TOWARDS ACTIVE AND INTERACTIVE VISUAL LEARNING

by

YAN ZHU

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Computer Science

Written under the direction of

Dimitris N. Metaxas

And approved by

____

____

____

____

New Brunswick, New Jersey

OCTOBER, 2017

ABSTRACT OF THE DISSERTATION

Towards Active and Interactive Visual Learning

By Yan Zhu

Dissertation Director:

Dimitris N. Metaxas

Modern computer vision models mostly rely on massive human annotated datasets for supervised training. The models are typically learned from the supervision of static datasets in a passive learning manner. As the performance on classical computer vision tasks tends to saturate, novel visual tasks emerged and posed challenges to the traditional passive learning paradigm. We explored such new settings where huge dataset supervisions are scarce, and novellearning paradigms beyond passive training are proposed. We specifically focused on the following three visual learning scenarios, in which we showed active and interactive learning paradigms are better suited than traditional passive learning.

First, we focused on histopathological image classification with a limited annotation budget. We proposed an active selection algorithm via constrained submodular function maximization. The proposed method encourages uncertainty reduction as well as selection diversity. We also show the greedy-like algorithm has near optimal theoretical guarantee and scalable to large scale unlabeled data. Second, we proposed a novel semantic amodal segmentation task in which occluded object segmentation masks are predicted. To address the challenge of inadequate hard examples, we proposed to actively generate hard synthetic examples for training. Experiment results demonstrate improved performance against baselines. We also show the amodal segmentation can be applied to spatial depth ordering. Third, we proposed an interactive learning approach to generate natural language dialogue between two conversation agents, in order to accomplish a visual ground task. Experiment results showed that the interactive learning significantly improved the supervised training baseline, and the performance gains most when multiple models are simultaneously updated through mutual interaction. The analysis on the generated conversations showed the thorough interactive training, two agents learned to evolve the communication towards a more efficient direction, and improved the task success rate.

Acknowledgements

I would like to thank Professor Dimitris N. Metaxas for his kind support and guidance during the past five years. I am deeply grateful for his insightful talks and encouragement, which have helped me tremendously.

I would like to thank my other committee members: Prof. Kostas Bekris, Prof. Konstantinos Michmizos and Prof. Dimitris Samaras for their valuable advice and comments regarding this thesis. It is my great honor to have each of them in my committee and help me to improve this work.

At Rutgers, I am fortunate to have collaborated with Prof. Shaoting Zhang, Prof. Chao Chen and Dr. Wei Liu. I am deeply grateful for their precious discussions and encouragement when I was still a junior graduate student.

I would like to sincerely thank my internship mentors at Facebook AI Research, Piotr Dollár and Yuandong Tian, for their tremendous support, patience and huge inspiration as academic role models to me.

Also, I would like to thank all the labmates at Rutgers CBIM lab, and fellow officemates at Hill center 4th floor. Thanks for the memories and sharing the excitements and frustrations along this unforgettable journey.

Last but not least, I never forget to thank my families, for their unselfish support and enduring me all these years. Please allow me to express my gratitude, now and always.

Dedication

To my parents

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Supervised learning has been the most dominant and widely applied machine learning paradigm in modern computer vision research. Many seminal supervised training models including AdaBoost[6], SVM [7], and convolutional neural network [8] have been adopted widely in practical computer vision tasks. And their theoretical properties such as PAC learning theory [9] have also been well studied. With the recent renaissance of deep neural network, we've witnessed large scale deep learning models achieving near human accuracy performance on several traditional computer vision tasks including image classification [10, 11, 12, 13], semantic segmentation [14], face recognition[15] and object detection [16].

The success of large scale supervised trained models typically requires massive human-annotated dataset, a differentiable supervised objective function and gradient descent updates, as illustrated in 1.1. For example, the award-winning image classification neural network architecture, Residual Network [13] is trained on ImageNet dataset [3], which contains about 14 million images and 22K annotated image concepts. Since image classification has been traditionally considered as a fundamental computer vision task, the community has been collaboratively invested considerable resources in creating large scale annotated datasets including ImageNet, SUN [17], COCO [18] and Youtube-8M [19]. The emergence of such large scale datasets enabled the breakthrough of large scale supervised training. As the performance saturation in above mentioned classical computer vision tasks, computer vision researchers began to focus on new tasks beyond traditional image classification or object detection, towards artificial generic intelligence (AGI) agents. However, such new tasks usually lack a proper large scale annotated dataset for training or not directly learnable with a straightforward supervised loss function. Even worse, if we want to train an agent with intelligent behavior under complex environments (either in physical environments or synthetic environments), the corresponding

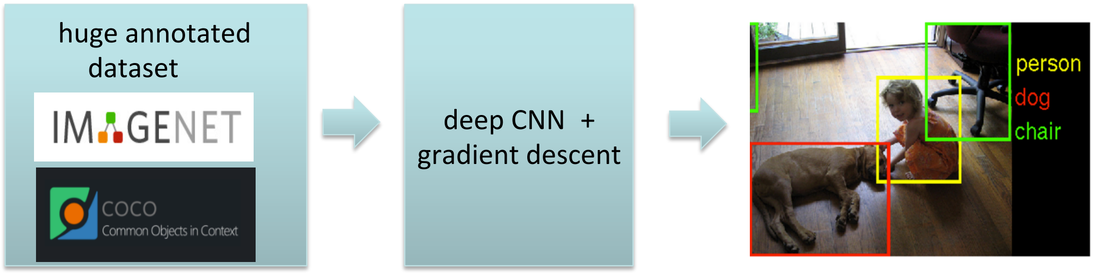
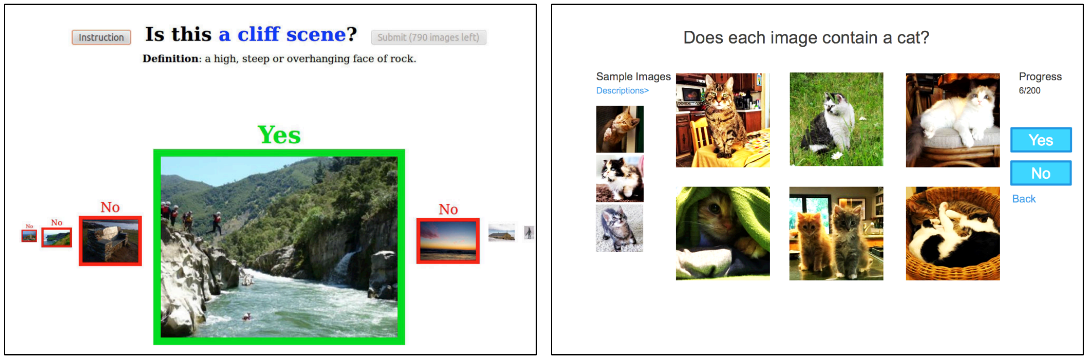


Figure 1.1: A conceptional scheme of an exemplary supervised trained computer vision model. Modern computer vision models typically rely on large scale annotated image dataset with millions of images and annotated instances, such as COCO [18] and ImageNet [3]. Deep learning models typically have large model capacity and trained on a differentiable supervised loss function via gradient descent updates. Despite good generalization, the supervised trained model is trained in a passive manner since the desired model behavior entirely come from static datasets.

knowledge can be hardly represented human annotation directly. The inherent difference for such new tasks is that the intelligent agents need to be able to reason within realistic visual scenes, while the traditional computer vision research more focuses on the low level perception problems. Such transitions also require researchers to explore new learning paradigms beyond static, passive supervised learning. This dissertation mainly addresses such new scenarios and proposed several approaches towards active and interactive learning paradigms, compared with traditional passive supervised learning.

1.1 Background

For traditional computer vision tasks such as image classification and face recognition, supervised learning paradigm generally considers these tasks as perception problems. These tasks can be reduced to a statistical classification or regression task, and thus can be properly learned by fitting a powerful parametric model with large scale training samples. Thanks to the availability of large scale annotated dataset and development in gradient back propagation techniques, deep neural network achieved near-human performance on these tasks. However, the supervised trained deep neural networks mostly follow the same underlying passive learning paradigm, which has been existed for years and well-studied both empirically and theoretically.



Is this a cliff scene?

Definition: a high, steep or overhanging face of rock.

Does each image contain a cat?

Figure 1.2: Example of typical annotation interfaces benchmark image datasets. The first one is the interface for Place dataset [2], and the second one is from ImageNet dataset. [3]. Both web interfaces are used on Amazon Mechanical Turk platform for crowd-sourcing annotators. Both tasks involve basic visual concept perception, thus can be accomplished by annotators without any domain expertise.

Humans in real world need to process and understand visual data more than low level perception. Also, many novel tasks cannot be properly addressed by simply fitting the model from huge amount of image annotation pairs. Such new tasks include visual reasoning, video game AI agent and visual conversational bot. We will particularly focus on the below three scenarios, in which traditional “passive” supervised learning fail to perform well, and new learning paradigms need to be explored.

1.1.1 Annotation budget

With the popularity of crowd-sourcing platform such as Amazon Mechanical Turk, the annotation cost for image dataset has been effectively reduced. Most large scale computer vision datasets are annotated via public crowd-sourcing platforms including COCO [18], MIT places [2] and Visual Genome [20]. Most natural image annotation tasks are simple visual perception task, and therefore the annotation cost is greatly reduced once crowd sourced via web platform. Since visual perception is a universal capability, these annotation tasks can be reliably completed by normal annotators without any specialized domain knowledge. Figure 1.2 shows two exemplary annotation interfaces for some benchmark computer vision datasets.

However, for some specific applications, such as medical imaging or satellite images, annotation requires domain experts with strong domain knowledge. Such requirement effectively makes crowd-sourcing annotation infeasible, and makes the annotation cost much higher than natural images. Under such scenario, the dataset size is constrained by a limited annotation budget. One practical problem is which subset of the unlabeled images we should annotate given a fixed annotation budget, in order to train a decent model with best performance? This will require the model to *actively* select what data for learning, rather than passively accept training data fed to it.

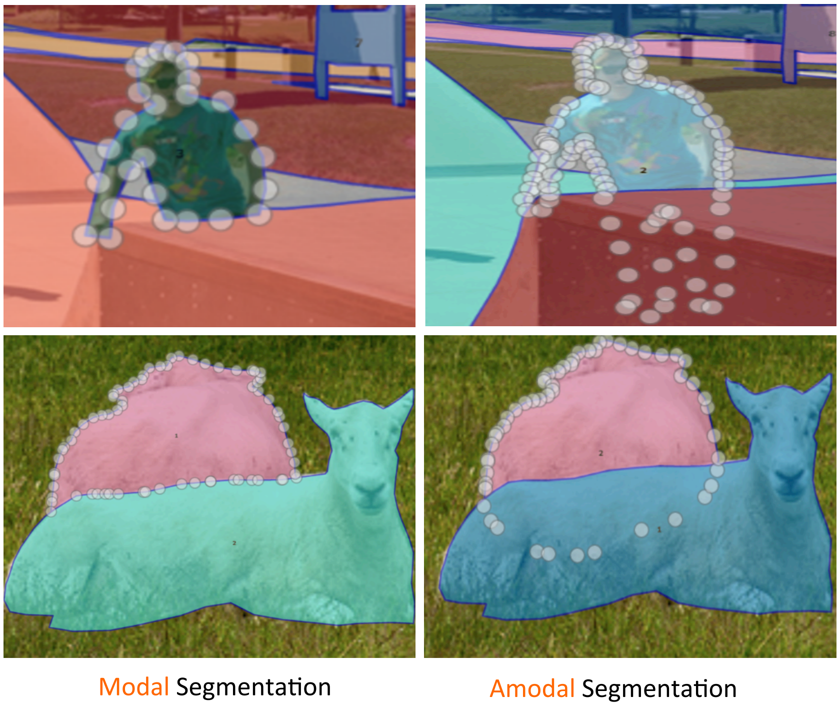
1.1.2 Inadequate hard examples

As a classical computer vision task, semantic segmentation has long been defined as a per-pixel classification problem: each visible pixel belongs to the foreground object and can be classified into one of the predefined categories. This definition is fundamentally different from how human perceive objects and scenes in real world: human beings are able to perceive both visible and occluded pixels as a whole object, a perception mechanism called "amodal perception" [21, 22]. Traditional semantic segmentation research focuses on "modal segmentation", in which models essentially learns to classify each pixel, even if contexts and high order potential models [23] were extensively studied. In contrast, amodal segmentation requires models able to reason the occlusion relationship, understand the spatial scene layout and infer the actual underlying shape. Figure 1.3 illustrate the difference between modal segmentation versus amodal segmentation.

To study the amodal shape completion, the proper training instance would be occluded object instances with canonical shapes. However, in natural images, high percentage of non-occluded or slight occluded object instances are not suitable to be used for training or testing the amodal segmentation task. Manually collecting such hard examples from real images become infeasible. Under such scenario, actively generating proper synthetic hard examples, and learning to deform easy examples become an important problem.

1.1.3 Learn from interaction

Human beings acquire knowledge not merely from supervision of examples, but also more importantly, from daily interactions and various feedbacks. One of the most primary interaction forms is through natural language conversation. Ideally, we also hope the intelligent agents can



Modal Segmentation

Amodal Segmentation

Figure 1.3: Comparison of *modal* segmentation (left column) versus *amodal* segmentation (right column). In *modal segmentation*, only objects' visual regions are segmented, while in *amodal segmentation*, human not only segment the visual regions, but also infer the underlying occluded shape, by reasoning the spatial occlusion relationship within the visual scene.

understand and articulate natural language. For example, we want to command robots via natural language conversation such as "Can you pick up the blue bottle from the bookshelf behind the table?". To accomplish the required command, the agent needs to translate the command from unconstrained language sentence into a set of executable commands, then corresponds the abstract linguistic concepts with the visual environments. To accurately pick up the right object, the agent needs to ground the concepts such as "blue bottle" and "the bookshelf behind the table" on the real world visual scenes. In NLP community, conversational agents have already been studied and related commercial products such as Apple Siri were widely used. However, such conversational bots are not capable to ground the conversation with visual concepts.

Due to huge variation and ambiguity exists in natural language conversations, such knowledge can not be effectively learned from human examples. As we will show in Chapter 4. Supervised trained conversational agents only mimic the human dialogue distributions, but fail to generate useful and effective conversation in order to achieve a common goal.

1.2 Contribution of the Dissertation

We mainly focused on the above mentioned challenging scenarios and proposed active and interactive learning paradigms for the following three new computer vision tasks, and showed the improved performance against the passive learning baselines.

First, we proposed an active selection algorithm for histopathological image classification. The proposed method is based on constrained submodular optimization, which is scalable and has approximated theoretical guarantee. The method is evaluated on a breast tissue histopathological image dataset and empirically outperform other active learning methods.

Second, we proposed a novel vision task called semantic amodal segmentation. We systematically proposed a dataset to research amodal segmentation, and also proposed strong baselines and evaluation metrics. To address the inadequacy of hard occluded examples, we also proposed to actively generate hard examples during training, and improved the baseline segmentation performance.

Third, we proposed a novel interactive learning approach to generate natural language conversations in order to accomplish an object grounding task. In the proposed method, two models are simultaneously improved during natural language interactions. We showed that both generative models not only managed to improve the collaborative task completion rate, but also learns to evolve with new language characteristics, and effectively arrived a new communication protocol variant. The resulting performance significantly improved the state-of-the-art and reduced the gap between human performances.

1.3 Outline of the Dissertation

The outline of the dissertation is as follows:

Chapter 2 introduced the constrained submodular optimization based active selection approach for histopathological images. It includes the review of active learning literature, constrained submodular maximization formulation, proof and experiment details on a breast tissue image dataset.

In Chapter 3, we focused on the semantic amodal segmentation problem and active generation of synthetic hard examples. It includes annotation details of the amodal segmentation dataset, the corresponding annotation consistency analysis and the evaluation of amodal segments and depth ordering task. Also, it introduced the active generation approach to address the problem of inadequate hard examples.

Chapter 4 addressed the interactive learning system to generate natural language conversation for an object grounding task. It includes the introduction and related work about visual conversation tasks, proposed methodology, the experiment results and the observations on the generated conversations.

Chapter 5 summarized this dissertation and discussed the current limitation and future directions.

Chapter 2

Active Learning for Medical Image Analysis

2.1 Introduction

Recent development of microscopical acquisition technology enables computerized analysis of histopathological images [24]. For example, in the context of breast cancer diagnosis, plenty of systems have been designed to conduct automatic and accurate analysis of high-resolution images digitized from tissue histopathology slides, where well-known machine learning and image processing techniques [25, 26, 27] have been exploited. Particularly, supervised learning models such as Support Vector Machines (SVMs) [28] have been extensively employed, because they are able to effectively bridge the so-called "semantic gap" between histopathological images and their diagnosis information [26, 29, 24, 30]. To train an accurate prediction model under a supervised manner, it is usually necessary to require a large amount of labeled data, *e.g.*, manual annotations from domain experts or pathologists. However, acquiring and collecting high quality annotations is a very expensive and tedious process. To alleviate this issue and reduce the labeling cost, active learning [31] has been suggested to intelligently select a small yet informative subset of the whole database, which requires only a few labeling operations from domain experts to build an accurate enough prediction model with a low training cost.

Active learning has been widely investigated in the machine learning community, aiming for progress in both theoretical aspects, *e.g.*, sample complexity bounds [32] and agnostic active learning [33], and valid methods solving practical applications, *e.g.*, image [34] and text[35] classification and retrieval (the related work in active learning is described below). However, for histopathological images, previous active learning methods have two main shortcomings: 1) Almost all of them assume that unlabeled data samples are *independently and identically distributed* (I.I.D.), which is not necessarily suitable for histopathological images. In fact, for each patient there are usually several images available which share common pathological characteristics, *e.g.*, images from different ROIs. Obviously, there are considerable correlations among such image samples. 2) Even if the I.I.D. property holds, previous active learning methods may disregard the structured information of histopathological images, *e.g.*, patient identity, which is easy to obtain but could be crucial for active learning to enforce diversity during sample selection.

In this work, we propose a novel batch mode active learning approach which is specifically designed for histopathological image analysis and leverages structured information to enforce diversity during intelligent sample selection. We formulate the active learning problem (essentially the sample selection problem) as a constrained submodular optimization problem and present a greedy algorithm to efficiently solve it. Notably, we provide a theoretical bound characterizing the quality of the submodular active learning strategy, which guarantees that our proposed greedy algorithm approximates the optimal batch mode active learning strategy for the adaptive submodular function maximization problem with a partition matroid constraint. In practice, our active learning driven histopathological image analysis approach outperforms state-of-the-art methods that are proposed recently to tackle histopathological image analysis. We perform experiments on a large database of histopathological images with high-dimensional features. The experimental results demonstrate the efficacy of our approach which achieves 83% prediction accuracy with merely 100 labeled samples among more than two thousand images (*i.e.*, less than 5% training data). This accuracy is 11% higher than passive learning and 6% higher than state-of-the-art active learning methods.

2.2 Related Work

Active learning can be considered as a combinational optimization problem which is typically difficult to exactly solve, so a variety of heuristics have been resorted to. For example, a number of active learning algorithms relax the original combinational problem that involves discrete constraints to a continuous optimization problem, and then employ regular convex or nonconvex optimization techniques to solve the relaxed problem. These algorithms usually suffer from prohibitively high computational complexities, and the deviation from the solution of the relaxed problem to the solution of the original problem remains unknown. In contrast, some latest work copes with the active learning problem via submodular set function maximization which is a direct combinational optimization method. While maximizing a submodular function appears NP-hard, a landmark result from Nemhauser *et al.* [36] certifies that a simple greedy optimization scheme is able to achieve the $(1-\frac{1}{e})$-approximation for the cardinality constraint and the $(\frac{1}{p+1})$-approximation for p matroid constraints, respectively. Built on this theoretic finding, Chen and Krause [37] propose a nearly optimal batch mode active learning algorithm by applying adaptive submodular optimization [38]. Our active learning method is motivated by this line of submodular optimization, and firstly explores and leverages structured information of histopathological images through imposing a partition matroid constraint on active learning.

2.3 Approach

2.3.1 Problem Definition

Given an unlabeled dataset $\mathcal{U} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$, each data sample $\mathbf{x}_i \in \mathcal{U}$ carries a random label variable $y_i \in \mathcal{Y}$ ($\mathcal{Y} = \{1, -1\}$) in our binary classification task for which the positive label '1' implies 'benign' and the negative label '-1' implies 'actionable'. Assume that there exists a joint probability distribution $P(\mathbf{y}_{\mathcal{U}})$ of the labels of the samples in $\mathcal{U}$, where $\mathbf{y}_{\mathcal{U}} = [y_1, \cdots, y_n]^\top \in \mathcal{Y}^n$. Batch mode active learning selects a small subset of $\mathcal{U}$, queries their labels from experts, and then trains a classifier using the chosen labeled samples. To be specific to histopathological image analysis, batch mode active learning works as follows: whenever a batch of k unlabeled images $\mathcal{B} \subseteq \mathcal{U}$ ($|\mathcal{B}| = k$) are selected, their associated labels $\mathbf{y}_{\mathcal{B}} \in \mathcal{Y}^k$ are requested from the diagnosis of pathologists and acquired simultaneously; the obtained labels are used to select next batches of images iteratively until the needed classification (*i.e.*, predicting 'benign' or 'actionable') accuracy is achieved.

