Greedy Inference Algorithms for Structured and Neural Models

Qing Sun

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical Engineering

Dhruv Batra, Chair
Jia-Bin Huang, Co-chair
Devi Parikh
Lynn Amos Abbott
Aditya Bodicherla Prakash
Harpreet Singh Dhillon

November 29, 2017
Blacksburg, Virginia

Keywords: Graph models, Diversity, Submodular, Beam Search, Active Learning

Greedy Inference Algorithms for Structured and Neural Models

Qing Sun

(ABSTRACT)

A number of problems in Computer Vision, Natural Language Processing, and Machine Learning produce structured outputs in high-dimensional space, which makes searching for the global optimal solution extremely expensive. Thus, greedy algorithms, making trade-offs between precision and efficiency, are widely used. In this thesis, we prove that greedy algorithms are effective and efficient to search for multiple top-scoring hypotheses from structured (neural) models: 1) Entropy estimation. We aim to find deterministic samples that are representative of Gibbs distribution via a greedy strategy. 2) Searching for a set of diverse and high-quality bounding boxes. We formulate this problem as the constrained maximization of a monotonic sub-modular function such that there exists a greedy algorithm having near-optimal guarantee. 3) Fill-in-the-blank. The goal is to generate missing words conditioned on context given an image. We extend Beam Search, a greedy algorithm applicable on unidirectional expansion, to bidirectional neural models when both past and future information have to be considered. We test our proposed approaches on a series of Computer Vision and Natural Language Processing benchmarks and show that they are effective and efficient.

Greedy Inference Algorithms for Structured and Neural Models

Qing Sun

(GENERAL AUDIENCE ABSTRACT)

The rapid progress has been made in Computer Vision (*e.g.*, detecting what and where objects are shown in an image), Natural Language Processing (*e.g.*, translating a sentence in English to Chinese), and Machine learning (*e.g.*, inference over graph models). However, a number of problems produce structured outputs in high-dimensional space, *e.g.*, semantic segmentation requires predicting the labels (*e.g.*, dog, cat, or person, etc) of all super-pixels, the search space is huge, say L^n, where L is the number of object labels and n is the number of super-pixels. Thus, searching for the global optimal solution is often intractable. Instead, we aim to prove that greedy algorithms that produce reasonable solutions, *e.g.*, near-optimal, are much effective and efficient. There are three tasks studied in the thesis: 1) Entropy estimation. We attempt to search for a finite number of semantic segmentations which are representative and diverse such that we can approximate the entropy of the distribution over output space by applying the existing model on the image. 2) Searching for a set of diverse bounding boxes that are most likely to contain an object. We formulate this problem as an optimization problem such that there exist a greedy algorithm having theoretical guarantee. 3) Fill-in-the-blank. We attempt to generate missing words in the blanks around which there are contexts available. We tested our proposed approaches on a series of Computer Vision and Natural Language Processing benchmarks, *e.g.*, MS COCO, PASCAL VOC, etc, and show that they are indeed effective and efficient.

Acknowledgments

Firstly, I would like to say a big thank to my advisor, Prof. Dhruv Batra, for the support and guidance during my Ph.D. study. He introduced me to Artificial Intelligence(AI) research field and showed me how to be a solid researcher. He helped me improve half-baked ideas, clarify my confusion, strength my skills, to make me a competitive researcher. His optimism and confidence in me made me not fear anything.

He was the leader, walked me through the dark. Because of him, I really enjoyed this journey. It was my great honor to have him as my advisor. I was lucky enough to work close with Prof. Devi Parikh. She is very smart and nice as well. She always proposed insightful ideas to my projects, making me dive deep. Thanks for doing that.

I had the opportunity to do my internship at Baidu IDL Lab, where I met my mentors Dr. Wei Xu and Dr. Yi Yang. I would like to thank them for introducing me to a very exciting research field which I have never touched before. And, I am happy that this research field inspired me a lot and I would like to pursue it in the future.

I would like to thank Prof. Jia-Bin Huang, Prof. Lynn Abbott, Prof. Aditya Prakash and Prof. Harpreet Dhillon for serving in my committee. Without their valuable suggestions, this dissertation would not be such comprehensive at all.

During my PhD studies, I made a lot of friends, who helped me in various ways. CVMLP lab at Virginia Tech has always been a family to me and I owe thanks to all my family members:

Xiao Lin, Peng Zhang, Jiasen Lu, Jianwei Yang, Stanislaw Antol, Ramakrishna Vedantam, Arjun Chandrasekaran, Ramprasaath Selvaraju, Yash Goyal and Aishwarya Agrawal, etc. I miss the old days when we hanging out, discussing projects together. All these are going to be precious memory in my whole life.

Last but not the least, I have a very warm and loving family. Since my childhood, my parents taught me to be diligent, curious and confident, which played quite important roles in my life. Their love and support enabled me to chase my dream and achieve what I have. I want to express my biggest appreciation and thanks to them.

Contents

Chapter 1

Introduction

A number of problems in Computer Vision, Natural Language Processing, and Machine Learning can be written as structured prediction problems involving a distribution with an exponentially large search space. For instance, in semantic segmentation [20, 22, 140]the size of the search space is L^n, where n is the number of (super-)pixels and L is the number of object labels that each (super-)pixel can take. In object detection [78, 96], each bounding box can be determined by the top-left point and bottom-right point. Thus, the number of possible bounding boxes in an image is $O(\#pixels^2)$. In image captioning/machine translation [64, 126], the word at every time step is chosen from the entire vocabulary. As a result, the search space of possible sequences describing an image is exponential in the sequence length. However, such huge search space makes exact inference in structured models intractable. So, we are interested in greedy algorithms which are effective and efficient to search for multiple top-scoring hypotheses from structured (neural) models.

In the following chapters, we will introduce a variety of works based on greedy algorithms to solve structured-output problems in Computer Vision and Natural Language Processing.

Entropy estimation. Maximum a Posterior (MAP) inference in graph models has been widely studied [16, 84]. However, compared with finding the highest mode or configuration,

computing entropy of structured-output models is more challenging due to the summation over an exponentially large output space. Classical stochastic sampling techniques, *e.g.*, Markov Chain Monte Carlo (MCMC) methods, approximate integrals or sums by replacing an exponentially large support with a finite number of samples. Unfortunately, MCMC sampling based methods often require a long burn-in period to transit out of one mode of the distribution to another. It is obvious that the layout of the distribution is crucial to estimate entropy. Motivated by this, we proposed a novel variational approach, which generates deterministic samples built upon a greedy strategy to visit as many modes as possible, to approximate the entropy of Gibbs distribution in structured models.

We tested our approach under active learning setting, in which we begin with a structured probabilistic model (CRF) trained on a small set of labeled images, then search the large unlabeled pool for a set of informative images to annotate where our current model is most uncertain, *i.e.*, has highest entropy. We show that our approach outperforms a number of baselines and results in a 90%-reduction in the number of annotations needed to achieve nearly the same accuracy as learning from the entire dataset.

Submodular Maximization. In general, greedy algorithms do not produce an optimal solution since they always make the choice that seems to be the best at that moment in the hope that this choice will lead to an optimal solution. Fortunately, there is a situation when greedy algorithms are able to guarantee a reasonably good approximation. This situation can be formulated as: given a set of items Y, and you're looking for the best subset $S \in Y$ which maximizes a pre-defined objective function. If the objective function is monotonic and submodular, there exists a greedy algorithm that iteratively selects the item with the largest marginal gain achieves a near-optimal approximation factor of $(1 - 1/e)$. This motivates us to formulate the search for a set of bounding boxes as an optimization problem. The "best" subset is a set of diverse and high-quality bounding boxes that have high likelihood

of containing an object and cover as many objects instances as possible.

We apply our proposed technique to the task of generating object proposals on the PASCAL VOC 2007 [37], PASCAL VOC 2012 [38], and MS COCO [77] datasets. Our results show that our approach outperforms all baselines.

Bidirectional Greedy Inference. Beam Search (BS) is a widely used approximate inference algorithm for decoding sequences from unidirectional neural sequence models. BS is a greedy algorithm that maintains the top-B most likely partial hypotheses through the search tree. At each time step, BS expands these B partial hypotheses to all possible beam expansions and then select the B highest scoring among the expansions. Unfortunately, this procedure is not applicable to Bidirectional Recurrent Neural Networks (RNNs). We simply can not extend a beam from left to right because it requires knowing future states and variables. To enable the use of bidirectional models, we proposed an approximation inference approach by extending BS to be able to consider both past and future dependencies.

We compare our proposed method against a number of baselines for fill-in-the-blank tasks on MS COCO [77] and Visual Madlibs [139]. Our results show that it is effective and efficient and consistently outperforms all baselines.

Chapter 2

Active Learning in Structured Probabilistic Models

A number of problems in Computer Vision – image segmentation, geometric labeling, human body pose estimation – can be written as a mapping from an input image $\mathbf{x} \in \mathcal{X}$ to an exponentially large space $\mathcal{Y}$ of *structured outputs*. For instance, in semantic segmentation, $\mathcal{Y}$ is the space of all possible (super-)pixel labelings, $|\mathcal{Y}| = L^n$, where n is the number of (super-)pixels and L is the number of object labels that each (super-)pixel can take.

As a number of empirical studies have found [50, 91, 141], the amount of training data is one of the most significant factors influencing the performance of a vision system. Unfortunately, unlike *unstructured* prediction problems – binary or multi-class classification – data annotation is a particularly expensive activity for structured prediction. For instance, in image segmentation annotations, we must label every (super-)pixel in every training image, which may easily run into millions. In pose estimation annotations, we must label 2D/3D locations of all body parts and keypoints of interest in thousands of images. As a result, modern dataset collection efforts such as PASCAL VOC [37], ImageNet [31], and MS COCO [76] typically involve spending thousands of human-hours and dollars on crowdsourcing websites such as Amazon Mechanical Turk.

Active learning [106] is a natural candidate for reducing annotation efforts by seeking

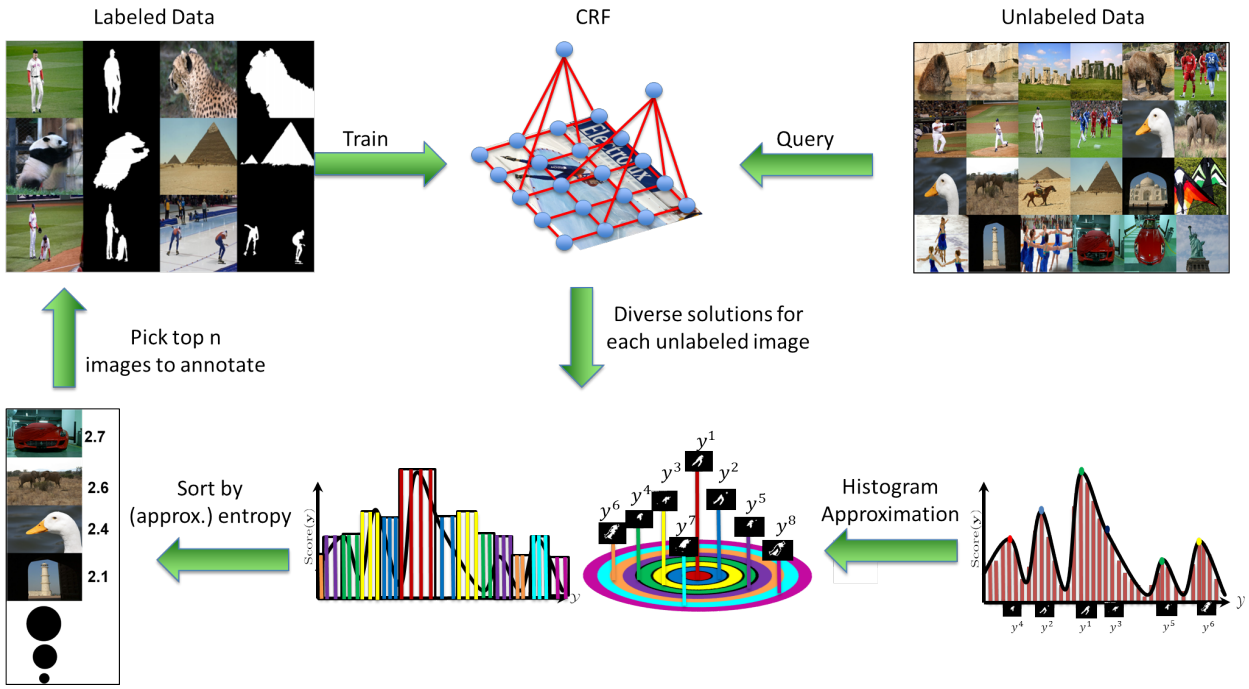


Figure 2.1: Overview of our approach. We begin with a structured probabilistic model (CRF) trained on a small set of labeled images; then search the large unlabeled pool for a set of informative images to annotate where our current model is most uncertain, *i.e.* has highest entropy. Since computing the exact entropy is NP-hard for loopy models, we approximate the Gibbs distribution with a *coarsened histogram* over M bins. The bins we use are 'circular rings' of varying hamming-ball radii around the highest scoring solution. This leads to a novel variational approximation of entropy in structured models, and an efficient active learning algorithm.

labels only on the most *informative* images, rather than the annotator passively labeling all images, many of which may be uninformative. Unfortunately, active learning for structured-output models is challenging. Perhaps even the simplest definition of "informative" involves computing the entropy of the learnt model over the output space:

$$H(\mathrm{P}) = -\mathbb{E}_{\mathrm{P}(\mathbf{y}|\mathbf{x})}[\log(\mathrm{P}(\mathbf{y}|\mathbf{x}))] = -\sum_{\mathbf{y}\in\mathcal{Y}} \mathrm{P}(\mathbf{y}|\mathbf{x}) \log \mathrm{P}(\mathbf{y}|\mathbf{x}), \tag{2.1a}$$

which is intractable due to the summation over an exponentially-large output space $\mathcal{Y}$.

Overview and Contributions. In this chapter, we study active learning for probabilistic models such as Conditional Random Fields (CRFs) that encode probability distributions over an exponentially-large structured output space.

Our main technical contribution is a variational approach [131] for approximate entropy computation in such models. Specifically, we present a crude yet surprisingly effective *histogram approximation* to the Gibbs distribution, which replaces the exponentially-large support with a *coarsened* distribution that may be viewed a histogram over M bins. As illustrated in Fig. 4.1, each bin in the histogram corresponds to a subset of solutions – for instance, all segmentations where size of foreground (number of ON pixels) is in a specific range $[L\ \ U]$. Computing the entropy of this coarse distribution is simple since M is a small constant ($\sim$10). Importantly, we prove that the *optimal histogram*, *i.e.* one that minimizes the KL-divergence to the Gibbs distribution, is composed of the mass of the Gibbs distribution in each bin, *i.e.* $\sum_{\mathbf{y}\in\text{bin}} P(\mathbf{y}|\mathbf{x})$. Unfortunately, the problem of estimating sums of the Gibbs distribution under general hamming-ball constraints continues to be #P-complete [119]. Thus, we upper bound the mass of the distribution in a bin with the maximum entry in a bin multiplied by the size of the bin. Fortunately, finding the most probable configuration in a hamming ball has been recently studied in the graphical models literature [11, 83, 94], and efficient algorithms have been developed, which we use in this work.

