Parsons, "On using arguments for reasoning about actions and values," in *Proceedings of the AAAI Spring Symposium on Qualitative Preferences in Deliberation and Practical Reasoning, Stanford*, 1997, pp. 55–63.

[67] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in Cognitive Sciences*, vol. 3, no. 4, pp. 128–135, 1999.

[68] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *Computer Vision - ECCV 2004*, T. Pajdla and J. Matas, Eds. Berlin, Heidelberg: Springer, 2004, pp. 224–237.

[69] H. Fujita and Y.-C. Ko, "A priori membership for data representation: Case study of spect heart data set," in *Recent Advances in Intelligent Engineering*. Springer, 2020, pp. 65–80.

[70] C. Gao, Y. Luo, H. Wu, and D. Wang, "Data-driven image completion for complex objects," *Signal Processing: Image Communication*, vol. 57, pp. 21–32, 2017.

[71] B. Graham, M. Engelcke, and L. van der Maaten, "3D semantic segmentation with submanifold sparse convolutional networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[72] B. Gu, V. S. Sheng, K. Y. Tay, W. Romano, and S. Li, "Incremental support vector learning for ordinal regression," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 7, no. 26, pp. 1403–1416, 2015.

[73] B. Gu, V. S. Sheng, Z. Wang, D. Ho, S. Osman, and S. Li, "Incremental learning for ν-support vector regression," *Neural Networks*, vol. 67, pp. 140–150, 2015.

[74] Y. Han, Z. Zhou, Z. Zhou, J. Blanchet, P. W. Glynn, and Y. Ye, "Sequential batch learning in finite-action linear contextual bandits," *arXiv preprint arXiv:2004.06321*, 2020.

[75] B. Hao, Y. Abbasi Yadkori, Z. Wen, and G. Cheng, "Bootstrapping upper confidence bound," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019, pp. 12 123–12 133. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/412758d043dd247bddea07c7ec558c31-Paper.pdf

[76] K. He and J. Sun, "Image completion approaches using the statistics of similar patches," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 12, pp. 2423–2435, 2014.

[77] M. Hoffman, F. R. Bach, and D. M. Blei, "Online learning for Latent Dirichlet Allocation," in *Advances in Neural Information Processing Systems 23*, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds. Curran Associates, Inc., 2010, pp. 856–864.

[78] Z. Hu, X. Bai, J. Shang, R. Zhang, J. Dong, X. Wang, G. Sun, H. Fu, and C.-L. Tai, "VMNet: Voxel-mesh network for geodesic-aware 3D semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 488–15 498.

[79] Q. Huang, Y. Wang, and Z. Yin, "View-based weight network for 3D object recognition," *Image and Vision Computing*, vol. 93, p. 103828, 2020.

[80] S. Huang, Y. Chen, T. Yuan, S. Qi, Y. Zhu, and S.-C. Zhu, "PerspectiveNet: 3D object detection from a single RGB image via perspective points," in *Advances in Neural Information Processing Systems*, 2019, pp. 8903–8915.

[81] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Transactions on Graphics*, vol. 36, no. 4, p. 107, 2017.

[82] M. Isard, "Pampas: Real-valued graphical models for computer vision," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, vol. 1. IEEE, 2003, pp. I–I.

[83] Y. Jiang, N. Walker, J. Hart, and P. Stone, "Open-world reasoning for service robots," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 29, 2019, pp. 725–733.

[84] S. Jo, J. Yoo, and U. Kang, "Fast and scalable distributed loopy belief propagation on real-world graphs," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 2018, pp. 297–305.

[85] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 433–449, 1999.

[86] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Machine Learning*, vol. 37, no. 2, pp. 183–233, 1999.

[87] K.-S. Jun, A. Bhargava, R. Nowak, and R. Willett, "Scalable generalized linear bandits: Online computation and hashing," in *Advances in Neural Information Processing Systems*, 2017, pp. 99–109.