2.3.2 Adaptive Submodular Optimization

Our goal is to learn a classifier $h : \mathcal{U} \rightarrow \mathcal{Y}$ from a set $\mathcal{H}$ of finite hypotheses. We write $\mathcal{S} = \{(\mathbf{x}_i, y_i)\} \subseteq \mathcal{U} \times \mathcal{Y}$ to denote the set of observed sample-label pairs. We define $\mathcal{H}(\mathcal{S}) =$

$\{h \in \mathcal{H} : y_i \equiv h(\mathbf{x}_i), \forall(\mathbf{x}_i, y_i) \in \mathcal{S}\}$ to denote the reduced hypothesis space consistent with the observed sample-label pairs in $\mathcal{S}$. We then define and aim to maximize the objective set function $f : 2^{\mathcal{U}\times\mathcal{Y}} \rightarrow \mathbf{R}$ as

$$f(\mathcal{S}) = |\mathcal{H}| - |\mathcal{H}(\mathcal{S})|, \tag{2.1}$$

where the operator $|\cdot|$ outputs the cardinality of an input set. In this paper, we study hyperplane hypotheses in the form of $h(\mathbf{x}) = \mathrm{sgn}(\mathbf{w}^\top\mathbf{x})$ in which the sign function $\mathrm{sgn}(x)$ returns 1 if $x > 0$ and -1 otherwise. Intuitively, the function $f(\mathcal{S})$ measures the number of hypotheses eliminated by the observed labeled data in $\mathcal{S}$. As a matter of fact, f satisfies the following properties:

- $f(\varnothing) = 0$; (**Normalized**)
- for any $\mathcal{S}_1 \subseteq \mathcal{S}_2 \subseteq \mathcal{U} \times \mathcal{Y}$, $f(\mathcal{S}_1) \leq f(\mathcal{S}_2)$; (**Monotonic**)
- for any $\mathcal{S}_1 \subseteq \mathcal{S}_2 \subseteq \mathcal{U} \times \mathcal{Y}$ and $(\mathbf{x}, y) \in (\mathcal{U} \times \mathcal{Y})\backslash\mathcal{S}_2$, we have $f(\mathcal{S}_2 \cup \{(\mathbf{x}, y)\}) - f(\mathcal{S}_2) \leq f(\mathcal{S}_1 \cup \{(\mathbf{x}, y)\}) - f(\mathcal{S}_1)$; (**Submodular**)
- for an unlabeled sample $\mathbf{x}$ and an observed data subset $\mathcal{S} \subseteq \mathcal{U} \times \mathcal{Y}$, define the conditional expected marginal gain of $\mathbf{x}$ with regard to $\mathcal{S}$ as

  $$\Delta_f(\mathbf{x} \mid \mathcal{S}) = \sum_{y\in\mathcal{Y}} P(y_i = y \mid \mathcal{S})[f(\mathcal{S} \cup \{(\mathbf{x}, y)\}) - f(\mathcal{S})], \tag{2.2}$$

  and then the function f along with the distribution $P(\mathbf{y}_{\mathcal{U}})$ is called adaptive submodular if $\Delta_f(\mathbf{x} \mid \mathcal{S}_2) \leq \Delta_f(\mathbf{x} \mid \mathcal{S}_1)$ holds for any $\mathcal{S}_1 \subseteq \mathcal{S}_2 \subseteq \mathcal{U} \times \mathcal{Y}$ and $P(\mathcal{S}_2) > 0$. (**Adaptive Submodular** [38])

To work under the batch mode setting, the *BatchGreedy* algorithm [37] generalizes the conditional marginal benefit in Eq. (2.2) to allow for conditioning on a set of selected but not yet observed sample-label pairs within the current batch. *BatchGreedy* greedily selects the samples within each batch and assembles batches in a sequential manner. Specifically, *BatchGreedy* selects the i-th sample in the j-th batch as follows:

$$\mathbf{x}^* = arg \max_{\mathbf{x}\in\mathcal{U}} \Delta_f(\mathbf{x} \mid \{\mathbf{x}_{1,j}, ..., \mathbf{x}_{i-1,j}\}, \mathcal{S}), \tag{2.3}$$

where $\mathcal{S}$ represents the observed labeled data from all previous $j-1$ batches, and $\{\mathbf{x}_{1,j}, \cdots, \mathbf{x}_{i-1,j}\}$ retains the selected $i-1$ samples whose labels are not observed yet within the current j-th batch. This algorithm is theoretically guaranteed to obtain an approximation to the optimal batch-mode active sampling strategy.

2.3.3 Modeling the Partition Matroid Constraint

Since images of the same patient are very likely to include large pathological information redundancy, we propose to explicitly enforce diversity within the selected images by imposing an additional partition matroid constraint on the original adaptive submodular function maximization problem in Eq. (3).

A partition matroid constraint is defined as follows: $\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_q$ are a partitioning of the set $\mathcal{U}$ if $\mathcal{U} = \bigcup_{1 \leq i \leq q} \mathcal{P}_i$ and $\mathcal{P}_1, \cdots, \mathcal{P}_q$ are disjoint with each other. We require the currently selected batch to include at most one sample from each subset $\mathcal{P}_i$.

More formally, our proposed constrained problem is defined as follows:

$$\begin{aligned} \mathcal{B}^* = \ & \arg\max_{\mathcal{B} \subseteq \mathcal{U}} \Delta_f(\mathcal{B} \mid \mathcal{S}) \\ & \text{subject to } |\mathcal{B}| = k, |\mathcal{B} \cap \mathcal{P}_i| \leq 1, \forall i \in \{1, ..., q\}, \end{aligned} \tag{2.4}$$

where $\mathcal{B}^*$ is the optimal k-cardinality batch selected from the current unlabeled dataset $\mathcal{U}$, $\mathcal{P}_1, \cdots, \mathcal{P}_q$ are q disjoint subsets partitioning $\mathcal{U}$, and $\mathcal{S}$ is the set composed of the previously observed labeled data. These disjoint subsets can be obtained through performing clustering according to the structured information of the annotated images.

Within each batch, the i-th sample of the j-th batch is selected as follows

$$\begin{aligned} \mathbf{x}^* = \ & \arg\max_{\mathbf{x} \in \mathcal{U}} \Delta_f(\mathbf{x} \mid \{\mathbf{x}_{1,j}, ..., \mathbf{x}_{i-1,j}\}, \mathcal{S}) \\ & \text{subject to } cluster(\mathbf{x}) \neq cluster(\mathbf{x}_{k,j}), \forall k \in \{1, \cdots, i-1\}, \end{aligned} \tag{2.5}$$

where $cluster(\mathbf{x})$ is the index of the cluster that $\mathbf{x}$ belongs to.

For the sequential version of this problem, Golovin and Krause[39] have proven that the greedy method can achieve a $(\frac{1}{p+1})$-approximation to the optimum when maximizing f subject to p matroid constraints, which motivates us to generalize this result to the batch mode setting.

Algorithm 1 BGAL-PMC (Batch Greedy Active Learning with a Partition Matroid Constraint)

```
Input: a set of disjoint clusters $\mathcal{P}_1, \mathcal{P}_2, ...\mathcal{P}_q$, previously selected dataset $\mathcal{B}$ and their observed labels $\mathbf{y}_\mathcal{B}$, unlabeled dataset $\mathcal{U}$, hypothesis set size $N$, and batch size $k$.
Ouput: the selected batch $\mathcal{S}$ and their labels $\mathbf{y}_\mathcal{S}$.
Sample a hypothesis set $\mathcal{H} = \{h_1, h_2, ...h_N\}$ using $\mathbf{y}_\mathcal{B}$;
initialize $\mathcal{S} = \emptyset$, $D = \emptyset$, and $\mathcal{T} = \emptyset$;
for $i = 1$ to $k$ do
  for $j = 1$ to $|\mathcal{U}|$ do
    $score(\mathbf{x}_j) = |\mathcal{H}(\{\mathbf{x}, h(\mathbf{x}) \mid \mathbf{x} \in \mathcal{S} \cup \{\mathbf{x}_j\}\})|$
  end for
  while true do
    $\mathbf{x}^* = arg \min_{\mathbf{x} \in \mathcal{U} \setminus \{\mathcal{S} \cup \mathcal{T}\}} score(\mathbf{x})$
    $ind = cluster(\mathbf{x}^*)$
    if $ind \notin D$ then
      $\mathcal{S} = \mathcal{S} \cup \{\mathbf{x}^*\}$, $D = D \cup \{ind\}$
      break
    else
      $\mathcal{T} = \mathcal{T} \cup \{\mathbf{x}^*\}$
    end if
  end while
end for
query the labels $\mathbf{y}_\mathcal{S}$ for $\mathcal{S}$.
```

We propose a practical batch mode active learning algorithm BGAL-PMC, as described in Algorithm 1. In what follows, we show that BGAL-PMC can well approximate the optimal batch selection strategy. Note that $\mathcal{H}$ is the hypothesis set.

The resulting active selection framework is conceptionally shown in Fig. 2.1.

Theorem 1. *Given a monotonic and submodular function f and a label distribution P such that (f, P) is adaptive submodular, when maximizing f subject to a partition matroid constraint, the expected cost of the BGAL-PMC algorithm is at most $2(\ln(|\mathcal{H}| - 1) - 1)$ times the expected cost of the optimal batch selection strategy.*

The proof of Theorem 1 is provided in the supplemental material. Importantly, this theorem guarantees that BGAL-PMC needs at most $2(\ln(|\mathcal{H}| - 1) - 1)$ times more batches than those required by the optimal batch selection strategy. Note that directly searching for the optimal selection strategy takes exponential time. To sample a finite hypothesis set $\mathcal{H}$, we employ the hit-and-run sampler [40] to generate a set of linear separators, which has been used by [37][41] and proven effective for active learning problems.

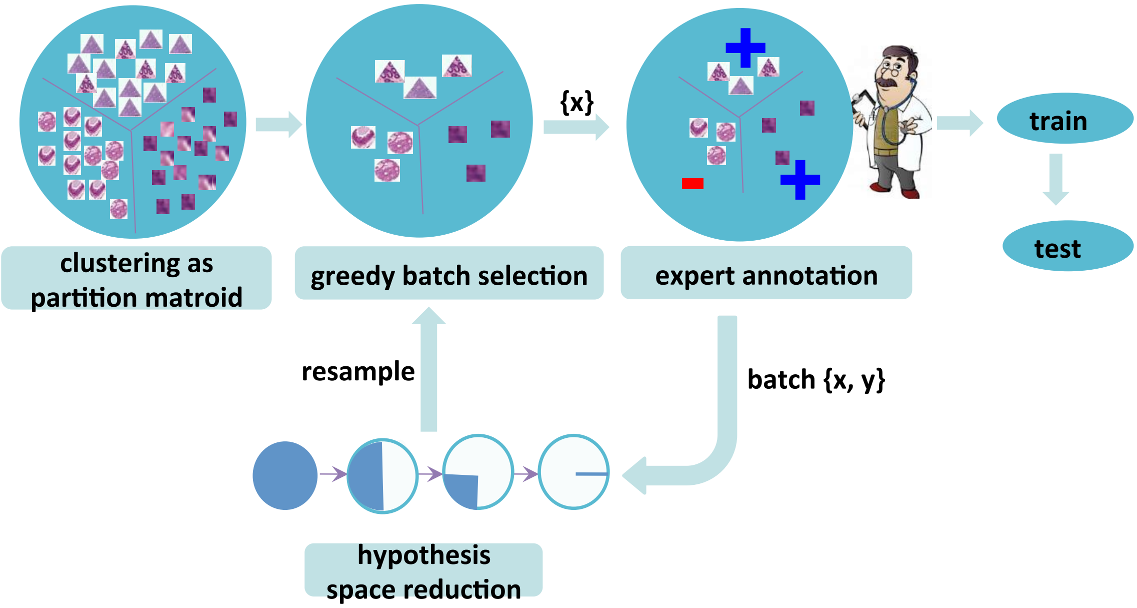


Figure 2.1: Algorithm framework for active selection

2.4 Proof of Theorem 1

Lemma 1. *(from Lemma 3 in [37]) Let* $\mathcal{V} = \{1, ..., n\}$*,* $\mathcal{Y}$ *be finite sets;* $f : 2^{\mathcal{V}\times\mathcal{Y}} \to \mathbf{N}$ *monotonic and submodular, and* $P(\mathbf{Y}_{\mathcal{V}})$ *such that* (f, P) *is adaptive submodular. Let* $\mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_m \subseteq \mathcal{V}$*, and define for* $i \in \{1, ..., m\}$*,* $Z_i = [\mathbf{Y}_{j_1,...,j_l}]$ *where* $\mathcal{A}_i = \{j_1, ..., j_l\}$*, and* l *is a constant integer. Let* $\mathcal{W} = \{1, ..., m\}$ *and* $Q(\mathbf{Z_W})$ *be the distribution over* $Z_1, Z_2, ..., Z_m$ *induced by* P*. Let* $\mathcal{Y}' = \bigcup_{i\in\mathcal{W}} range(Z_i)$*. Define the function*

$$\gamma : 2^{\mathcal{W}\times\mathcal{Y}'} \to 2^{\mathcal{V}\times\mathcal{Y}}, \gamma(\{(a_1, \mathbf{z}_1), ..., (a_t, \mathbf{z}_t)\}) = \bigcup_{j=1}^{t} \{(i, o) : i \in A_j, o = [\mathbf{z}_j]_i\} \tag{2.6}$$

and define $g : 2^{\mathcal{W}\times\mathcal{Y}'} \to \mathbf{N}$ *by* $g(\mathcal{S}) = f(\gamma(\mathcal{S}))$*. Then* g *is submodular, and* (g, Q) *is adaptive submodular.*

Lemma 2. *(from Theorem 7 in [39]) For an adaptive monotonic submodular function* $f : 2^E \times \mathcal{Y}^E \to \mathbf{R}_{\leq 0}$ *and a p-independent system* $(E, \mathcal{I})$*. Fix a policy* π *which is* $\alpha-$ *approximate greedy with respect to* f *for constraint* $\mathcal{I}$*. Then* π *yields an* $\frac{\alpha}{p+\alpha}$ *approximation, meaning*

$$f_{avg}(\pi) \leq (\frac{\alpha}{p+\alpha}) \max_{feasible \pi^*} f_{avg}(\pi^*) \tag{2.7}$$

where π^* is feasible iff $E(\pi^*, \Phi) \in \mathcal{I}$ for all Φ.

Below is the proof of theorem 1. We adopt the similar proving technique as [37]. Basically, we transfer from a batch mode policy for the original problem to a sequential policy to the superset of the original problem instance.

Proof

Suppose we are given $f, \mathcal{V}, \mathcal{Y}$ and P satisfying Lemma 1. Also we are given a set of disjoint ground sets $\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_n$ partitioning $\mathcal{V}$, therefore it gives a partition matroid constraint $\mathcal{M}$. Let $\{S_1, ...S_M\}$ are the superset of all possible size k subsets, where $M = \binom{n}{k}$. According to Lemma 1, an induced problem instance for $\{S_1, ...S_M\}$ is (g, Q), where Q is the distribution of the observations for all possible size k subsets $\{S_1, ...S_M\}$. From Lemma 1, (g, Q) is adaptive submodular. For every batch mode policy for problem (f, P) subject to $\mathcal{M}$, there is a corresponding sequential policy for problem (g, Q) subject to $\mathcal{M}$.

According to Theorem 11 in [38], the greedy policy π satisfies

$$cost_{avg}(\pi) \leq OPT_{avg,k}(ln(|\mathcal{H}| - 1) + 1) \tag{2.8}$$

where $|\mathcal{H}|$ is the size of the hypothesis space, and $OPT_{avg,k}$ is the optimal policy for size k batch selection. However, policy π is assuming that within each batch the seelection is optimal. The proposed algorithm BGAL-PGM greedily select samples within each batch. Notice that a partition matroid constraint is a special case of p-independent systemm when $p = 1$. So According to Lemma 2, the policy adopting BGAL-PMC maximizes function g with a $\frac{1}{2}$-approximation to the optimal policy. Therefore, we prove that

$$cost_{avg}(\pi_{BGAL-PMC}) \leq OPT_{avg,k} \times 2 \times (ln(|\mathcal{H}| - 1) + 1) \tag{2.9}$$

as stated in Theorem 1. ■

2.5 Experiments

In this section, we discussed details of our experiments and results on a breast microscopic tissue images, and compares selection accuracy, efficiency and diversity with state-of-the-art active selection methods.

Experimental Settings: Our experiments are conducted on a large database of histopathological images from breast microscopic tissues. This database contains 2377 images, sampled from 657 larger region-of-interests images, which are gathered from hundreds of patients. Each image is labeled as benign category (usual ductal hyperplasia (UDH)) or actionable category (atypical ductal hyperplasia (ADH) and ductal carcinoma in situ (DCIS)) by pathologists, which are development procedures from a normal terminal duct-lobular unit to an invasive cancer.

Classifying these two categories is an important clinical problem since the therapy planning and management relies on the diagnosis of UDH and ADH/DCIS. It is also very challenging due to the subtle differences between categories.

Sift feature descriptors from each image and quantized into cluster centers using bag-of-words. Then, we represent each feature by a 10000-dimensional feature vector, where 10000 corresponds to the number of clusters. The final histogram representation according to 10000 clustering centers. So each image is represented by 10000-dimensional bag-of-words sift histogram.

We randomly split the dataset into 50% training to actively select candidate images and 50% testing to test the learned classifier. We also ensure that images for a particular patient are either in the training set or in the testing set. We randomly split 10 times and the average performance is reported.

Five active learning methods are compared, i.e., Random Selection, Min Margin [35], Fbatch [34], BMDR [42], and BatchGreedy [37]. Note that the Random Selection is equivalent to the passive learning setting.

In our method, we partitioned the dataset into 20 disjoint subsets using both the structured information and image texture features by K-means. For fair comparison, we use SVM classifier for all methods, with the same parameters tuned via five-fold cross validation. We set batch size at 5 throughout the experiments. Two positive images and two negative images are randomly selected for initialization. The size of the hypothesis set is set at 300, which is empirically large enough in our experiments. All experiments are conducted on a 2.80GHz i7 CPU with 16 cores and 16G RAM in Matlab implementation.

Results: Fig. 2.2 shows the classifier learning curves as selected samples increase. Not

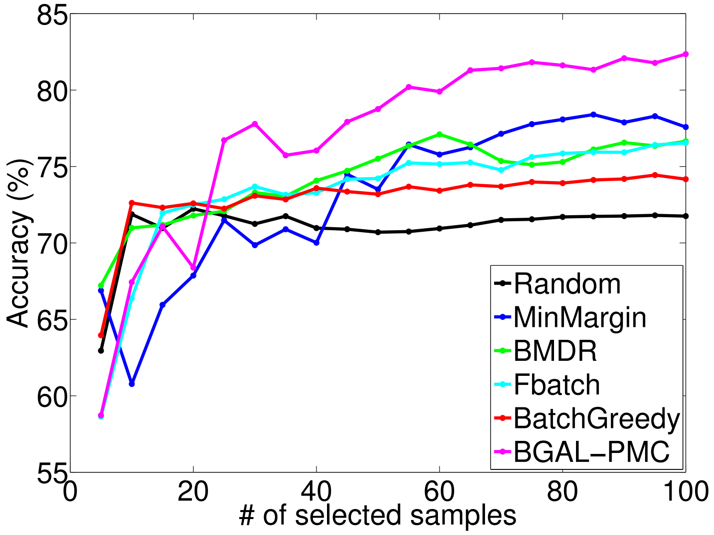


Figure 2.2: Learning curves of the proposed BGAL-PMC and other 5 methods on the breast microscopic tissues image dataset. X-axis is the number of selected images while Y-axis is the accuracy as the number of selected training images increases. BGAL-PMC (the pink curve) outperforms the other 5 methods significantly;

surprisingly, all five active learning methods perform better than random selection, which manifest the effectiveness of active learning. In particular, the proposed BGAL-PMC performs significantly better than all other four active learning methods. Min Margin method as a classical active learning baseline is the second-best in our experiments. Although Fbatch, BMDR and BatchGreedy perform well in the first 20 selected samples, the improvement of their accuracy is less substantial when more batches are selected. The reason is that all other methods do not take the information of clusters into consideration. Therefore, their selected images may include information redundancy, which downgrades their performances. On the other hand, trivially using cluster information cannot achieve accuracy either. We tested sampling from randomly-chosen distinct clusters, as an alternative baseline, and this is still significantly worse than our proposed method. It proves the efficacy of unifying the prtition matroid constraint with active learning. With less than 5% data labeled, our method achieves 83% prediction accuracy. This accuracy is at least 6% higher than all compared methods. In fact, when 80% data is labeled, the prediction accuracy is 87%, which is merely 4% higher than our method but use much more labeled samples than us. Therefore, this scheme considerably reduces the label effort from pathologists, without significantly sacrificing the accuracy.

