

Learning Neural Parsers with Deterministic Differentiable Imitation Learning

Tanmay Shankar

CMU-RI-TR-18-44

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Thesis Committee:

Kris M. Kitani, Co-Chair
Katharina Muelling, Co-Chair
David Held,
Nicholas Rhinehart

*Submitted in partial fulfillment of the
requirements for the degree of Master in Robotics*

©Tanmay Shankar, 2018

Abstract

We explore the problem of learning to decompose spatial tasks into segments, as exemplified by the problem of a painting robot covering a large object. Inspired by the ability of classical decision tree algorithms to construct structured partitions of their input spaces, we formulate the problem of decomposing objects into segments as a parsing approach. We make the insight that the derivation of a parse-tree that decomposes the object into segments closely resembles a decision tree constructed by ID3, which can be done when the ground-truth is available. We learn to imitate an expert parsing oracle, such that our neural parser can generalize to parse natural images without ground truth. We introduce a novel deterministic policy gradient update, DRAG, in the form of a deterministic actor-critic variant of AggreVaTeD [28], to train our neural parser. Alternatively, our approach may be seen as a variant of the Deterministic Policy Gradient [18, 27] suitable for the imitation learning setting. Training our neural parser via DRAG allows it to outperform several existing imitation and reinforcement learning approaches.

Acknowledgements

I would like to thank my advisors Kris and Katharina, for the inputs and guidance they have provided me over the course of the past two years. I have been extremely fortunate to return to CMU for my masters, and having been advised by Kris and Katharina during this time has contributed greatly to my experience during my master's.

I would also like to thank David, for his quick inputs and questions as a part of my committee. My sincere thanks to Nick and Arjun, who have continually listened and challenged my incessant ideas, eventually helping me find the good ones that go into this thesis. A shout out to Anirudh and Wen for discussing some of these ideas is also in order. I owe thanks to a number of people for livening up my life outside of the lab, Ratnesh, Arjun, Abhijat, Nilesh, and many others.

Finally, I'd like to thank my family, who have supported me and encouraged me in every way possible to help me reach where I am today.

List of Figures

- 3.1 Constructing an equivalent hierarchical decomposition of an object image by two methods. The Information Gain Maximization algorithm creates a decision tree (legend on the left), while the Shape Parsing approach to the right constructs a parse tree (legend to the right). The equivalent hierarchical decomposition (depicted in the center), shows correspondence of each node in the decision tree and parse tree to a segment in the image. 8

- 4.1 Depiction of constructed sample parses for a metal plate (rows 1 & 2), a window frame (rows 3 & 4), and a car door (rows 5 & 6), from the oracle (column 2), the proposed DRAG(column 3), and the various baselines (columns 4-7). The first column shows the original image (odd rows) with ground truth labels (even rows, red object pixels are to be painted, blue are not to be painted). Each column shows the segmented object image along with the predicted label assignment. . 24

List of Tables

4.1	Parsing Accuracies of Proposed model and various baselines	21
-----	--	----

Contents

Abstract	I
Acknowledgements	II
List of Figures	III
List of Tables	V
1 Introduction	1
1.1 Motivation	1
1.2 Thesis Contributions	3
1.3 Thesis Outline	3
2 Related Work	4
2.1 Coverage Planning	4
2.2 Facade Parsing	5
2.3 Semantic Segmentation	5
2.4 Policy Gradient Reinforcement Learning	5
2.5 Imitation Learning	6
3 Approach	7
3.1 Problem Setting	7
3.2 Parsing Objects by Imitating Maximal Information Gain	8

3.3	Shape Parsing Objects	9
3.4	Shape Parsing as a Markov Decision Process	11
3.5	Learning the Shape Parser via Imitation Learning	13
4	Experimental Results	20
4.1	Experimental Objective and Setup	20
4.2	Experimental Results and Analysis	23
5	Conclusions and Future Work	27
5.1	Conclusion	27
5.2	Future Work	28
	Bibliography	29

Chapter 1

Introduction

1.1 Motivation

Consider the task of a robot painting an object or an aerial robot surveying a large field. These spatial tasks represent a coverage problem that the robot may not be able to address in a single shot. For instance, a robot may not be able to paint the entirety of a large object with a single stroke, being limited by the footprint of its paint brush. Instead, the robot must decompose the spatial task of painting objects into smaller segments that it can cover in single stroke. However, discovering or learning an appropriate decomposition of such tasks into segments is challenging. In the object-painting problem, there may be several constraints upon the resultant segments, such as the overall paint coverage. Further, there exist multiple ways to decompose an object into constituent segments - for example, an object may be decomposed into length or breadth-wise segments.

A few well studied algorithms are able to somewhat circumvent these challenges. In particular, when applied to the task of classifying a set of points, ID3 [21] and C4.5 [25] recursively partition the input space in order to achieve an accurate classification of the points. As we demonstrate in ??, an ID3-like Information Gain maximizing algorithm, IGM, builds a similar partitioning of an object as in

Fig. 3.1. However, this IGM algorithm (like the ID3 and C4.5 algorithms it is derived from) requires ground truth classification labels, and cannot be applied to decompose novel objects for which the ground truth labels are unavailable, or are expensive to obtain.

To bypass this issue, we approach the problem of decomposing objects into segments as a *parsing* problem, based on the insight that the derivation of a parse-tree (Fig. 3.1) that decomposes a given object into segments closely resembles a decision tree constructed by IGM (Fig. 3.1). Rather than learn to parse objects by reinforcement learning as in [32], we propose to learn how to parse objects into such structured decompositions via *imitation learning*, treating the IGM algorithm as a *parsing oracle*. Our neural parser is trained to parse objects by imitating the IGM oracle *observing only raw object images* as input, while the IGM oracle exploits access to ground truth information to demonstrate how to parse a particular object. This strategy allows our neural parser to construct structured decompositions of novel unseen objects despite lacking ground truth information. As expected, our imitation learning approach significantly outperforms reinforcement learning baselines in practice.

We further introduce a novel deterministic policy gradient update, DRAG, suitable for training deterministic policies in the hybrid imitation-reinforcement learning setting. The DRAG policy gradient update serves to train the deterministic policy component of our neural parser, eliminating the complexity of maintaining probability distributions specific to the parsing setting. By rephrasing the AggreVaTeD [28] objective in the deterministic policy case, we retrieve a gradient update to a deterministic policy that relies on a differentiable approximation to the oracle’s cost-to-go. This policy gradient update may be viewed as a *deterministic actor-critic variant of AggreVaTeD*, which we refer to as DRAG (i.e. DeteRministically AGgre-vate). DRAG may also be viewed as a variant of the Deterministic Policy Gradient [18] suitable for imitation learning, and replaces an approximation to the true gradi-

ent in the original Deterministic Policy Gradient [18], with the true policy gradient.

Training our parser via DRAG allows our parser to outperform several baselines on the task of parsing novel objects, showcasing its potential to achieve performance closer to that of the oracle than several other existing imitation and reinforcement learning approaches.

1.2 Thesis Contributions

To enumerate the contributions made in this thesis, we introduce:

1. A framework for learning to parse (or decompose) objects into segments, while observing only natural object images as input.
2. An imitation learning approach to learning neural parsers, based on the equivalence between decision-tree algorithms and derivation of parse-trees.
3. A novel Deterministic Policy Gradient update for the hybrid Reinforcement-Imitation Learning setting.

1.3 Thesis Outline

Having briefly introduced the contributions we make in this thesis, we describe where our work stands in relation to existing research works that solve similar problems, or adopt similar approaches in 2. 3 describes our approach in the following manner. We first describe how we formulate the problem, and how one may address this problem via a classical decision tree like algorithm. We then explain the a shape-parsing approach to our problem, and how one may combine the decision-tree like algorithm with the shape-parsing approach in an imitation learning setting. Finally, we introduce our novel deterministic policy-gradient update, DRAG, that we employ to learn a parsing policy via imitation learning.