We perform experiments on figure-ground image segmentation and coarse 3D geometric labeling [56]. Our proposed algorithm significantly outperforms a large number of baselines and can help save hours of human annotation effort.

2.1 Related Work

Large-scale data annotation efforts in computer vision have typically involved thousands of hours of human effort, either for pay (Mechanical Turk) or motivated by the task being fun [102] or a game [129].

Learning from weak annotations. One theme in reducing annotation effort in recognition tasks is to learn from weak annotations – where the annotation provides only the

name of the object in the image [5, 133, 135], or partial labelings where the annotations for some pixels are missing [53, 121], or learning in an interactive setting where the annotator repeatedly provides scribbles [9, 10, 17, 100], or propagating labels from annotated images to unannotated images [73]. In contrast, we focus on the fully-supervised active learning setting where the goal is to identify *which* images to label; once an image is chosen, we receive full annotations.

Active learning is a vast sub-field of machine learning, with a number of approaches for quantifying the informativeness of an as yet unlabeled example – based on disagreement among a committee of classifiers [43], version space of an SVM [116], and expected informativeness [80] for probabilistic models. We focus on entropy-based active learning, which is a natural definition of informativeness, but is intractable to compute for structured models such as CRFs.

In computer vision, active learning has been used for scene classification [95], object/image categorization [60, 63], and annotating large image and video datasets [27, 40, 123, 130, 136]. Notice that these are all instances of *unstructured* prediction – binary or multi-class classification. We focus on structured prediction where the space of possible outcomes and thus the support of the distribution of our model is exponentially large. Two previous works address the problem of exact computation of entropy in such models [28, 107]. However, both these works assume chain/tree-structured graphical models, which is understandable in natural language processing, but is an unreasonable assumption for computer vision problems. We make no such assumptions.

The closest to our goal is the recent work of Luo *et al.* [79], on active learning in latent structured models. There are a number of subtle but important differences w.r.t. to our work. The algorithm in [79] estimates the *local entropy* of the marginal distribution of each variable via convex belief propagation [51], and asks the user to annotate the single variable/pixel

that is most marginally uncertain. In comparison, the focus of our work is to estimate the entropy of the joint distribution not the entropy of the marginal. Thus, we are able to find an *image* where the model is most uncertain, rather than a pixel. This matches the natural annotation modality, where annotators are shown full images and asked to provide polygonal annotations [31, 76, 102]. In our experiments, we compare against an adapted version of the algorithm from [79], which estimates the entropy of the joint distribution by summing entropies of the marginals (thereby assuming that pixels are independent). To its credit, [79] studies a more general setting (learning under partial supervision) than the one studied in this chapter (full supervision). However, within this narrower but important domain, we find that our approach outperforms all baselines, including our adaption of [79].

The work of Maji *et al.* [81] defines a novel uncertainty measure for structured models, called 'MAP perturbation uncertainty,' which upper-bounds the true entropy of the Gibbs distribution via MAP perturbations [89, 114] under Gumbel noise. Note that for entropy-based active learning, ideally we require *lower bounds* on entropy, not upper bounds. Inspired by the MAP perturbation literature [89, 114], we compare to (and outperform) a baseline that directly approximates the entropy by treating the MAP perturbation solutions as samples.

Finally, a number of recent works have looked into active learning with multiple modalities of annotator feedback and rich learner-supervisor interactions beyond simply asking for class-labels [92, 108, 122]. Combining such rich interactions with our approach is an interesting direction for future.

2.2 Preliminaries and Notation

Notation. For any positive integer n, let $[n]$ be shorthand for the set $\{1, 2, \ldots, n\}$. Given an input image $\mathbf{x} \in \mathcal{X}$, our goal is to make a prediction about $\mathbf{y} \in \mathcal{Y}$, where $\mathbf{y}$ may be a figure-ground segmentation, or a category-level semantic segmentation. Specifically, let $\mathbf{y} =$

$\{y_1 \dots y_n\}$ be a set of discrete random variables, each taking value in a finite label set, $y_u \in Y_u$. In semantic segmentation, u indexes over the (super-)pixels in the image, and these variables are the labels assigned to each (super-)pixel, *i.e.* $y_u \in Y_u = \{\text{sky, building, road, car}, \dots\}$.

CRF Model. Let $G = (\mathcal{V}, \mathcal{E})$ be a graph defined over the output variables $\mathbf{y}$, *i.e.* $\mathcal{V} = [n]$, $\mathcal{E} \subseteq \binom{[n]}{2}$. Let $\theta_u(y_u)$ be the unary term expressing the local confidence at site u for the label y_u, and $\theta_{uv}(y_u, y_v)$ be the pairwise term expressing compatibility of label y_u and y_v at adjacent vertices. The *score* for any configuration $\mathbf{y}$ is given by the sum $S(\mathbf{y}) = \sum_{u \in \mathcal{V}} \theta_u(y_u) + \sum_{(u,v) \in \mathcal{E}} \theta_{uv}(y_u, y_v)$, and its probability is given by the Gibbs distribution: $P(\mathbf{y}|\mathbf{x}) = \frac{1}{\mathcal{Z}} e^{S(\mathbf{y})}$, where $\mathcal{Z}$ is the partition function or normalization constant. The techniques proposed in this chapter are naturally applicable to higher-order CRFs. However, to simplify the exposition we only consider pairwise energies.

These unary and pairwise terms are derived from a weighted combination of features extracted at vertices and edges, *i.e.*, $\theta_u(y_u) = w_u^\intercal \phi(\mathbf{x}, y_u)$ and $\theta_{uv}(y_u, y_v) = w_{uv}^\intercal \phi(\mathbf{x}, y_u, y_v)$. Thus, this is a log-linear model, with $S(\mathbf{y}) = \mathbf{w}^\intercal \boldsymbol{\phi}(\mathbf{x}, \mathbf{y})$, where $\mathbf{w}$ are all the model parameters concatenated into a long vector, and $\boldsymbol{\phi}(\mathbf{x}, \mathbf{y})$ are all the features concatenated.

2.3 Approach: Approximate Entropy for Gibbs

We now describe our proposed active learning approach. We begin with a small number of labelled images from which an initial estimate of $\mathbf{w}$ is trained. Given a pool of unlabeled images, our goal is to find and seek annotation for the image where our current model is most uncertain.

Exact Entropy. For each unlabeled image $\mathbf{x}$, we need to compute the entropy of the conditional distribution $\mathrm{P}(\mathbf{y}|\mathbf{x})$:

$$H(\mathrm{P}) = -\mathbb{E}_{\mathrm{P}(\mathbf{y}|\mathbf{x})}[\log(\mathrm{P}(\mathbf{y}|\mathbf{x}))] = -\sum_{\mathbf{y}\in\mathcal{Y}} \mathrm{P}(\mathbf{y}|\mathbf{x}) \log \mathrm{P}(\mathbf{y}|\mathbf{x}) \tag{2.2a}$$

Computing this entropy exactly is intractable due to the summation over an exponentially-large output space $\mathcal{Y}$.

Variational Inference for Approximate Entropy. At a high level, the goal of any variational method is to construct a surrogate distribution $Q(\mathbf{y})$, and measure its entropy as an approximation to the entropy of $\mathrm{P}(\mathbf{y}|\mathbf{x})$. There are two desiderata for constructing a good surrogate:

- **Efficiency:** We should be able to quickly construct the surrogate distribution $Q(\mathbf{y})$ and compute its entropy since this computation needs to be repeatedly performed as the model learns, and the unlabeled pool of images may be very large. Thus, $Q(\mathbf{y})$ should be *compact*, and allow computation of entropy in a small number of (say $O(M)$) operations:

$$H(Q) = -\sum_{m=1}^{M} Q(\mathbf{y}^m) \log Q(\mathbf{y}^m) \tag{2.3}$$

- **Approximation Quality:** The surrogate $Q(\mathbf{y})$ should faithfully approximate the Gibbs distribution $\mathrm{P}(\mathbf{y}|\mathbf{x})$ and lead to an accurate entropy approximation, even for high level of compactness.

In the next few subsections, we look at a few different notations of compactness – first considering a standard technique and then proposing our own notion of compactness.

2.3.1 Surrogate with Stochastic Samples

Classical techniques such as Monte Carlo methods for numerically approximating integrals involve replacing the exponentially large summation with a finite sum over a small set of

M solutions $\mathbf{Y} = \{\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^M\}$, which corresponds to *sum-of-weighted-delta (SOWD) approximation* to the Gibbs distribution:

$$H(P) \approx -\sum_{m=1}^{M} \frac{e^{S(\mathbf{y}^m)}}{\mathcal{Z}_\delta} \log \frac{e^{S(\mathbf{y}^m)}}{\mathcal{Z}_\delta}, \tag{2.4}$$

where $\mathcal{Z}_\delta = \sum_{i=1}^{M} e^{S(\mathbf{y}^i)}$ is the normalizing constant of the delta-approximation.

Broadly speaking, there are two main families of methods for constructing $\mathbf{Y}$:

- **Classical Monte Carlo:** where $\mathbf{y}^i$ are samples from the distribution $\mathrm{P}(\mathbf{y}|\mathbf{x})$. Since direct sampling from undirected graphical models is hard [68], typically Markov Chain Monte Carlo (MCMC) methods such as Gibbs sampling are used, which sample from a Markov Chain whose stationary distribution is $\mathrm{P}(\mathbf{y}|\mathbf{x})$.
- **Quasi Monte Carlo:** where $\mathbf{y}^i$ are stochastic *low-discrepancy* points that try to cover the space as uniformly as possible without creating any regions with high or low density.

However, both these methods fall short. MCMC sampling based methods are slow, often requiring a long *burn-in* period before the Markov Chain converges to the stationary distribution, and even after that a large number of samples may be needed before they transition out of one mode of the distribution $\mathrm{P}(\mathbf{y}|\mathbf{x})$ to another mode. Since our goal is to estimate entropy, it is crucial that we see samples from as many modes of the distribution as possible. In our experiments, we compare to Gibbs sampling and confirm that it performs poorly. On the other hand, Quasi Monte Carlo methods completely ignore the function being summed, and may end up summing terms with insignificant effect, especially if the distribution $\mathrm{P}(\mathbf{y}|\mathbf{x})$ is non-uniform.

2.3.2 Surrogate with Deterministic Samples

Instead of running a long Markov Chain to convergence, can we efficiently find *deterministic* samples that are representative of $\mathrm{P}(\mathbf{y}|\mathbf{x})$? Specifically, can we construct a surrogate $Q(\mathbf{y})$ with support on exactly M solutions $\mathbf{Y} = \{\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^M\}$ that *optimally* approximates $\mathrm{P}(\mathbf{y}|\mathbf{x})$?

Let $Q(\mathbf{y}) = \sum_{m=1}^{M} q_m [\![\mathbf{y} = \mathbf{y}^m]\!]$, where $[\![\cdot]\!]$ is the Iverson bracket, which is 1 when the input argument is true, and 0 otherwise. Thus, $Q(\mathbf{y})$ is a sum of weighted delta functions. This surrogate is parameterized by (i) the location of the support $\mathbf{Y}$, and (ii) the weights $\mathbf{q} = \{q_1, \ldots, q_m\}$ that must clearly sum to 1. Lemma 2.1 shows that optimal support location and weights correspond to the top M highest scoring configurations in P.

Lemma 2.1. *Let* $Q(\mathbf{y}; \mathbf{Y}, \mathbf{q}) = \sum_{m=1}^{M} q_m [\![\mathbf{y} = \mathbf{y}^m]\!]$ *be a SOWD-approximation parameterized by* $\mathbf{Y}$ *and* $\mathbf{q}$*. Let* $KL(Q||\mathrm{P})$[1] $= \sum_{\mathbf{y}\in\mathcal{Y}} Q(\mathbf{y}) \log \frac{Q(\mathbf{y})}{\mathrm{P}(\mathbf{y}|\mathbf{x})}$ *denote the KL-divergence between the two distributions. The parameters* $\hat{\mathbf{Y}}, \hat{\mathbf{q}}$ *that minimize* $KL(Q||\mathrm{P})$ *are:*

$$\hat{\mathbf{y}}^m = \operatorname*{argmax}_{\mathbf{y}\in\mathcal{Y}} \ \mathrm{P}(\mathbf{y}|\mathbf{x}) \tag{2.5a}$$

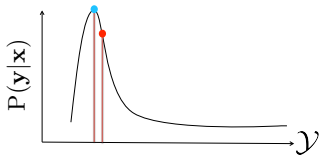
$$s.t. \ \ \mathbf{y} \neq \hat{\mathbf{y}}^{m'} \quad \forall m' < m \tag{2.5b}$$

$$\hat{q}_m = \frac{e^{S(\hat{\mathbf{y}}^m)}}{\sum_{m'=1}^{M} e^{S(\hat{\mathbf{y}}^{m'})}} \tag{2.6}$$

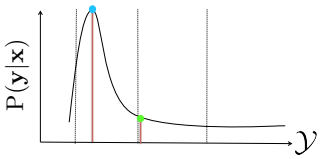
Proof 1. Using the method of Lagrangian multipliers and solving the system of partial derivates of the Lagrangian. More details in section 4.7.

Lemma 2.1 matches what we would intuitively expect – that if we need to approximate P with a set of points, these should be placed at the top M most probable locations in P. (2.13) corresponds to a problem known in the graphical models literature as the M-Best MAP [8, 88, 138]. (2.38) corresponds to normalizing the unnormalized probabilities of the M-Best MAP points to form a valid distribution.

[1] We work with $KL(Q||\mathrm{P})$ because $KL(\mathrm{P}||Q)$ is not defined when Q has more restrictive support than P.



(a) Delta approximation.



(b) Histogram approximation.

Figure 2.2: Delta vs Histogram Approximation.