[88] A. Kanezaki, Y. Matsushita, and Y. Nishida, "RotationNet for joint object categorization and unsupervised pose estimation from multi-view images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2019.

[89] J. Kappes, B. Andres, F. Hamprecht, C. Schnorr, S. Nowozin, D. Batra, S. Kim, B. Kausler, J. Lellmann, N. Komodakis *et al.*, "A comparative study of modern inference techniques for discrete energy minimization problems," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 1328–1335.

[90] S. H. Kasaei, L. S. Lopes, and A. M. Tomé, "Coping with context change in open-ended object recognition without explicit context information," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018, pp. 1–7.

[91] S. H. M. Kasaei, L. F. Seabra Lopes, and A. M. Tomé, "Local-LDA: Open-ended learning of latent topics for 3D object recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 10, pp. 2567–2580, 2020.

[92] S. H. Kasaei, L. S. Lopes, A. M. Tomé, and M. Oliveira, "An orthographic descriptor for 3D object learning and recognition," in *2016*

IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2016, pp. 4158–4163.

[93] S. H. Kasaei, M. Oliveira, G. H. Lim, L. S. Lopes, and A. M. Tomé, "An adaptive object perception system based on environment exploration and Bayesian learning," in *2015 IEEE International Conference on Autonomous Robot Systems and Competitions*. IEEE, 2015, pp. 221–226.

[94] S. H. Kasaei, M. Oliveira, G. H. Lim, L. Seabra Lopes, and A. M. Tomé, "Interactive open-ended learning for 3D object recognition: An approach and experiments," *Journal of Intelligent & Robotic Systems*, vol. 80, no. 3, pp. 537–553, Dec 2015. [Online]. Available: https://doi.org/10.1007/s10846-015-0189-z

[95] S. H. Kasaei, A. M. Tomé, L. S. Lopes, and M. Oliveira, "GOOD: A global orthographic object descriptor for 3D object recognition and manipulation," *Pattern Recognition Letters*, vol. 83, pp. 312–320, 2016.

[96] S. H. Kasaei, A. M. Tomé, and L. S. Lopes, "Hierarchical object representation for open-ended object category learning and recognition," in *Advances in Neural Information Processing Systems*, 2016, pp. 1948–1956.

[97] R. Kashef, "A boosted SVM classifier trained by incremental learning and decremental unlearning approach," *Expert Systems with Applications*, vol. 167, p. 114154, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417420308976

[98] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[99] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, pp. 2149–2154.

[100] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1568–1583, 2006.

[101] N. Komodakis and G. Tziritas, "Image completion using efficient belief propagation via priority scheduling and dynamic pruning," *IEEE Transactions on Image Processing*, vol. 16, no. 11, pp. 2649–2661, 2007.

[102] Y. Kong, M. Zhang, and D. Ye, "A belief propagation-based method for task allocation in open and dynamic cloud environments," *Knowledge-Based Systems*, vol. 115, pp. 123–132, 2017.

[103] N. Kotonya and F. Toni, "Gradual argumentation evaluation for stance aggregation in automated fake news detection," in *Proceedings of the 6th Workshop on Argument Mining*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 156–166. [Online]. Available: https://aclanthology.org/W19-4518

[104] H. Laga, Y. Guo, H. Tabia, R. B. Fisher, and M. Bennamoun, *3D shape Analysis: Fundamentals, Theory, and Applications*. John Wiley & Sons, 2018.

[105] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 1817–1824.

[106] T. Lattimore and C. Szepesvári, *Bandit Algorithms*. Cambridge University Press, 2020.

[107] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on X-transformed points," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 820–830. [Online]. Available: http://papers.nips.cc/paper/7362-pointcnn-convolution-on-x-transformed-points.pdf

[108] Y. Li, S. Pirk, H. Su, C. R. Qi, and L. J. Guibas, "Fpnn: Field probing neural networks for 3D data," in *Advances in Neural Information Processing Systems*, 2016, pp. 307–315.