Chapter 2

Related Work

Our approach draws ideas from a number of fields. We describe some representative work of these fields, and where our work stands in relation to these works that solve similar problems, or adopt similar approaches.

2.1 Coverage Planning

Coverage Planning has traditionally referred to finding a constrained path that while maximizing some notion of “coverage”. While the particular form of the coverage objective and the constraints placed on the path may differ with the application at hand, various solutions to coverage problems bear similarities with one another. Coverage planning has been well explored in traditional literature [6, 12]. The Boustrophedon decomposition [7], and its non-polygonal extension the Morse decomposition [1], adopt cellular decomposition approaches to coverage problems. Alternate approaches include graph or grid coverage approaches, as adopted by [3, 11, 14]. Such problems may be solved exactly for small instances of the problem, or approximately via dynamic programming solutions via the Bellman-Held-Karp algorithm [3, 14]. We direct the reader to [6, 12] for a more thorough review of approaches to coverage planning in robotics.

2.2 Facade Parsing

Facade parsing attempts to identify the topology of a building facade, by parsing an image of the facade into its various components [16, 20, 31, 32, 34]. [31, 32] learn to applying production rules of a grammar to reduce a shape into its constituent segments in a reinforcement learning setting. We build on the insights made in [32] in our approach to the problem of decomposing objects. In contrast to other facade parsing approaches that use labels of the *resulting parse* [16, 20, 34], our method instead seeks to imitate the sequential decision making of an expert parser.

2.3 Semantic Segmentation

The problem of semantic segmentation addresses assigning semantic labels to pixels in a given image. [17] employed a graph partitioning approach for image segmentation, to construct more coherent regions in the image, known as superpixels. Several recent works [2, 17, 19, 26] have trained end-to-end models on large scale datasets for semantic segmentation. We direct readers to [13] for a more complete review of semantic segmentation literature. While the notion of a resultant set of segments with label assignments is common to our parsing approach and semantic segmentation, our approach builds a hierarchical decomposition of the object image as the end result. Another difference is that in our problem, constraints may be imposed that exclude arbitrary results such as the aforementioned stroke coverage constraint of a painting robot.

2.4 Policy Gradient Reinforcement Learning

Stochastic policy gradient approaches [10, 15, 30, 33] have been used to learn control policies in the reinforcement learning (RL) setting. Silver et al. [27] introduced the Deterministic Policy Gradient (DPG), a deterministic counterpart to the stochastic

policy gradient [30]. Lillicrap et al. [18] later extended DPG to the function approximator case. We introduce a variant to these deterministic policy gradient algorithms [18, 27] suitable for imitation learning, that removes the approximation to the true gradient of the policy used in DPG [18, 27].

2.5 Imitation Learning

Recent imitation learning algorithms [9, 23, 24] have addressed the setting when one has access to an expert policy that may be queried. Ross et al. [24] demonstrated an *interactive* imitation learning paradigm, DAgger, is preferable over a naive behavioural cloning approach in this setting. Ross and Bagnell [22] further introduced AggreVaTe, a cost sensitive algorithm that uses estimates of the cost-to-go of the expert to better learn control policies. The AggreVaTeD framework [28] subsequently derived a stochastic policy gradient update of the AggreVaTe [22] objective, enabling its use on complex neural network policies. [5, 29] further explore the realm between imitation and reinforcement learning, by reward shaping using an oracle and switching to policy gradient RL after initial imitation respectively. We follow [22, 28], a paradigm of training agents with partial information to imitate oracles with full information at train time, as in [4, 8].

Chapter 3

Approach

3.1 Problem Setting

We seek to learn how to parse objects directly using object images as input, by learning to imitate an expert parsing oracle. The connection between object parsing and the decision trees constructed by IGM Fig. 3.1 afford us such a parsing oracle, that makes the imitation setting preferable over the reinforcement learning setting followed in [32]. To successfully learn a policy capable of imitating a parsing oracle, we introduce DRAG, a deterministic actor-critic variant of AggreVaTeD [28]. We explain our approach to learning this parsing policy by first describing the problem setting of parsing objects, then describing the IGM algorithm that serves as a ground truth parsing oracle. We then demonstrate how the object parsing problem may be framed as an MDP, finally highlighting how we use DRAG to train a neural parser to decompose objects.

Consider the following problem setting. Given an image of an object to be painted, our objective is to decompose the object image into a set of segments, while maximizing the paint coverage of the object, and ideally minimizing the “paint wasted”. We may view this as assigning labels of whether or not to paint each pixel of the object image, where painting object pixels increases coverage, while not paint-

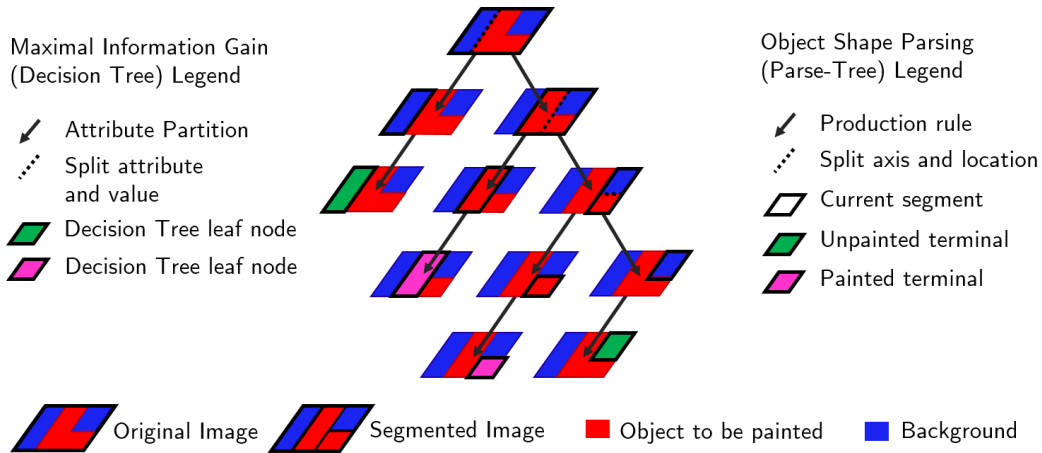


Figure 3.1: Constructing an equivalent hierarchical decomposition of an object image by two methods. The Information Gain Maximization algorithm creates a decision tree (legend on the left), while the Shape Parsing approach to the right constructs a parse tree (legend to the right). The equivalent hierarchical decomposition (depicted in the center), shows correspondence of each node in the decision tree and parse tree to a segment in the image.

ing the surroundings corresponds to decreasing “wasted paint”. Under this label assignment problem, our objective translates to constructing an object decomposition or a set of segments constrained to mutually compose the object image, while *accurately* assigning labels of whether or not to paint each of the resultant segments.

3.2 Parsing Objects by Imitating Maximal Information Gain

Representing the painting process as a labeling problem affords us the ability to employ decision tree algorithms such as ID3 [21] and C4.5 [25], that naturally address labeling tasks by partitioning the space (i.e. the object image) into regions that each contain pixels of a single class. ID3 achieves this by using ground truth information to select the partitioning with the maximum information gain over the resultant segments of the object image. By modifying ID3 to allow multiple splits (i.e. partitions) along a particular axis (or attributes in ID3), we construct a ground-truth

oracle that is able to perfectly label any object image allowed sufficient partitions. We refer to this oracle as the Information Gain Maximization algorithm (IGM), or π_{IGM} . Fig. 3.1 depicts such a “decision tree” constructed from IGM on a toy image.