Unfortunately, while this delta-surrogate may be intuitive, it suffer from some pathological behaviors. One such counter-intuitive behavior is illustrated in Fig. 2.2. Consider $M = 2$. Lemma 2.1 asks us to pick the two points shown in Fig. 2.2a since they have the highest probability under P. The scores of these two configurations are very similar and thus Q seems nearly uniform, with $H(Q) \simeq \log_2 2 = 1$. However, P is extremely peaky, and this will lead to wasted effort in annotating this image. The reason for this discrepence is that $\min_Q KL(Q||\mathrm{P})$ attempts to approximate the entire distribution P, while our primary interest is in approximating the entropy. Even if we changed the objective function, there is a second pathology. Recall that the entropy of any discrete distribution with support on M points is upper-bounded by $\log_2 M$ bits. This is a *significantly* smaller quantity than $\log_2 |\mathcal{Y}|$. For instance, in a binary image segmentation problem, the size of the state space is $|\mathcal{Y}| = 2^n$, and the maximum entropy possible is $\log_2 2^n = n$ bits. Here n is the number of super-pixels and typically around $200 - 2000$, while the number of points M is typically around 10 ($\log_2 10 = 3.29$). While we do not expect distributions over real image segmentation instances to be nearly uniform, it is clear that as soon as the entropy of P becomes larger than 3.29 bits, any uniform distribution Q that places support on *any* M points is an *optimal* entropy approximator. Clearly, such an approximation cannot be used to perform active learning.

2.3.3 Surrogate with Histogram Bins

Based on these intuitions, we propose a different notion of compactness of Q – one that still requires the same number of M parameters to represent Q, but is more representative of P globally. As illustrated in Fig. 2.2b, we partition $\mathcal{Y}$ into M non-overlapping bins and make Q a normalized histogram over these bins. Specifically, let $\{\bar{\mathbf{y}}^1, \bar{\mathbf{y}}^2, \ldots, \bar{\mathbf{y}}^M\}$ denote the bin centers, $\Delta(\mathbf{y}^1, \mathbf{y}^2)$ denote the Hamming distance between $\mathbf{y}^1$ and $\mathbf{y}^2$, and $\mathcal{Y}^m = \{\mathbf{y} \mid \Delta(\mathbf{y}, \bar{\mathbf{y}}^m) \leq r\}$ be the set of configurations that lie within bin m, which is to say that they lie within an appropriately defined r-radius distance ball of $\bar{\mathbf{y}}^m$. With this notation, we define the surrogate Q to be:

$$Q(\mathbf{y}) = \sum_{m=1}^{M} q_m [\![\mathbf{y} \in \mathcal{Y}^m]\!] = \sum_{m=1}^{M} q_m [\![\Delta(\mathbf{y}, \bar{\mathbf{y}}^m) \leq r]\!] \tag{2.7}$$

i.e. if $\mathbf{y}$ lies in the the m^{th} bin, it is assigned a probability of q_m. Note that this formulation can contain the delta-approximation as a special case with $r = 0$, if we allow for the bins to not be a complete cover of $\mathcal{Y}$ (*i.e.* $\mathcal{Y} \neq \cup_m \mathcal{Y}^m$).

We can now ask a similar question as in the previous section – what is the *optimal* histogram approximation?

Lemma 2.2. *Let* $Q(\mathbf{y}; \{\mathcal{Y}^m\}, \mathbf{q}) = \sum_{m=1}^{M} \frac{q_m}{|\mathcal{Y}^m|} [\![\mathbf{y} \in \mathcal{Y}^m]\!]$ *be a histogram-approximation parameterized by bins* $\{\mathcal{Y}^m\}$ *and weights* $\mathbf{q}$*. Let* $KL(\mathrm{P}||Q) = \sum_{\mathbf{y} \in \mathcal{Y}} \mathrm{P}(\mathbf{y}|\mathbf{x}) \log \frac{\mathrm{P}(\mathbf{y}|\mathbf{x})}{Q(\mathbf{y})}$ *denote the KL-divergence between the two distributions. For any fixed set of non-overlapping (potentially unequally sized) bins* $\{\mathcal{Y}^m\}$*, such that* $\mathcal{Y} = \cup_m \mathcal{Y}^m$*, the weights* $\hat{\mathbf{q}}$ *that minimize* $KL(\mathrm{P}||Q)$ *are:*

$$\hat{q}_m = \sum_{\mathbf{y} \in \mathcal{Y}^m} \mathrm{P}(\mathbf{y}|\mathbf{x}) = \frac{1}{\mathcal{Z}} \sum_{\mathbf{y} \in \mathcal{Y}^m} e^{S(\mathbf{y})} \tag{2.8}$$

Proof 2. Using the method of Lagrangian multipliers and solving the system of partial derivates of the Lagrangian. More details in section 4.7.

Lemma 2.2 is also intuitive in its prescription – the optimal histogram is one that represents the *mass of the Gibbs distribution* over each bin. Note that the partition function $\mathcal{Z}$ is a constant and does not depend on the bin, thus we can equivalently compute the mass of the *unnormalized* Gibbs distribution, *i.e.* $\tilde{q}_m = \sum_{\mathbf{y} \in \mathcal{Y}^m} e^{S(\mathbf{y})}$, and then simply normalize these to compute $q_m = \frac{\tilde{q}_m}{\sum_{m'} \tilde{q}_{m'}}$.

Unfortunately, the problem of estimating sums of the Gibbs distribution under general hamming-ball constraints continues to be #P-complete. Thus, we proposed to compute a simple upper-bound on the unnormalized mass:

$$\tilde{q}_m \leq |\mathcal{Y}_m| \cdot \max_{\mathbf{y} \in \mathcal{Y}^m} e^{S(\mathbf{y})}, \tag{2.9}$$

i.e. we upper-bound the mass of a bin with the maximum entry in a bin multiplied by the size of the bin. This upper bound is a good approximation if P is nearly flat over the bin, and a poor approximation if P is very peaky in the bin.

In order to compute this upper-bound, we build on recent advances in the graphical models literature for producing a set of diverse high-scoring solutions in CRFs [11, 21, 83], specifically the Parallel Diverse MAPs (PDivMAP) formulation of Meier *et al.* [83]. Let $\mathbf{y}^1 = \max_{\mathbf{y} \in \mathcal{Y}} e^{S(\mathbf{y})}$ be the MAP solution. We define M *circular bins or rings* around the MAP solution, with inner and outer radii of the rings given by L_m and U_m. This allows our histogram approximation to be distribution-specific and be "centered" around the MAP solution, which is where P places most mass. Formally, we search for the the highest scoring

configuration in a bin via the following optimization problem:

$$\mathbf{y}^m \;=\; \underset{\mathbf{y}\in\mathcal{Y}}{\operatorname{argmax}} \quad \sum_{u\in\mathcal{V}} \theta_u(y_u) + \sum_{(u,v)\in\mathcal{E}} \theta_{uv}(y_u, y_v), \;\; s.t.\; L_m \leq \Delta(\mathbf{y}, \mathbf{y}^1) \leq U_m. \tag{2.10a}$$

We set $U_m = L_{m+1}$, and chose L_m's by evenly dividing the range $[0 \;\; \max \Delta(\cdot,\cdot)]$, so that the rings cover the entire output space $\mathcal{Y}$. Meier *et al.* [83] showed that the partial Lagrangian dual of this modified formulation is easily optimizable:

$$\begin{aligned} f(\alpha_m, \beta_m) = \max_{\mathbf{y}\in\mathcal{Y}} \;\; & S_{\alpha,\beta}(\mathbf{y}) \doteq \sum_{u\in\mathcal{V}} \theta_u(y_u) + \sum_{(u,v)\in\mathcal{E}} \theta_{uv}(y_u, y_v) \\ & + \alpha_m \; \big(\Delta(\mathbf{y}, \mathbf{y}^1) - L_m\big) - \beta_m \; \big(\Delta(\mathbf{y}, \mathbf{y}^1) - U_m\big) \end{aligned} \tag{2.11}$$

where α_m, β_m are the two Lagrangian multipliers for the inner and outer radius constraints respectively. This Lagrangian dual function is easy to evaluate (and consequently minimize) since the Hamming distance function is absorbed into the node terms:

$$S_{\alpha,\beta}(\mathbf{y}) = \sum_{u\in\mathcal{V}} \underbrace{\Big(\boldsymbol{\theta}_u(y_u) + (\alpha_m - \beta_m)[\![y_u \neq y_u^1]\!]\Big)}_{\text{Perturbed Unary Score}} + \sum_{(u,v)\in\mathcal{E}} \boldsymbol{\theta}_{uv}(y_{uv}).$$

Thus, this maximization can be performed simply by feeding a perturbed unary term to the algorithm used for MAP inference (*e.g.* α-expansion or TRW-S). Lagrangian multipliers α_m, β_m can be optimized via subgradient descent, and the update rules are described in [83].

2.3.4 Summary of the algorithm

To summarize the entire algorithm, we initialize the weights of the CRF $\mathbf{w}$ by training on a small set of labeled images. Then these weights are used to compute the node and edge potentials for each image in the unlabeled pool. For each unlabeled image, we produce the

highest scoring configurations in the M circular bins $\{\mathcal{Y}^m\}$, use these to estimate the entropy and pick the unlabeled image with the highest estimated entropy. The parameters of the model $\mathbf{w}$ are then retrained and this process is repeated.

2.4 Experiments

We evaluate our approach on one synthetic and two real problems. The goal of the synthetic experiments is to perform sanity-checks and compare the performance of our algorithm when the entropy of the Gibbs distribution can be exactly computed. The goal of the real experiments is to show broad applicability and performance gains relative to other approximate inference techniques that may be used to estimate entropy.

Practical Considerations. For all experiments, we used the PDivMAP algorithm to find the highest scoring solutions in the bins, with L_m, U_m set by breaking the diversity range $[0 \ \max \Delta(\cdot,\cdot)]$ evenly into $M = 10$ bins and α_m, β_m optimized via subgradient descent. Naïvely computing $\sum_{i=1}^{M} e^{S(\mathbf{y}^i)}$ results in loss of numerical precision (underflow or overflow depending on whether $S(\mathbf{y}^i)$ were positive or negative). We used the "log-sum-exp trick", where the re-normalized distribution is computed as:

$$\tilde{q}_i = e^{\frac{S(\mathbf{y}^i)-S_{\min}}{T}} / \sum_{j=1}^{M} e^{\frac{S(\mathbf{y}^j)-S_{\min}}{T}} \tag{2.12}$$

where $S_{\min} = \min_{j\in[M]} S(\mathbf{y}^j)$ is the smallest score, and T is a "temperature" parameter. When $T = 1$, the role of $S_{\min}$ is simply to avoid numerical underflow/overflow and otherwise does not change the entropy approximation. When $T \leq 1$, the delta-approximate distribution is sharpened around the MAP (thus decreasing the estimated entropy), and when $T \geq 1$ the approximate distribution is flatted towards uniform (thus increasing the estimated entropy). We tried two different approaches to set T: (i) cross-validation (where we pick T to maximize

performance of active learning on a fully-annotated held-out set), and (ii) scaling by the score of the MAP solution, *i.e.* $T = |S(\mathbf{y}^1)|$. Interestingly, they both performed similar, suggesting that only a normalization of the scores was needed. All results reported in the chapter are from (ii).

Parameter learning: We learn $\mathbf{w}$ via Maximum (Conditional) Likelihood Estimation, optimized via Stochastic Gradient Ascent. In order to compute gradients of the likelihood, we computed marginals via sum-product loopy Belief Propagation (without damping) from Mark Schmit's UGM package [104]. We observed that BP converged in all our experiments. We also tried MCMC for computing gradients, but did not find any significant differences in the results.

In our preliminary experiments, we also tried parameter learning with max-margin objectives such as N/1-slack Structured SVMs [48, 61, 117], however the performance was not as good as MLE. We believe this is due to the fact that SSVMs are not probabilistic and lead to weight vectors and scores that are not "calibrated" probabilities. Similar observations [87, 93] have been made in the context of SVMs and Logistic Regression (the unstructured analogues of SSVM and CRFs). We believe this uncalibrated nature of scores/probabilities leads to a model whose peak (1-best) is generally accurate, but the entropy is unreliable.

Baselines. We compare our approach `Active-PDivMAP` against 7 baselines:

- `Gibbs`: we run a Gibbs sampler to produce 500 samples, and then use the delta-approximation over these samples. The burn-in period was 1000 samples.
- `Perturb-and-MAP`: we inject Gumbel noise into the node potentials, followed by MAP inference, as proposed by [89] to produce approximate samples, and then perform delta-approximation over these samples.
- `Mean-Field`: we perform variational mean-field approximation to find the fully-factorized distribution $Q_{mf}(\mathbf{y}|\mathbf{x}) = \prod_i Q_{mf}(y_i|\mathbf{x})$, which is closest to $\mathrm{P}(\mathbf{y}|\mathbf{x})$ in terms of KL-

divergence. Then we compute exact entropy in this mean-field approximation.

- `Min-Marginals`: we use the ideas from interactive segmentation literature – we compute min-marginals [67] for each super-pixel, treat this min-marginal as a measure of uncertainty at this super-pixel, and use the entropies of (normalized) min-marginals averaged over all super-pixels.
- `Marginals`: we approximate the approach of Luo *et al.* [79] by calculate the marginal probabilities at each variable, and then summing these entropies to estimate the entropy for an image. The key difference is that [79] uses convex BP and we use loopy BP (we observed that BP always converged in our experiments). To be precise, this is only an approximation of the "separate" algorithm from [79]. Unfortunately, a direct comparison against all algorithms proposed by [79] is not possible because their code is not available.
- `Margin-based` [99]: we calculate the margin (difference in scores) between the best solution and the second-best solution, and select those unlabeled images with the smallest margin.
- `Rand`: we pick an unlabeled image uniformly at random to annotate.

2.4.1 Synthetic Experiment

Setup. We generated random spanning trees on 100 nodes. All variables took two states. The node and edge potentials were sampled from Gaussians such that the true entropy lied in the range of [5 20], which represents low- and mid-level of entropy (maximum entropy possible in this setting is 100 bits). Since the graph is a tree, exact entropy can be computed via sum-product message passing. Table 2.1 compares the three sampling-based approaches – `PDivMAP`, `Gibbs`, and `Perturb-and-MAP` – in terms of:

(a) *Rank correlation:* correlation between their predicted ordering of trees and the correct

ordering according to true entropy (higher is better)

(b) *True-Rank-of-Pred:* the average rank of the tree picked by the methods to be annotated, according to true entropy (lower is better);

(c) *Pred-Rank-of-True:* average rank of the tree with the true highest entropy in the lists generated by the methods (lower is better).