[109] K. Liang, H. Chang, B. Ma, S. Shan, and X. Chen, "Unifying visual attribute learning with object recognition in a multiplicative framework," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1747–1760, July 2019.

[110] P. Liang, S. Petrov, M. Jordan, and D. Klein, "The infinite PCFG using hierarchical Dirichlet processes," in *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, 2007, pp. 688–697.

[111] M. R. Loghmani, M. Planamente, B. Caputo, and M. Vincze, "Recurrent convolutional fusion for RGB-D object recognition," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2878–2885, July 2019.

[112] V. Losing, B. Hammer, and H. Wersing, "Incremental on-line learning: A review and comparison of state of the art algorithms," *Neurocomputing*, vol. 275, pp. 1261–1274, 2018.

[113] J. Lu, F. Shen, and J. Zhao, "Using self-organizing incremental neural network (soinn) for radial basis function networks," in *2014 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2014, pp. 2142–2148.

[114] V. N. Lu, J. Wirtz, W. H. Kunz, S. Paluch, T. Gruber, A. Martins, and P. G. Patterson, "Service robots, customers and service employees: what can we learn from the academic literature and where are the gaps?" *Journal of Service Theory and Practice*, 2020.

[115] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[116] M. Mende, M. L. Scott, J. van Doorn, D. Grewal, and I. Shanks, "Service robots rising: How humanoid robots influence service experiences and elicit compensatory consumer responses," *Journal of Marketing Research*, vol. 56, no. 4, pp. 535–556, 2019.

[117] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. Su, "PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[118] M. Moens, "Argumentation mining: How can a machine acquire common sense and world knowledge?" *Argument and Computation*, vol. 9, no. 1, pp. 1–14, 2018.

[119] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, and K.-R. Müller, "Layer-wise relevance propagation: an overview," *Explainable AI: interpreting, explaining and visualizing deep learning*, pp. 193–209, 2019.

[120] M. Mozina, J. Zabkar, and I. Bratko, "Argument based machine learning," *Artificial Intelligence*, vol. 171, no. 10-15, pp. 922–937, 2007.

[121] R. M. Neal, "Markov chain sampling methods for Dirichlet process mixture models," *Journal of Computational and Graphical Statistics*, vol. 9, no. 2, pp. 249–265, 2000.

[122] M. Oliveira, L. S. Lopes, G. H. Lim, S. H. Kasaei, A. M. Tomé, and A. Chauhan, "3D object perception and perceptual learning in the RACE project," *Robotics and Autonomous Systems*, vol. 75, pp. 614–626, 2016.

[123] J. Paisley, C. Wang, D. M. Blei, and M. I. Jordan, "Nested hierarchical Dirichlet processes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 2, pp. 256–270, Feb 2015.

[124] G. Papagni and S. Koeszegi, "Understandable and trustworthy explainable robots: a sensemaking perspective," *Paladyn, Journal of Behavioral Robotics*, vol. 12, no. 1, pp. 13–30, 2021.

[125] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2536–2544.

[126] A. Pazienza, S. Ferilli, and F. Esposito, "On the gradual acceptability of arguments in bipolar weighted argumentation frameworks with degrees of trust," in *International Symposium on Methodologies for Intelligent Systems*. Springer, 2017, pp. 195–204.

[127] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Elsevier, 2014.

[128] K. Polat, "Similarity-based attribute weighting methods via clustering algorithms in the classification of imbalanced medical datasets," *Neural Computing and Applications*, vol. 30, no. 3, pp. 987–1013, 2018.

[129] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[130] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/d8bf84be3800d12f74d8b05e9b89836f-Paper.pdf

[131] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[132] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, Mar 1986. [Online]. Available: https://doi.org/10.1007/BF00116251

[133] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[134] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 2001–2010.