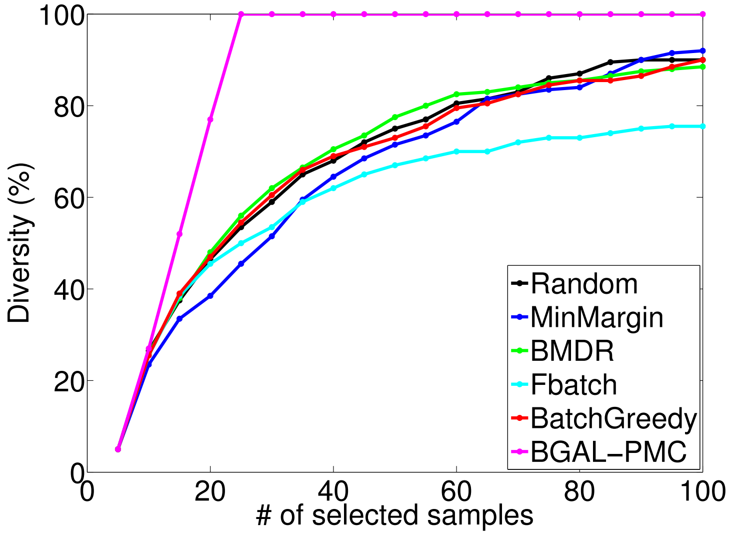


Figure 2.3: The diversity curves of all 6 methods. X-axis is the number of selected images while Y-axis is the diversity of the selected set as the number of selected images increases. Note that the diversity here is defined as the percentage of partitioning clusters being covered.

Table 2.1: Comparison of the average time to select a single batch of images for 5 active learning algorithms (batch size=5)

Method	Time(s)
MinMargin[35]	3.13
BMDR[42]	17.63
FBatch[34]	128.13
BatchGreedy[37]	1.97
BGAL-PMC	**1.98**

We further investigated the diversity of all methods, as shown in Fig. 2.3. The diversity here is defined as the coverage rate of the clusters. Since we enforce the partition matroid constraint explicitly, BGAL-PMC covered all the clusters in much fewer iterations than other methods. Fig. 2.4 is one selected batch using our proposed method, to show the diversity of our selections visually. We also compared the running time, as shown in Table 2.1.

Aside of accuracy, the selection efficiency is a very important metric for active learning, as the selection process is employed multiple times in a batched sequential fashion. Ideally, the batch selection speed should be instance so the annotators do not have to wait too long to get the next batch annotation tasks.

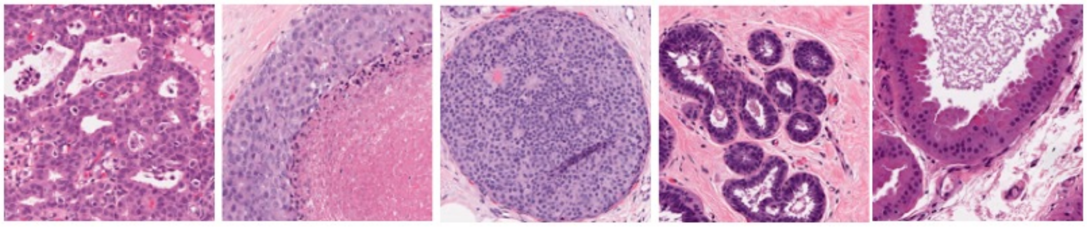


Figure 2.4: One example batch of selected images using our proposed method. The first 3 are actionable, and the last 2 are benign. 5 images are selected from distinct clusters.

In our experiments, BatchGreedy and BGAL-PMC are much more efficient than other active learning algorithms. BatchGreedy is slightly faster than ours (1.97s vs. 1.98s), both of which are negligible in the practical use of active learning.

2.6 Conclusion

In this paper, we proposed a novel batch mode active learning approach which leverages the structured information of annotated histopathological images. We formulated the batch mode active learning problem as a submodular function maximization problem with a partition matroid constraint, which prompts us to design an efficient greedy algorithm for approximate optimization. We further provided a theoretic bound characterizing the quality of the solution achieved by our algorithm. We compared the proposed active learning approach against several state-of-the-art active learning methods on a large database of histopathological images, and demonstrated the performance superiority of our approach. The spirit of our active learning method capitalizing on submodular optimization is generic, and can thus be applicable to other problems in medical image analysis.

The method can be potentially extended to wider settings in future work. First, the current active selection formulation is still developed and evaluated on SVM classifiers. Recently deep learning models have already exhibited strong potentials in medical imaging [43, 44]. Typically neural network requires much larger dataset than SVM type classifiers. It would be helpful to adapt the proposed active selection with convolutional neural networks. The difficult part is how to estimate the uncertainty space as we've done in this work using the hit and run sampler.

Other than active selection, another promising direction to deal with small training set is to explore recent proposed low shot learning methods [45, 46]. Hopefully, the low shot learning paradigms can better utilize the external meta knowledge in histopatholgical images and accelerate training with a modest labeled set.

Chapter 3

Semantic Amodal Segmentation and Active Generation

3.1 Introduction

In recent years, visual recognition tasks such as image classification [10, 13], object detection [47, 48, 49, 50], edge detection [51, 52, 53], and semantic segmentation [54, 55, 56] have witnessed dramatic progress. This has been driven by the availability of large scale image datasets [57, 3, 18] coupled with a renaissance in deep learning techniques with massive model capacity [10, 11, 12, 13]. Given the pace of recent advances, one may conjecture that techniques for many of these tasks will rapidly approach human levels of performance. Indeed, preliminary evidence exists this is already the case for ImageNet classification [58].

In this work we ask: what are the next set of challenges in visual recognition? What capabilities do we expect future visual recognition systems to possess?

We take our inspiration from the study of the human visual system. A remarkable property of human perception is the ease with which our visual system interpolates information not directly visible in an image [22]. A particularly prominent example of this, and one on which we focus, is *amodal perception*: the phenomenon of perceiving the whole of a physical structure when only a portion of it is visible [21, 22, 59]. Humans can readily perceive partially occluded objects and guess at their true shape.

To encourage the study of machine vision systems with similar capabilities, we ask human subjects to annotate regions in images *amodally*. Specifically, annotators are asked to mark the full extent of each region, not just the visible pixels. Annotators outline and name all salient regions in the image and specify a partial depth order. The result is a rich scene structure, including visible and occluded portions of each region, figure-ground edge information, semantic labels, and object overlap. See Figure 3.1.

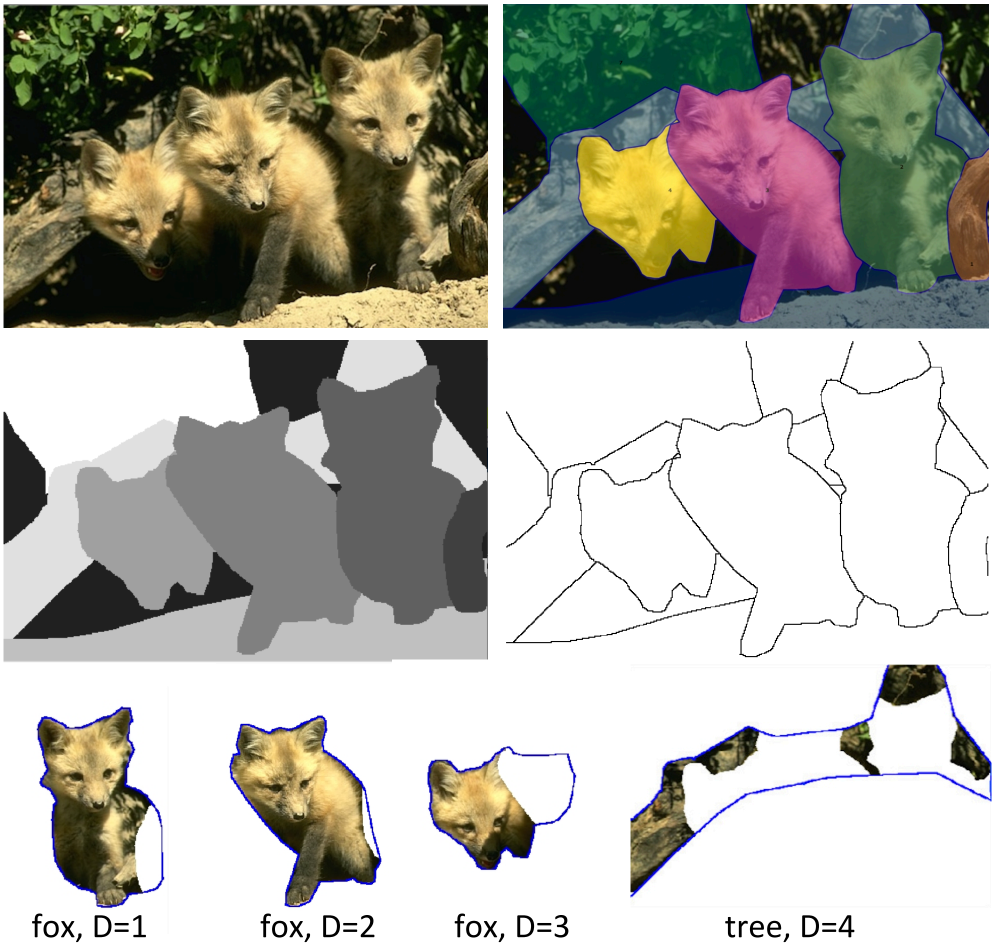


Figure 3.1: Example of *Semantic Amodal Segmentation*. Given an image (top-left), annotators segment each region (top-right) and specify a partial depth order (middle-left). From this, visible edges can be obtained (middle-right) along with figure-ground assignment for each edge (not shown). All regions are annotated *amodally*: the full extent of each region is marked, not just the visible pixels. Four annotated regions along with their semantic label and depth order are shown (bottom); note that both visible and occluded portions of each region are annotated.

An astute reader may ask: is amodal segmentation even a well-posed annotation task? More precisely, will multiple annotators agree on the annotation of a given image?

To study these questions, we asked multiple annotators to label all 500 images in the BSDS dataset [51]. We designed the annotation task in a manner that encouraged annotators to consider object relationships and reason about scene geometry. This resulted in agreement between annotators that is surprisingly strong. In particular, our data has higher region and edge consistency than the original BSDS labels. Likewise, annotators tend to agree on the amodal completions. We report a thorough study of human performance on amodal segmentation using this data and also use it to train and evaluate state-of-the-art edge detectors.

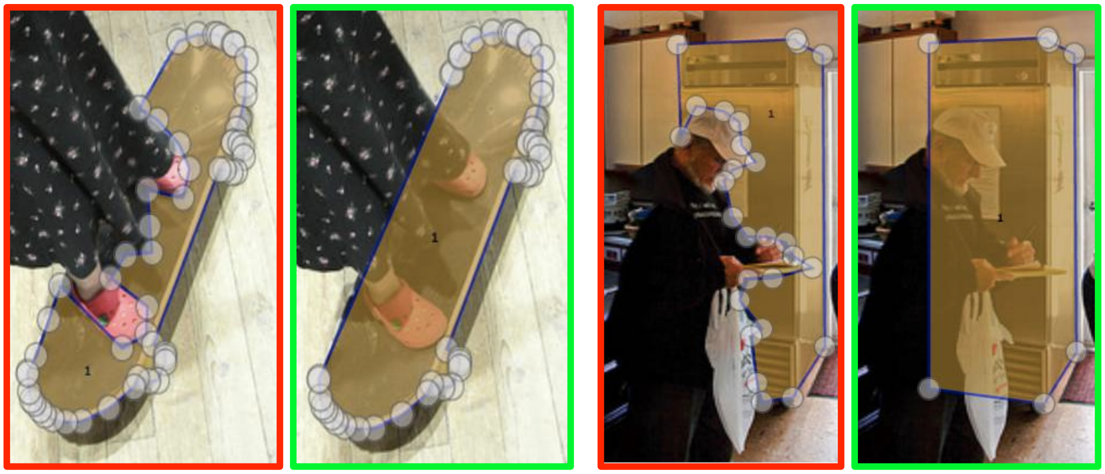


Figure 3.2: *Amodal versus modal segmentation*: The left (red frame) of each image pair shows the modal segmentation of a region (visible pixels only) while the right (green frame) shows the amodal segmentation (visible and interpolated region). In this work we ask annotators to segment regions amodally. Note that the amodal segments have simpler shapes than the modal segments.

In addition to the BSDS data, we annotate a second larger semantic amodal segmentation dataset using 5000 images from COCO [18]. To achieve this scale, each image in COCO was annotated with just one expert annotator plus strict quality control. The dataset is divided into 2500/1250/1250 images for train/val/test, respectively. We introduce novel evaluation metrics for measuring amodal segment quality and pairwise depth-ordering of region segments. We do not currently use the semantic labels for evaluation as they come from an open vocabulary; nevertheless, we show that collecting these labels is key for obtaining high-quality amodal annotations. All train and val annotations along with evaluation code will be publicly released.

Finally, the larger collection of annotations on COCO allows us to train strong baselines for amodal segmentation and depth ordering. To perform amodal segmentation, we extend recent modal segmentation algorithms [60, 61] to the amodal setting. We train two baselines: first, we train a deep net to directly predict amodal masks, second, motivated by [62], we train a model that takes a modal mask and attempts to expand it. Both variants achieve large gains over their modal counterparts, especially under heavy occlusion. We also experiment with deep nets for depth ordering and achieve accuracy over 80%.

Our challenging new dataset, metrics, and strong baselines define concrete new challenges for the community and we hope that they will help spur novel research directions.

3.1.1 Related Work

Amodal perception [21] has been studied extensively in the psychophysics literature, for a review see [59, 22]. However, amodal completion, along with many of the principles of perceptual grouping, are often demonstrated via simple illustrative examples such as the famous Kanizsa's triangle [21]. To our knowledge, there is no large scale dataset of amodally segmented natural images.

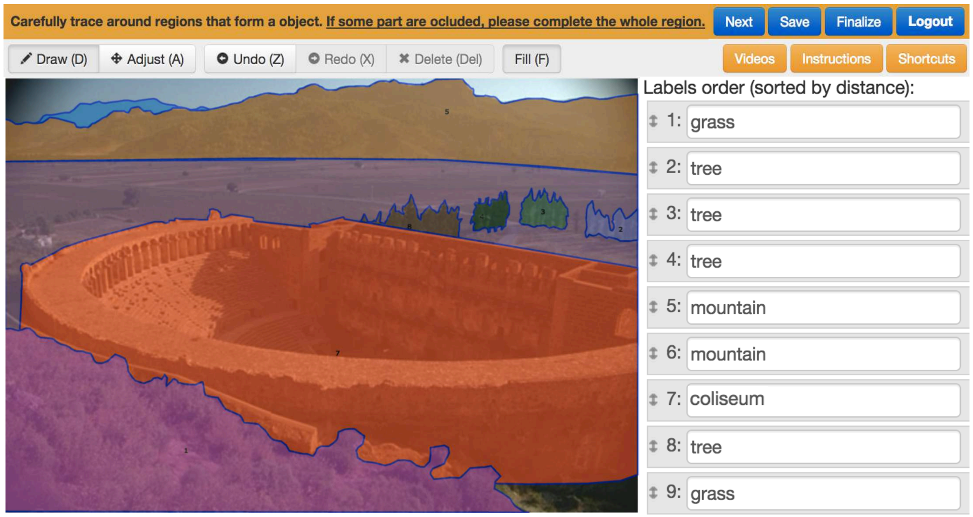
Modal segmentation[1] datasets are more common. The most well known of these is the BSDS dataset [51], which has been used extensively for training and evaluating edge detection [63, 52, 53] and segmentation algorithms [51]. BSDS was later extended with figure-ground edge labels [64]. A drawback of this annotation style is that it lacks clear guidelines, resulting in inconsistencies between annotators.

An alternative to unrestricted modal segmentation is *semantic segmentation* [54, 65, 66], where each image pixel is assigned a unique label from a fixed category set (for instance, grass, sky, person). Such datasets have higher consistency than BSDS. However, the label set is typically small, individual objects are not delineated, and the annotations are modal. Notable exception are the StreetScenes dataset [67], which contains a few categories which are labeled amodally, and PASCAL context [68], which uses a large category set.

The closest dataset to ours is the hierarchical scenes dataset from Maire *et al* [69], which aims to captures occlusion, figure-ground ordering, and object-part relations. The dataset consists of incredibly rich and detailed annotations for 100 images. Our dataset shares some similarities but is easier to collect, allowing us to scale. Likewise, Visual Genome [70] also provides rich annotations, including depth ordering, but does not include segmentation.

Compared to *object detection* datasets [57, 3, 18], our annotation is dense, amodal, and covers both objects and regions. Related datasets such as Sun [71] have objects annotated modally. LabelMe [72] does have some amodal annotations but not consistently annotated. Only for pedestrian detection [73] are objects often annotated amodally (with both visible and amodal bounding boxes).

[1] In an abuse of terminology, we use *modal segmentation* to refer to an annotation of only the visible portions of a region. This lets us easily differentiate it from *amodal segmentation* (full region extent annotated).



Carefully trace around regions that form a object. If some part are ocluded, please complete the whole region.

Labels order (sorted by distance):

1: grass
2: tree
3: tree
4: tree
5: mountain
6: mountain
7: coliseum
8: tree
9: grass

Figure 3.3: A screenshot of our annotation tool for semantic amodal segmentation (adopted from the Open Surfaces tool [4]).

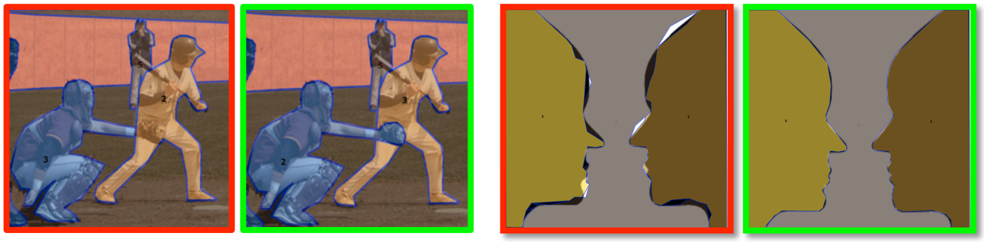
We note that our annotation scheme subsumes modal segmentation [51], edge detection [51], and figure-ground edge labeling [64]. As our COCO annotations (5000 images) are an order of magnitude larger than BSDS (500 images) [51], the previous de-facto dataset for these tasks, we expect our data to be quite useful for these classic tasks.

Finally there has been some algorithmic work on amodal completion [74, 75, 76, 77] and depth ordering [78, 79]. Of particular interest, Ke *et al* [62] recently proposed a general approach for amodal segmentation that serves as the foundation for one of our baselines (see §3.4). Most existing recognition systems, however, operate on a per-patch or per-window basis, or with a limited receptive field, including for object detection [47, 48, 49], edge detection [63, 52, 53], and semantic segmentation [54, 55, 56]. Our dataset will present challenges to such methods as amodal segmentation requires reasoning about object interactions.

3.2 Dataset Details

3.2.1 Annotation Details

For our semantic amodal segmentation, we extend the Open Surfaces annotation tool from Bell *et al* [4], see Figure 3.3. The original tool allows for labeling multiple regions in an image by specifying a closed polygon for each; the same tool was also adopted for annotation of COCO [18]. We extend the tool in a number of ways, including for region ordering, naming,



(a) depth ordering

(b) edge sharing

Figure 3.4: (a) We ask annotators to arrange region depth order. The right panel gives a correct depth order of the two people in the foreground while in the left panel the order is reversed. (b) Shared region edges must be marked to avoid duplicate edges. Unlike regular edges, shared edges do not have a figure-ground side.

and improved editing. For full details, including handling of corner cases, we refer readers to the supplementary. We will open-source the updated tool.