Table 2.1

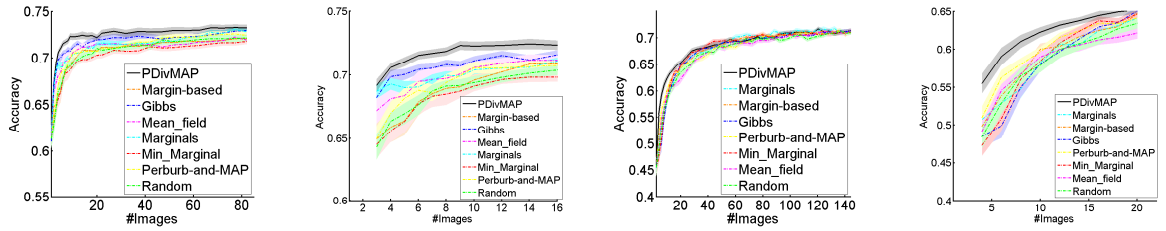
	Rank correlation (↑)	True-Rank-of-Pred (↓)	Pred-Rank-of-True (↓)
`PDivMAP`	$\mathbf{0.47 \pm 0.03}$	$\mathbf{1.9 \pm 0.18}$	$\mathbf{1.7 \pm 0.13}$
`Gibbs`	0.32 ± 0.04	6.6 ± 0.37	4.2 ± 0.18
`Perturb-and-MAP`	0.17 ± 0.04	9.5 ± 0.26	6.1 ± 0.25

In all three metrics, we can see that `PDivMAP` significantly outperforms the baselines.

2.4.2 Foreground-Background Segmentation

Setup. We tested our algorithm on the problem of binary (foreground-background) image segmentation. We replicated the experimental setup of [47, 49]. We used the co-segmentation dataset iCoseg [10], which consists of 37 groups of related images mimicking typical consumer photograph collections. Each group may be thought of as an "event" (*e.g.*, images from a baseball game, a safari, *etc.*). The dataset provides pixel-level ground-truth foreground-background segmentations for each image. We used 9 difficult groups containing 166 images in total. These images were then split into `train` and `test` sets of equal size. We initialize with 1 annotated image, perform active learning on the `train` set, and use the `test` to report accuracies.

Model and Features. The segmentation task is modeled as a binary pairwise CRF where each node corresponds to a superpixel [1] in the image. We extracted 12-dim color features at each superpixel (mean RGB; mean HSV; 5 bin Hue histogram; Hue histogram entropy). The edge features, computed for each pair of adjacent superpixels, correspond to a standard Potts model and a contrast sensitive Potts model. The weights at each edge were constrained to be positive so that the resulting supermodular potentials could be maximized via graph-



(a) Binary Segmentation (b) Binary Segmentation: Zoomed (c) Geometric Labeling (d) Geometric Labeling: Zoomed

Figure 2.3: Accuracy vs number of images annotated. Shaded regions indicate confidence intervals, achieved from 20 (top) and 30 runs (bottom). We can see that our approach `Active-PDivMAP` outperforms all baselines and is very quickly able to reach the same performance as annotating the entire dataset.

cuts [15, 69].

2.4.3 Geometric Labeling

Dataset. We used CMU Geometric Context dataset of Hoiem *et al*. [56], where every region is categorized into one of three main classes: "ground", "sky", and "vertical". The "vertical" class is further divided into 5 subclasses: "left", "center", "right", "porous", and "solid". These images were then split into `train` and `test` sets with 150 and 50 images respectively. We initialized with 2 annotated images, performed active learning on the `train` set, and use the `test` to report accuracies. The segmentation task is modeled as a pairwise CRF where each node corresponds to a superpixel in the image that can take 7 states.

2.4.4 Results and Analysis

Quantitative Results and Take-Home Message. For both experiments, we ran multiple runs with different initial images (30 runs for binary segmentation and 20 runs for geometric labeling). Fig. 2.3 shows the accuracy of various methods vs the number of images annotated for both datasets (shaded regions indicate confidence intervals). Note that the performance of a "fully supervised" approach is the rightmost point on the curve. We can see that

our approach `Active-PDivMAP` significantly outperforms all the baselines, with no overlap in confidence intervals. Moreover, `Active-PDivMAP` is able to reach within 1%-points of the final accuracy (where all images have been annotated) with less than 9% of the data annotated. Based on Mechanical Turk annotation statistics reported in previous work [110], a simple back-of-the-envelope calculation – assuming 3-minutes per image, 10-cents per image $\times$ 5 MTurk annotations per image – show that our approach saved approximately 45 hours of human-effort and \$35 – even for these medium-sized dataset.

Overall, outperforming `Rand` shows that even though it may be crude, the histogram approximation does capture enough information about the entropy to be useful. Outperforming `Gibbs` and `Perturb-and-Map` shows the power of using non-overlapping bins as opposed to IID samples, and outperforming `Mean-Field`, `Min-Marginals` and `Marginals` shows that it is better to approximate the entropy computation with a histogram approximation than with a fully factorized model.

Efficiency and Runtime. Due to reliance on efficient MAP solvers (e.g dynamic graph-cuts in binary segmentation), our implementation has fairly low overhead. Specifically, 60 subgradient iterations $\times$ 10 solutions takes 1.8s, which is much less than Gibbs (40s for 500 samples), comparable to LoopyBP (1.2s), MeanField (1.4s), and slower than Margin-based (0.12s), and MinMarginals (0.08s).

2.5 Appendix

2.5.1 Proof of Lemma 1

Lemma 2.3. *Let* $Q(\mathbf{y}; \mathbf{Y}, \mathbf{q}) = \sum_{m=1}^{M} q_m [\![\mathbf{y} = \mathbf{y}^m]\!]$ *be a SOWD-approximation parameterized by* $\mathbf{Y}$ *and* $\mathbf{q}$*. Let* $KL(Q||\mathrm{P}) = \sum_{\mathbf{y}\in\mathcal{Y}} Q(\mathbf{y}) \log \frac{Q(\mathbf{y})}{\mathrm{P}(\mathbf{y}|\mathbf{x})}$ *denote the KL-divergence between the two distributions. The parameters* $\hat{\mathbf{Y}}, \hat{\mathbf{q}}$ *that minimize* $KL(Q||\mathrm{P})$ *are:*

$$\hat{\mathbf{y}}^m = \underset{\mathbf{y}\in\mathcal{Y}}{\text{argmax}} \;\; \mathrm{P}(\mathbf{y}|\mathbf{x}) \tag{2.13a}$$

$$s.t. \;\; \mathbf{y} \neq \hat{\mathbf{y}}^{m'} \quad \forall m' < m \tag{2.13b}$$

$$\hat{q}_m = \frac{e^{S(\hat{\mathbf{y}}^m)}}{\sum_{m'=1}^{M} e^{S(\hat{\mathbf{y}}^{m'})}} \tag{2.14}$$

Proof 3.

$$KL(Q||\mathrm{P}) = \sum_{\mathbf{y}\in\mathcal{Y}} Q(\mathbf{y}) \log \frac{Q(\mathbf{y})}{\mathrm{P}(\mathbf{y}|\mathbf{x})} \tag{2.15a}$$

$$= \sum_{\mathbf{y}\in\mathbf{Y}} Q(\mathbf{y}) \log \frac{Q(\mathbf{y})}{\mathrm{P}(\mathbf{y}|\mathbf{x})} + \sum_{\mathbf{y}\in\mathcal{Y}\setminus\mathbf{Y}} Q(\mathbf{y}|\mathbf{x}) \log \frac{Q(\mathbf{y})}{\mathrm{P}(\mathbf{y}|\mathbf{x})} \tag{2.15b}$$

$$= \sum_{\mathbf{y}\in\mathbf{Y}} Q(\mathbf{y}) \log \frac{Q(\mathbf{y})}{\mathrm{P}(\mathbf{y}|\mathbf{x})} \tag{2.15c}$$

Thus, we have

$$\min_{\mathbf{Y},\mathbf{q}} KL(Q||\mathrm{P}) \Rightarrow \begin{cases} \min\limits_{q_m,\mathbf{y}^m} & \sum\limits_{m=1}^{M} q_m \log \frac{q_m}{\mathrm{P}(\mathbf{y}^m|\mathbf{x})} \\ s.t. & \sum\limits_{m=1}^{M} q_m = 1 \end{cases} \tag{2.16}$$

We can write the Lagrangian for (2.16) as

$$L(\mathbf{Y}, \mathbf{q}, \lambda) = \sum_{m=1}^{M} q_m \log \frac{q_m}{\mathrm{P}(\mathbf{y}^m|\mathbf{x})} + \lambda \cdot \left(\sum_{m=1}^{M} q_m - 1 \right) \tag{2.17}$$

Method of Lagrangian multipliers involves setting the derivative of L w.r.t q_m to 0,

$$\frac{\partial L}{\partial q_m} = \log \frac{q_m}{\mathrm{P}(\mathbf{y}^m|\mathbf{x})} + 1 + \lambda = 0 \tag{2.18}$$

Thus,

$$q_m = e^{-1-\lambda} \mathrm{P}(\mathbf{y}^m|\mathbf{x}) \tag{2.19}$$

Using the fact that $\sum_{m=1}^{M} q_m = 1$, we can show that $\lambda = \log\left(\sum_{m=1}^{M} \mathrm{P}(\mathbf{y}^m|\mathbf{x})\right) - 1$. Thus,

$$q_m = \frac{\mathrm{P}(\mathbf{y}^m|\mathbf{x})}{\sum_{m'=1}^{M} \mathrm{P}(\mathbf{y}^{m'}|\mathbf{x})} \tag{2.20}$$

Plugging this definition of q_m in objective function of (2.16), we get

$$\sum_{m=1}^{M} q_m \log \frac{q_m}{\mathrm{P}(\mathbf{y}^m|\mathbf{x})} \tag{2.21}$$

$$= \sum_{m=1}^{M} \frac{\mathrm{P}(\mathbf{y}^m|\mathbf{x})}{\sum_{m'=1}^{M} \mathrm{P}(\mathbf{y}^{m'}|\mathbf{x})} \log \frac{\frac{\mathrm{P}(\mathbf{y}^m|\mathbf{x})}{\sum_{m'=1}^{M} \mathrm{P}(\mathbf{y}^{m'}|\mathbf{x})}}{\mathrm{P}(\mathbf{y}^m|\mathbf{x})} \tag{2.22}$$

$$= \sum_{m=1}^{M} \frac{\mathrm{P}(\mathbf{y}^m|\mathbf{x})}{\sum_{m'=1}^{M} \mathrm{P}(\mathbf{y}^{m'}|\mathbf{x})} \log \frac{1}{\sum_{m'=1}^{M} \mathrm{P}(\mathbf{y}^{m'}|\mathbf{x})} \tag{2.23}$$

$$= \left(\sum_{m=1}^{M} \mathrm{P}(\mathbf{y}^m|\mathbf{x})\right) \cdot \left(\frac{1}{\sum_{m'=1}^{M} \mathrm{P}(\mathbf{y}^{m'}|\mathbf{x})} \log \frac{1}{\sum_{m'=1}^{M} \mathrm{P}(\mathbf{y}^{m'}|\mathbf{x})}\right) \tag{2.24}$$

$$= -\log \sum_{m'=1}^{M} \mathrm{P}(\mathbf{y}^{m'}|\mathbf{x}) \tag{2.25}$$

Thus,

$$\min_{\mathbf{Y},\mathbf{q}} KL(Q||\mathrm{P}) \Rightarrow \max_{\mathbf{Y}} \sum_{\mathbf{y}\in\mathbf{Y}} \mathrm{P}(\mathbf{y}|\mathbf{x}) \tag{2.26}$$

Clearly, $\sum_{\mathbf{y}\in\mathbf{Y}} \mathrm{P}(\mathbf{y}|\mathbf{x})$ is maximized by picking the top M probability locations in P,

$$\hat{\mathbf{y}}^m = \operatorname*{argmax}_{\mathbf{y}\in\mathcal{Y}} \mathrm{P}(\mathbf{y}|\mathbf{x}) \tag{2.27}$$

$$s.t.\ \ \mathbf{y} \neq \hat{\mathbf{y}}^{m'} \quad \forall m' < m \tag{2.28}$$

$$\hat{q}_m = \frac{\mathrm{P}(\hat{\mathbf{y}}^m|\mathbf{x})}{\sum_{m'=1}^{M} \mathrm{P}(\hat{\mathbf{y}}^{m'}|\mathbf{x})} = \frac{e^{S(\hat{\mathbf{y}}^m)}}{\sum_{m'=1}^{M} e^{S(\hat{\mathbf{y}}^{m'})}} \tag{2.29}$$

This completes the proof. We can see that the optimal Q is a normalized distribution over the top M most probable locations in P.

2.5.2 Proof of Lemma 2

Lemma 2.4. *Let* $Q(\mathbf{y}; \{\mathcal{Y}^m\}, \mathbf{q}) = \sum_{m=1}^{M} \frac{q_m}{|\mathcal{Y}^m|} [\![\mathbf{y} \in \mathcal{Y}^m]\!]$ *be a histogram-approximation parameterized by bins* $\{\mathcal{Y}^m\}$ *and weights* $\mathbf{q}$*. Let* $KL(\mathrm{P}||Q) = \sum_{\mathbf{y}\in\mathcal{Y}} \mathrm{P}(\mathbf{y}|\mathbf{x}) \log \frac{\mathrm{P}(\mathbf{y}|\mathbf{x})}{Q(\mathbf{y})}$ *denote the KL-divergence between the two distributions. For any fixed set of non-overlapping (potentially unequally sized) bins* $\{\mathcal{Y}^m\}$*, such that* $\mathcal{Y} = \cup_m \mathcal{Y}^m$*, the weights* $\hat{\mathbf{q}}$ *that minimize* $KL(\mathrm{P}||Q)$ *are:*

$$\hat{q}_m = \sum_{\mathbf{y}\in\mathcal{Y}^m} \mathrm{P}(\mathbf{y}|\mathbf{x}) = \frac{1}{\mathcal{Z}} \sum_{\mathbf{y}\in\mathcal{Y}^m} e^{S(\mathbf{y})} \tag{2.30}$$

Proof 4.

$$KL(\mathrm{P}||Q) = \sum_{m=1}^{M} \sum_{\mathbf{y}\in\mathcal{Y}^m} \mathrm{P}(\mathbf{y}|\mathbf{x}) \log \left(\frac{\mathrm{P}(\mathbf{y}|\mathbf{x})}{\frac{q_m}{|\mathcal{Y}^m|}} \right) \tag{2.31}$$

$$= \sum_{m=1}^{M} \sum_{\mathbf{y}\in\mathcal{Y}^m} \mathrm{P}(\mathbf{y}|\mathbf{x}) \log \left(\mathrm{P}(\mathbf{y}|\mathbf{x}) \cdot |\mathcal{Y}^m|\right) - \sum_{m=1}^{M} \sum_{\mathbf{y}\in\mathcal{Y}^m} \mathrm{P}(\mathbf{y}|\mathbf{x}) \log q_m \tag{2.32}$$

$$= h - \sum_{m=1}^{M} \sum_{\mathbf{y}\in\mathcal{Y}^m} \mathrm{P}(\mathbf{y}|\mathbf{x}) \log q_m \tag{2.33}$$

$$= h - \sum_{m=1}^{M} p_m \log q_m \tag{2.34}$$

where, h is a constant and $p_m = \sum_{\mathbf{y} \in \mathcal{Y}^m} \mathrm{P}(\mathbf{y}|\mathbf{x})$, i.e., the mass of the distribution in the bin m. Thus,

$$\min_{\mathbf{q}} KL(\mathrm{P}||Q) \Rightarrow \begin{cases} \min\limits_{q_m} & -\sum\limits_{m=1}^{M} p_m \log q_m \\ s.t., & \sum\limits_{m=1}^{M} q_m = 1 \end{cases} \tag{2.35}$$

We can write the Lagrangian for (2.35) as

$$L(\mathbf{q}, \lambda) = -\sum_{m=1}^{M} p_m \log q_m + \lambda \left(\sum_{m=1}^{M} q_m - 1 \right) \tag{2.36}$$

Method of Lagrangian multipliers involves setting the derivative of L w.r.t q_m to 0,

$$\frac{\partial L}{\partial q_m} = -\frac{p_m}{q_m} + \lambda = 0 \Rightarrow q_m = \frac{p_m}{\lambda} \tag{2.37}$$

Using the fact that $\sum_{m=1}^{M} q_m = \sum_{m=1}^{M} \frac{p_m}{\lambda} = \frac{1}{\lambda} = 1$, we can show that $\lambda = 1$. Thus, $\hat{q}_m = p_m$, i.e.,

$$\hat{q}_m = \sum_{\mathbf{y} \in \mathcal{Y}^m} \mathrm{P}(\mathbf{y}|\mathbf{x}) = \frac{1}{\mathcal{Z}} \sum_{\mathbf{y} \in \mathcal{Y}^m} e^{S(\mathbf{y})} \tag{2.38}$$

This completes the proof. We can see that the optimal Q is a normalized histogram over M bins.