The IGM oracle uses ground truth labeling of object images, which are unavailable for novel objects. To address this issue, we draw inspiration from facade parsing literature [31, 32], where images of facades were decomposed into various components via *shape-parsing*. We observe that parse-trees that decompose an image into segments in [32] resemble the decision trees constructed by IGM (Fig. 3.1). Motivated by this insight, we decompose objects into segments via a *parsing* approach.

3.3 Shape Parsing Objects

To learn how to decompose object images into their constituent segments, we require an appropriate representation of the recursive object decompositions that arise in the divide-and-conquer paradigm. Shape parsing [32] provides us a compact representation of such potentially complex decompositions of an object. We adopt a *shape parsing* approach similar to [32], using a binary split grammar to represent the hierarchical object decomposition. Formally, we use a probabilistic context-free grammar \mathcal{G} , defined as a tuple $\mathcal{G} = (\mathcal{V}, \mathcal{T}, \mathcal{R}, V_0, \mathcal{P})$, where \mathcal{V} is a set of non-terminal symbols, \mathcal{T} is a set of terminal symbols, V_0 is a starting symbol, \mathcal{R} is a set of production rules, and \mathcal{P} defines a set of probabilities of applying the production rules on a given non-terminal V .

Set of non-terminal symbols \mathcal{V} : A non-terminal symbol $V \in \mathcal{V}$ is defined as an axis-aligned rectangle over the input image. Each non-terminal is specified with a set of attributes (x, y, w, h) , where (x, y) defines the origin of the rectangle and (w, h) define the spatial extents of the rectangle along the horizontal and vertical directions respectively. The starting symbol $V_0 \in \mathcal{V}$ is a rectangular region encompassing the entire object image.

Set of terminal symbols \mathcal{T} : A terminal symbol $T \in \mathcal{T}$ is an axis-aligned rectangle image segment, with an additional attribute b denoting whether a region is to be painted or not.

Set of production rules \mathcal{R} : We consider binary split rules, which split a non-terminal V along its axes into two constituent non-terminals. A split rule is specified by a horizontal or vertical split axis (h or v), and split location l , as $V \xrightarrow{h:l} V V$, or $V \xrightarrow{v:l} V V$. An instance of a horizontal split rule is visualized in the top row of Fig. 3.1, resulting in two non-terminals $(V_{\text{left}}, V_{\text{right}})$. Our grammar also includes terminal production rules that assign a particular precedent non-terminal symbol to one of the two terminal symbols, a region b_p to be painted, or a region not to be painted, b_{np} . The full set of production rules is as follows:

$$\mathcal{R} = \{V_0 \rightarrow V, V \xrightarrow{x:l} (V_{\text{left}}, V_{\text{right}}), V \xrightarrow{y:l} (V_{\text{top}}, V_{\text{bottom}}), V \rightarrow b_p, V \rightarrow b_{np}\}$$

Rule probabilities \mathcal{P} : Production rules r have an associated probability p_r of applying rule r on the current non-terminal (i.e. an image segment). In our problem, we seek to learn these probabilities p_r of applying the production rules, along with the associated split location attribute l of the rules.

Recursively applying production rules \mathcal{R} of the grammar \mathcal{G} described above on an object image I (and the resultant segments) decomposes the image into its constituent segments, a process known as shape-parsing. Parsing an object image results in a hierarchical decomposition of the object, or an object parse-tree, as depicted for a toy object image in Fig. 3.1.

Starting with the entire image (represented by starting symbol V_0), the object parse-tree is constructed by expanding each node in the tree in a top-down and depth-first manner. While expanding a node N of the parse-tree, we sample a rule $r \in \mathcal{R}$ with probability p_r , from the set of rules applicable on N . We add the antecedents of this rule to the parse tree as children nodes of the expanded node. We then continue to derive the tree in a depth-first fashion, moving on to the next unex-

panded node in the tree. The labels assigned to these leaf nodes of a fully expanded image parse tree yield a segmentation of the image. We present a sample object image, the corresponding image parse tree derived via shape parsing, and the final segmented image in Fig. 3.1.

3.4 Shape Parsing as a Markov Decision Process

The process of shape parsing described above can be seen as a sequential decision making process, where a parsing agent find a sequence of partitions and label assignments that maximizes the paint coverage of the object, while affording us a decomposition of the object. More formally, we can describe the sequential process of shape parsing as a Markov Decision Process (MDP) \mathcal{M} .

Applied to our scenario, the current image segment ρ_t corresponds to node N in the parse tree as the current state $s \in S$ of \mathcal{M} . Actions $a \in \mathcal{A}$ correspond to applying production rules $r \in \mathcal{R}$ with a particular split location $l \in [0, 1]$. Upon taking an action a from state s , we “transition” to the next unexpanded node s' in the parse tree. Here, s' is specified by the deterministic transition dynamics $p(s_{t+1}|a_t, s_t)$ enforced by the top-down, depth-first expansion of the tree τ .

The sequence of nodes expanded during the expansion of the image parse-tree corresponds to the sequence of states observed by our agent. This may be incorporated elegantly in the definition of both the one-step reward and the cumulative discounted reward (returns) G of the agent. The one-step reward function encodes the coverage objective of our object parsing problem. For every terminal symbol T , we evaluate the image correlation between the predicted label assignments \mathbb{P} over each of the terminal segments, and the ground truth paint labels of the objects \mathbb{L} . For any non-terminal node V in the parse tree, the return $G(V)$ is defined *recursively* in Eq. (3.1) as the discounted sum of the returns of all child nodes of V . This recursion propagates rewards up the tree in a bottom-up manner, starting from the

terminal leaf-nodes T , where the return is the one-step reward.

$$G(N) = \begin{cases} \sum_{(x,y) \in N} \mathbb{L}(x,y) \mathbb{P}(x,y) & \text{if } N \in \mathcal{T} \\ \sum_{c \in \text{Children}(N)} G(c) & \text{if } N \in \mathcal{V} \end{cases} \quad (3.1)$$

Here, x and y represent pixel locations in the image, $\text{Children}(N)$ is the set of children nodes of N in the tree, C indexes these child nodes.

We seek to learn a policy $\pi : s \rightarrow a$ mapping the current state s to one of the possible actions a available in the current state. In our setting, the policy must select (1) which production rule to apply, and (2) a corresponding split location. To predict these facets of an action a directly from visual input of the current image segment, we represent our policy as a deep convolutional network. Our policy network thus takes in as input an image region $\rho_t(x, y, w, h)$ defined by the current non-terminal V that is being expanded.

The policy then predicts (1) a categorical probability distribution over the valid production rules, $\pi(r|s_t, \theta)$, and (2) a split-location $l = \mu(s_t|\theta)$ within the current image segment in case of applying a split rule. Since the size of the image segment varies at every step, the valid split locations *vary* with the current state. While maintaining a valid probability distribution over such a varying range of values is possible, it is notably challenging, due to the normalizing a distribution across changing scale and limits of the distribution at every step. Instead, we employ a *deterministic* representation of the split location policy, thus $\mu(s_t|\theta)$ is *deterministically* predicted as a scaled logistic function of the deep network features. The two components of our policy network represent the *mixed* deterministic and stochastic nature of the our policy.

3.5 Learning the Shape Parser via Imitation Learning

While Teboul et al. [32] learn a shape-parser via reinforcement learning, it is known that the imitation learning paradigm is preferable to reinforcement learning if an expert agent may be easily obtained [22]. In our case, the connection object parsing and the decision trees constructed by IGM afford us such an expert. We hence consider learning this mixed deterministic-stochastic policy in the imitation learning setting. The stochastic component of the policy $\pi(r|s, \theta)$ may be learned via *off-policy* Monte-Carlo [33] or actor-critic [10] policy gradient algorithms (we point the reader towards [27] for a review of these algorithms). However, existing algorithms for learning the deterministic component of the policy $l = \mu(s|\theta)$ (notably the Deterministic Policy Gradient introduced in [27]), have only been developed in the reinforcement learning setting, not the imitation learning setting.