[135] P. Reiner and B. M. Wilamowski, "Efficient incremental construction of RBF networks using quasi-gradient method," *Neurocomputing*, vol. 150, pp. 349–356, 2015.

[136] S. Ren, K. He, R. Girshick, X. Zhang, and J. Sun, "Object detection networks on convolutional feature maps," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 7, pp. 1476–1481, July 2017.

[137] S. Ren, Y. Lian, and X. Zou, "Incremental naïve Bayesian learning algorithm based on classification contribution degree." *JCP*, vol. 9, no. 8, pp. 1967–1974, 2014.

[138] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should I trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[139] C. Riquelme, G. Tucker, and J. Snoek, "Deep Bayesian bandits showdown: An empirical comparison of Bayesian deep networks for Thompson sampling," in *International Conference on Learning Representations*, 2018.

[140] L. Rizzo and L. Longo, "An empirical evaluation of the inferential capacity of defeasible argumentation, non-monotonic fuzzy reasoning and expert systems," *Expert Systems with Applications*, vol. 147, p. 113220, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417420300464

[141] H. Robbins and S. Monro, "A stochastic approximation method," *The Annals of Mathematical Statistics*, pp. 400–407, 1951.

[142] R. S. V. Rodrigues, J. F. M. Morgado, and A. J. P. Gomes, "Part-based mesh segmentation: A survey," *Computer Graphics Forum*, vol. 37, no. 6, pp. 235–274, 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13323

[143] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D recognition and pose using the Viewpoint Feature Histogram," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 2155–2162.

[144] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 3212–3217.

[145] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof, "On-line random forests," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 1393–1400.

[146] E. Sangineto, M. Nabi, D. Culibrk, and N. Sebe, "Self paced deep learning for weakly supervised object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 3, pp. 712–725, March 2019.

[147] P. Schermerhorn and M. Scheutz, "Using logic to handle conflicts between system, component, and infrastructure goals in complex robotic architectures," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 392–397.

[148] S. Schneider, F. Hegger, A. Ahmad, I. Awaad, F. Amigoni, J. Berghofer, R. Bischoff, A. Bonarini, R. Dwiputra, G. Fontana *et al.*, "The RoCKIn@Home Challenge," in *ISR/Robotik 2014; 41st International Symposium on Robotics*, June 2014, pp. 1–7.

[149] S. Sen, M. Das, and R. Chatterjee, "Estimation of incomplete data in mixed dataset," in *Progress in Intelligent Computing Techniques: Theory, Practice, and Applications*. Springer, 2018, pp. 483–492.

[150] L. Shen, L. Wu, Y. Dai, W. Qiao, and Y. Wang, "Topic modelling for object-based unsupervised classification of VHR panchromatic satellite images based on multiscale image segmentation," *Remote Sensing*, vol. 9, no. 8, p. 840, 2017.

[151] B. Sheng, P. Li, X. Fang, P. Tan, and E. Wu, "Depth-aware motion deblurring using loopy belief propagation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 4, pp. 955–969, 2019.

[152] A. Soula, K. Tbarki, R. Ksantini, S. B. Said, and Z. Lachiri, "A novel incremental Kernel Nonparametric SVM model (iKN-SVM) for data classification: An application to face detection," *Engineering Applications of Artificial Intelligence*, vol. 89, p. 103468, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197619303501

[153] H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M.-H. Yang, and J. Kautz, "SPLATNet: Sparse lattice networks for point cloud processing," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[154] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, second edition ed., ser. Adaptive computation and machine learning series. Cambridge, Massachusetts: The MIT Press, 2018.

[155] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A comparative study of energy minimization methods for Markov random fields with smoothness-based priors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 6, pp. 1068–1080, 2008.

[156] K. Talamadupula, G. Briggs, M. Scheutz, and S. Kambhampti, "Architectural mechanisms for handling human instructions for open-world mixed-initiative team tasks and goals," *Advances in Cognitive Systems*, vol. 5, pp. 37–56, 2017.