2.5.3 Qualitative Results

Fig. 2.4,2.5 show example images with the most uncertainty/certainty according to our approach `Active-PDivMAP` and `Gibbs`, under a model trained with 5 images from 2 random trails in the experiment section.

We can see that `Gibbs` has difficulty transitioning out of one mode to reach another mode.

As a result, almost all sampled segmentations look visually very similar, and the estimated distribution/histogram is nearly uniform. From these two examples, we can see that `Gibbs` will typically pick images where the MAP is already pretty accurate – the model will seem uncertain because `Gibbs` is picking samples that are all very similar to MAP.

In contrast, our approach `Active-PDivMAP` can pick images (first row) for which the set of plausible segmentations (or histogram bin centers) are truly diverse, but have similar energies. Such images are much more helpful in updating the beliefs of the model in an active learning setting.

Note that in both examples, our approach estimates the entropy of the most uncertain image to be ≈ 2.29 (compared to the maximum possible entropy of a 10-D probability mass function $\log 10 = 2.30$).

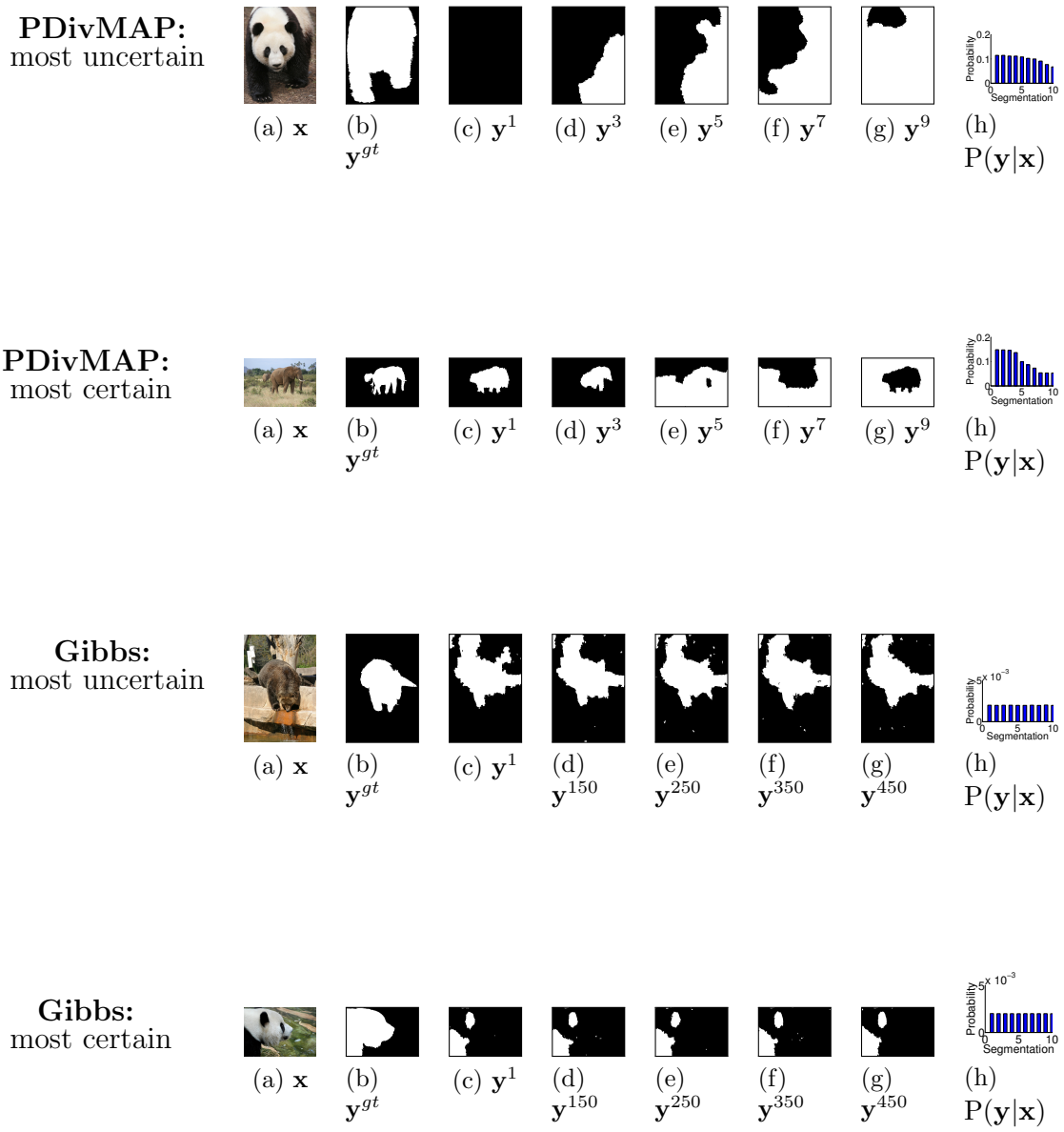


Figure 2.4: Example 1: First/second row shows the image with the most uncertainty/certainty, as estimated by our approach `Active-PDivMAP`. Third/fourth row shows the image with the most uncertainty/certainty, as estimated by `Gibbs`. We can see that `Gibbs` has difficulty transitioning out of one mode to reach another mode. Thus, almost all sampled segmentations of the most uncertain image look visually very similar. In contrast, our approach `Active-PDivMAP` can pick images (first row) for which the set of plausible segmentations (or histogram bin centers) are truly diverse, but have similar energies. This identifies images where the model is truly uncertain, and such images are helpful in updating the beliefs of the model.

Figure 2.5: Example 2: First/second row shows the image with the most uncertainty/certainty, as estimated by our approach `Active-PDivMAP`. Third/fourth row shows the image with the most uncertainty/certainty, as estimated by `Gibbs`. We can see that `Gibbs` has difficulty transitioning out of one mode to reach another mode. Thus, almost all sampled segmentations of the most uncertain image look visually very similar. In contrast, our approach `Active-PDivMAP` can pick images (first row) for which the set of plausible segmentations (or histogram bin centers) are truly diverse, but have similar energies. This identifies images where the model is truly uncertain, and such images are helpful in updating the beliefs of the model.

Chapter 3

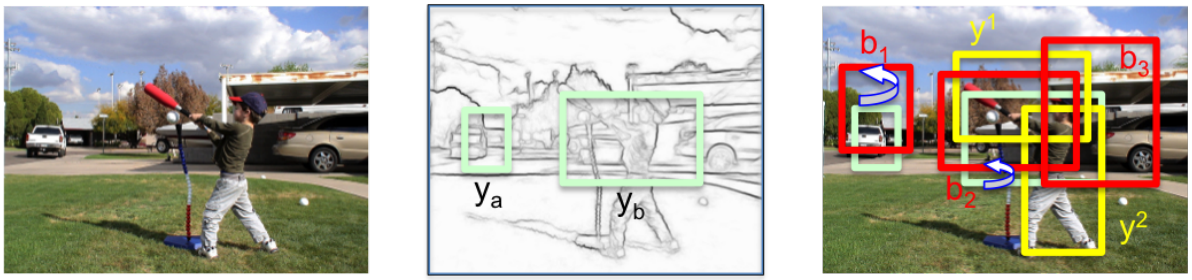
Near-Optimal Search for a Set of Diverse Object Proposals.

A number of problems in Computer Vision and Machine Learning involve searching for a set of bounding boxes or rectangular windows. For instance, in object detection [30, 42, 44, 52, 105, 112, 113], the goal is to output a set of bounding boxes localizing all instances of a *particular* object category. In object proposal generation [4, 20, 118, 142], the goal is to output a set of candidate bounding boxes that may potentially contain an object (of *any* category). Other scenarios include face detection, multi-object tracking and weakly supervised learning [32].

Classical Approach: Enumeration + Diverse Subset Selection. In the context of object detection, the classical paradigm for searching for a set of bounding boxes used to be:

- **Sliding Window** [30, 42, 128]: *i.e.*, enumeration over all windows in an image with some level of sub-sampling, followed by
- **Non-Maximal Suppression (NMS)**: *i.e.*, picking a spatially-diverse set of windows by suppressing windows that are too close or overlapping.

As several previous works [14, 74, 128] have recognized, the problem with this approach is inefficiency – the number of possible bounding boxes or rectangular subwindows in an image is $O(\#pixels^2)$. Even a low-resolution (320 x 240) image contains more than *one billion*



(a) Image (b) Relevance term (c) Diversity term

Figure 3.1: Overview of our formulation: Submodboxes. We formulate the selection of a set of boxes as a constrained submodular maximization problem. The objective and marginal gains consists of two parts: relevance and diversity. Figure (b) shows two candidate windows $\mathbf{y}_a$ and $\mathbf{y}_b$. Relevance is the sum of edge strength over all edge groups (black curves) wholly enclosed in the window. Figure (c) shows the diversity term. The marginal gain in diversity due to a new window ($\mathbf{y}_a$ or $\mathbf{y}_b$) is the ability of the new window to *cover* the reference boxes that are currently not well-covered with the already chosen set $Y = \{\mathbf{y}^1, \mathbf{y}^2\}$. In this case, we can see that $\mathbf{y}_a$ covers a new reference box b_1. Thus, the marginal gain in diversity of $\mathbf{y}_a$ will be larger than $\mathbf{y}_b$.

rectangular windows [74]!

As a result, modern object detection pipelines [44, 52, 113] often rely on object proposals as a pre-processing step to reduce the number of candidate object locations to a few hundreds or thousands (rather than billions).

Interestingly, this migration to object proposals has simply *pushed the problem (of searching for a set of bounding boxes) upstream.* Specifically, a number of object proposal techniques [24, 97, 142] involve the same enumeration + NMS approach – except they typically use cheaper features to be a fast proposal generation step.

Goal. The goal of this chapter is to formally study the search for a set of bounding boxes as an optimization problem. Clearly, enumeration + post-processing for diversity (via NMS) is one widely-used heuristic approach. Our goal is to formulate a formal optimization objective and propose an efficient algorithm, ideally with guarantees on optimization performance.

Challenge. The key challenge is the exponentially-large search space – the number of possible M-sized sets of bounding boxes is $\binom{O(\#pixels^2)}{M} = O(\#pixels^{2M})$ (assuming $M \leq \#pixels^2/2$).

Overview of our formulation: Submodboxes. Let $\mathcal{Y}$ denote the set of all possible bounding boxes or rectangular subwindows in an image. This is a structured output space [13, 61, 115], with the size of this set growing quadratically with the size of the input image, $|\mathcal{Y}| = O(\#pixels^2)$.

We formulate the selection of a set of boxes as a search problem on the power set $2^{\mathcal{Y}}$. Specifically, given a budget of M windows, we search for a set Y of windows that are both *relevant* (*e.g.*, have high likelihood of containing an object) and *diverse* (to cover as many objects instances as possible):

$$\underbrace{\underset{Y \in 2^{\mathcal{Y}}}{\operatorname{argmax}}}_{\text{search over power-set}} \quad \underbrace{F(Y)}_{\text{objective}} = \underbrace{R(Y)}_{\text{relevance}} + \underbrace{\lambda}_{\text{trade-off parameter}} \underbrace{D(Y)}_{\text{diversity}} \qquad s.t. \quad \underbrace{|Y| \leq M}_{\text{budget constraint}} \tag{3.1}$$

Crucially, when the objective function $F : 2^{\mathcal{Y}} \rightarrow \mathbb{R}$ is *monotone* and *submodular*, then a simple greedy algorithm (that iteratively adds the window with the largest *marginal gain* [71]) achieves a near-optimal approximation factor of $(1 - \frac{1}{e})$ [71, 86].

Unfortunately, although conceptually simple, this greedy augmentation step requires an enumeration over the space of all windows $\mathcal{Y}$ and thus a naïve implementation is intractable.

In this work, we show that for a broad class of relevance and diversity functions, this greedy augmentation step may be efficiently formulated as a Branch-and-Bound (B&B) step [35, 74], with easily computable upper-bounds. This enables an efficient implementation of greedy, with significantly fewer evaluations than a linear scan over $\mathcal{Y}$.

Finally, in order to speed up repeated application of B&B across iterations of the greedy algorithm, we present a novel generalization of Minoux's 'lazy greedy' algorithm [85] to the B&B tree, where different branches are explored in a lazy manner in each iteration.

We apply our proposed technique Submodboxes to the task of generating object proposals [4, 20, 118, 142] on the PASCAL VOC 2007 [36], PASCAL VOC 2012 [38], and MS COCO [77]

datasets. Our results show that our approach outperforms all baselines.