To learn the deterministic component of the policy $\mu(s|\theta)$, we introduce a deterministic policy gradient update suitable for training deterministic policies in the cost sensitive imitation learning setting. DRAG (DeteRministically AGgregate) may be viewed as a deterministic actor-critic variant of AggreVaTeD [28], or alternatively, a variant of the Deterministic Policy Gradient [18] suitable for imitation learning. DRAG replaces an approximation to the true gradient in the original Deterministic Policy Gradient [18], with the correct gradient.

We present DRAG by first describing the AggreVaTe / AggreVaTeD setting - an ideal starting point given we have an oracle (IGM) that we may *query* for the optimal action to execute from any state. AggreVaTe [22] and AggreVaTeD [28] approach the problem of learning a policy π_θ by training the policy π_{θ_n} at training iteration n to minimize the cost-to-go Q^* of the oracle π^* , over the aggregated distribution of states $d_{\pi_n}^t$ induced by the current learner's policy, π_{θ_n} . To do so, they roll-out a trajectory with a mixture policy $\pi_n(s) = \beta\pi^*(s) + (1-\beta)\pi_\theta(s)$ till time step $t \in [1, \dots, H]$, and subsequently follow the expert π^* then onwards. β simply represents the mixing coefficient, and H is the horizon length of the MDP, and the aggregated distri-

Algorithm 1 Train Parser via DRAG

Input: $\mathcal{D}, \pi^*, \beta, N_{\text{iterations}}$ \triangleright Require a dataset, expert parser, mixing parameter, iterations

Output: π_θ \triangleright Output the learned policy

- 1: $\theta \leftarrow \mathbf{0}, \mathcal{M} \leftarrow \{\}$ \triangleright Initialize Policy Parameters, Initialize Memory
- 2: **for** $i \in [1, 2, \dots, N_{\text{iterations}}]$ **do**
- 3: $\pi_i \leftarrow \beta\pi^* + (1 - \beta)\pi_\theta$
- 4: **for** $j \in [1, 2, \dots, N_{\text{images}} = |\mathcal{D}|]$ **do**
- 5: $t \sim \mathbb{U}[1, H]$ \triangleright Sample a switching index
- 6: $\tau_j = \text{Parse}(\mathcal{D}_j)$ \triangleright Parse the image, following π_i till step t , and π^* thereafter.
- 7: $G_t = G(\tau_j^t)$ \triangleright Evaluate returns at node τ_j^t via expert's cost to go in Eq. (3.1).
- 8: $\mathcal{M} \leftarrow \mathcal{M} \cup \{(s_t, r_t, l_t, t, G_t)\}$ \triangleright Store the transition at index t in memory
- 9: $\mathcal{B} \sim \mathbb{U}[\mathcal{M}], |\mathcal{B}| = B$ \triangleright Sample a minibatch from memory
- 10: $\theta \leftarrow \theta + \alpha \nabla_\theta|_{\mathcal{B}}$ \triangleright Update θ via Eq. (3.7) approximated at \mathcal{B}
- 11: $\omega \leftarrow \omega - \alpha \nabla_\omega|_{\mathcal{B}}$ \triangleright Update ω by gradient of objective in Eq. (3.6) approximated at \mathcal{B}

bution of states $d_{\pi_n}^t$ is defined as $\sum_{\{s_i, a_i\}_{i \leq t-1}} \rho_0(s_1) \prod_{i=1}^{t-1} \pi_n(a_i | s_{t-1}) p(s_t | s_{t-1}, a_{t-1})$, and ρ_0 is the initial state distribution. Sun et al. [28] demonstrate that the objective that is optimized is represented as:

$$J_n(\theta) = \mathbb{E}_{t \sim U(1, \dots, H), s_t \sim d_{\pi_n}^t, a_t \sim \pi_n(a|s)} \left[Q_t^*(s_t, a_t) \right] \quad (3.2)$$

Sun et al. [28] assume a stochastic policy $\pi(a|s, \theta)$ to derive a stochastic policy gradient update to the parameters of the policy θ . However, as mentioned in section 3.4, maintaining a valid probability distribution over split locations in the stochastic policy case is challenging. We hence employ a *deterministic* policy $\mu(s|\theta)$ for split locations - making the stochastic policy gradient update derive in [28] unsuitable for learning $\mu(s|\theta)$. The AggreVaTe objective [22] of minimizing the cost-to-go of the oracle in the deterministic policy setting may be expressed as:

$$J_n(\theta) = \mathbb{E}_{t \sim U(1, \dots, H), s_t \sim d_{\mu_n}^t} \left[Q_t^*(s_t, \mu(s|\theta)) \right]. \quad (3.3)$$

Rather than sampling a split location l_t from a stochastic policy, we retrieve the split location deterministically from the policy $l_t = \mu(s|\theta)$. As in [28], we may improve the policy by updating its parameters θ in the direction of improvement of $J_n(\theta)$, given by the gradient of equation 3.3:

$$\nabla_{\theta} J_n(\theta) = \mathbb{E}_{t \sim U(1, \dots, H), s_t \sim d_{\mu_n}^t} \left[\nabla_{\theta} Q_t^*(s_t, \mu(s|\theta)) \right] \quad (3.4)$$

The Deterministic Policy Gradient [27] allows us to evaluate this gradient, applying the chain rule:

$$\nabla_{\theta} J_n(\theta) = \mathbb{E}_{t \sim U(1, \dots, H), s_t \sim d_{\mu_n}^t} \left[\nabla_a Q_t^*(s_t, a)|_{a=\mu(s|\theta)} \nabla_{\theta} \mu(s|\theta) \right] \quad (3.5)$$

The AggreVaTeD framework [28] uses Monte Carlo samples G_t of the oracle’s cost to go, directly estimating $Q_t^*(s_t, a)$ by sampling. While this provides an unbiased estimate of Q_t^* , we cannot compute the gradient $\nabla_a Q_t^*(s_t, a)$ using non-differentiable samples of the oracle’s cost-to-go. Instead, we construct a *differentiable* approximation of the oracle’s cost to go, in the form of a critic network $Q(s, a|\omega)$, parametrized by ω . While the notion of the critic network is similar to that present in DPG [27] and DDPG [18], note that our critic network approximates the cost-of-go of the *oracle*, rather than the *learner’s* policy μ , i.e. the critic is trained to optimize:

$$\min_{\omega} \mathbb{E}_{s_t \sim d_{\mu_n}^t, a = \mu_n(s_t|\theta)} \left[(Q(s_t, a|\omega) - G_t)^2 \right]. \quad (3.6)$$

Using a critic network $Q_t(s_t, a|\omega)$ to approximate the oracle’s cost-to-go allows us to perform an update to the policy $\mu(s|\theta)$ by replacing $Q_t^*(s_t, a)$ in equations 3.4 and 3.5 with critic network’s estimate, $Q_t(s_t, a|\omega)$, leading to the following deterministic policy gradient update:

$$\nabla_{\theta} J_n(\theta) = \mathbb{E}_{t \sim U(1, \dots, H), s_t \sim d_{\mu_n}^t} \left[\nabla_a Q_t(s_t, a|\omega)|_{a=\mu(s|\theta)} \nabla_{\theta} \mu(s|\theta) \right] \quad (3.7)$$

Employing a differentiable critic network to estimate the oracle’s cost-to-go thus introduces an *deterministic actor critic variant of AggreVaTeD*, which we refer to as DRAG. DRAG hence serves as a deterministic policy gradient update that we use for training our deterministic split location policy $\mu(s|\theta)$ in the cost sensitive imitation learning setting.