We found four guidelines to be key for obtaining high-quality and consistent annotations: (1) only semantically meaningful regions should be annotated, (2) images should be annotated densely, (3) all regions should be ordered in depth, and (4) shared region boundaries should be marked. These guidelines encouraged annotators to consider object relationships and reason about scene geometry, and have proven to be effective in practice as we show in §3.3.

(1) Semantic annotation: Annotators are asked to name all annotated regions. Perceptually, the fact that a segment can be named implies that it has a well-defined prototype and corresponds to a semantically meaningful region. This criterion leads to a natural constraint on the granularity of the annotation: material boundaries and object parts (*i.e*interior edges) should not be annotated if they are not namable. Moreover, under this constraint, annotators are more likely to have a consistent prior on the occluded part of a region. In practice, we found that enforcing region naming led to more consistent and higher-quality amodal annotations.

(2) Dense annotation: Annotators are asked to label an image densely, in particular all foreground object over a minimum size (600 pixels) should be labeled. Of particular importance is that if an annotated region is occluded, the occluder should also be annotated. When all foreground regions are annotated and a depth order specified, the visible and occluded portions of each annotated region are determined, as are the visible and hidden edges.

(3) Depth ordering: Annotators are asked to specify the relative depth order of all regions, see Figure 3.4a. In particular, for two overlapping regions, the occluder should precede the occludee. In ambiguous cases, the depth order is specified so that edges are correctly 'rendered'

(*e.g*, eyes go in front of the face). For non-overlapping regions any depth order is acceptable. Depth ordering encourages annotators to reason about scene geometry, including occlusion, and therefore improves the quality of amodal annotation.

(4) Edge sharing: When one region occludes another, the figure-ground relation is clear, and an edge separating the regions belongs to the foreground region. However, when two regions are adjacent, an edge is shared and has no figure-ground side. We require annotators to explicitly mark shared edges, thus avoiding duplicate edges, see Figure 3.4b. As with the other criteria, this encourages annotators to reason about object interactions and scene geometry.

For our task we adopt the Open Surfaces [4] annotation tool developed by Bell *et al*for material segmentation. The original tool allows for labeling multiple regions in an image by specifying a closed polygon for each region. The same tool was also adopted for annotation of COCO [18]. The interface is simple and intuitive.

We extend the tool in a number of ways to support semantic amodal segmentation and facilitate annotation (see Figure 3.3). We have added the following features:

Depth ordering: An ordered list next to the image indicates the segment depth order. Annotators can rearrange the order by dragging items up and down in this list (see Figure 3.3). Moreover, visual feedback is given about depth order through the region fill overlaid on the image, allowing annotators to quickly determine the correct order, see Fig. 3.4a.

Semantic annotation: The same list used for specifying depth ordering is also used for naming each segment. The annotators enter free-form text for the segment names. All segments must be named for an annotation to be complete.

Edge sharing: We extended polygon annotation to allow for 'snapping' of a new polygon vertex to the closest existing polygon edge or vertex. This mechanism allows for easily annotating shared edges, see Figure 3.4b.

Polygon editing: Finally, we add control for adding and removing vertices while editing existing polygons.

The code for the modified annotation tool is released on github.

Although our annotation instructions are sufficient for most images, the following cases require special treatment:

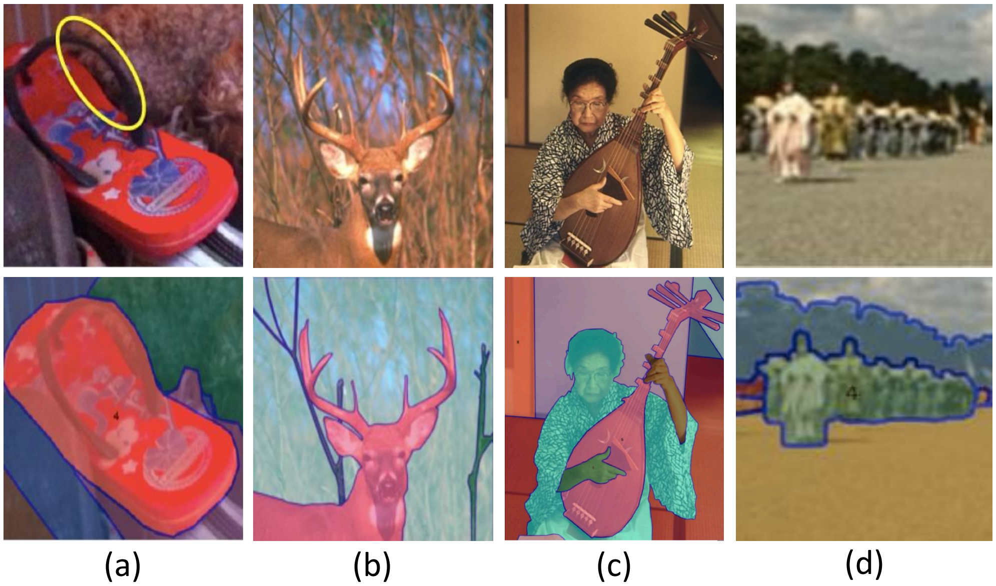


Figure 3.5: A few corner cases in annotation: (a) Annotators only label exterior boundaries, leaving holes as part of the region. (b) Annotators only label the most salient objects in blurry and cluttered backgrounds. (c) For regions with intertwined depth ordering, annotators are instructed to pick the depth ordering which is ‘least wrong’ or to annotate object parts. (d) Annotators can mark a group of similar objects using a single segment.

Regions with holes: We only annotate the exterior region boundaries, therefore each region is represented by a single segment. Holes are ignored (Figure 3.5a).

Background objects: For blurry objects in the background, annotators are asked to label only the most salient objects individually, rather than every detail (Figure 3.5b).

Intertwined depth: Two regions might not have a valid depth ordering (e.g., the woman holding the musical instrument in Figure 3.5c). In such cases we instruct the annotators to pick the depth ordering which is ‘least wrong’. In extreme cases, annotators may label parts of an object so that visibility and occlusion information are correctly specified (e.g., by marking the woman’s hands in Figure 3.5c).

Groups: For groups of similar objects (e.g. a crowd of people or bunch of bananas), annotators are instructed to mark a single region enclosing the entire group (Figure 3.5d). Note that groups are often perceived as a single visual entity, so this form of annotation is quite natural.

Truncation: Segments must be fully contained within the image boundaries, *i.e*regions extending beyond the image are *not* annotated amodally (annotation outside the image is particularly challenging as the occluder is not visible).

Rather than rely on a crowdsourcing platform, we utilize a pool of expert workers to perform all annotations. This allows us to specify more complex instructions than is typically possible with crowdsourcing platforms and iterate with workers until annotations reach a sufficient quality. We note, however, that if necessary we could move our annotation onto a crowdsourcing platform. This would require splitting a single image annotation into multiple separate and possibly redundant tasks, similarly to how annotation was performed on COCO [18].

While every image in BSDS is annotated by multiple workers, we also monitor individual worker quality. We differentiate between *obvious errors*, which we ask workers to correct, and *subjective judgments*, which differ between individuals and for which a clear criterion is harder to define. Each image annotation is manually checked, and obvious errors are sent back to the annotators for improvement. Subjective judgements, on the other hand, are left to annotators' discretion. Checking annotations for errors is a quick and lightweight process (and can also be crowdsourced).

Common obvious errors include incorrect depth ordering, missing foreground objects, regions annotated modally, and low quality polygons. These errors all explicitly violate the annotation instructions and are easily identifiable. On the other hand, common subjective judgements include the semantic label used, the exact location of hidden edges, and whether a region was sufficiently salient to warrant annotation. As mentioned, annotators are asked to correct obvious errors but not subjective judgements.

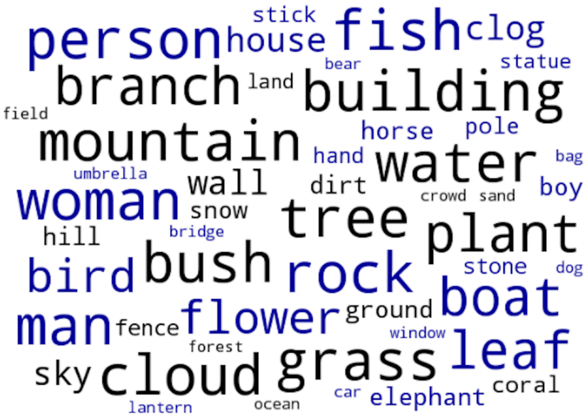
3.2.2 Dataset Statistics

The analysis in this section is primarily based on the 500 images in the BSDS dataset [51], which has been used extensively for edge detection and modal segmentation. Annotating the same images amodally allows us to compare our proposed annotations to the original annotations. While all following analysis is based on these images, we note that the statistics of our annotations on COCO [18] are similar (they differ slightly as COCO images are more cluttered).

Figure 3.6a summarizes the statistics of our data. Each of the 500 BSDS images was annotated independently by 5 to 7 annotators. On average each image annotation consists of 7.3 labeled regions, and each region polygon consists of 64 points. About 84% of image pixels are

	BSDS	COCO
ann/image	5-7	1
regions/ann	7.3	9.2
points/region	64	46
pixel coverage	84%	69%
occlusion rate	62%	61%
occ/region	21%	31%
time/polygon	68s	41s
time/region	2m	2m
time/ann	15m	18m

(a) dataset summary statistics



(b) most common semantic labels

Figure 3.6: (a) Dataset summary statistics on BSDS and COCO. COCO images are more cluttered, leading to some differences in statistics (*e.g.*higher regions/ann and lower pixel coverage). (b) The top 50 semantic labels in our BSDS annotations. Roughly speaking, the blue words indicate 'things' (person, fish, flower) while the black words indicate 'stuff' (grass, cloud, water).

covered by at least one region polygon. Of all regions, 62% are partially occluded and average occlusion is 21%.

Annotating a single region takes ~2 minutes. Of this, half the time is spent on the initial polygon and the rest on naming, depth ordering, and polygon refinement. Annotating an entire image takes ~15m, although this varies based on image complexity and annotator skill.

Semantic labels: Figure 3.6b shows the top 50 semantic labels in our data with word size indicating region frequency. The labels give insight into the regions being labeled as well as the granularity of the annotation. Most labels correspond to basic level categories and refer to entire objects (not object parts). Using common terminology [80, 81], we explicitly classify the labels into two categories: 'things' and 'stuff', where a 'thing' is an object with a canonical shape (person, fish, flower) while 'stuff' has a consistent visual appearance but can be of arbitrary spatial extent (grass, cloud, water). Both 'thing' and 'stuff' labels are prevalent in our data (stuff composes about a quarter of our regions).

Shape complexity: One important property of amodal segments is that they tend to have a relatively simple shape compared to modal segments that is independent of scene geometry and occlusion patterns (see Figure 3.2). We verify this observation with the following two statistics, shape *convexity* and *simplicity*, defined on a segment S:

	BSDS			**COCO**	
	original	modal	amodal	modal	amodal
simplicity	.801	.718	.834	.746	.856
convexity	.664	.616	.643	.658	.685
density	1.80%	1.57%	1.97%	1.71%	2.10%

Table 3.1: Comparison of shape and edge statistics between modal and amodal segments on BSDS and COCO. Amodal segments tend to have a relatively simpler shape that is independent of scene geometry and occlusion patterns (see also Figure 3.2). Interestingly, the original BSDS annotations (first column) are even simpler than our modal annotations. Finally the last row reports edge density.

$$convexity(S) = \frac{Area(S))}{Area(ConvexHull(S))} \tag{3.1}$$

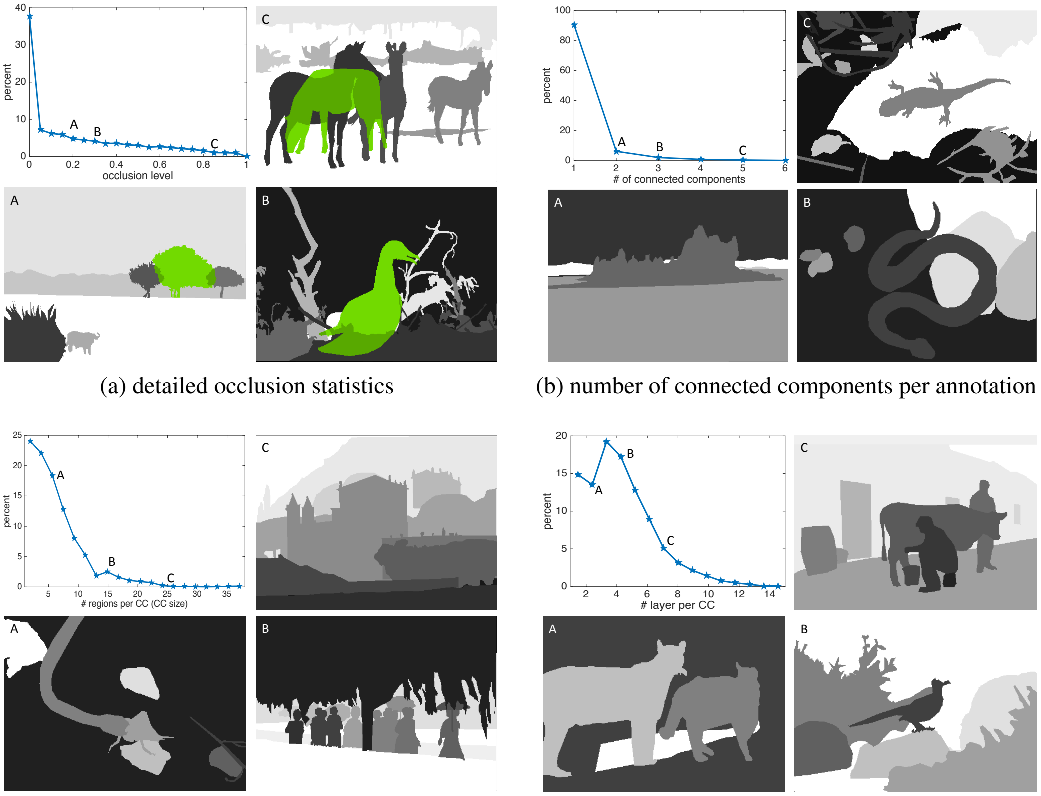
$$simplicity(S) = \frac{\sqrt{4\pi * Area(S)}}{Perimeter(S)} \tag{3.2}$$

A segment with a large convexity and simplicity value means it is simple (and both metrics achieve their maximum value of 1.0 for a circle). Table 3.1 shows that amodal regions are indeed simpler than modal ones, which verifies our hypothesis. Due to their simplicity, amodal regions can actually be more efficient to label than modal regions.

We also compare to the original (modal) BSDS annotations (first column of Table 3.1). Interestingly, the original BSDS annotations are even simpler than our modal annotations. Qualitatively it appears that the original annotators had a bias for simpler shapes and smoother boundaries.

Edge density: The last row of Table 3.1 shows that our dataset has fewer visible edges marked than the original BSDS annotation (edge density is the percentage of image pixels that are edge pixels). This is necessarily the case as material boundaries and object parts (i.e. interior edges) are not annotated in our data. Note that in §3.3 we demonstrate that although our edge maps are slightly less dense, they can be used to effectively train state-of-the-art edge detectors.

Occlusion: Figure 3.7a shows a histogram of occlusion level (defined as the fraction of region area that is occluded). Most regions are slightly occluded, while a small portion of regions are heavily occluded. We additionally display 3 occluded examples at different occlusion levels.



(a) detailed occlusion statistics

(b) number of connected components per annotation

(c) connected components size

(d) number of depth layers per connected component

Figure 3.7: Detailed dataset statistics. See text for details.

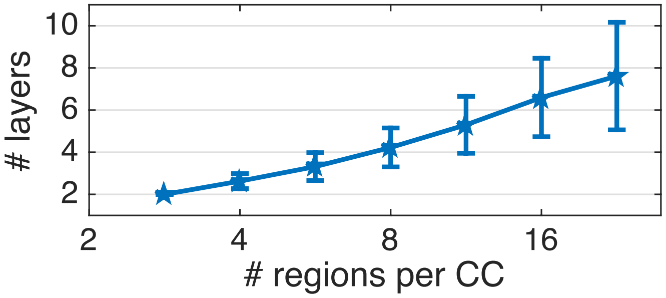


Figure 3.8: The minimum number of depth layers necessary to represent a connected component (CC). See text for details.

Scene complexity: With the help of depth ordering, we can represent regions using a Directed Acyclic Graph (DAG). Specifically, we draw a directed edge from region R_1 to region R_2 if R_1 spatially overlaps R_2 and R_1 precedes R_2 in depth ordering. Given the DAG corresponding to an image annotation, a few quantities can be analyzed.

First, Figure 3.7b shows the number of connected components (CC) per DAG. Most annotations have only one CC, as shown in example A. If regions are scattered and disconnected an image will have more CC's, as in B and C.

The size of a CC measures how many regions are mutually overlapped, which in turns gives an implicit measure of scene complexity. Figure 3.7c shows a number of examples. More complex scenes (examples B and C) have large CC's.

Finally, the longest directed path of any CC in a DAG characterizes the minimum number of depth layers required to properly order all regions in the DAG. Note that the number of depth layers is often smaller than the size of a CC: e.g. a large CC with numerous non-overlapping foreground objects and a single common background only requires two depth layers. Figure 3.7d shows the distribution of number of depth layers needed per CC. Most components require only a few depth layers although some are far more complex.

Figure 3.8 further investigates the correlation between CC size and the minimum number of depth layers necessary to order all regions. We observe that the number of depth layers necessary appears to grow logarithmically with CC size.

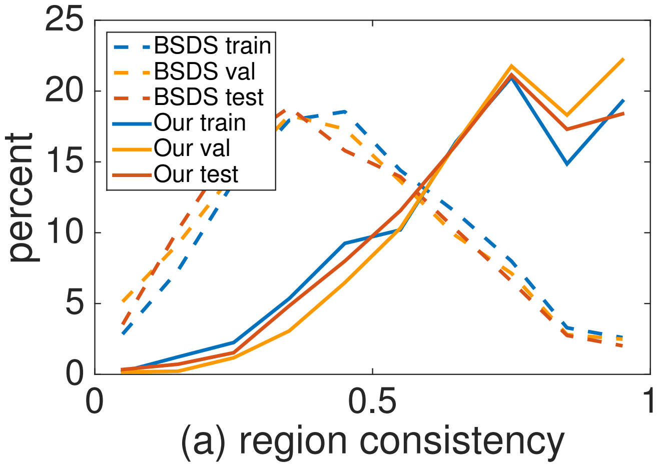


Figure 3.9: (a) Histogram of pairwise *region consistency* scores for the original *modal* BSDS annotations and our *amodal* regions.

3.3 Consistency

We next aim to show that semantic amodal segmentation is a well-posed annotation task. Specifically, we show that agreement between independent annotators is high. Consistency is a key property of any human-labeled dataset as it enables machine vision systems to learn a well defined concept. In the next two sub-sections we analyze our dataset's region and edge consistency on BSDS. As a baseline, we compare to the original (modal) BSDS annotations.

3.3.1 Region Consistency

To measure region consistency, we use Intersection over Union (IoU) to match regions. The IoU between two segments is the area of their intersection divided by the area of their union. We threshold IoU at 0.5 and use bipartite matching to match two sets of regions. We set each annotation as the ground truth in turn, and for every other annotator we compute precision (P) and recall (R) and summarize the result via the F measure: $F = 2PR/(P + R)$. For n annotators this yields $n(n - 1)$ F scores per image.