[157] X. Tao, X. Hong, X. Chang, S. Dong, X. Wei, and Y. Gong, "Few-shot class-incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 183–12 192.

[158] A. Tarski, "A lattice-theoretical fixpoint theorem and its applications." *Pacific Journal of Mathematics*, vol. 5, no. 2, pp. 285–309, 1955.

[159] Y. W. Teh, K. Kurihara, and M. Welling, "Collapsed variational inference for HDP," in *Advances in Neural Information Processing Systems*, 2008, pp. 1481–1488.

[160] Y. Teh, M. Jordan, M. Beal, and D. Blei, "Hierarchical Dirichlet processes," *Journal of the American Statistical Association*, vol. 101, no. 476, 2006.

[161] P. Theologou, I. Pratikakis, and T. Theoharis, "A comprehensive overview of methodologies and performance evaluation frameworks in 3D mesh segmentation," *Computer Vision and Image Understanding*, vol. 135, pp. 49 – 82, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314215000028

[162] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *European conference on computer vision*. Springer, 2010, pp. 356–369.

[163] G. M. Van de Ven and A. S. Tolias, "Three scenarios for continual learning," *arXiv preprint arXiv:1904.07734*, 2019.

[164] F. H. Van Eemeren, B. Garssen, E. C. Krabbe, A. F. S. Henkemans, B. Verheij, and J. H. Wagemans, *Handbook of Argumentation Theory*. Dordrecht: Springer, 2014.

[165] A. Vassiliades, N. Bassiliades, and T. Patkos, "Argumentation and explainable artificial intelligence: a survey," *The Knowledge Engineering Review*, vol. 36, 2021.

[166] G. A. Vreeswijk, "Abstract argumentation systems," *Artificial Intelligence*, vol. 90, no. 1-2, pp. 225–279, 1997.

[167] M. J. Wainwright, T. Jaakkola, and A. S. Willsky, "Tree-based reparameterization for approximate inference on loopy graphs," in *Advances in Neural Information Processing Systems*, 2002, pp. 1001–1008.

[168] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, "Map estimation via agreement on trees: message-passing and linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 11, pp. 3697–3717, 2005.

[169] C. Wang, J. Paisley, and D. Blei, "Online variational inference for the hierarchical Dirichlet process," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, G. Gordon, D. Dunson, and M. Dudík, Eds. Fort Lauderdale, FL, USA: PMLR, 11–13 Apr 2011, vol. 15, pp. 752–760. [Online]. Available: http://proceedings.mlr.press/v15/wang11a.html

[170] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong, "O-cnn: Octree-based convolutional neural networks for 3D shape analysis," *ACM Transactions on Graphics*, vol. 36, no. 4, Jul. 2017. [Online]. Available: https://doi.org/10.1145/3072959.3073608

[171] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Computing Surveys*, vol. 53, no. 3, pp. 1–34, 2020.

[172] Z. Wang and M. Zhou, "Thompson sampling via local uncertainty," in *International Conference on Machine Learning*. PMLR, 2020, pp. 10 115–10 125.

[173] G. I. Webb, L. K. Lee, B. Goethals, and F. Petitjean, "Analyzing concept drift and shift from sample data," *Data Mining and Knowledge Discovery*, vol. 32, no. 5, pp. 1179–1199, 2018.

[174] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, "Chapter 4 - algorithms: The basic methods," in *Data Mining (Fourth Edition)*, fourth edition ed., I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, Eds. Morgan Kaufmann, 2017, pp. 91–160. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128042915000040

[175] . H. Witten, E. Frank, M. A. Hall, and C. J. Pal, "Chapter 6 - trees and rules," in *Data Mining (Fourth Edition)*, fourth edition ed., I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, Eds. Morgan Kaufmann, 2017, pp. 209–242. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128042915000064

[176] W. Wohlkinger and M. Vincze, "Ensemble of shape functions for 3D object classification," in *2011 IEEE international conference on robotics and biomimetics*. IEEE, 2011, pp. 2987–2992.