We further note that applying the Deterministic Policy Gradient Theorem [27] typically requires an approximation of the true gradient $\nabla_{\theta} J_n(\theta)$ from [10], due to the implicit dependence of $Q(s_t, a_t|\omega)$ on the parameters of the policy θ . However, in DRAG, the learned estimates of $Q(s_t, a_t|\omega)$ estimate the cost-to-go of the *oracle* π^* , and not the learner’s policy $\pi(a_t|s_t, \theta)$. The true cost-to-go of the oracle $Q^*(s_t, a_t)$ (and any estimate $Q(s_t, a_t|\omega)$ of this cost) are both *independent* of the parameters θ of the learner’s policy. DRAG hence removes the dependence of $Q(s_t, a_t|\omega)$ on the learner’s policy θ by virtue of following the oracle after time t , hence the gradient update we present in equation 3.7 is no longer an approximation to the true gradient based on [10].

DRAG allows us to compute a gradient update to deterministic split-location component of our neural parser, while we employ a standard stochastic actor-critic policy gradient update to for the rule policy $\pi(a|s, \theta)$. This leads to a *mixed* stochastic-deterministic policy gradient update in Eq. (3.18):

$$\begin{aligned} \nabla_{\theta} J_n(\theta) = \mathbb{E}_{t \sim U(1, \dots, H), s_t \sim d_{\pi_n, \mu_n}, r_t \sim \pi_n(r|s_t, \theta)} & \left[\nabla_{\theta} \log \pi(r_t|s_t, \theta) \cdot Q_t(s_t, r_t, \mu(s_t|\theta)|\omega) \right. \\ & \left. + \nabla_l Q_t(s_t, r_t, l|\omega)|_{l=\mu(s_t|\theta)} \cdot \nabla_{\theta} \mu(s_t|\theta) \right] \quad (3.8) \end{aligned}$$

The full derivation of this mixed policy gradient update is provided below. This *mixed* stochastic-deterministic policy gradient update is based on DRAG, and is applicable to any policy with both a deterministic and stochastic component. We utilize this mixed update to train our neural parser, as described in Algorithm 1.

3.5.1 Mixed Policy Gradient

Above, we derived a deterministic policy gradient update to deterministic policy trained in the AggreVaTeD [28] setting.

The specific policy representation that we employ in our neural parser has a stochastic component predicting which rules to apply, $\pi(r|s_t, \theta)$, as well as a deterministic component to predict split locations, $\mu(s_t|\theta)$. Given such a *mixed* stochastic-deterministic policy representation, we derive a corresponding *mixed* stochastic-deterministic policy gradient update that we employ to train our neural parser.

Consider that at any time step t , rules r_t are sampled from $\pi(r|s_t, \theta)$, and split locations l_t are given by $\mu(s_t|\theta)$. Following the AggreVaTeD [28] training paradigm, we seek to train the components of our policy $\pi(r|s_t, \theta)$ and $\mu(s_t|\theta)$, to maximize the cost to go of the expert, $Q^*(s_t, r_t, l_t)$. Formally, we seek to maximize:

$$J_n(\theta) = \mathbb{E}_{t \sim U(1, \dots, H), s_t \sim d_{\pi_n, \mu_n}^t, r_t \sim \pi_n(r|s_t, \theta)} \left[Q_t^*(s_t, r_t, \mu(s_t|\theta)) \right] \quad (3.9)$$

As described in section 3.4 of the main paper, the actor critic variant of AggreVaTeD uses a learnt *estimate* $Q(s_t, r_t, l_t|\omega)$ of the cost-to-go of the expert $Q^*(s_t, r_t, l_t)$. The objective equation 3.9 thus becomes:

$$J_n(\theta) = \mathbb{E}_{t \sim U(1, \dots, H), s_t \sim d_{\pi_n, \mu_n}^t, r_t \sim \pi_n(r|s_t, \theta)} \left[Q_t(s_t, r_t, \mu(s_t|\theta)|\omega) \right] \quad (3.10)$$

The expectation of $r_t \sim \pi(r|s_t, \theta)$ may be represented as follows:

$$J_n(\theta) = \mathbb{E}_{t \sim U(1, \dots, H), s_t \sim d_{\pi_n, \mu_n}^t} \left[\sum_{r \in \mathcal{R}} \pi(r_t|s_t, \theta) Q_t(s_t, r_t, \mu(s_t|\theta)|\omega) \right] \quad (3.11)$$

To compute an update to the policy, we may compute the gradient of this objective $J_n(\theta)$ with respect to the parameters of the policy θ :

$$\nabla_{\theta} J_n(\theta) = \nabla_{\theta} \mathbb{E}_{t \sim U(1, \dots, H), s_t \sim d_{\pi_n, \mu_n}^t} \left[\sum_{r \in \mathcal{R}} \pi(r_t|s_t, \theta) Q_t(s_t, r_t, \mu(s_t|\theta)|\omega) \right] \quad (3.12)$$

Considering the linearity of expectations, and taking the gradient ∇_θ inside the sum $\sum_{r \in \mathcal{R}}$, this gives us:

$$\nabla_\theta J_n(\theta) = \mathbb{E}_{t \sim U(1, \dots, H), s_t \sim d_{\pi_n, \mu_n}^t} \left[\sum_{r \in \mathcal{R}} \nabla_\theta \left\{ \pi(r_t | s_t, \theta) Q_t(s_t, r_t, \mu(s_t | \theta) | \omega) \right\} \right] \quad (3.13)$$

Applying the product rule, we have:

$$\begin{aligned} \nabla_\theta J_n(\theta) = \mathbb{E}_{t \sim U(1, \dots, H), s_t \sim d_{\pi_n, \mu_n}^t} \left[\sum_{r \in \mathcal{R}} \left\{ \nabla_\theta \pi(r_t | s_t, \theta) \cdot Q_t(s_t, r_t, \mu(s_t | \theta) | \omega) \right. \right. \\ \left. \left. + \pi(r_t | s_t, \theta) \cdot \nabla_\theta Q_t(s_t, r_t, \mu(s_t | \theta) | \omega) \right\} \right] \quad (3.14) \end{aligned}$$

The first term in the expectation, $\nabla_\theta \pi(r_t | s_t, \theta) \cdot Q_t(s_t, r_t, \mu(s_t | \theta) | \omega)$ may be simplified by applying the importance sampling trick:

$$\begin{aligned} \nabla_\theta J_n(\theta) = \mathbb{E}_{t \sim U(1, \dots, H), s_t \sim d_{\pi_n, \mu_n}^t} \left[\sum_{r \in \mathcal{R}} \left\{ \pi(r_t | s_t, \theta) \cdot \frac{\nabla_\theta \pi(r_t | s_t, \theta)}{\pi(r_t | s_t, \theta)} \cdot Q_t(s_t, r_t, \mu(s_t | \theta) | \omega) \right. \right. \\ \left. \left. + \pi(r_t | s_t, \theta) \cdot \nabla_\theta Q_t(s_t, r_t, \mu(s_t | \theta) | \omega) \right\} \right] \quad (3.15) \end{aligned}$$

This becomes the gradient of the *log-probability* of the policy $\pi(r | s_t, \theta)$:

$$\begin{aligned} \nabla_\theta J_n(\theta) = \mathbb{E}_{t \sim U(1, \dots, H), s_t \sim d_{\pi_n, \mu_n}^t} \left[\sum_{r \in \mathcal{R}} \left\{ \pi(r_t | s_t, \theta) \cdot \nabla_\theta \log \pi(r_t | s_t, \theta) \cdot Q_t(s_t, r_t, \mu(s_t | \theta) | \omega) \right. \right. \\ \left. \left. + \pi(r_t | s_t, \theta) \cdot \nabla_\theta Q_t(s_t, r_t, \mu(s_t | \theta) | \omega) \right\} \right] \quad (3.16) \end{aligned}$$