In Figure 3.9 we display a histogram of F scores for both the original BSDS *modal* annotations from [51] and the *amodal* annotations in our proposed dataset across each split of the dataset. The region consistency of our amodal regions is substantially higher than the consistency of the original modal regions: median of 0.723 versus 0.425. This is in spite of the fact that our amodal regions include both the visible and occluded portions of each region. We note

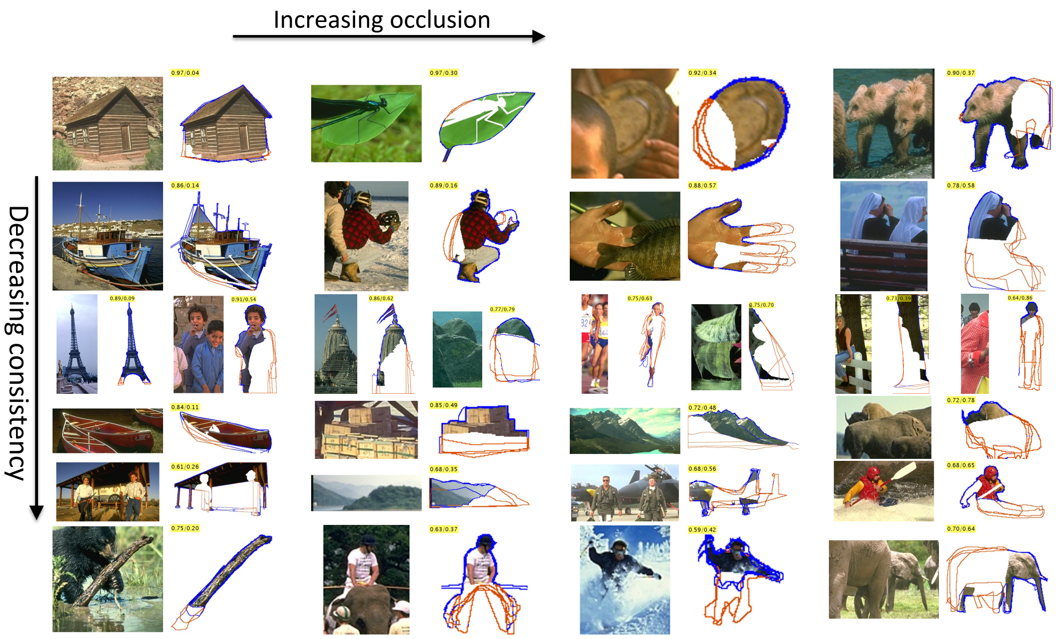


Figure 3.10: Visualizations of amodal region consistency. The blue edges are the visible edges, while the red edges are the occluded edges. Ground truth is determined by a single randomly chosen annotator. The region consistency score (average IoU score) and the occlusion rate are displayed. Examples are roughly sorted by decreasing consistency vertically and increasing occlusion horizontally.

that the modal region consistency of our annotations is 0.756, slightly higher than for amodal regions, as expected.

A number of factors contribute to the consistency of our regions. Most importantly, we gave more focused instructions to the annotators; specifically, we asked annotators to label only semantically meaningful regions and to label all foreground objects, see §3.2. Thus there was less inherent ambiguity in the task. Moreover, in modal segmentation, annotation level of detail substantially impacts region agreement.

Figure 3.10 shows qualitative examples of annotator agreement on individual regions for both visible and occluded portions of a region. Naturally, annotations are most consistent for regions with simple shapes and little occlusions. On the other hand, when the object is highly articulated and/or severely occluded, annotators tend to disagree more.

3.3.2 Edge Consistency

Given the amodal annotations and depth ordering, along with the constraint that all foreground regions are annotated, we can compute the set of visible image edges. We next verify the quality

	SE [52]			**HED** [53]		
train / test	ODS	AP	R50	ODS	AP	R50
bsds / bsds	.744	.795	.921	**.787**	.790	.855
ours / bsds	**.747**	**.802**	**.923**	.775	**.793**	**.868**
bsds / ours	.619	.603	.761	.657	**.578**	.697
ours / ours	**.630**	**.630**	**.785**	**.694**	.572	**.752**

Table 3.2: Cross-dataset performance of two state-of-the-art edge detectors. For SE, training on our dataset improves performance even when testing on the original BSDS edges. For HED, using the same train/test combination maximizes performance. These results indicate that our dataset is valid for edge detection.

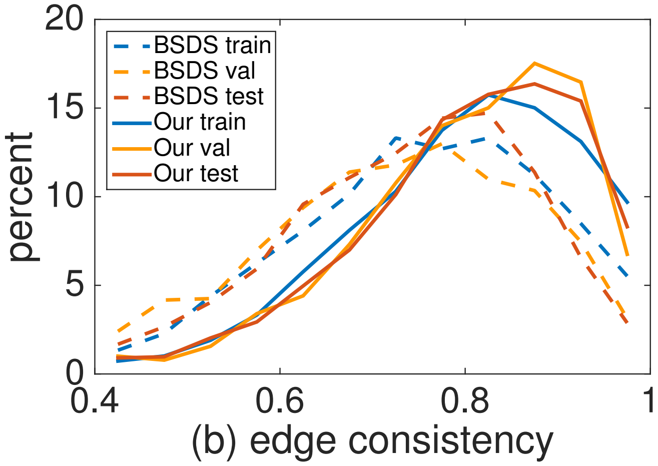


Figure 3.11: Histogram of pairwise *edge consistency* scores for visible edges.

of the obtained edge maps.

First, to measure edge consistency among annotators, we compute the F score between each pair of annotations, for details see [51]. Figure 3.11 shows the distribution of the boundary consistency scores. The edges in our amodal dataset are more consistent than edges in the original BSDS annotations (median consistency of 0.795 versus 0.728).

While our edges are more consistent, the edges are also less dense (see Table 3.1). To evaluate the efficacy of using our data for edge detection, we test two popular state-of-the-art edge detectors: structured edges (SE) [52] and the holistically-nested edge detector (HED) [53]. Results for cross-dataset generalization are shown in Table 3.2. For SE, training on our dataset improves performance even when testing on the original BSDS edges. For HED, using the same train/test combination maximizes performance by a slight margin. These results indicate that our dataset is valid for edge detection. Note, however, that our test set is substantially harder as only semantic boundaries are annotated.

Finally, we measure human performance. As in [51], we take one annotation as the detection and the union of the others as ground truth (note that this differs from the 1-vs-1 methodology used for Figure 3.11). On the original BSDS test set, precision/recall/F-Score are .92/.73/.81. Human performance is much higher on our test set, the scores are .98/.83/.90. Of particular interest, however, is the gap between human and machine. On the original BSDS annotations, HED achieves ODS of .79 while human F score is .81, leaving a gap of just .02. On our annotations, however, HED drops to .69 while human F score increases to .90. Thus, unlike the original annotations, our dataset leaves substantial room for improvement of the state-of-the-art.

3.4 Method and Evaluation

We aim to develop measures to quantify algorithm performance on our data. We begin by reiterating that our rich annotations subsume many classic grouping tasks, including modal segmentation, edge detection, and figure-ground edge labeling. Indeed, our COCO dataset (5000 images) is an order of magnitude larger than BSDS (500 images), the previous de-facto dataset for these tasks. We encourage researchers to use our data to study these classic tasks; for well-established metrics we refer readers to [51].

Here we propose two simple metrics that focus on the most salient aspect of our dataset: the amodal nature of the segmentations. Predicting amodal segments requires understanding object interaction and reasoning about occlusion. Specifically, we propose to evaluate: (1) amodal segment quality and (2) pairwise depth ordering between regions. We additionally define strong baselines for each task.

All experiments are on the 5000 COCO annotations, split into 2500/1250/1250 images for train/val/test, respectively. We evaluate on val and reserve the test images for use in a possible future challenge as is best practice on COCO.

	all regions				things only				stuff only			
	AR	AR$^\text{N}$	AR$^\text{P}$	AR$^\text{H}$	AR	AR$^\text{N}$	AR$^\text{P}$	AR$^\text{H}$	AR	AR$^\text{N}$	AR$^\text{P}$	AR$^\text{H}$
DeepMask [60]	.378	.456	.407	.248	.422	.470	.473	.279	.248	.367	.242	.199
SharpMask [61]	.396	.493	.428	.242	.448	.510	.501	.275	.246	.384	.243	.187
ExpandMask$^\text{S}$	.384	.460	.415	.256	.427	.474	.480	.284	.258	.374	.250	.212
AmodalMask$^\text{S}$	.395	.457	.424	.289	.435	.468	.487	.316	.282	.388	.268	.246
ExpandMask	.417	.480	.428	.327	.456	.495	.488	.351	.305	.387	.278	.289
AmodalMask	4̇34	.470	.460	.364	.458	**.479**	.498	.376	.366	.414	.365	.346
A^*modalMask	**.459**	**.501**	**.487**	**.383**	**.480**	.478	**.518**	**.388**	**.398**	**.445**	**.403**	**.375**

Table 3.3: Amodal segmentation quality on the COCO validation set for multiple baselines and under no, partial, and heavy occlusion (AR$^\text{N}$, AR$^\text{P}$, AR$^\text{H}$).

3.4.1 Amodal Segment Quality

Metrics: To evaluate amodal segments, we adopt a popular metric for object proposals: average recall (AR), proposed in [82] and used in the COCO challenges. To compute AR, segment recall is computed at multiple IoU thresholds (0.5-0.95), then averaged. To extend to our setting, we simply measure the IoU against the *amodal* masks. We measure AR for 1000 segments per image and also separately for things and stuff. Finally, we report AR for varying occlusion levels q: none (q=0), partial (0<q≤.25), and heavy (q>.25), comprising 39%, 31% and 30% of the data.

Baselines: We use *DeepMask* [60] and *SharpMask* [61], current state-of-the-art methods for *modal* class-agnostic object segmentation, as our first baselines. Next, inspired by Ke et al. [62] (which is not directly applicable to our setting), we propose a deep network we call *ExpandMask*. ExpandMask takes an image patch and a modal mask generated by SharpMask as input and outputs an amodal mask. Finally, we train a network, which we call *AmodalMask*, to directly predict amodal masks from image patches. ExpandMask and AmodalMask share an identical network architecture with SharpMask (except ExpandMask adds an extra input channel and uses a slightly larger input size). However, while AmodalMask is run convolutionally, ExpandMask is evaluated on top of SharpMask segments.

We use the DeepMask and SharpMask publicly available code and pre-trained models. We implement ExpandMask and AmodalMask on top of the same codebase. Our models are initialized from the SharpMask network trained on the original modal COCO data. We finetune using our amodal training set. We also attempted to fine-tune our models using synthetic amodal data

(*ExpandMask*^{S} and *AmodalMask*^{S}) by randomly overlaying objects masks from the original COCO dataset. For reproducibility, and to elucidate design and network choices, all source code are released on github.

Active Generation: Despite rich annotation, the amount of amodal instances in our proposed dataset is still very limited, especially for VGG-scale neural network training. The baseline methods mentioned above only use the amodal training set for passive fine-tuning, which means the amodal completion capacity is strictly confined by the limited annotation. Beyond passive training, generating meaningful synthetic samples would be crucial to improve the amodal segmentation model.

We consider two types of synthetic amodal generations: 1) directly generating occluded samples in image space and 2) learning to generate hard occluded samples in feature space.

To generate synthetic training samples in image space, we adopt a similar overlaying method mentioned in [62]: random select a foreground segmented mask and overlay on another object which is to be segmented amodally. The occlusion level is carefully controlled to make sure slight / heavy occlusion ratio are well balanced. A generated sample is show in 3.12, and we can find the artificial occlusion can not mimic the real world occlusion pattern, and the nontrivial distribution difference indeed affect the value of generated training samples, thus making the synthetic sample training useless.

To generate hard samples in feature space, the aim is to adapt to the occlusion distribution implicitly from learning, so the generated sample can mimic the real world occlusion distribution. Deep generative models [83, 84, 85, 86] are widely studied in recent years and achieved impressive photo-realistic level image generation. Aside of generation task, deep generative models are also applied to discriminative tasks including segmentation [87] and object detection [88]. We modified the SharpMask model and insert an occlusion-generating module to generate an occlusion mask. The learned occlusion mask is used to dropout all channels of network output features. Following [88], suppose a network intermediate layer output' spatial size is $d*d$, the occlusion module split it into 9 squares of size $\frac{d}{3}*\frac{d}{3}$. Then the occlusion generating module pick the square which lead to the highest cross entropy loss increase. The occlusion generation training is only applied on the non-occluded instances. Since at most 10% pixels are dropped out, the hardness of the training samples are kept within a controlled range. Fig

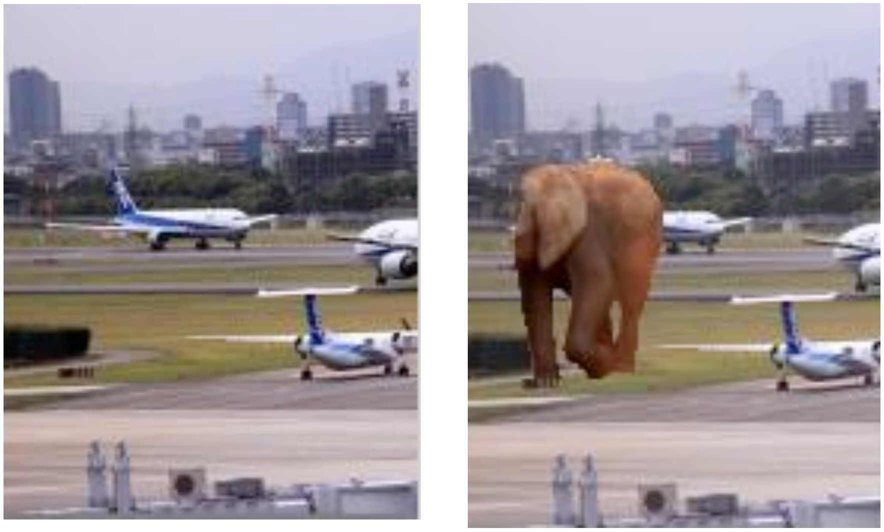


Figure 3.12: Generating synthetic amodal samples in image space by random overlaying: Left is the original image, right is random synthetic occlusion by overlaying a random selected foreground object: the elephant occluded the plane.

	Sharp Mask	Expand Mask	Amodal Mask	Ground Truth	Ground Truth
train-recall	45%	56%	59%	50%	100%
test-recall	41%	51%	54%	100%	100%
area	.696	.703	.719	.715	.715
y-axis	.711	.708	.706	.702	.702
OrderNet^{B}	.753	.764	.770	.770	.765
OrderNet^{M}	.786	.785	.791	.810	.817
OrderNet^{M+I}	**.793**	**.802**	**.814**	**.869**	**.883**

Table 3.4: Accuracy of pairwise depth ordering baselines applied to various segmentations results. See text for details.

3.13 illustrate the dropout stage of occlusion generating module.

Results: AR for all methods is given in Table 3.3 and qualitative results are shown in Figure 3.14. SharpMask is a strong baseline, especially for things and under limited occlusion, which is its training setup. With more occlusion, the amodal baselines are superior, indicating these models can predict amodal masks (however, they are worse on unoccluded objects). Using overlaying synthetic data improved AR on occluded regions over SharpMask but lagged the accuracy of using real training data. However, using occlusion generating module (dubbed A^*modalMask) improve the AmodalMask baseline, which shows the effectiveness of actively generation for training. Finally, we note that human accuracy on this task is still substantially higher (see §3.3).

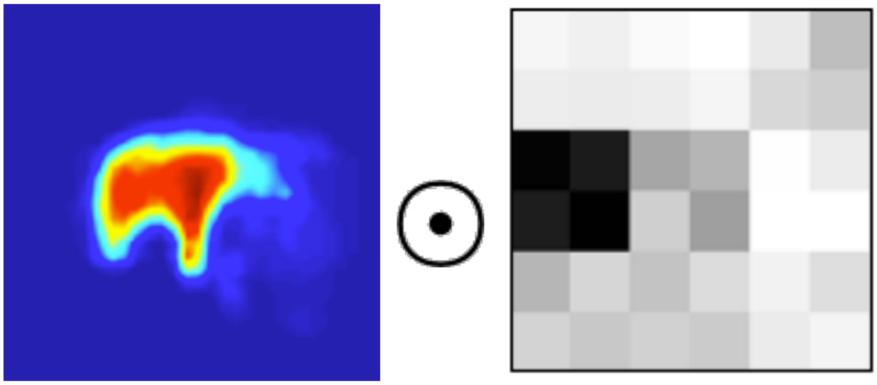


Figure 3.13: Actively generating amodal samples in feature space: actively selecting an occlusion mask which lead to highest cross entropy loss, and dropout all channels of the network output during training. Left side is an example network output visualization, while right side is an illustration occlusion mask.

3.4.2 Pairwise Depth Ordering

Metrics: Understanding full scene structure is challenging. Instead, we focus on evaluating pairwise depth ordering, which still requires reasoning about object interactions and spatial layout. Specifically, we report the accuracy of predicting which of two overlapping masks is in front. There are 36k/23k overlapping masks in the train/val sets.

Note that we have decoupled depth ordering from mask prediction. Since higher quality masks should be easier to order, we test each ordering algorithm with masks from multiple segmentation approaches. Specifically, for each ground truth mask we first find the best matching mask generated by a segmenter (with IoU of at least 0.5), we then evaluate the depth ordering only on these matched masks.

Baselines: We start with two trivial baselines: order by area (smaller mask in front) and order by y-axis (mask closest to top in back). Next, we implemented a number of deep nets for this binary prediction task: OrderNet$^{\mathrm{B}}$ which takes two bounding boxes as input, OrderNet$^{\mathrm{M}}$ which takes two masks as input, and OrderNet$^{\mathrm{M+I}}$ which takes two masks and an image patch. OrderNet$^{\mathrm{B}}$ uses a 3 layer MLP while the other variants use pre-trained ResNet50 models [13] (modified slightly to account for varying number of input channels). We train and test a separate OrderNet model for each set of masks. For each prediction we run inference twice (with input order reversed) and average the results.

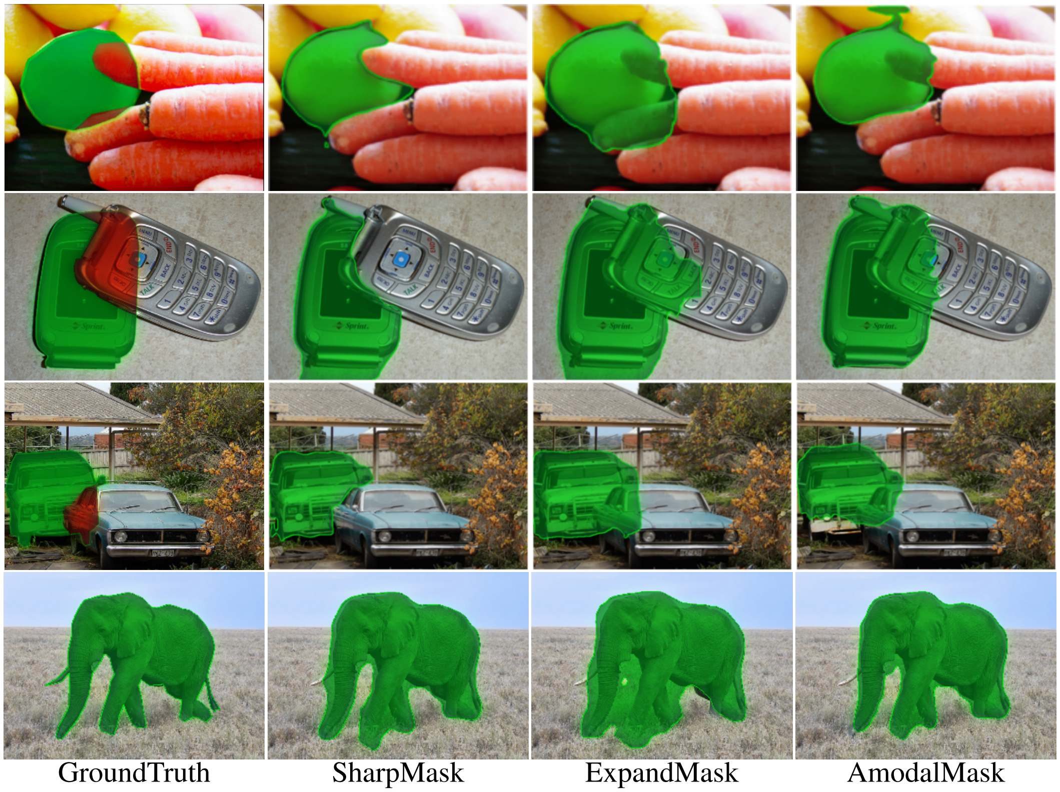


Figure 3.14: Examples of amodal mask prediction (red indicates occlusion). SharpMask predicts *modal* masks; ExpandMask and AmodalMask predict *amodal* masks. The last row shows an unoccluded object, for which ExpandMask is overzealous.

Results: We report results in Table 3.4. In addition to ordering masks from multiple segmentation algorithms, we also train and test OrderNet on ground truth masks (with varying amount of training data) to capture the role of mask quality and data quantity on ordering accuracy. The naive heuristics (area and y-axis) both achieve about 70% accuracy. OrderNet performs much better, with OrderNet$^{\text{M+I}}$ achieving $\sim$80% accuracy on generated masks and $\sim$90% on ground truth. OrderNet benefits from better masks (performance increases in each row moving from left to right), and the percent of recalled pairs also affects results slightly (as there is more data for training). Considering the simplicity of our approach, these results are surprisingly strong.

3.4.3 Edge Detection Evaluation

To allow for the study of edge detectors on COCO, in this appendix we report the performance of the structured edges (SE) [52] and the holistically-nested edge detector (HED) [53]

	SE [52]			**HED** [53]		
train / test	bsds-5	bsds-1	coco-1	bsds-5	bsds-1	coco-1
bsds-5	.630	.543	.522	.694	.615	.583
bsds-1	.628	.540	.520	.690	.609	.575
coco-1	.622	.536	.524	.686	.607	.609

Table 3.5: Edge detection accuracy (ODS) versus the *number of annotators per image.* Each row shows a different train setup and each column a different test setup. The number of annotators per image heavily affects test accuracy, but it makes little difference for training. Finally, switching the training set from BSDS to COCO has only a minor effect on SE but impacts HED more.