[177] M. Wozniak, "A hybrid decision tree training method using data streams," *Knowledge and Information Systems*, vol. 29, no. 2, pp. 335–347, 2011.

[178] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.

[179] K. Xu, V. G. Kim, Q. Huang, N. Mitra, and E. Kalogerakis, "Data-driven shape analysis and processing," in *SIGGRAPH ASIA 2016 Courses*, ser. SA '16. New York, NY, USA: Association for Computing Machinery, 2016. [Online]. Available: https://doi.org/10.1145/2988458.2988473

[180] R. Xu, X. Li, B. Zhou, and C. C. Loy, "Deep flow-guided video inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3723–3732.

[181] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Generalized belief propagation," in *Advances in Neural Information Processing Systems*, 2001, pp. 689–695.

[182] F. Yu, K. Liu, Y. Zhang, C. Zhu, and K. Xu, "PartNet: A recursive part decomposition network for fine-grained and hierarchical shape segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[183] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5505–5514.

[184] T. Yu, J. Meng, and J. Yuan, "Multi-View Harmonized Bilinear Network for 3D Object Recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[185] J. Zhang, J. Zhang, S. Ghosh, D. Li, S. Tasci, L. Heck, H. Zhang, and C.-C. J. Kuo, "Class-incremental learning via deep model consolidation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 1131–1140.

[186] S. V. Zhidkov and R. Dinis, "Belief propagation receivers for near-optimal detection of nonlinearly distorted ofdm signals," in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*. IEEE, 2019, pp. 1–6.

[187] Y. Zhong, "Intrinsic shape signatures: A shape descriptor for 3D object recognition," in *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*. IEEE, 2009, pp. 689–696.

[188] B. Zhou, D. Bau, A. Oliva, and A. Torralba, "Interpreting deep visual representations via network dissection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 9, pp. 2131–2145, Sep. 2019.

[189] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3D object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4490–4499.

[190] Z. Zhou, M. Bloem, and N. Bambos, "Infinite time horizon maximum causal entropy inverse reinforcement learning," *IEEE Transactions on Automatic Control*, vol. 63, no. 9, pp. 2787–2802, 2018.

[191] Z. Zhou, S. Athey, and S. Wager, "Offline multi-action policy learning: Generalization and optimization," *arXiv preprint arXiv:1810.04778*, 2018.

[192] Z. Zhou, R. Xu, and J. Blanchet, "Learning in generalized linear contextual bandits with stochastic delays," in *Advances in Neural Information Processing Systems*, 2019, pp. 5197–5208.

List of Publications

Publications

1. H. Ayoobi, M. Cao, R. Verbrugge, and B. Verheij, "Handling unforeseen failure conditions using argumentation-based learning," in *International Conference on Automation Science and Engineering*. IEEE, 2019

2. H. Ayoobi and M. Rezaeian, "Swift distance transformed belief propagation using a novel dynamic label pruning method," *IET Image Processing*, vol. 14, no. 9, pp. 1822–1831, 2020

3. H. Ayoobi, M. Cao, R. Verbrugge, and B. Verheij, "Argumentation-based online incremental learning," *IEEE Transactions on Automation Science and Engineering (TASE)*, 2021

4. H. Ayoobi, M. Cao, R. Verbrugge, and B. Verheij, "Argue to learn: Accelerated argumentation-based learning," in *20th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2021

5. H. Ayoobi, H. Kasaei, M. Cao, R. Verbrugge, and B. Verheij, "Local-HDP: Interactive open-ended 3D object category recognition in real-time robotic scenarios," *Robotics and Autonomous Systems (RAS)*, vol. 147, p. 103911, 2022

In preparation

H. Ayoobi, H. Kasaei, M. Cao, R. Verbrugge, and B. Verheij, "Explain What You See: 3D Object Recognition and Parts Segmentation," 2022, (Under Preparation)