The second term in the expectation, $\pi(r_t | s_t, \theta) \cdot \nabla_\theta Q_t(s_t, r_t, \mu(s_t | \theta) | \omega)$, may be computed using the Deterministic Policy Gradient Theorem [27] (i.e. essentially apply-

ing the chain rule):

$$\begin{aligned} \nabla_{\theta} J_n(\theta) = \mathbb{E}_{t \sim U(1, \dots, H), s_t \sim d_{\pi_n, \mu_n}^t} & \left[\sum_{r \in \mathcal{R}} \left\{ \pi(r_t | s_t, \theta) \cdot \nabla_{\theta} \log \pi(r_t | s_t, \theta) \cdot Q_t(s_t, r_t, \mu(s_t | \theta) | \omega) \right. \right. \\ & \left. \left. + \pi(r_t | s_t, \theta) \cdot \nabla_l Q_t(s_t, r_t, \mu(s_t | \theta) | \omega) \Big|_{l=\mu(s_t | \theta)} \cdot \nabla_{\theta} \mu(s_t | \theta) \right\} \right] \quad (3.17) \end{aligned}$$

We may now convert the expression $\sum_{r \in \mathcal{R}} \pi(r | s_t, \theta)$ back to an expectation of $r_t \sim \pi(r | s_t, \theta)$, leading to the following policy gradient update to our neural parser:

$$\begin{aligned} \nabla_{\theta} J_n(\theta) = \mathbb{E}_{t \sim U(1, \dots, H), s_t \sim d_{\pi_n, \mu_n}^t, r_t \sim \pi_n(r | s_t, \theta)} & \left[\nabla_{\theta} \log \pi(r_t | s_t, \theta) \cdot Q_t(s_t, r_t, \mu(s_t | \theta) | \omega) \right. \\ & \left. + \nabla_l Q_t(s_t, r_t, \mu(s_t | \theta) | \omega) \Big|_{l=\mu(s_t | \theta)} \cdot \nabla_{\theta} \mu(s_t | \theta) \right] \quad (3.18) \end{aligned}$$

Note that applying the Deterministic Policy Gradient Theorem typically requires an approximation from [10], due to the implicit dependence of $Q(s_t, a_t | \omega)$ on the parameters of the policy θ . However, in the deterministic variant of AggreVaTeD, the learnt estimates of $Q(s_t, a_t | \omega)$ estimate the cost-to-go of the *expert policy* π^* , and not the learner’s policy $\pi(a_t | s_t, \theta)$.

The true cost-to-go of the expert $Q^*(s_t, a_t)$ (and any estimate $Q(s_t, a_t | \omega)$ of this cost) are both *independent* of the parameters of the policy θ . The deterministic actor-critic variant of AggreVaTeD removes the dependence of $Q(s_t, a_t | \omega)$ on θ , thus removes the necessity for the approximation from Degris et al. [10].

Chapter 4

Experimental Results

4.1 Experimental Objective and Setup

The objective of our experiments is to evaluate the following:

1. The idea of learning to parse objects by imitation learning.
2. The ability of a learnt parser to generalize to unseen objects.
3. The relative benefit of Reinforcement Learning, Imitation Learning, and hybrid Imitation and Reinforcement Learning approaches in the problem of learning a parser.
4. The ability of our policy update, DRAG , to learn a parsing policy.

In order to be able to evaluate our approach with regard to the above points, we evaluate DRAG relative to several baseline imitation learning (IL), reinforcement learning (RL) and hybrid IL+RL approaches, and the IGM oracle.

We describe the baseline algorithms, and the exact policy representation they use below. Each model uses a convolutional neural network with 7 convolutional layers and 2 dense layers as a base model. The baseline approaches then represent their respective policies as follows:

Table 4.1: Parsing Accuracies of Proposed model and various baselines

Model	Train Accuracy	Test Accuracy
IGM Oracle with GT Access (Depth 7)	98.50%	—
Monte-Carlo Policy Gradient (RL)	53.54%	51.23%
DDPG (RL)	51.94%	48.78%
Behavior Cloning (IL)	75.11%	75.10%
Dagger (IL)	84.01%	84.03%
Stochastic AggreVaTeD (IL+RL)	81.85%	81.07%
Off-Policy Monte-Carlo Policy Gradient (IL+RL)	84.85%	83.65%
DRAG (IL+RL) (Ours)	88.05%	86.86%

Reinforcement Learning baselines:

In the pure RL setting, the learner is provided with evaluations of the quality of the parses it constructs via the reward function, and does not have access to the IGM oracle agent in any form. We consider two RL baselines, where the objective is to simply maximize the cumulative reward achieved by the learner:

- *Monte-Carlo Policy Gradient (MCPG)*: We consider an on-policy stochastic Monte-Carlo Policy Gradient approach, similar to REINFORCE. The policy maintains a categorical distribution over valid rules, predicted as a softmax of deep network features over the valid rules applicable at the current image segment, and a logit-normal distribution over valid split locations within the boundaries of the current image segment.
- *Deterministic Policy Gradient (DPG)*: We then consider a DDPG [18] style approach, where the policy deterministically predicts split locations as scaled logistic function of the object image features. The rules here are still predicted stochastically as a softmax output of the deep policy network.

IL baselines:

In contrast with the RL setting, the learner in the pure IL setting has access to the actions taken by the IGM oracle, and not the reward function. Here, the learner sim-

ply tries to copy the actions executed by the expert; this corresponds to maximizing the likelihood of the rules selected by the expert under the learner’s policy, and *regressing* to the split locations selected by the expert. We consider two IL baselines:

- *Behavior Cloning*: Here, the agent minimizes the categorical cross entropy between the rules selected by the IGM oracle, and the softmax probability distribution over rules predicted by the policy network. This is equivalent to maximizing the log-likelihood of the rules selected by the IGM oracle. The split locations are predicted as a scaled logistic function of the deep network features. A L2 norm loss between the predicted and IGM oracle split is used to train the agent’s split location policy.
- *Dagger*: Following the interactive learning paradigm DAgger [24], objects are parsed according to a *mixture* of the expert and the learner’s current policy. The policy representation is identical to that used in the Behavior Cloning case.

Hybrid IL+RL baselines:

Of particular interest to us is the hybrid IL+RL case, where the learner has access to both the actions executed by the expert, as well as samples of the reward function for the parses it constructs.

- *Off-Policy Monte-Carlo Policy Gradient (Off-MCPG)*: As in the case of the vanilla MCPG, the policy representation is a categorical distribution over the valid rules, and the splits with a logit-normal distribution over valid splits.
- *AggreVaTeD*: We finally consider the original stochastic policy gradient training paradigm of AggreVaTeD [28]. As in the Off-MCPG case, the rules are predicted via a categorical distribution from the deep policy network features, and the split locations are predicted as a logit-normal distribution over valid splits.

The IL and IL+RL baselines are provided access to the IGM Oracle, with a maximum allowed parse tree depth of 7 enforced for computational reasons.

Experimental Setup:

To evaluate our proposed DRAG approach against the above baselines, we use the following experimental setup. We collect a set of 300 RGB training object images of size 256×256 pixels, each annotated with a per-pixel binary label of 1 (to be painted), or 0 (not to be painted), serving as ground truth object labels. We evaluate our models on a set of 62 test images annotated with similar ground truth labels, measuring performance as the pixel accuracy between the predicted assignment of labels of the object image against these ground truth object labels, presented for each of the above baselines and our approach in Table 4.1. An ideal object parse assigns paint labels to all portions of the object to be painted over contiguous segments, ensuring that parts of the object do not go unpainted. We present the parses created for 3 sample images in Fig. 4.1 for the oracle, our DRAG approach, and the various baselines.