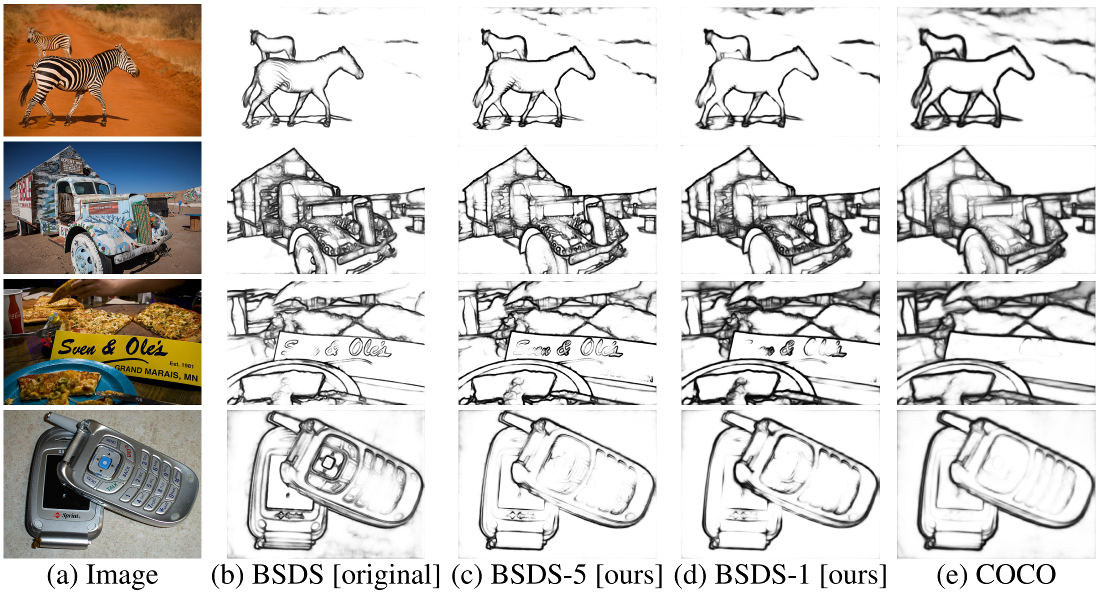


Figure 3.15: Edge detections for HED learned with different *training* sets. (b) Using the original BSDS annotations results in dense edge maps with interior edges being detected. (c,d) Training with our BSDS edges (with either 1 or 5 annotators per image) results in sparser, more semantically meaningful edges. (e) Finally, training with our COCO edges yields qualitatively similar albeit slightly better results.

on COCO. Results of these detectors on the BSDS dataset [51] (for both the original annotations and our annotations) were presented in §3.3.2. Here we train these state-of-the-art edge detectors on the 2500 COCO train images and test them on the 1250 image COCO val set.

We begin by noting that edge detection metrics [51] are heavily impacted by the *number of annotators per image*. The ground truth edges used for evaluation are the union of the human annotations and using more annotators per image results in denser edges for testing. In Table 3.5, we report edge detection accuracy versus the number of annotators per image using our annotations. During *testing*, reducing the number of annotators per image lowers ODS substantially (even though the evaluated models are identical). On the other hand, reducing the

	ODS	AP	R50
SE [52]	.524	.474	.519
HED [53]	.609	.493	.741

Table 3.6: Edge evaluation for SE and HED on the COCO val set.

number of annotations per image during *training* leaves results largely unchanged.

From Table 3.5 we also observe that results between COCO and BSDS are quite similar once the number of annotators per image is accounted for. We thus emphasize that while the edge detection accuracy on COCO appears to be worse than on BSDS (both using our annotations), this is an artifact of how accuracy is measured. We also note that while COCO only has one annotator per image, it has 10× more images than BSDS (5000 versus 500). Thus, more data-hungry approaches should benefit from COCO.

In Table 3.6, we report complete SE and HED edge detection results on the COCO validation set (training performed on the COCO train set). Our dataset provides a substantial challenge for current state-of-the-art edge detectors. Finally, in Figure 3.15, we show qualitative HED edge detection results using different options for the training data.

3.5 Discussion

We presented a new dataset to study perceptual grouping tasks. Moreover, we formally proposed a new vision task: amodal segmentation, which is to segment object's both visible and invisible shape. The most distinctive feature of our dataset is that regions are annotated amodally: both the visible and occluded portions of regions are marked. The motivation is to encourage amodal perception, and reasoning about object interactions and scene structure. Extensive analysis shows that semantic amodal segmentation is a well-posed annotation task. We also provided evaluation metrics and strong baselines for the proposed tasks. We hope our dataset will help stimulate new research directions for the community.

To address the issue of limited amodal training instances, we also developed an approach to actively generate occlusion in feature space, used as augmented training data. The proposed method improves over both baselines and synthetic generating occlusion directly in image space.

In the future, we would like to further explore actively generated occlusion: combining adversarial training (GANs) with amodal segmentation to generate more realistic and useful training data. Another potential research is to use the learned occlusion prediction model to reason about the full scene geometric layout.

Chapter 4

Interactive Learning for Visual Grounding

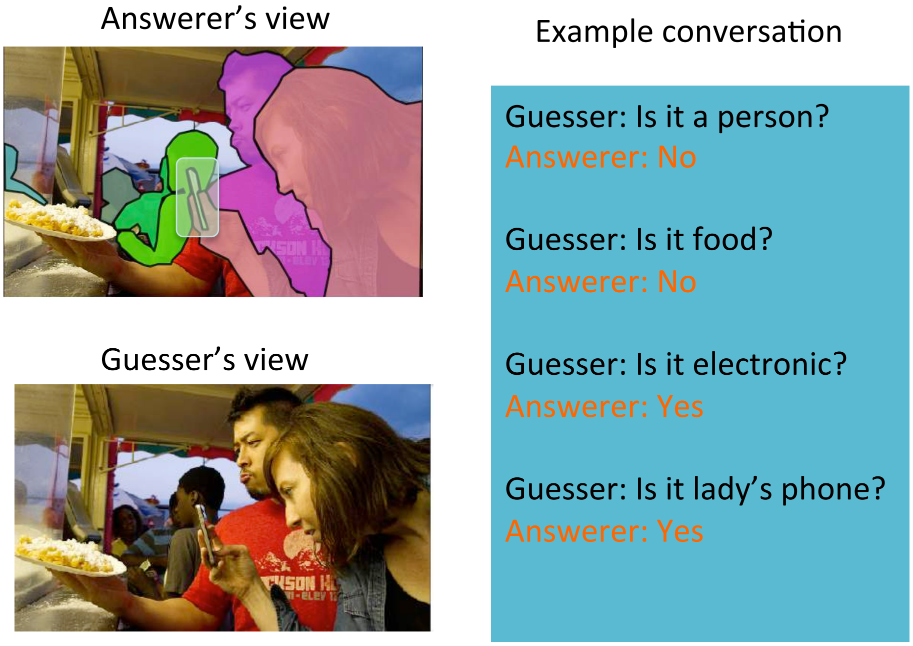
4.1 Introduction

Interaction via natural language is the most efficient and natural for human to acquire knowledge and complete daily tasks. In artificial intelligence, developing intelligent agents which are able to communicate via natural language and correspond the linguistic concepts with visual contents.

With the rapid advance of neural networks in computer vision since 2012, traditional vision tasks including image classification, object detection[50] and semantic segmentation [56] achieved near human level accuracy. Novel vision task towards higher level understanding is becoming the new focused area, including image captioning[89], visual question answering [90] and referring expression[91]. Above new tasks are all heavily rely on the natural language processing techniques.

Image captioning, VQA and referring expression explored different aspects of vision model's capability to understand and articulate meaningful sentences, but these tasks still lack multi-round interactions. Moving forward, visual dialogue tasks are recently proposed to test if a neural model is able to conduct complete and meaningful conversation with another model or human agent, in order to achieve a common goal. We are particularly interested in the scenario when the dialogue content is related to the visual content and the common goal is defined by the image object grounding.

Two recent visual dialogue tasks along with datasets are proposed: visDial[92] and GuessWhat?! [93]. In visDial, two players are allowed to conduct free form conversation (chit-chat), in which the questioners only see the caption of the image while the answerer see the image. The evaluation is done by the ranking of ground truth responses. GuessWhat?! task instead



Answerer’s view

Guesser’s view

Example conversation

Guesser: Is it a person?
Answerer: No

Guesser: Is it food?
Answerer: No

Guesser: Is it electronic?
Answerer: Yes

Guesser: Is it lady’s phone?
Answerer: Yes

Figure 4.1: GuessWhat game is a two-player game, in which the guesser(also the questioner) only sees the images but the answerer also sees the candidate regions and one ground truth referred region. Both players cooperate to accomplish the guessing task by performing a conversation. The answerer is only allowed to answer the question by using (Yes, No, N/A).

chose an environment setup for cooperative goal-ended conversation: The questioners see an image and need to ask a ground truth object in the image. The answerer sees the ground truth object mask and bounding box, but only allowed to answer (yes, no, n/a) to help the questioner. In the end the of conversation, the questioner see a list of candidate regions and need to select one according to the conversation. The guess success rate is used as the evaluation metric. The object grounding task need both agents being able to detect and recognize objects, understand the spatial layout, and using positional word to distinguish between candidate regions. The evaluation is straightforward and directly testify the problem-solving capability through natural language. In this work we focus on the GuessWhat?! Task. An example of GuessWhat?! game is shown in Figure 4.1.

The existing dialogue generation methods are mostly relied on the passive supervised training using the static collected human conversation training data. Reinforcement learning methods are explored in chit-chat form dialogue in [1]. Also, [5] shows the question generators are able to improve using reinforcement learning. Inspired by this work, we explore the interactive training of multiple models in a dynamic environment, and we show the state-of-the-art results benefiting from interactive training. We also compare and analyze the dialogue generation, and show the interesting observation that the phenomenon of diverging from natural language to more effective artificial language during the interactive training.

In summary, the contribution of this chapter is threefold:

- We introduced an interactive reinforcement learning method to generate visual conversation, for a cooperative object grounding task, and we report the significant improvement over state-of-the-art task success rate on GuessWhat?! dataset.
- We introduced seq2seq model with attention module into visual question generator architecture, and the results showed the proposed architecture outperformed the LSTM baselines.
- We analyze the conversation generation results in detail and observe the emergence of more efficient usage of human vocabulary.

In the following, we first review the related work in visual conversation and reinforcement learning in Section 4.2. Then in Section 4.3, we discuss our architecture and interactive learning. In Section 4.4, we showed the experiment results and concluded in 4.5.

4.2 Related Work

4.2.1 From Captioning, VQA to Visual Conversation

The intersection of computer vision and natural language has long been studied and evolved from basic tasks including image captioning[94, 89], visual question answering [95, 90] and referring expressions [96, 91], to more thorough tasks such as visual conversation. Image captioning and VQA can be regarded as single round visual conversation, which mainly focuses on bridging visual concept with visual concepts descriptively. However, in real world applications, the ideal intelligent agents should be able to conduct multiple round conversations to reach the common goal. Such setting is more complicated than VQA and captioning since the agents are required to be able to reasoning based on the conversation history and generate the proper questions to narrow down the uncertainty space. In natural language processing community, the goal ended dialogue is also widely studied [97, 98]. Most models rely on the recent success of LSTM architecture based sequential generation models [99, 100]. The visual conversation can be viewed as a direct counterpart of dialogue systems grounded by visual content.

The existing visual conversation datasets can be divided into task-oriented dialogue [93] and free form dialogue (chit-chat) [92]. The GuessWhat?! task is a two-player game, where the questioner need to guess which object the answerer is referring to, by asking discriminative questions to narrow down the candidate regions, while the answerer is only allowed to answer "yes, no, n/a". The task is evaluated directly by the success rate of object grounding task. The annotation is based on the existing object detection benchmark COCO dataset [18]. The VisDial [92] dataset is also based on COCO dataset, but focusing on image level description: the questioners only have access to the caption of images, while the answerers see the raw images. The conversation is free form with no clearly defined goal. Both works introduced strong baseline using supervised training. In the following works [1] showed reinforcement learning improves supervised pre-trained model in an image retrieval task. [5] also showed reinforcement learning can improve the question generator's performance in GuessWhat?! task. Inspired by this line of work, and further extended the reinforcement training into multiple model interactive learning.

4.2.2 Deep Reinforcement Learning and NLP

Deep Reinforcement learning has been successfully applied to Go game [101, 102], video games [103] and robotics. Consider vocabulary as action spaces and conversation as an environment, with a proper-defined reward function, conversation systems can also use reinforcement learning to improve the agents' performance. [104] first adopted seq2seq model for end-to-end dialogue system, which encodes the conversation history with the encoder and outputs a response with decoder. [97] adopted deep reinforcement learning into dialogue systems by defining rewards to encourage desirable conversation properties including informativity, coherence, and easy to answer. Due to the large action spaces, policy gradient methods [105] are usually used for reinforcement training.

4.2.3 Emergence of Artificial Language

Along with natural language generation, researchers [106, 107] also found that multiple agents can form their own communication protocols (artificial languages) during cooperation training. This line of research shows that the autonomous generated language could potentially be more effective in achieving human created tasks.

4.3 Methods

In this section, we will discuss the detailed architecture of the proposed methods. We will first overview the overall model schemes, then discuss the question generator's seq2seq architecture, then we will discuss the supervised pre-training and interactive training method.

4.3.1 Overview

Following the baseline [93], the conversation system is consists of 3 disconnected models: answerer, guesser and question generator (qgen). All 3 models have access to the raw image and conversation history. The answerer model has access to the ground truth referred object mask / bounding box, and output a token from a 3-vocabulary dictionary: yes, no, n/a. The guesser model has access to the candidate object regions, and output a score for each region. The question generator does not have additional knowledge, and need to articulate questions to narrow down the uncertainty space accordingly. The detailed scheme is shown in 4.4. We can see that 3 models are disconnected although the task-specific knowledge (*e.g*vocabulary, spatial region encoding) is clearly shared. The intuition is that all 3 models should be co-trained together to improve based on interactions. This is the primary intuition of this work as we will see in the next section.

4.3.2 Seq2Seq Question Generator

We used sequence to sequence architecture as the basis of the question generator in our visual conversation setting. The seq2seq model is originally introduced for machine translation [100] and yielding state-of-the-art performances, as shown in Fig 4.2. Compared with vanilla RNN

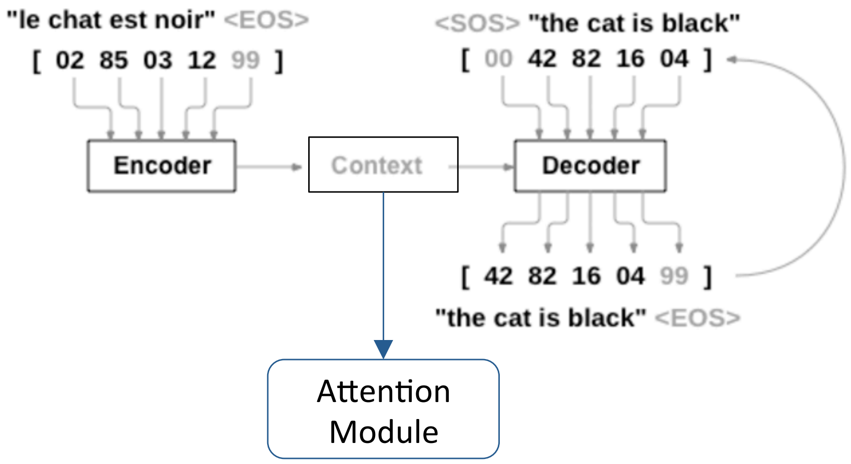


Figure 4.2: seq2seq model for machine translation: (French to English). The encoder take source sentence as input and encode into an embedding space. The attention layer re-weight the encoding and the decoder learns to predict next word in the target translation sequentially in the target language.

model such as LSTMs, sequence to sequence model is consists of an encoder-decoder architecture, and more crucially, various attention modules can be inserted into the bottleneck, thus the model is more powerful in long distance reasoning. The seq2seq models have previously used in conversation systems [104] but not in visual conversations to our best knowledge. The architecture is shown in Fig 4.3

The input is a partial conversation history: Assume the previous 2 round conversations are follows: “Q: Is this a person? A: No. Q: Is it an elephant? A: Yes.”. The ground truth next round question is “Q: Is it the white one on the left?” The seq2seq model will first encode the partial history tokens using the LSTM encoder into an embedding space, then concatenated with the image embeddings (*e.g*VGG feature in our experiments) as the visual context. To mixing the visual context with the language embeddings, we used a global dot product attention layer [108]. The illusionary scheme is shown in 4.3. During reinforcement training stage, we only take at most 2 round recent conversations as the input to the seq2seq model, instead of the whole conversation history due to efficiency consideration.

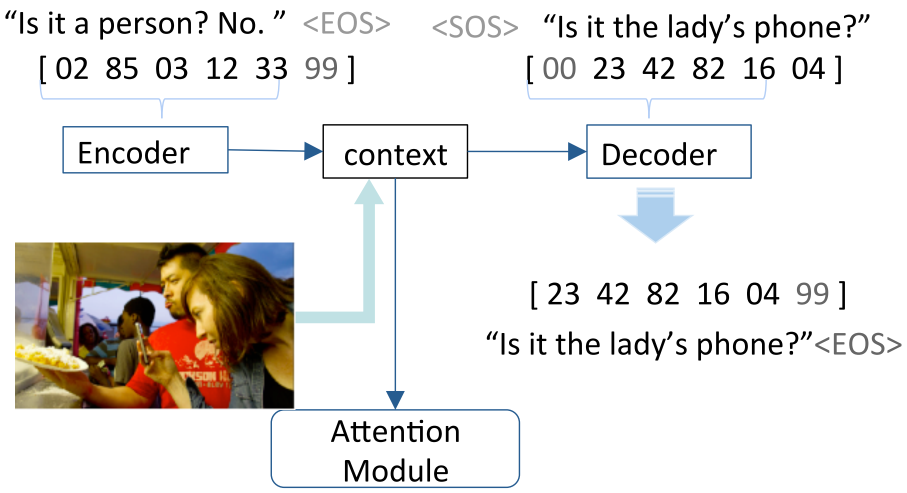


Figure 4.3: seq2seq model for visual conversations: The encoder take **partial conversation history** as input, and encode the tokens along with the image into an embedding space. The decoder learns to predict next word in the target **next question** sequentially in the human conversation annotations.

4.3.3 Interactive Reinforcement Training

After the supervised pre-training, 3 models only obtained knowledge from the static dataset. Despite the decent accuracy in separate evaluation, when 3 models are hooked up together, the weakness of each model are enlarged and thus yielding sub-optimal overall success rate. Based on this intuition, we enable 3 models to actively learning in a self-talking environment with an interactive manner.

The goal-oriented conversation can be viewed as a Markov Decision Process (MDP), with the action space of answerer and question generator can be defined by their vocabulary respectively. For the question generator, the state can be defined as:

$$\mathbf{s}_{qgen}^{t} = (\mathbf{q_t}, (\mathbf{q}, a)_{:t-1}, \mathbf{I}) \tag{4.1}$$

, where t is the number of current QA rounds, q_t is the current *incomplete* questions, $(\mathbf{q}, a)_{:t-1}$ are the previous $t-1$ rounds conversation history and $\mathbf{I}$ is the referred image. The action space of question generator is to append a new word $token_i \in V_{qgen}$. In our implementation, V_{qgen} is of size 4.8K (replacing words with frequency ≤ 2 with a special token $< UNK >$). The maximum question length is set to 12 tokens and the question is truncated after the last $<? >$ token. Each token of the question is randomly sampled from the question generator's last softmax layer, thus yielding a stochastic policy function $\pi_{qgen}(q_t|\mathbf{s}_t; \theta_{qgen})$.

Similarly, for the answerer model, we can also define the state and action space as follows:

$$\mathbf{s}_{ans}^{t} = (\mathbf{q_t}, (\mathbf{q}, a)_{:t-1}, \mathbf{I}, \mathbf{O}_{gt}) \tag{4.2}$$

, where n is the number of current QA rounds, q_t is the current $complete$ questions, $(\mathbf{q}, a)_{:t-1}$ are the previous $t-1$ rounds conversation history; $\mathbf{I}$ is the referred image and $\mathbf{O}_{gt}$ is referred ground truth object. The action space for the answerer is to sample a single response word $ans_i \in V_{ans}$. In our setting, V_{ans} only contains 3 words: $yes, no, n/a$. The response is similarly sampled from the answerer's last softmax layer, therefore the neural network also defines a stochastic policy function $\pi_{ans}(a_t|\mathbf{s}_t; \theta_{ans})$.