Contributions. This chapter makes the following contributions:

1. We formulate the search for a set of bounding boxes or subwindows as the constrained maximization of a monotone submodular function. To the best of our knowledge, despite the popularity of object recognition and object proposal generation, this is the first such formal optimization treatment of the problem.
2. Our proposed formulation contains existing heuristics *as special cases*. Specifically, Sliding Window + NMS can be viewed as an instantiation of our approach under a specific definition of the diversity function $D(\cdot)$.
3. Our work can be viewed as a generalization of the 'Efficient Subwindow Search (ESS)' of Lampert *et al*. [74], who proposed a B&B scheme for finding the *single* best bounding box in an image. Their extension to detecting multiple objects consisted of a heuristic for 'suppressing' features extracted from the selected bounding box and re-running the procedure. We show that this heuristic is a special case of our formulation under a specific diversity function, thus providing theoretical justification to their intuitive heuristic.
4. To the best of our knowledge, our work presents the first generalization of Minoux's 'lazy greedy' algorithm [85] to structured-output spaces (the space of bounding boxes).
5. Finally, our experimental contribution is a novel diversity measure which leads to state-of-art performance on the task of generating object proposals.

3.1 Related Work

Our work is related to a few different themes of research in Computer Vision and Machine Learning.

Submodular Maximization and Diversity. The task of searching for a diverse high-

quality subset of items from a ground set has been well-studied in a number of application domains [19, 33, 66, 72, 75, 94], and across these domains submodularity has emerged as an a fundamental property of set functions for measuring diversity of a subset of items. Most previous work has focussed on submodular maximization on *unstructured spaces*, where the ground set is efficiently enumerable.

Our work is closest in spirit to Prasad *et al*. [94], who studied submodular maximization on *structured* output spaces, *i.e*. where each item in the ground set is itself a structured object (such as a segmentation of an image). Unlike [94], our ground set $\mathcal{Y}$ is not exponentially large, only 'quadratically' large. However, enumeration over the ground set for the greedy-augmentation step is still infeasible, and thus we use B&B. Such structured output spaces and greedy-augmentation oracles were not explored in [94].

Bounding Box Search in Object Detection and Object Proposals. As we mention in the introduction, the search for a set of bounding boxes via heuristics such as Sliding Window + NMS used to be the dominant paradigm in object recognition [30, 42, 128]. Modern pipelines have shifted that search step to object proposal algorithms [44, 52, 113]. A comparison and overview of object proposals may be found in [57]. Zitnick *et al*. [142] generate candidate bounding boxes via Sliding Window + NMS based on an "objectness" score, which is a function of the number of contours wholly enclosed by a bounding box. We use this objectness score as our relevance term, thus making Submodboxes directly comparable to NMS. Another closely related work is [46], which presents an 'active search' strategy for reranking selective search [118] object proposals based on a contextual cues. Unlike this work, our formulation is not restricted to any pre-selected set of windows. We search over the entire power set $2^{\mathcal{Y}}$, and may generate any possible set of windows (up to convergence tolerance in B&B).

Branch-and-Bound. One key building block of our work is the 'Efficient Subwindow Search

(ESS)' B&B scheme *et al*. [74]. ESS was originally proposed for single-instance object detection. Their extension to detecting multiple objects consisted of a heuristic for 'suppressing' features extracted from the selected bounding box and re-running the procedure. In this work, we extend and generalize ESS in multiple ways. First, we show that relevance (objectness scores) and diversity functions used in object proposal literature are amenable to upper-bound and thus B&B optimization. We also show that the 'suppression' heuristic used by [74] is a special case of our formulation under a specific diversity function, thus providing theoretical justification to their intuitive heuristic. Finally, [14] also proposed the use of B&B for NMS in object detection. Unfortunately, as we explain later in the chapter, the NMS objective is submodular *but not monotone*, and the classical greedy algorithm does not have approximation guarantees in this setting. In contrast, our work presents a general framework for bounding-box subset-selection based on *monotone* submodular maximization.

3.2 Formulation and Approach

We begin by establishing the notation used in the chapter.

Preliminaries and Notation. For an input image $\mathbf{x}$, let $\mathcal{Y}_\mathbf{x}$ denote the set of all possible bounding boxes or rectangular subwindows in this image. For simplicity, we drop the explicit dependance on $\mathbf{x}$, and just use $\mathcal{Y}$. Uppercase letters refer to set functions $F(\cdot), R(\cdot), D(\cdot)$, and lowercase letter refer to functions over individual items $f(\mathbf{y}), r(\mathbf{y})$.

A set function $F : 2^\mathcal{Y} \rightarrow \mathbb{R}$ is submodular if its *marginal gains* $F(b|S) \equiv F(S \cup b) - F(S)$ are decreasing, *i.e.* $F(b|S) \geq F(b|T)$ for all sets $S \subseteq T \subseteq \mathcal{Y}$ and items $b \notin T$. The function F is called *monotone* if adding an item to a set does not hurt, *i.e.* $F(S) \leq F(T),\ \forall S \subseteq T$.

Constrained Submodular Maximization. From the classical result of Nemhauser [86], it is known that cardinality constrained maximization of a monotone submodular F can be performed near-optimally via a greedy algorithm. We start out with an empty set $Y^0 = \emptyset$,

and iteratively add the next 'best' item with the largest marginal gain over the chosen set :

$$Y^t = Y^{t-1} \cup y^t, \qquad \text{where} \qquad y^t = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \; F(y \mid Y^{t-1}). \tag{3.2}$$

The score of the final solution Y^M is within a factor of $(1 - \frac{1}{e})$ of the optimal solution. The computational bottleneck is that in each iteration, we must find the item with the largest marginal gain. In our case, $|\mathcal{Y}|$ is the space of all rectangular windows in an image, and exhaustive enumeration is intractable. Instead of exploring subsampling as is done in Sliding Window methods, we will formulate this greedy augmentation step as an optimization problem solved with B&B.

Sets vs Lists. For pedagogical reasons, our problem setup is motivated with the language of sets $(\mathcal{Y}, 2^{\mathcal{Y}})$ and subsets (Y). In practice, our work falls under submodular *list* prediction [33, 98, 111]. The generalization from sets to lists allows reasoning about an ordering of the items chosen and (potentially) repeated entries in the list. Our final solution Y^M is an (ordered) list not an (unordered) set. All guarantees of greedy remain the same in this generalization [33, 98, 111].

3.2.1 Parameterization of $\mathcal{Y}$ and Branch-and-Bound Search

In this subsection, we briefly recap the Efficient Subwindow Search (ESS) of Lampert *et al*. [74], which is used a key building block in this work. The goal of [74] is to maximize a (potentially non-smooth) objective function over the space of all rectangular windows: $\max_{\mathbf{y} \in \mathcal{Y}} f(\mathbf{y})$.

A rectangular window $\mathbf{y} \in \mathcal{Y}$ is parameterized by its top, bottom, left, and right coordinates $\mathbf{y} = (t, b, l, r)$. A *set* of windows is represented by using intervals for each coordinate instead of a single integer, for example $[T, B, L, R]$, where $T = [t_{low}, t_{high}]$ is a range. In this parameterization, the set of all possible boxes in an $(h \times w)$-sized image can be written as

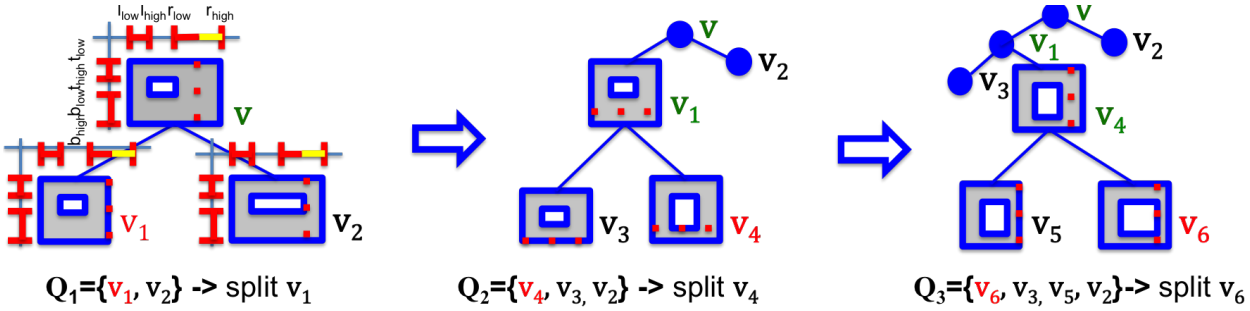


Figure 3.2: Priority queue in B&B scheme. Each vertex in the tree represents a set of windows. Blue rectangles denote the largest and the smallest window in the set. Gray region denotes the rectangle set $\mathcal{Y}_v$. In each case, the priority queue consists of all leaves in the B&B tree ranked by the upper bound U_v. *Left:* shows vertex v is split along the *right* coordinate interval into equal halves: v_1 and v_2. *Middle:* The highest priority vertex v_1 in Q_1 is further split along *bottom* coordinate into v_3 and v_4. *Right:* The highest priority vertex v_4 in Q_2 is split along *right* coordinate into v_5 and v_6. This procedure is repeated until the highest priority vertex in the queue is a single rectangle.

$\mathcal{Y} = [[1, h], [1, h], [1, w], [1, w]]$.

Branch-and-Bound over $\mathcal{Y}$. ESS creates a B&B tree, where each vertex v in the tree is a rectangle set $\mathcal{Y}_v$ and an associated upper-bound on the objective function achievable in this set, *i.e.* $\max_{\mathbf{y}\in\mathcal{Y}_v} f(\mathbf{y}) \leq U_v$. Initially, this tree consists of a single vertex, which is the entire search space $\mathcal{Y}$ and (typically) a loose upper-bound. ESS proceeds in a best-first manner [74]. In each iteration, the vertex/set with the highest upper-bound is chosen for branching, and then new upper-bounds are computed on each of the two children/sub-sets created. In practice, this is implemented with a priority queue over the vertices/sets that are currently leaves in the tree. Fig. 3.2 shows an illustration of this procedure. The parent rectangle set is split along its largest coordinate interval into two equal halves, thus forming disjoint children sets. B&B explores the tree in a best-first manner till a single rectangle is identified with a score *equal* to the upper-bound at which point we have found a global optimum. In our experiments, we show results with different convergence tolerances.

Objective. In our setup, the objective (at each greedy-augmentation step) is the marginal gain of the window $\mathbf{y}$ w.r.t. the currently chosen list of windows Y^{t-1}, *i.e.* $f(\mathbf{y}) = F(\mathbf{y} \mid Y^{t-1}) = R(\mathbf{y} \mid Y^{t-1}) + \lambda D(\mathbf{y} \mid Y^{t-1})$. In the following subsections, we describe the relevance

and diversity terms in detail, and show how upper bounds can be efficiently computed over the sets of windows.

3.2.2 Relevance Function and Upper Bound

The goal of the relevance function $R(Y)$ is to quantify the "quality" or "relevance" of the windows chosen in Y. In our work, we define $R(Y)$ to be a *modular* function aggregating the quality of all chosen windows *i.e.* $R(Y) = \sum_{\mathbf{y} \in Y} r(\mathbf{y})$. Thus, the marginal gain of window $\mathbf{y}$ is simply its individual quality regardless of what else has already been chosen, *i.e.* $R(\mathbf{y} \mid Y^{t-1}) = r(\mathbf{y})$.

In our application of object proposal generation, we use the objectness score produced by EdgeBoxes [142] as our relevance function. The main intuition of EdgeBoxes is that the number of contours or "edge groups" wholly contained in a box is indicative of its objectness score. Thus, it first creates a grouping of edge pixels called edge groups, each associated with a real-valued edge strength s_i.

Abstracting away some of the domain-specific details, EdgeBoxes essentially defines the score of a box as a weighted sum of the strengths of edge groups contained in it, normalized by the size of the box *i.e.* $\text{EdgeBoxesScore}(\mathbf{y}) = \frac{\sum_{\text{edge group } i \in \mathbf{y}} w_i s_i}{\text{size-normalization}}$, where with a slight abuse of notation, we use 'edge group $i \in \mathbf{y}$' to mean the edge groups contained the rectangle $\mathbf{y}$.

These weights and size normalizations were found to improve performance of EdgeBoxes. In our work, we use a simplification of the EdgeBoxesScore which allow for easy computation of upper bounds:

$$r(\mathbf{y}) = \frac{\sum_{\text{edge group } i \in \mathbf{y}} s_i}{\text{size-normalization}}, \tag{3.3}$$

i.e., we ignore the weights. One simple upper-bound for a set of windows $\mathcal{Y}_v$ can be computed

by accumulating *all possible* positive scores and *the least necessary* negative scores:

$$\max_{\mathbf{y}\in\mathcal{Y}_v} r(\mathbf{y}) \leq \frac{\sum_{\text{edge group } i\in\mathbf{y}_{\max}} s_i \cdot [\![s_i \geq 0]\!] + \sum_{\text{edge group } i\in\mathbf{y}_{\min}} s_i \cdot [\![s_i \leq 0]\!]}{\text{size-normalization}(\mathbf{y}_{\min})}, \tag{3.4}$$

where $\mathbf{y}_{\max}$ is the largest and $\mathbf{y}_{\min}$ is the smallest box in the set $\mathcal{Y}_v$; and $[\![\cdot]\!]$ is the Iverson bracket.

Consistent with the experiments in [142] , we found that this simplification indeed hurts performance in the EdgeBoxes Sliding Window + NMS pipeline. However, interestingly we found that even with this weaker relevance term, Submodboxes was able to outperform EdgeBoxes. Thus, the drop in performance due to a weaker relevance term was more than compensated for by the ability to perform B&B jointly on the relevance and diversity terms.

3.2.3 Diversity Function and Upper Bound

The goal of the diversity function $D(Y)$ is to encourage non-redundancy in the chosen set of windows and potentially capture different objects in the image. Before we introduce our own diversity function, we show how existing heuristics in object detection and proposal generation can be written as special cases of this formulation, under specific diversity functions.

Sliding Window + NMS. Non-Maximal Suppression (NMS) is the most popular heuristic for selecting diverse boxes in computer vision. NMS is typically explained *procedurally* – select the highest scoring window $\mathbf{y}^1$, suppress all windows that overlap with $\mathbf{y}^1$ by more than some threshold, select the next highest scoring window $\mathbf{y}^2$, rinse and repeat.