4.2 Experimental Results and Analysis

Based on the results presented in Table 4.1, and the sample parses as depicted in Figure 4.1, we make the following observations.

(1) Reinforcement learning applied to our task is unsuccessful.

As seen in Table 4.1, both the stochastic MCPG and deterministic DPG baselines achieve no better than random performance. This is likely because RL relies on sampling good rules *and* split locations via random exploration to learn. The recursive nature of parsing means that constructing a good parse after selecting initially bad actions is difficult. Conversely, good initial actions do not guarantee an appropriate parse; leaf node assignments must be made appropriately in order to obtain a high-reward parse. Good initial rule and split location selections must also be followed by correctly labeling the parsed segments. These challenges contribute to the failure of the MCPG and DPG baselines to parse object images at all.

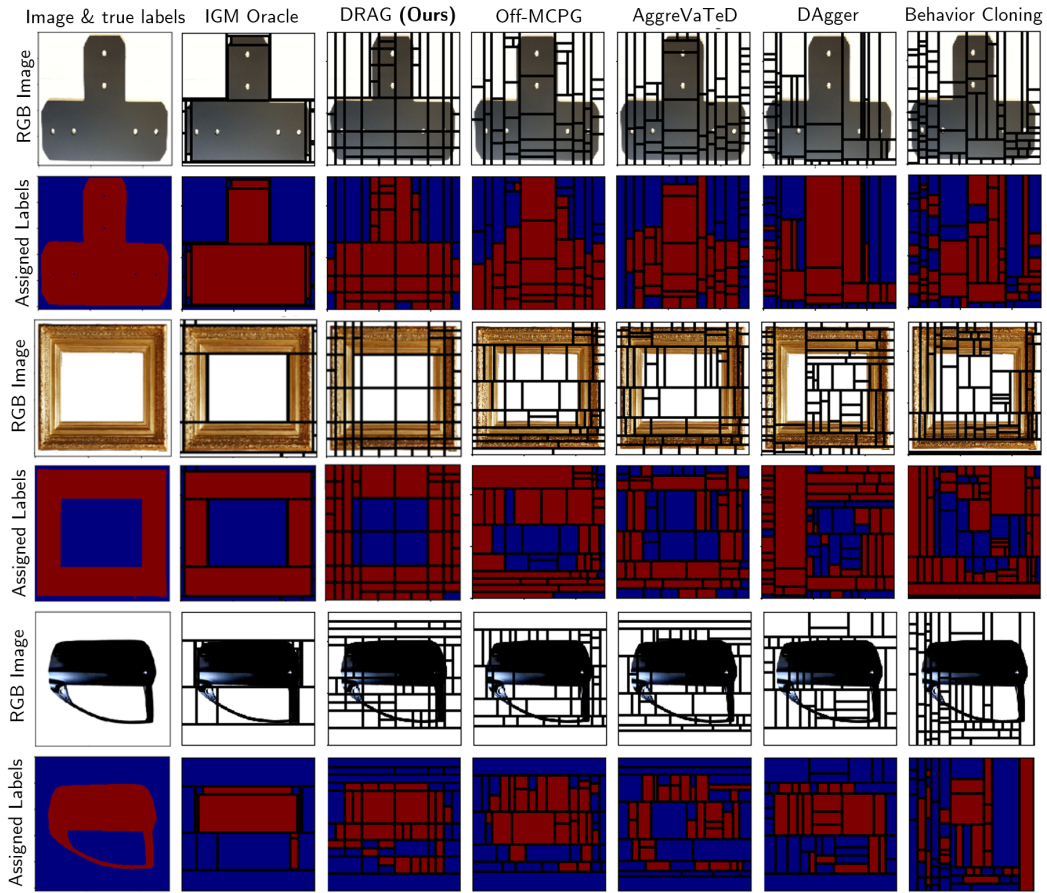


Figure 4.1: Depiction of constructed sample parses for a metal plate (rows 1 & 2), a window frame (rows 3 & 4), and a car door (rows 5 & 6), from the oracle (column 2), the proposed DRAG(column 3), and the various baselines (columns 4-7). The first column shows the original image (odd rows) with ground truth labels (even rows, red object pixels are to be painted, blue are not to be painted). Each column shows the segmented object image along with the predicted label assignment.

(2) Imitation learning applied to our parsing task is successful.

The structural similarity between the “decision-trees” built by the IGM oracle and the parse trees constructed by parsing allows the IGM oracle to *guide sampling* towards good rules and split locations, rather than naively explore all possible rules and splits via RL. In contrast with the RL setting, the IL setting hence allows for

actually learning a parser. As empirically validated in table Table 4.1, even our most naive IL baseline, Behavior Cloning, significantly outperforms the RL baselines achieving 75.10% test accuracy. The DAgger baseline reconciles with the state distribution mismatch [24], boosting the parser to 84.03% test accuracy. The IL paradigm is thus able to learn reasonable parses of object images, overcoming the issues of learning a parser present in the RL setting. The IL setting however is not perfect - it learns parses of objects that are often not coherent, or mislabels large portions of the object.

(3) DRAG outperforms state-of-the-art IL and RL baselines on the task of parsing novel objects.

We observe that the relative benefits of the IL paradigm transfer over to the hybrid IL+RL paradigm, as validated by both the quantitative (Table 4.1) and visual parse results (Figure 4.1). The hybrid IL+RL paradigm in theory moves towards optimizing the underlying cost function, in addition to imitating the expert. This facilitates learning particularly when the expert is suboptimal with respect to the underlying cost function, which occurs in our case due to the restricted depth of the parse tree allowed to the expert.

We note that the hybrid IL+RL baselines (stochastic AggreVaTeD and Off-MCPG) perform nearly as well as the best performing IL baseline DAgger. Further, the hybrid IL+RL paradigm is demonstrated to be Quantitatively, our proposed DRAG approach achieves significantly higher train and test set accuracies (88.05 % and 86.86% respectively) than all other baseline approaches. While other stochastic hybrid IL+RL baselines (Off-MCPG and AggreVaTeD) are able to capture coarse object structure, they fail to capture splits aligned with prominent image gradients. In contrast, DRAG is able to assign labels that correlate strongly with that of the ground truth (and the IGM oracle). Further, DRAG’s deterministic representation of split locations enables DRAG to construct more regular parses of the object im-

ages by selecting splits that align with object boundaries, as compared to baseline approaches. This is particularly suitable for axis-aligned images, or images with small aberrations (such as rows 3 & 4 in Figure 4.1). In the case of irregular images such as the car door in rows 5 & 6, when the IGM oracle misses out on the correctly labeling the rim of the door (due to a restricted parse tree depth), DRAG is able to correctly label this portion of the door, in contrast with all other baseline approaches.

Chapter 5

Conclusions and Future Work

5.1 Conclusion

In this paper, we address the problem of learning to parse objects into hierarchical decompositions via imitation learning. By treating an Information Gain Maximizing algorithm as an expert parsing oracle, our neural parser learns to parse objects by imitating this IGM oracle, observing only raw object images as input. We further introduce a novel deterministic policy gradient update, DRAG, suitable for generic imitation learning tasks with a deterministic policy representation. The proposed DRAG may be seen as both as a deterministic actor-critic variant of AggreVaTeD [28] and a variant of DDPG [18] suitable for imitation learning. Training our neural parser to parse objects using DRAG outperforms existing RL, IL and IL+RL baselines, leading to more accurate and coherent parses. Our experimental results demonstrate the capability of our approach to successfully parse objects, and potentially address more generic spatially decomposable tasks.