The reward function is defined by the outcome of guesser model: once the whole QA trajectory is sampled from current (θ_{qgen}, θ_{ans}) until M_{max} rounds are reached, the conversation is finished. Then the guesser model will read the whole conversation history and scoring each candidate regions. If the highest scored region $\mathbf{O}_{pred}$ matches the ground truth region $\mathbf{O}_{gt}$, the task is considered as successfully completed.

$$R_T(\mathbf{s}_{ans}^T, a^T) = R_T(\mathbf{s}_{qgen}^T, \mathbf{q}^T) = \begin{cases} 1, & if \quad \mathbf{O}_{pred}(\theta_{guesser}) == \mathbf{O}_{gt}; \\ 0, & if \quad \mathbf{O}_{pred}(\theta_{guesser}) \neq \mathbf{O}_{gt}; \end{cases} \tag{4.3}$$

With policy function and action spaces defined, we used REINFORCE algorithm to update (θ_{qgen}, θ_{ans}) to maximize the expected reward under two models' policies:

$$J(\theta_{ans}, \theta_{qgen}) = \mathop{\mathbb{E}}_{\pi_{ans}, \pi_{qgen}} [\Sigma_{t=0}^{T} R_t(\mathbf{s}_t, (q_t, a_t))] \tag{4.4}$$

During optimization, we first sample a mini-batch (denote the batch size as B) of trajectories using (π_{ans}, π_{qgen}). To reduce the gradient variance, we additionally add a base branch for both of the question generator and the answerer model to estimate the reward scores. The baseline estimation in policy gradient algorithm was used in [109] to stabilize optimization, and was also adopted in [5]. We extend baseline estimation to both the question generator and the answerer model. The baseline branch takes the LSTMs hidden output, as input and use a single linear + ReLU layer to estimate the reward score.

We used the likelihood ratio trick [105], thus the formulation of policy gradient updates for the question generator and answerer can be written as follows:

$$\nabla_{\theta_{qgen}} J = \mathop{\mathbb{E}}_{\pi_{qgen},\pi_{ans}} [\sum_{b=1}^{B} \sum_{t=1}^{T} \nabla_{\theta_{qgen}} log\pi_{qgen}(\mathbf{q_t}|\mathbf{s_t}) * (r - b_{qgen}))] \tag{4.5}$$

$$\nabla_{\theta_{ans}} J = \mathop{\mathbb{E}}_{\pi_{qgen},\pi_{ans}} [\sum_{b=1}^{B} \sum_{t=1}^{T} \nabla_{\theta_{ans}} log\pi_{ans}(a_t|\mathbf{s_t}) * (r - b_{ans}))] \tag{4.6}$$

To optimize the base branch, we use MSE loss to regress the baseline estimation to the reward scores. Notice that the gradient of baseline branch is detached from the main models, so the learning the baseline estimation will not affect the question/answer generation.

Note that the reward function R is actually parametrized by the guesser model. As we can see in the supervised pre-training results, the guesser model is indeed imperfect even in human dialogue setting (the best supervised trained guesser model has about 30% error rate). This inherent noisy reward will inevitably affect the quality of policy gradient. We thus suspect there's potential for the guesser model to improve in the self-talking environment. The guesser model can potentially adapt to the machine dialogue distribution during self-talking and provide more accurate reward guidance to the question generator and the answerer.

The overall interactive training can be summarized in the above algorithm. The interactive learning illustrational graph is shown in Fig 4.4

4.4 Experiments

In this section, we discussed the detailed implementation and experimental results on GuessWhat?! dataset and analyze the generated conversations both quantitatively and qualitatively. Our codebase is developed using Pytorch [110].

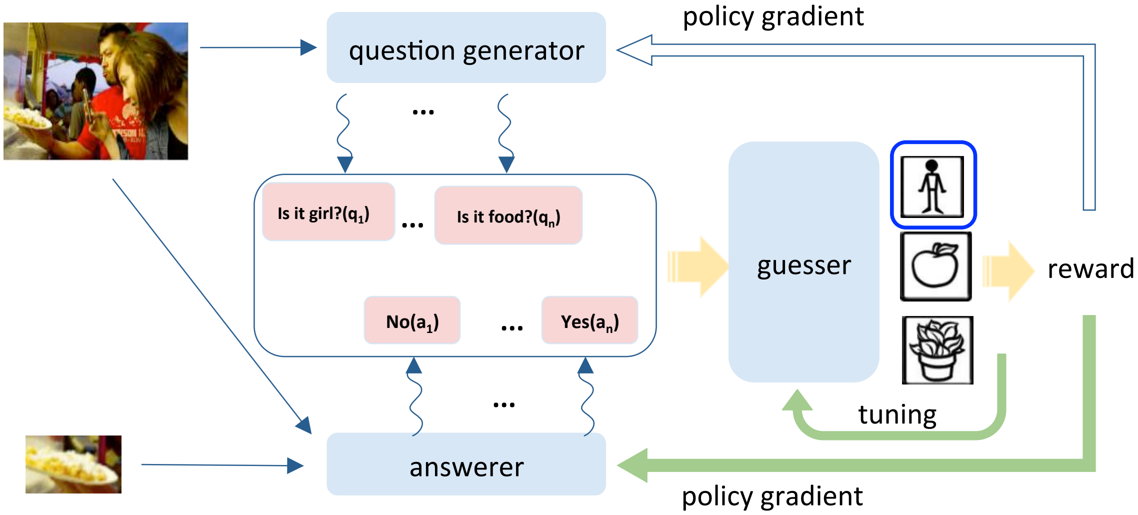


Figure 4.4: In the interactive reinforcement learning stage, the question generator and the answerer model are put into a self-talking environment. The randomly sampled conversations are collected and sent to the guesser model to compute the binary reward. Once the reward is computed, we can compute the policy gradient of the question generator and the answerer model respectively, and update their parameters using ADAM optimizer. Also, we can tune the guesser model as well to help the guesser provide more accurate rewards. Note that the baseline model [5] only used policy gradient for the question generator, but not tuning for the answerer model and guesser model (denoted by the green arrows on the right).

Algorithm 3 algorithm sketch for the interactive reinforcement training

```
for $i = 1$ to $B$ do
    sample image $I$, and ground truth referred region $O_{gt}$
    for $j = 1$ to $M_{max}$ do
        generate question $q_{ij}$ using $\pi_{qgen}$
        generate answer $a_{ij}$ using $\pi_{ans}$
    end for
end for
get rewards: $r_{1:B}$ using $f_{guess}(<\mathbf{qa}>, \mathbf{O_{gt}}|\theta_{guesser})$
optimize the question generator using Equation. 4.5 and MSE loss for baseline branch
optimize the answerer model using Equation. 4.6 and MSE loss for baseline branch
optimize the guesser model using cross entropy loss.
```

4.4.1 Implementation details

Dataset Preprocessing First we pre-process the corpus of the whole dataset to generate a vocabulary of size 4.8K words. The low-frequency words are discarded and replaced with a special token $< UNK >$. To make the comparison fair, we also use VGG network as the image feature extractor. We filtered out the incomplete conversations from the dataset, to avoid data ambiguity.

Supervised Pre-training For guesser and answerer model, we generally follow the similar supervised pre-training setup as in [93]. Specifically, for answerer model, we use an LSTM encoder with embedding size of 256 and a hidden layer of size 512. The category of the object is encoded using one hot encoding and embedded into a 512-dimensional space. The spatial feature is simply the 8-dimensional bounding box positional feature. The concatenation of 3 embeddings is passed through a hidden linear layer of size 800 with ReLU activation, then mapped to 3 dimensional output for (yes, no, n/a). The baseline branch uses 800 dimensional hidden layer as input and go through a single linear layer with ReLU activation, and output a single value estimation for the reward score. The answerer model is trained using SGD optimizer at learning rate of 0.0001 for 20 epochs. For the guesser model, we use 2 LSTM layers with 1024 embedding size, and a hidden layer of size 2048. The guesser model maps the region local information (category and spatial) and global information (vgg image feature and conversation history LSTM embedding) into a common embedding space and use dot-product as the score for each region. The guesser model is trained using RMSProp optimizer with learning rate of 0.0001 for 40 epochs.

	[1]	ours
answerer	21.50	21.31
guesser	38.70	38.02
Question generator	58.40	58.10

Table 4.1: The error rate of supervised-pretrained models. The answerer model and the guesser model is evaluated on the test set data. The question generator is evaluated on the test set images, and the conversation is generated from the pretrained answer model and the guesser model. Note that for the question generator, we also compared with [1] under random sampling setting.

For the question generator, we use the seq2seq architecture with following design: The encoder concatenates the VGG image feature with the 512-dimensional text embedding, and feed into a 512-dimensional LSTM encoding space. Then we use a dot global attention layer [108], and connects with the decoder. The decoder is consists of a LSTM decoder and a linear layer which maps the 512-dimensional embedding feature back into vocabulary size space. The baseline branch takes 512-dimensional hidden vector as input and also use one linear layer to output the estimated base value. During training, we use the teacher forcing method: always feed the ground truth target as the input of next round prediction. We also use a dropout layer $dropprobability = 0.1$ in the encoder to avoid over-fitting. During evaluation, we use multinomial sampling to generate questions. During training, we used a batch size of 16 and padded with special token when preparing the batched sample.

The supervised evaluation of the pretrained models are shown in the Table 4.1. Note that the evaluation is based on the held out test set. Thus, the scores only manifest how similar does the models behave compared to the human conversation, rather than the model performance for the actual guess task.

Interactive Learning To stabilize the reinforcement learning, both the question generator and the answerer models are equipped with a baseline branch to estimate the reward value. During both training and evaluation, we consistently use multinomial sampling to generate both questions and answers. We also tried to use greedy method for decoding during evaluation, but find that the performance degraded since many generated questions are identical. When preparing batch to sample trajectories, we only take the recent 2 round conversation histories for the question generators, The max length of a question is set to be 12 and the maximum of QA rounds is set to 6. The question is truncated after the last appearance of $<? >$. Following the setting in [1], during self-talking training, images are from the train set and the target region is randomly

method	Success rate (on test set)
[1] (samping)	58.50
[1] (greedy)	60.30
SL pretrained	41.70
IRL^{qgen}	58.21
IRL^{qa}	65.14
IRL^{qg}	63.10
IRL^{qag}	**82.97**
Human score	84.40

Table 4.2: The task success rate for different interactive reinforcement learning results.IRL^{qa} (both qgen and answerer are trained) and IRL^{qg} (both qgen and guesser are trained) outperform IRL^{qgen}. We experimented different configuration of interactive learning, and find that the best performed setting is when qgen, answerer and guesser models are simultaneously tuned (IRL^{qag}).

selected. During evaluation, the images come from the test set and use the assigned ground truth region as target. We set the batch size to be 64 and learning rate to be 0.0001. All three models are updated using Adam optimizer [111] for at most 20 epochs.

4.4.2 Evaluation Results

We've extensively evaluated different combination of interactive training. The results are summarized in Table 4.2. The result shows the when question generator and the answerer are learned interactively (IRL^{qa}), the resulting success rate significantly improves from 58.21 to 65.14. More surprisingly, when all three models are trained interactively (IRL^{qag}), the task success rate raised to 82.97, which is very close to the human success rate. We also experimented with different combinations of interactive learning, for instance, freezing updates for the guesser and question generator and only updating answerer model (IRL^{ans}). The result shows the improvement from IRL^{qag} is even greater than the improvement sum of each single tuned model, which validate the benefits of interactive learning.

Qualitative Analysis and Observation

To understand why interactive reinforcement learning improving the task completion, in this section we look at some exemplary generated conversations qualitatively as shown in Figure 4.5 and Figure 4.6. Both success and failure examples are given.

Spatial Reasoning: From examples, we can find that the question generator learns to use spatial reasoning to distinguish objects. The positional words such as “left”, “right”, and “front” are used extensively and the answerer model in most cases give the correct answer. Aside from simply using absolute positional word, the question generator also learns the relative position phrases, for instance, “second from left?” in the 2nd example in Figure 4.6. The positional questions effectively reduce the image scene by half, which is very similar to the KD-tree spatial splitting fashion, and also shrink the candidate regions greatly. We notice that the usage of the positional questions in our trained model is even more frequently than the human generated conversations. (In human conversations, enumerating questions like “Is it the blue cup next to the white book?” are more frequent.) We believe during the reinforcement interactive training, such spatial reasoning capability is strengthened compared with supervised pretrained baselines.

Divergence from Natural Language: We also observe that after the interactive reinforcement learning, the answerer model and the question generators adapt the original language grammar towards a seemingly more effective communication protocols. Some generated conversations are difficult to interpret from the standard English perspective, but well perceived by the guesser model and finally yielding correct prediction. One typical example is that after the reinforcement learning, the answerer model tends to use “N/A” more aggressively, even if the question should be better answered with “Yes” or “No”, for example, as shown in the 3rd example in Figure 4.5. Notice that in the annotated training data, the appearance of “N/A” is much less frequent compared with our generated conversations. We suspect that the answerer tries to use “N/A” to convey more information beyond binary responses. Surprisingly the answerer’s communication selection is well understood by the other two models, and finally improve the task completion rate. When three models are trained together, the diverging phenomenon is even more radical: the generated questions become degenerated into completely non-informative sentences, but surprisingly, the guesser presumably decode the message purely from answerer’s outputs, and the result turns out extremely successful, as shown in the bottom example in Figure 4.5.

Similar phenomenons are also observed by recent works [106, 107]. In their settings, the emergence of artificial language is generated from scratch, while in our setting, the evolved language still constrained by the basic English grammar, but diverge towards a more effective communication protocol variant.

Limitation: Despite the improved task success rate, there are still limitations of the current generation results. The first limitation is that the question generator tends to forget previous information as the conversation proceeds, and ask repetitive or highly similar questions. Such phenomenons could be potentially improved by utilizing external memory modules. Second, right now we manually fix the rounds of the QA. Ideally the question generator and the answerer should adaptively determine the conversation rounds based on the scene complexity, for example, the question generator should learn to early stop for an easy scene (for instance, only 2 or 3 candidate regions), or extend the conversation for a crowded scene. Third, the question generator still has trouble to detect small objects, thus failed to ask discriminative questions when small objects are placed close to each other, as shown in the last example in Figure 4.6. This is fundamentally limited by the current vectorized image representation, and could be potentially improved by using the convolutional image features to enhance the local spatial information.

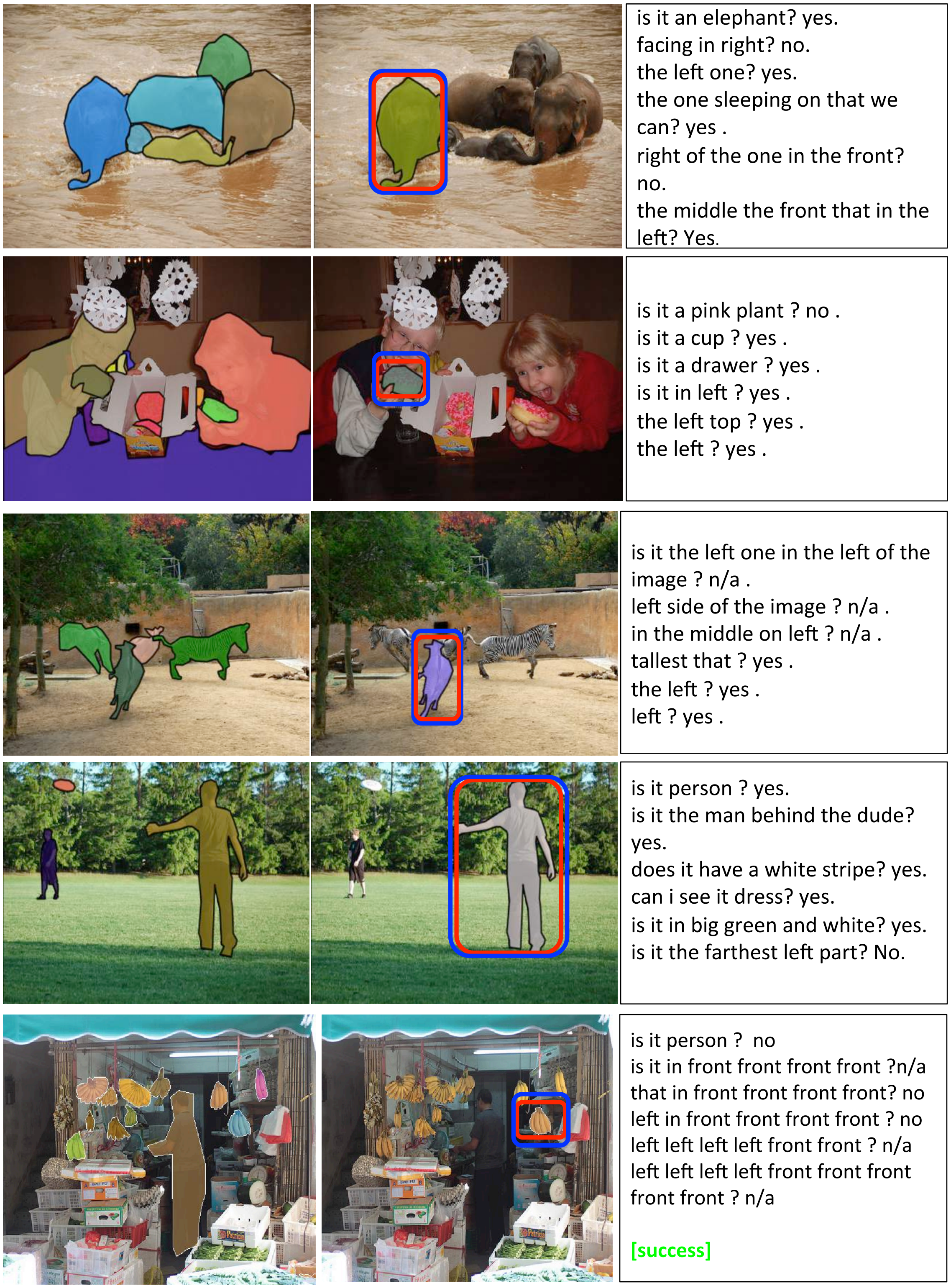
4.5 Conclusion

In this chapter, we've researched on the visual conversation task, in which neural network models are trained to generated questions and answers to complete a visual object grounding task. We extend the reinforcement learning to interactively trained reinforcement learning. During the interactive training stage, multiple models are required collaboratively evolving the communication, and finally achieving state-of-the-art task completion rate. The experiment result shows that when 3 models are trained collectively, the overall improvement is even greater than the linear combination of single model tuning improvement.

We also qualitatively analyze the generated conversation and observe the interesting evolution during the interactive learning. We find that the answerer model learns to use “n/a” more efficiently and the guesser also learns to adapt to answerer's new encoding and improve prediction dramatically. As the evolution continues, the generated conversation drifts from natural language distribution towards a more efficient direction, but also more difficult to understand for human. The phenomenon echoes with the recent research on the emergence of artificial language during cooperation [106, 107].

The current generated conversation lacks interpretability although improved task success rate. To overcome this limitation, in the future, we would like to constrain the generated conversation is within a certain semantic distance from the human interpretable natural language manifold. One possible solution is to use semantic regularization during optimization. The other idea is to adopt adversarial training, in which a discriminator is responsible to tell if the generated conversation is similar to the human conversation, thus we can get rid of repetitions like "Is he on the left on the left?". Another direction is to apply spatial attention module with the seq2seq model. Right now we only use the vectorized VGG image feature, so the spatial local encodings could be sacrificed and making the spatial reasoning hard. By explicitly train a spatial attention module, we can not only potentially improve the models' reasoning capability but also visualize the reasoning procedure from the attention heat map, thus making the model more interpretable.

Finally, we would like to extend the training method on other visual conversation tasks other than object grounding. Object grounding can be viewed as a specialized testbed for the spatial reasoning capability. Other logic / factual reasoning are still untouched but would be highly exciting to explore.



is it an elephant? yes.
facing in right? no.
the left one? yes.
the one sleeping on that we can? yes .
right of the one in the front? no.
the middle the front that in the left? Yes.

is it a pink plant ? no .
is it a cup ? yes .
is it a drawer ? yes .
is it in left ? yes .
the left top ? yes .
the left ? yes .