This procedure can be explained as a special case of our formulation. Sliding Window corresponds to enumeration over $\mathcal{Y}$ with some level of sub-sampling (or stride), typically with a fixed aspect ratio. Each step in NMS is precisely a greedy augmentation step under

the following marginal gain:

$$\underset{\mathbf{y}\in\mathcal{Y}_{\text{sub-sampled}}}{\text{argmax}} \quad r(\mathbf{y}) + \lambda D_{NMS}(\mathbf{y} \mid Y^{t-1}), \quad \text{where} \tag{3.5a}$$

$$D_{NMS}(\mathbf{y} \mid Y^{t-1}) = \begin{cases} 0 & \text{if} \quad \max_{\mathbf{y}'\in Y^{t-1}} \text{IoU}(\mathbf{y}', \mathbf{y}) \leq \text{NMS-threshold} \\ -\infty & \text{else.} \end{cases} \tag{3.5b}$$

Intuitively, the NMS diversity function imposes an infinite penalty if a new window $\mathbf{y}$ overlaps with a previously chosen $\mathbf{y}'$ by more than a threshold, and offers no reward for diversity beyond that. This explains the NMS procedure of suppressing overlapping windows and picking the highest scoring one among the unsuppressed ones. Notice that this diversity function is submodular but *not monotone* (the marginals gains may be negative). A similar observation was made in [14]. For such non-monotone functions, greedy does not have approximation guarantees and different techniques are needed [18, 41]. This is an interesting perspective on the classical NMS heuristic.

ESS Heuristic [74]. ESS was originally proposed for single-instance object detection. Their extension to detecting multiple instances consisted of a heuristic for suppressing the features extracted from the selected bounding box and re-running the procedure. Since their scoring function was linear in the features, this heuristic of suppressing features and rerunning B&B can be expressed as a greedy augmentation step under the following marginal gain:

$$\underset{\mathbf{y}\in\mathcal{Y}}{\text{argmax}} \; r(\mathbf{y}) + \lambda D_{ESS}(\mathbf{y} \mid Y^{t-1}), \text{ where } D_{ESS}(\mathbf{y} \mid Y^{t-1}) = -r\left(\mathbf{y} \cap (\mathbf{y}^1 \cup \mathbf{y}^2 \ldots \mathbf{y}^{t-1})\right) \tag{3.6}$$

i.e., the ESS diversity function *subtracts* the score contribution coming from the intersection region. If the $r(\cdot)$ is non-negative, it is easy to see that this diversity function is monotone

and submodular – adding a new window never hurts, and since the marginal gain is the score contribution of the *new regions* not covered by previous window, it is naturally diminishing. Thus, even though this heuristic not was presented as such, the authors of [74] did in fact formulate a near-optimal greedy algorithm for maximizing a monotone submodular function. Unfortunately, while $r(\cdot)$ is always positive in our experiments, this was not the case in the experimental setup of [74].

Our Diversity Function. Instead of hand-designing an explicit diversity function, we use a function that implicitly measures diversity in terms of *coverage* of a set of *reference* set of bounding boxes B. This reference set of boxes may be a uniform sub-sampling of the space of windows as done in Sliding Window methods, or may itself be the output of another object proposal method such as Selective Search [118]. Specifically, each greedy augmentation step under our formulation given by:

$$\underset{\mathbf{y}\in\mathcal{Y}}{\operatorname{argmax}}\ r(\mathbf{y}) + \lambda D_{\text{coverage}}(\mathbf{y} \mid Y^{t-1}),\ \text{where}\ D_{\text{coverage}}(\mathbf{y} \mid Y^{t-1}) = \max_{b\in B} \delta\text{IoU}(\mathbf{y}, b \mid Y^{t-1}) \tag{3.7a}$$

$$\delta\text{IoU}(\mathbf{y}, b \mid Y^{t-1}) = \max\{\text{IoU}(\mathbf{y}, b) - \max_{\mathbf{y}'\in Y^{t-1}} \text{IoU}(\mathbf{y}', b), 0\}. \tag{3.7b}$$

Intuitively speaking, the marginal gain of a new window $\mathbf{y}$ under our diversity function is the largest gain in coverage of exactly one of the references boxes. We can also formulate this diversity function as a maximum bipartite matching problem between the reference proposal boxes Y and the reference boxes B (in our experiments, we also study performance under top-k matches). We show in the supplement that this marginal gain is always non-negative and decreasing with larger Y^{t-1}, thus the diversity function is monotone submodular. All that remains is to compute an upper-bound on this marginal gain. Ignoring constants, the key term to bound is $\text{IoU}(\mathbf{y}, b)$. We can upper-bound this term by computing the intersection

w.r.t. the largest window in the window set $\mathbf{y}_{\max}$, and computing the union w.r.t. to the smallest window $\mathbf{y}_{\min}$, *i.e.* $\max_{\mathbf{y}\in\mathcal{Y}_v} \text{IoU}(\mathbf{y}, b) \leq \frac{area(\mathbf{y}_{\max}\cap b)}{area(\mathbf{y}_{\min}\cup b)}$.

3.3 Speeding up Greedy with Minoux's 'Lazy Greedy'

In order to speed up repeated application of B&B across iterations of the greedy algorithm, we now present an application of Minoux's 'lazy greedy' algorithm [85] to the B&B tree.

The key insight of classical lazy greedy is that the marginal gain function $F(\mathbf{y} \mid Y^t)$ is a non-increasing function of t (due to submodularity of F). Thus, at time $t-1$, we can *cache* the priority queue of marginals gains $F(\mathbf{y} \mid Y^{t-2})$ for all items. At time t, lazy greedy does not recompute all marginal gains. Rather, the item at the front of the priority queue is picked, its marginal gain is updated $F(\mathbf{y} \mid Y^{t-1})$, and the item is reinserted into the queue. Crucially, if the item remains at the front of the priority queue, lazy greedy can stop, and we have found the item with the largest marginal gain.

Interleaving Lazy Greedy with B&B. In our work, the priority queue does not contain single items, rather sets of windows $\mathcal{Y}_v$ corresponding to the vertices in the B&B tree. Thus, we must interleave the lazy updates with the Branch-and-Bound steps. Specifically, we pick a set from the front of the queue and compute *the upper-bound* on its marginal gain. We reinsert this set into the priority queue. Once a set remains at the front of the priority queue after reinsertion, we have found the set with the highest upper-bound. This is when perform a B&B step, *i.e.* split this set into two children, compute the upper-bounds on the children, and insert them into the queue.

Fig. 3.3 illustrates how the priority queue and B&B tree are updated in this process. Suppose at the end of iteration $t-1$ and the beginning of iteration t, we have the priority queue shown on the left. The first few updates involve recomputing the upper-bounds on the window sets (v_6, v_5, v_3), following by branching on v_3 because it continues to stay on top of the queue,

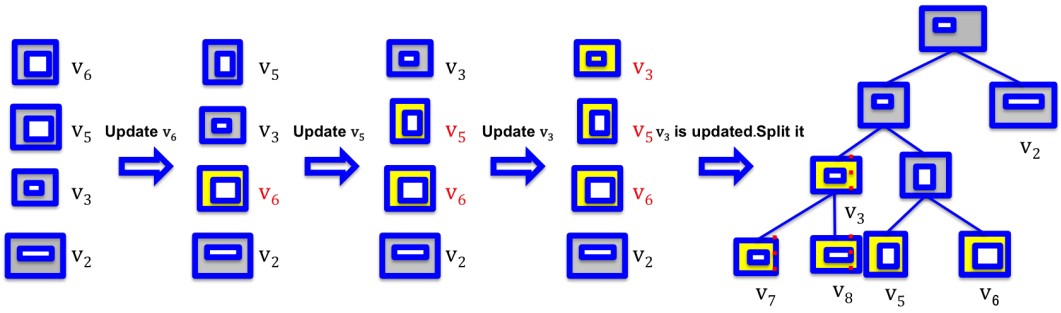


Figure 3.3: Interleaving Lazy Greedy with B&B. The first few steps update upper-bounds, following by finally branching on a set. Some sets, such as v_2 are never updated or split, resulting in a speed-up.

creating new vertices v_7, v_8. Notice that v_2 is never explored (updated or split), resulting in a speed-up.

3.4 Experiments

Setup. We evaluate Submodboxes for object proposal generation on three datasets: PASCAL VOC 2007 [36], PASCAL VOC 2012 [38], and MS COCO [77]. The goal of experiments is to validate our approach by testing the accuracy of generated object proposals and the ability of handling different kinds of reference boxes, and observe trends as we vary multiple parameters.

Evaluation. To evaluate the quality of our object proposals, we use Mean Average Best Overlap (MABO) score. Given a set of ground-truth boxes GT^c for a class c, ABO is calculated by averaging the best IoU between each ground truth bounding box and all object proposals:

$$\mathrm{ABO}^c = \frac{1}{|\mathrm{GT}^c|} \sum_{g \in \mathrm{GT}^c} \max_{\mathbf{y} \in Y} \mathrm{IoU}(g, \mathbf{y}) \tag{3.8}$$

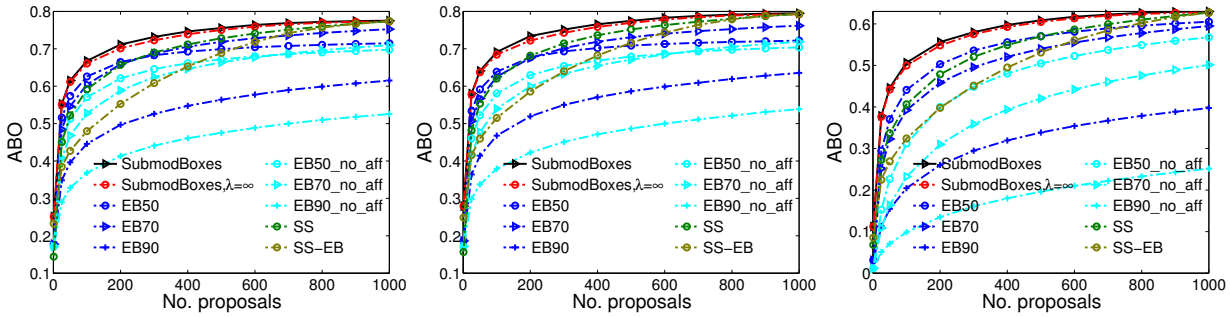
MABO is a mean ABO over all classes.

Weighing the Reference Boxes. Recall that the marginal gain of our proposed diversity function rewards covering the reference boxes with the chosen set of boxes. Instead of weighing all reference boxes equally, we found it important to weigh different reference boxes differently. The exact form the weighting rule is provided in the section 3.5.3. In our experiments, we present results with and without such a weighting to show impact of our proposed scheme.

3.4.1 Accuracy of Object Proposals

In this section, we explore the performance of our proposed method in comparison to relevant object proposal generators. For the two PASCAL datasets, we perform cross validation on 2510 validation images of PASCAL VOC 2007 for the best parameter λ, then report accuracies on 4952 test images of PASCAL VOC 2007 and 5823 validation images of PASCAL VOC 2012. The MS COCO dataset is much larger, so we randomly select a subset of 5000 training images for tuning λ, and test on complete validation dataset with 40138 images. We use 1000 top ranked selective search windows [118] as reference boxes. In a manner similar to [70], we chose a different λ_M for $M = 100, 200, 400, 600, 800, 1000$ proposals. We compare our approach with several baselines: 1) $\lambda = \infty$, which essentially involves re-ranking selective search windows by considering their ability to coverage other boxes. 2) Three variants of EdgeBoxes [142] at IoU $= 0.5, 0.7$ and 0.9, and corresponding three variants without affinities in (3.3). 3) Selective Search: compute multiple hierarchical segments via grouping superpixels and placing bounding boxes around them. 4) SS-EB: use EdgeBoxesScore to re-rank Selective Search windows.

Fig. 3.4 shows that our approach at $\lambda = \infty$ and validation-tuned λ both outperform all baselines. At $M = 25, 100,$ and 500, our approach is $20\%, 11\%,$ and 3% better than Selective Search and $14\%, 10\%,$ and 6% better than EdgeBoxes70, respectively.



(a) Pascal VOC 2007 (b) Pascal VOC 2012 (c) MS COCO

Figure 3.4: ABO vs. No. Proposals.

3.4.2 Ablation Studies.

We now study the performance of our system under different components and parameter settings.

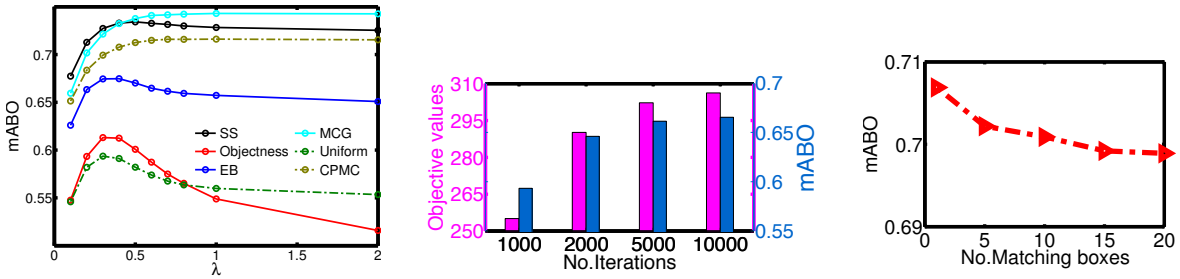
Effect of λ and Reference Boxes. We test performance of our approach as a function of λ using reference boxes from different object proposal generators (all reported at M=200 on PASCAL VOC 2012). Our reference box generators are: 1) Selective Search [118]; 2) MCG [4]; 3) CPMC [20]; 4) EdgeBoxes [142] at IoU = 0.7; 5) Objectness [2]; and 6) Uniform-sampling [57]: *i.e.* uniformly sample the bounding box center position, square root area and log aspect ratio.

Table 3.1 shows the performance of Submodboxes when used with these different reference box generators. Our approach shows improvement (over corresponding method) for *all reference boxes*. Our approach outperforms the current state of art MCG by 2% and Selective Search by 5%. This is significantly larger than previous improvements reported in the literature.

Fig. 3.5a shows more fine-grained behavior as λ is varied. At $\lambda = 0$ all methods produce the same (highest weighted) box M times. At $\lambda = \infty$, they all perform a reranking of the reference set of boxes. In nearly all curves, there is a peak at some intermediate setting of λ. The only exception is EdgeBoxes, which is expected since it is being used in both the

	Selective-Search	MCG	EB	CPMC	Objectness	Uniform-sampling
$\lambda \approx 0.4$, weighting	**0.7342**	0.7377	**0.6747**	0.7125	**0.6131**	**0.5937**
$\lambda \approx 0.4$, without weighting	0.5697	0.5042	0.6350	0.5681	0.6220	0.5136
$\lambda = 10$, weighting	0.7233	**0.7417**	0.6467	**0.7130**	0.5006	0.5478
$\lambda = 10$, without weighting	0.5844	0.5534	0.6232	0.5849	0.5920	0.5115
$\lambda = \infty$, weighting	0.7222	0.7409	0.6558	0.7116	0.4980	0.5453
Original method	0.6817	0.7206	0.6755	0.7032	0.6038	0.5295

Table 3.1: Comparison with/without weighting scheme (rows) with different reference boxes (columns). 'Original method' row shows performance of directly using object proposals from these proposal generators. '≈' means we report the best performance from $\lambda = 0.3, 0.4$ and 0.5 considering the peak occurs at different λ for different object proposal generators.