5.2 Future Work

We present several avenues of future work that are potentially interesting directions to extend the work presented in this thesis. Note that we present the results of our novel deterministic policy gradient update, DRAG , in the context of the parsing problem. We note, however, that DRAG is indeed a more general policy gradient update suitable, and is applicable to generic hybrid IL+RL problems. We believe evaluating DRAG on other such hybrid IL+RL problems, such as on the OpenAI Gym-Robotics manipulation tasks, could be an interesting direction to pursue. The parsing approach to coverage problems that we derive here is generally applicable to other coverage problems. However, it would be interesting to observe whether a similar approach may be adopted to other spatial tasks, in a hierarchical RL setting.

Bibliography

- [1] Ercan U. Acar, Howie Choset, Alfred A. Rizzi, Prasad N. Atkar, and Douglas Hull. Morse decompositions for coverage tasks. *I. J. Robotics Res.*, 21:331–344, 2002.
- [2] Vijay Badrinarayanan, Ankur Handa, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *CoRR*, abs/1505.07293, 2015. URL <http://arxiv.org/abs/1505.07293>.
- [3] Richard Bellman. Dynamic programming treatment of the travelling salesman problem. *J. ACM*, 9:61–63, 1962.
- [4] Mohak Bhardwaj, Sanjiban Choudhury, and Sebastian Scherer. Learning heuristic search via imitation. *arXiv preprint arXiv:1707.03034*, 2017.
- [5] C.-A. Cheng, X. Yan, N. Wagener, and B. Boots. Fast Policy Learning through Imitation and Reinforcement. *ArXiv e-prints*, May 2018.
- [6] Howie Choset. Coverage for robotics – a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31(1):113–126, Oct 2001. ISSN 1573-7470. doi: 10.1023/A:1016639210559. URL <https://doi.org/10.1023/A:1016639210559>.
- [7] Howie Choset and Philippe Pignon. Coverage path planning: The boustrophedon cellular decomposition. In Alexander Zelinsky, editor, *Field and Service Robotics*, pages 203–209, London, 1998. Springer London. ISBN 978-1-4471-1273-0.
- [8] Sanjiban Choudhury, Ashish Kapoor, Gireeja Ranade, Sebastian Scherer, and Debadeepta Dey. Adaptive Information Gathering via Imitation Learning. 2017. URL <http://arxiv.org/abs/1705.07834>.

- [9] Hal Daumé, John Langford, and Daniel Marcu. Search-based structured prediction. *Machine learning*, 75(3):297–325, 2009.
- [10] Thomas Degris, Martha White, and Richard S. Sutton. Off-policy actor-critic. *CoRR*, abs/1205.4839, 2012. URL <http://arxiv.org/abs/1205.4839>.
- [11] Yoav Gabriely and Elon Rimon. Spanning-tree based coverage of continuous areas by a mobile robot. *Annals of Mathematics and Artificial Intelligence*, 31(1):77–98, Oct 2001. ISSN 1573-7470. doi: 10.1023/A:1016610507833. URL <https://doi.org/10.1023/A:1016610507833>.
- [12] Enric Galceran and Marc Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61(12):1258 – 1276, 2013. ISSN 0921-8890. doi: <https://doi.org/10.1016/j.robot.2013.09.004>. URL <http://www.sciencedirect.com/science/article/pii/S092188901300167X>.
- [13] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and José García Rodríguez. A review on deep learning techniques applied to semantic segmentation. *CoRR*, abs/1704.06857, 2017. URL <http://arxiv.org/abs/1704.06857>.
- [14] M. Held and R. Karp. A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied Mathematics*, 10(1):196–210, 1962. doi: 10.1137/0110015. URL <https://doi.org/10.1137/0110015>.
- [15] Tang Jie and Pieter Abbeel. On a connection between importance sampling and the likelihood ratio policy gradient. In *Advances in Neural Information Processing Systems*, pages 1000–1008, 2010.
- [16] Mateusz Kozinski, Raghudeep Gadde, Sergey Zagoruyko, Guillaume Obozinski, and Renaud Marlet. A mrf shape prior for facade parsing with occlusions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2820–2828, 2015.
- [17] Zhenguo Li, Xiao-Ming Wu, and Shih-Fu Chang. Segmentation using superpixels: A bipartite graph partitioning approach. In *CVPR*, pages 789–796. IEEE Computer Society, 2012. ISBN 978-1-4673-1226-4. URL <http://dblp.uni-trier.de/db/conf/cvpr/cvpr2012.html#LiWC12>.

- [18] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. 2015. ISSN 1935-8237. doi: 10.1561/22000000006. URL <http://arxiv.org/abs/1509.02971>.
- [19] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014. URL <http://arxiv.org/abs/1411.4038>.
- [20] Anelo Martinović, Markus Mathias, Julien Weissenberg, and Luc Van Gool. A three-layered approach to facade parsing. In *European conference on computer vision*, pages 416–429. Springer, 2012.
- [21] J. R. Quinlan. Induction of decision trees. *MACH. LEARN*, 1:81–106, 1986.
- [22] Stéphane Ross and J. Andrew Bagnell. Reinforcement and Imitation Learning via Interactive No-Regret Learning. pages 1–14, 2014. URL <http://arxiv.org/abs/1406.5979>.
- [23] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635, 2011.
- [24] Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. *Proceedings of AIS-TATS*, 15:627–635, 2011. ISSN <null>. URL <http://arxiv.org/abs/1011.0686>.
- [25] Steven L. Salzberg. C4.5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993. *Machine Learning*, 16(3):235–240, Sep 1994. ISSN 1573-0565. doi: 10.1007/BF00993309. URL <https://doi.org/10.1007/BF00993309>.
- [26] Abhishek Sharma, Oncel Tuzel, and David W Jacobs. Deep hierarchical parsing for semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 530–538. IEEE, 2015.
- [27] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic Policy Gradient Algorithms. *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 387–395, 2014. ISSN 1938-7228.

- [28] Wen Sun, Arun Venkatraman, Geoffrey J. Gordon, Byron Boots, and J. Andrew Bagnell. Deeply AggreVaTeD: Differentiable Imitation Learning for Sequential Prediction. 2017. doi: arXiv:1504.01391. URL <http://arxiv.org/abs/1703.01030>.
- [29] Wen Sun, J. Andrew Bagnell, and Byron Boots. TRUNCATED HORIZON POLICY SEARCH: DEEP COMBINATION OF REINFORCEMENT AND IMITATION. In *International Conference on Learning Representations*, 2018. URL <https://openreview.net/forum?id=ryUlhzWCZ>.
- [30] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.
- [31] Olivier Teboul, Loïc Simon, Panagiotis Koutsourakis, and Nikos Paragios. Segmentation of building facades using procedural shape priors. In *CVPR*, pages 3105–3112. IEEE Computer Society, 2010. ISBN 978-1-4244-6984-0. URL <http://dblp.uni-trier.de/db/conf/cvpr/cvpr2010.html#TeboulSKP10>.
- [32] Olivier Teboul, Iasonas Kokkinos, Loïc Simon, Panagiotis Koutsourakis, and Nikos Paragios. Shape grammar parsing via reinforcement learning. In *CVPR*, pages 2273–2280. IEEE Computer Society, 2011. ISBN 978-1-4577-0394-2. URL <http://dblp.uni-trier.de/db/conf/cvpr/cvpr2011.html#TeboulKSKP11>.
- [33] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, May 1992. ISSN 1573-0565. doi: 10.1007/BF00992696. URL <https://doi.org/10.1007/BF00992696>.
- [34] Hao Zhang, Kai Xu, Wei Jiang, Jinjie Lin, Daniel Cohen-Or, and Baoquan Chen. Layered analysis of irregular facades via symmetry maximization. *ACM Trans. Graph.*, 32(4):121–1, 2013.