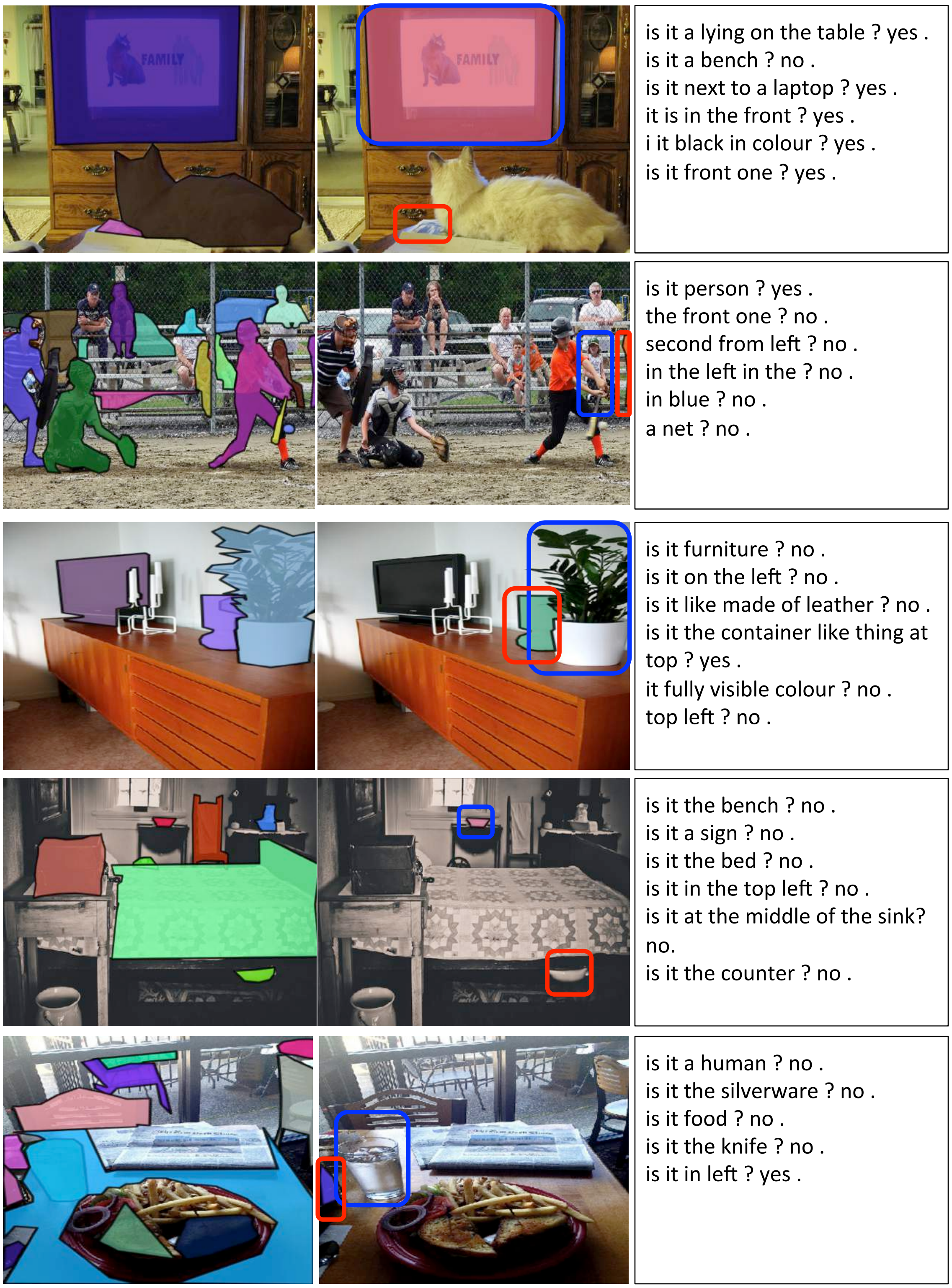
is it the left one in the left of the image ? n/a .
left side of the image ? n/a .
in the middle on left ? n/a .
tallest that ? yes .
the left ? yes .
left ? yes .

is it person ? yes.
is it the man behind the dude? yes.
does it have a white stripe? yes.
can i see it dress? yes.
is it in big green and white? yes.
is it the farthest left part? No.

is it person ? no
is it in front front front front ?n/a
that in front front front front? no
left in front front front front ? no
left left left left front front ? n/a
left left left left front front front front front ? n/a

[success]

Figure 4.5: Examples of **success** guessing tasks: the left column is the raw image with all highlighted candidate regions; the middle column marks the ground truth target region (highlighted by blue bounding boxes), along with the guesser model's predicted region (highlighted by red bounding boxes). The right column is the generated conversations from interactively trained answerer model and question generator.



is it a lying on the table ? yes .
is it a bench ? no .
is it next to a laptop ? yes .
it is in the front ? yes .
i it black in colour ? yes .
is it front one ? yes .

is it person ? yes .
the front one ? no .
second from left ? no .
in the left in the ? no .
in blue ? no .
a net ? no .

is it furniture ? no .
is it on the left ? no .
is it like made of leather ? no .
is it the container like thing at top ? yes .
it fully visible colour ? no .
top left ? no .

is it the bench ? no .
is it a sign ? no .
is it the bed ? no .
is it in the top left ? no .
is it at the middle of the sink? no.
is it the counter ? no .

is it a human ? no .
is it the silverware ? no .
is it food ? no .
is it the knife ? no .
is it in left ? yes .

Figure 4.6: Examples of **failed** guessing tasks. The layout is same with Figure 4.5. In the middle column, the ground truth target regions are highlighted by **blue** bounding boxes, while the guesser model's predicted regions are highlighted by **red** bounding boxes.

Chapter 5

Conclusions

In this dissertation, we addressed the challenging computer vision tasks in which traditional passive learning don't work well. We specifically focused on three typical scenarios, and proposed new learning paradigms including active selection under constrained annotation budget, actively generating hard examples and learning through natural language interactions.

- We proposed an active selection for histopathological Image classifications when the annotation budget is constrained. The proposed algorithm is based on submodular optimization with a partition matroid constraint. The proposed method encourage the uncertainty reduction as well as the selection diversity. We also show the greedy-like algorithm has near optimal theoretical guarantee and potentially scalable to large scale unlabeled dataset.

- We proposed a novel visual perception task called semantic amodal segmentation, in which the algorithms are required to not only segment out objects‘ visual region but also extend to its occluded parts. We proposed a systematically annotated amodal segmentation dataset for this new task, along with strong baselines and evaluation. To address the challenge of inadequate hard examples, we proposed to actively generate hard example by learning to deform easy examples. The experiment results demonstrate the improved performance against baselines. We additionally showed the amodal segmentation can be potentially used for spatial depth ordering inference.

- We proposed an interactive learning approach to generate natural language dialogue between two conversation agents, in order to accomplish a visual object grounding task. We proposed to use seq2seq architecture in the question generator along with attention module. The experiment on the Guesswhat?! dataset shows that interactively learned agents achieve state-of-the-art task success rate. We also analyzed the generated conversation and discussed the evolution of interactively learned conversation.

This thesis work was an attempt to go beyond traditional passive supervised learning and towards active and interactive learning paradigms. The encouraged potential future work includes the following several directions: For the active selection problem, the learner assumption in Chapter 2 is still SVM type classifier. It would be very interesting to rethink active selection problem from a neural network perspective. For the amodal segmentation, the current proposed method still far from human performance for amodal perception. The challenging setting of open dictionary semantic segmentation requires the learner to be able to acquire new concepts in few shot learning [45] paradigms. Also, amodal perception essentially requires model to be able to reason about the visual environment, which by itself is a challenging AI problem. Similarly, such visual reasoning is closely related to generating goal-oriented natural language conversation: how to construct questions in a logical manner and approach to the common goal, and also ground the visual concepts with linguistic concepts is still an open question.

References

[1] A. Das, S. Kottur, J. M. Moura, S. Lee, and D. Batra, "Learning cooperative visual dialog agents with deep reinforcement learning," *arXiv preprint arXiv:1703.06585*, 2017.

[2] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in neural information processing systems*, 2014, pp. 487–495.

[3] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[4] S. Bell, P. Upchurch, N. Snavely, and K. Bala, "Opensurfaces: A richly annotated catalog of surface appearance," *SIGGRAPH*, 2013.

[5] F. Strub, H. de Vries, J. Mary, B. Piot, A. Courville, and O. Pietquin, "End-to-end optimization of goal-driven and visually grounded dialogue systems," *arXiv preprint arXiv:1703.05423*, 2017.

[6] Y. Freund and R. Schapire, "A short introduction to boosting," *Journal-Japanese Society For Artificial Intelligence*, vol. 14, no. 771-780, p. 1612, 1999.

[7] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 1998.

[9] L. G. Valiant, "A theory of the learnable," *Communications of the ACM*, vol. 27, no. 11, pp. 1134–1142, 1984.

[10] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural nets," in *Annual Conference on Neural Information Processing Systems*, 2012.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[12] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[14] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[15] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701–1708.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[17] J. Xiao, K. A. Ehinger, J. Hays, A. Torralba, and A. Oliva, "Sun database: Exploring a large collection of scene categories," *International Journal of Computer Vision*, vol. 119, no. 1, pp. 3–22, 2016.

[18] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft COCO: Common objects in context," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[19] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "Youtube-8m: A large-scale video classification benchmark," *arXiv preprint arXiv:1609.08675*, 2016.

[20] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," 2016. [Online]. Available: https://arxiv.org/abs/1602.07332

[21] G. Kanizsa, *Organization in vision: Essays on Gestalt perception*. Praeger Publishers, 1979.

[22] S. E. Palmer, *Vision science: Photons to phenomenology*. MIT press Cambridge, MA, 1999.

[23] S. Vicente, V. Kolmogorov, and C. Rother, "Graph cut based image segmentation with connectivity priors," in *Computer vision and pattern recognition, 2008. CVPR 2008. IEEE conference on*. IEEE, 2008, pp. 1–8.

[24] M. N. Gurcan, L. E. Boucheron, A. Can, A. Madabhushi, N. M. Rajpoot, and B. Yener, "Histopathological image analysis: A review," *IEEE Reviews in Biomedical Engineering*, vol. 2, pp. 147–171, 2009.

[25] S. Petushi, F. U. Garcia, M. M. Haber, C. Katsinis, and A. Tozeren, "Large-scale computations on histology images reveal grade-differentiating parameters for breast cancer," *BMC Medical Imaging*, vol. 6, no. 1, p. 14, 2006.

[26] S. Doyle, S. Agner, A. Madabhushi, M. Feldman, and J. Tomaszewski, "Automated grading of breast cancer histopathology using spectral clustering with textural and architectural image features," in *Proc. ISBI*, 2008.

[27] M. M. Dundar, S. Badve, G. Bilgin, V. Raykar, R. Jain, O. Sertel, and M. N. Gurcan, “Computerized classification of intraductal breast lesions using histopathological images,” *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 7, pp. 1977–1984, 2011.

[28] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, Cambridge, MA, 2002.

[29] D. J. Foran, L. Yang *et al.*, “Imageminer: a software system for comparative analysis of tissue microarrays using content-based image retrieval, high-performance computing, and grid technology,” *Journal of the American Medical Informatics Association*, vol. 18, no. 4, pp. 403–415, 2011.

[30] L. Yang, W. Chen, P. Meer, G. Salaru, L. A. Goodell, V. Berstis, and D. J. Foran, “Virtual microscopy and grid-enabled decision support for large-scale analysis of imaged pathology specimens,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 4, pp. 636–644, 2009.

[31] B. Settles, “Active learning literature survey,” in *Technical Report, University of Wisconsin, Madison*, 2010.

[32] M.-F. Balcan, S. Hanneke, and J. W. Vaughan, “The true sample complexity of active learning,” *Machine learning*, vol. 80, no. 2-3, pp. 111–139, 2010.

[33] M.-F. Balcan, A. Beygelzimer, and J. Langford, “Agnostic active learning,” in *International Conference on Machine Learning*, 2006.

[34] S. C. Hoi, R. Jin, J. Zhu, and M. R. Lyu, “Batch mode active learning and its application to medical image classification,” in *International Conference on Machine Learning*, 2006.

[35] S. Tong and D. Koller, “Support vector machine active learning with applications to text classification,” *Journal of Machine Learning Research*, vol. 2, pp. 45–66, 2002.

[36] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey, “An analysis of approximations for maximizing submodular set functions–ii,” *Polyhedral Combinatorics*, pp. 73–87, 1978.

[37] Y. Chen and A. Krause, “Near-optimal batch mode active learning and adaptive submodular optimization,” in *International Conference on Machine Learning*, 2013.

[38] D. Golovin and A. Krause, “Adaptive submodularity: Theory and applications in active learning and stochastic optimization,” *Journal of Artificial Intelligence Research*, vol. 42, no. 1, pp. 427–486, 2011.

[39] ——, “Adaptive submodular optimization under matroid constraints,” in *arXiv preprint arXiv:1101.4450*, 2011.

[40] L. Lovász, “Hit-and-run mixes fast,” *Mathematical Programming*, vol. 86, no. 3, pp. 443–461, 1999.

[41] A. Gonen, S. Sabato, and S. Shalev-Shwartz, “Active learning of halfspaces under a margin assumption,” in *arXiv preprint arXiv:1112.1556*, 2011.

[42] Z. Wang and J. Ye, "Querying discriminative and representative samples for batch mode active learning," in *Proc. KDD*, 2013.

[43] A. de Brebisson and G. Montana, "Deep neural networks for anatomical brain segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 20–28.

[44] K. Kamnitsas, C. Ledig, V. F. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker, "Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation," *Medical image analysis*, vol. 36, pp. 61–78, 2017.

[45] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," in *Advances in Neural Information Processing Systems*, 2016, pp. 3630–3638.

[46] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 4, pp. 594–611, 2006.

[47] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[48] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in *International Conference on Learning Representations*, 2014.

[49] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[50] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Annual Conference on Neural Information Processing Systems*, 2015.

[51] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[52] P. Dollár and C. L. Zitnick, "Fast edge detection using structured forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[53] S. Xie and Z. Tu, "Holistically-nested edge detection," in *IEEE International Conference on Computer Vision*, 2015.

[54] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "*TextonBoost*: Joint appearance, shape and context modeling for multi-class object recognition and segm." in *European Conference on Computer Vision*, 2006.

[55] P. O. Pinheiro and R. Collobert, "Recurrent convolutional neural networks for scene labeling," in *International Conference on Machine Learning*, 2014.

[56] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[57] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL visual object classes (VOC) challenge,” *International Journal of Computer Vision*, 2010.

[58] A. Karpathy, 2015, http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/.

[59] J. Wagemans, J. H. Elder, M. Kubovy, S. E. Palmer, M. A. Peterson, M. Singh, and R. von der Heydt, “A century of Gestalt psychology in visual perception,” *Psychological Bulletin*, 2012.

[60] P. O. Pinheiro, R. Collobert, and P. Dollar, “Learning to segment object candidates,” in *Annual Conference on Neural Information Processing Systems*, 2015.

[61] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, “Learning to refine object segments,” in *European Conference on Computer Vision*, 2016.

[62] K. Li and J. Malik, “Amodal instance segmentation,” in *European Conference on Computer Vision*, 2016.

[63] P. Dollár, Z. Tu, and S. Belongie, “Supervised learning of edges and object boundaries,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[64] C. Fowlkes, D. Martin, and J. Malik, “Local figure–ground cues are valid for natural images,” *Journal of Vision*, 2007.

[65] C. Liu, J. Yuen, and A. Torralba, “Nonparametric scene parsing via label transfer,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[66] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, “Indoor segmentation and support inference from rgbd images,” in *European Conference on Computer Vision*, 2012.

[67] S. M. Bileschi, “Streetscenes: Towards scene understanding in still images,” Ph.D. dissertation, Citeseer, 2006.

[68] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, “The role of context for object detection and semantic segm. in the wild,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[69] M. Maire, S. X. Yu, and P. Perona, “Hierarchical scene annotation,” in *British Machine Vision Conference*, 2013.

[70] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei, “Visual genome: Connecting language and vision using crowdsourced dense image annotations,” *International Journal of Computer Vision*, 2017.

[71] J. Xiao, J. Hays, K. Ehinger, A. Oliva, A. Torralba *et al.*, “Sun database: Large-scale scene recognition from abbey to zoo,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[72] B. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, “LabelMe: A database and web-based tool for image annotation,” *International Journal of Computer Vision*, 2008.

[73] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[74] R. Guo and D. Hoiem, "Beyond the line of sight: labeling the underlying surfaces," in *European Conference on Computer Vision*, 2012.

[75] S. Gupta, P. Arbelaez, and J. Malik, "Perceptual organization and recognition of indoor scenes from RGB-D images," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[76] N. Silberman, L. Shapira, R. Gal, and P. Kohli, "A contour completion model for augmenting surface reconstructions," in *European Conference on Computer Vision*, 2014.

[77] A. Kar, S. Tulsiani, J. Carreira, and J. Malik, "Amodal completion and size constancy in natural scenes," in *IEEE International Conference on Computer Vision*, 2015.

[78] J. Tighe, M. Niethammer, and S. Lazebnik, "Scene parsing with object instances and occlusion ordering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[79] Y. Yang, S. Hallman, D. Ramanan, and C. Fowlkes, "Layered object detection for multi-class segm." in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[80] E. H. Adelson and J. R. Bergen, "The plenoptic function and the elements of early vision," in *Computational Models of Visual Processing*. MIT Press, 1991.

[81] D. A. Forsyth, J. Malik, M. M. Fleck, H. Greenspan, T. Leung, S. Belongie, C. Carson, and C. Bregler, *Finding pictures of objects in large collections of images*. Springer, 1996.

[82] J. Hosang, R. Benenson, P. Dollár, and B. Schiele, "What makes for effective detection proposals?" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[83] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[84] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[85] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[86] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2016, pp. 2172–2180.

[87] P. Luc, C. Couprie, S. Chintala, and J. Verbeek, "Semantic segmentation using adversarial networks," *arXiv preprint arXiv:1611.08408*, 2016.

[88] X. Wang, A. Shrivastava, and A. Gupta, "A-fast-rcnn: Hard positive generation via adversary for object detection," *arXiv preprint arXiv:1704.03414*, 2017.

[89] A. Karpathy and L. Fei-Fei, “Deep visual-semantic alignments for generating image descriptions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3128–3137.

[90] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, “Vqa: Visual question answering,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2425–2433.

[91] J. Mao, J. Huang, A. Toshev, O. Camburu, A. L. Yuille, and K. Murphy, “Generation and comprehension of unambiguous object descriptions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 11–20.

[92] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh, and D. Batra, “Visual dialog,” *arXiv preprint arXiv:1611.08669*, 2016.

[93] H. de Vries, F. Strub, S. Chandar, O. Pietquin, H. Larochelle, and A. Courville, “Guesswhat?! visual object discovery through multi-modal dialogue,” *arXiv preprint arXiv:1611.08481*, 2016.

[94] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, “Long-term recurrent convolutional networks for visual recognition and description,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2625–2634.

[95] M. Malinowski, M. Rohrbach, and M. Fritz, “Ask your neurons: A neural-based approach to answering questions about images,” in *IEEE International Conference on Computer Vision*, 2015, pp. 1–9.

[96] S. Kazemzadeh, V. Ordonez, M. Matten, and T. L. Berg, “Referitgame: Referring to objects in photographs of natural scenes.”

[97] J. Li, W. Monroe, A. Ritter, M. Galley, J. Gao, and D. Jurafsky, “Deep reinforcement learning for dialogue generation,” *arXiv preprint arXiv:1606.01541*, 2016.

[98] A. Bordes and J. Weston, “Learning end-to-end goal-oriented dialog,” *arXiv preprint arXiv:1605.07683*, 2016.

[99] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[100] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[101] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, “Mastering the game of go with deep neural networks and tree search,” *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[102] Y. Tian and Y. Zhu, “Better computer go player with neural network and long-term prediction,” *arXiv preprint arXiv:1511.06410*, 2015.

[103] A. Dosovitskiy and V. Koltun, “Learning to act by predicting the future,” *arXiv preprint arXiv:1611.01779*, 2016.

[104] O. Vinyals and Q. Le, “A neural conversational model,” *arXiv preprint arXiv:1506.05869*, 2015.

[105] R. J. Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[106] I. Mordatch and P. Abbeel, “Emergence of grounded compositional language in multi-agent populations,” *arXiv preprint arXiv:1703.04908*, 2017.

[107] K. Evtimova, A. Drozdov, D. Kiela, and K. Cho, “Emergent language in a multi-modal, multi-step referential game,” *arXiv preprint arXiv:1705.10369*, 2017.

[108] M.-T. Luong, H. Pham, and C. D. Manning, “Effective approaches to attention-based neural machine translation,” *arXiv preprint arXiv:1508.04025*, 2015.

[109] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, “Sequence level training with recurrent neural networks,” *arXiv preprint arXiv:1511.06732*, 2015.

[110] 2017, http://pytorch.org.

[111] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.