(a) Performance vs. λ with different reference box generators.

(b) Objective and performance vs. No. of iterations.

(c) Performance vs. No. of matching boxes.

Figure 3.5: Experiments on different parameter settings.

relevance and diversity terms.

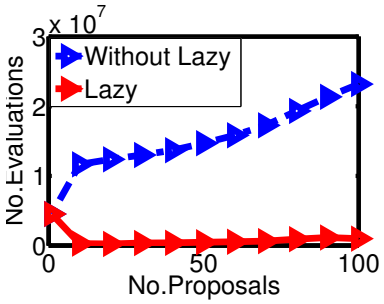


Figure 3.6: Comparison of the number of B&B iterations of our Lazy Greedy generalization and independent B&B runs.

Effect of No. B&B Steps. We analyze the convergence trends of B&B. Fig. 3.5b shows that both the optimization objective function value and the mABO increase with the number of B&B iterations.

Effect of No. of Matching Boxes. Instead of allowing the chosen boxes to cover exactly one reference box, we analyze the effect of matching top-k reference boxes. Fig. 3.5c shows that the performance decreases monotonically bit as more matches are allowed.

Speed up via Lazy Greedy. Fig. 3.6 compares the num-

ber of B&B iterations required with and without our proposed Lazy Greedy generalization (averaged over 100 randomly chosen images) – we can see that Lazy Greedy significantly reduces the number of B&B iterations required. The cost of each B&B evaluation is nearly the same, so the iteration speed-up is directly proportional to time speed-up.

3.5 Appendix

3.5.1 Monotonity and Submodularity

In this section, we will prove that the marginal gain in our formulation is non-negative and decreasing with larger Y. It is not hard to see that $r(y) \geq 0$ due to positive edge strength s_i and $\delta\text{IoU}(y, b \mid Y^{t-1}) \geq 0$ based on it's definition. Thus, $D(\mathbf{y} \mid Y)$ is non-negative. Now, we will prove $D(\mathbf{y} \mid Y \cup \hat{\mathbf{y}}) \leq D(\mathbf{y} \mid Y)$

$$D(\mathbf{y} \mid Y \cup \hat{\mathbf{y}}) = \max_{b \in B} \delta\text{IoU}(\mathbf{y}, b \mid Y \cup \hat{\mathbf{y}}) \tag{3.9}$$

$$= \max_{b \in B} \underbrace{\max\{\text{IoU}(y, b) - \max_{\mathbf{y}' \in Y \cup \hat{\mathbf{y}}} \text{IoU}(y', b), 0\}}_{\alpha(b, \mathbf{y})} \tag{3.10}$$

$$\leq \max_{b \in B} \underbrace{\max\{\text{IoU}(y, b) - \max_{\mathbf{y}' \in Y} \text{IoU}(y', b), 0\}}_{\beta(b, \mathbf{y})} \tag{3.11}$$

$$= D(\mathbf{y} \mid Y) \tag{3.12}$$

We know that $\max_{\mathbf{y}' \in Y \cup \hat{\mathbf{y}}} \text{IoU}(y', b) \geq \max_{\mathbf{y}' \in Y} \text{IoU}(y', b)$, then $\alpha(b, \mathbf{y}) \leq \beta(b, \mathbf{y})$ for each b. Thus, we get (3.11) from (3.10).

3.5.2 Algorithm

According to the explanation in the main paper, our SubmodBoxes algorithm is the following:

Algorithm 1: SubmodBoxes

Input: Image $\mathbf{x}$; **Output:** Y^M
begin
  $Q = v_0$, v_0 is the entire image. $Y^0 = \emptyset$ and $v^* = \emptyset$
  for $t = 1$ **to** M **do**
    repeat
      `/* Find the top region` v^* `evaluated in the current iteration` t `*/`
      while $v^* = \emptyset$ *or not updated in iteration* t **do**
        if $v^* \neq \emptyset$ **then**
          $U(v^*) = U(v^*|Y^{t-1})$
        $v^* = \text{argmax}_{v \in Q} U(v)$
      Split v^* into v_1^* and v_2^*; $Q = Q \setminus v^* \cup v_1^* \cup v_2^*$; $v^* = \text{argmax}_{v \in Q} U(v)$
    until v^* *has only one window*
    `/* The optimal window has been searched,` $\mathbf{y}^t = v^*$ `*/`
    $Y^t = Y^{t-1} \cup \mathbf{y}^t$

SubmodBoxes does not recompute all marginal gains. Rather update the items at the front of the priority queue. If these items are still at the front of the queue, algorithm can stop. Thus, we will proof that the best window output from SubmodBoxes in iteration t is the optimal window by maximizing $F(\mathbf{y} \mid Y^{t-1})$. That means it is exactly the same window if we re-compute all marginal gains and select the best one.

Lemma 3.1. *In the iteration t, the finally searched window $\mathbf{y}^t$ using SubmodBoxes is the* optimal *window by maximizing $F(\mathbf{y} \mid Y^{t-1})$.*

Proof. At the end of iteration t, we get priority queue Q. Then, $\mathbf{y}^t$ is a special vertex, say $v^t \in Q$. Let $\mathbf{y}^*$ is the optimal window by maximizing $F(\mathbf{y} \mid Y^{t-1})$. Now, we assume $\mathbf{y}^t \neq y^*$. Without loss of generality, we let $\mathbf{y}^* \in v^* \subseteq Q, v^* \cup v^t = \emptyset$. There are two cases for v^*:

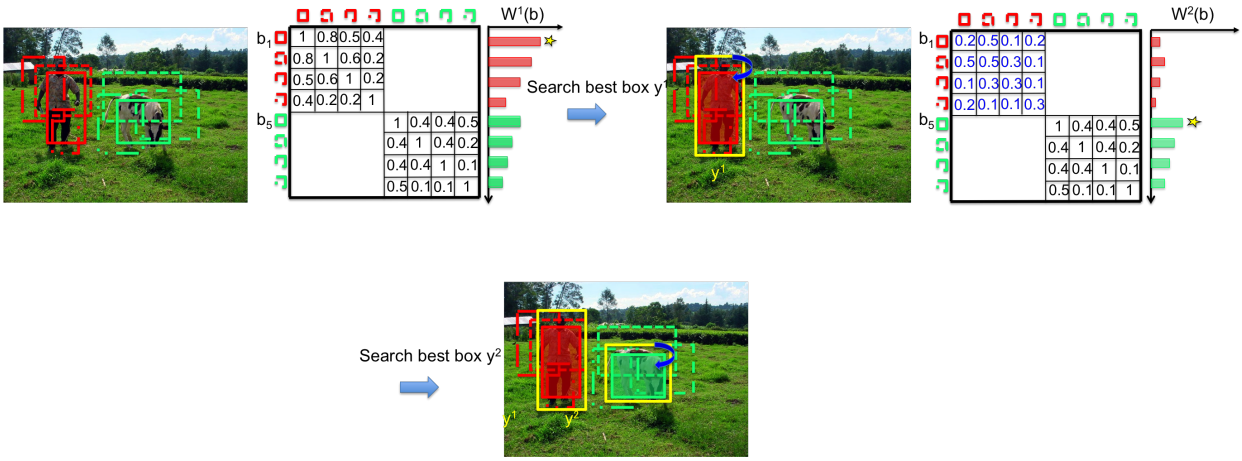


Figure 3.7: Weighting rule.

1. v^* is updated during iteration t.

$$F(\mathbf{y}^* \mid Y^{t-1}) \leq U(v^* \mid Y^{t-1}) \leq U(v^t \mid Y^{t-1}) = F(\mathbf{y}^t \mid Y^{t-1})$$

2. v^* is not updated during iteration t.

$$\underbrace{F(\mathbf{y}^*|Y^{t-1}) \leq F(\mathbf{y}^*|Y^i)}_{\text{Diminishing reward property}} \underbrace{\leq U(v^*|Y^i) \leq U(v^t|Y^{t-1})}_{\text{current upper bound in } Q} = F(\mathbf{y}^t|Y^{t-1})$$

where $i \leq t-1$ is the lastest iteration when v^* being updated.

Thus, we can see $F(\mathbf{y}^*|Y^{t-1}) \leq F(\mathbf{y}^t|Y^{t-1})$. We got contradiction since $\mathbf{y}^*$ should be the optimal window.

3.5.3 Weighting the Reference Boxes.

What kind of reference boxes should be matched in priority? In this section, we try to weight a reference box b in the following way:

$$w^t(b) = \sum_{b' \in B} \delta P(b, b'|Y^{t-1}), where \tag{3.13}$$

$$\delta P(b, b'|Y^{t-1}) = \max\{P(b, b') - \max_{\mathbf{y} \in Y^{t-1}} P(y, b'), 0\} \tag{3.14}$$

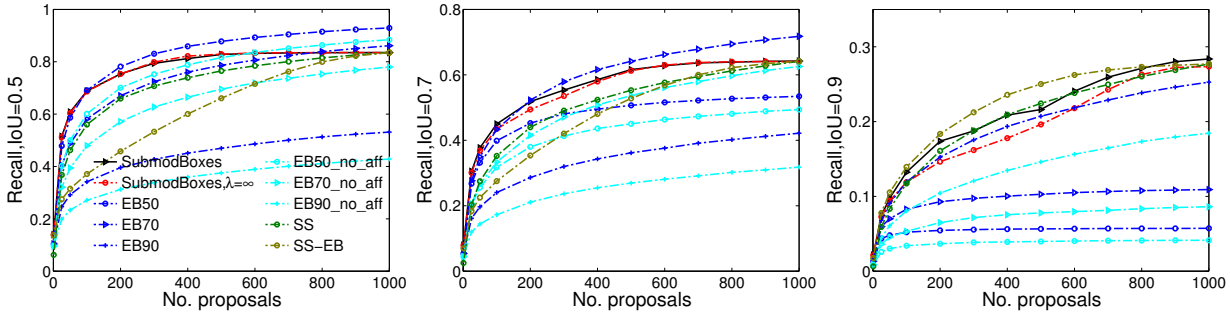
(3.14) indicates that the box which has higher coverage of other reference boxes should give higher weight to be matched. Here, $P(b, b')$ denotes the ability of b to cover b'. From another perspective, these boxes should obtain the more *support* from other reference boxes,

$$w^t(b) = \sum_{b' \in B} \underbrace{p(b')}_{\text{Probability of } b'} \underbrace{\delta P(b, b'|Y^{t-1})}_{\text{Support from } b'} \tag{3.15}$$

Where, we regard $p(b') \sim U[0, 1]$ that we believe all boxes equally. Thus, $w(b)$ denotes the expected support all reference boxes in B.

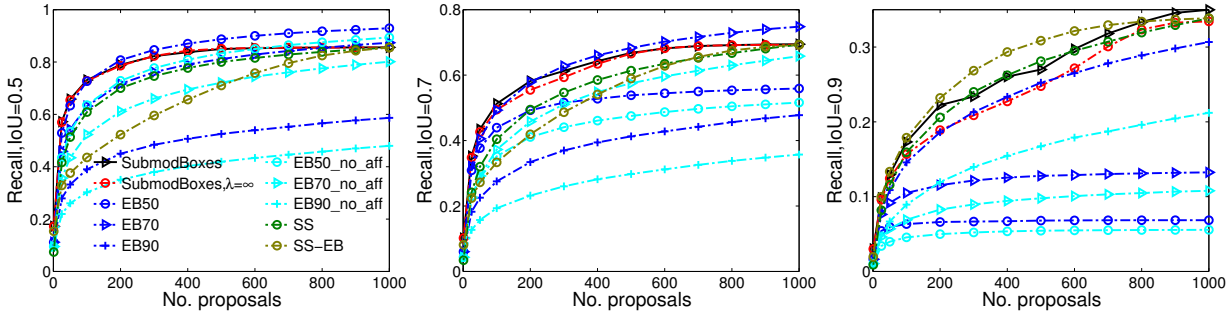
In this paper, we specify IoU metric as P just due to simplicity. Fig. 3.7 shows an example to illustrate the weighting rule. In the first iteration, we have a set of reference boxes (in color *red* and *green*), we calculate $P(b, b'|\emptyset)$ according to (3.14), we see b_1 has the highest weight. After $\mathbf{y}^1$ is selected which is high likely be around b_1, $w^2(b)$ where b in red color will be much lower since all its neighbors is covered by $\mathbf{y}^1$ a lot.

3.5.4 Experiments



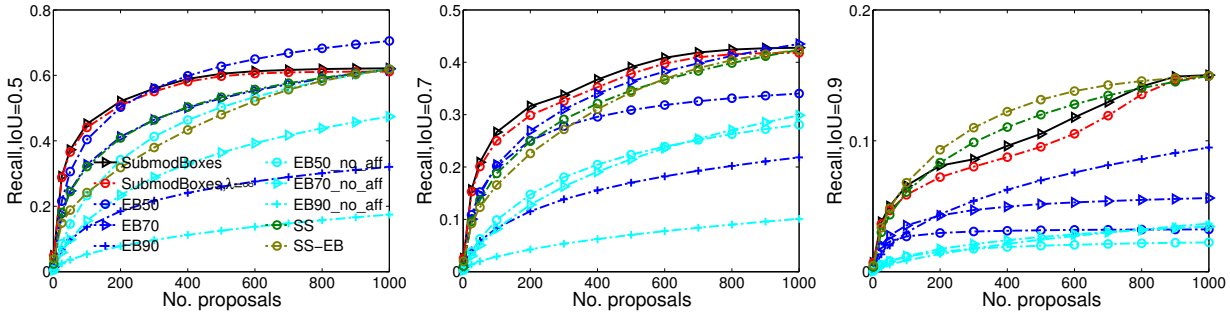
(a) IoU = 0.5 (b) IoU = 0.7 (c) IoU = 0.9

Figure 3.8: Recall vs. No. Proposals on PASCAL VOC 2007.



(a) IoU = 0.5 (b) IoU = 0.7 (c) IoU = 0.9

Figure 3.9: Recall vs. No. Proposals on PASCAL VOC 2012.



(a) IoU = 0.5 (b) IoU = 0.7 (c) IoU = 0.9

Figure 3.10: Recall vs. No. Proposals on PASCAL VOC MS COCO.

3.5.5 Recall of object proposals.

We further test recall at IoU $= 0.5, 0.7$ and 0.9 on three object detection datasets: PASCAL VOC 2007 [36], PASCAL VOC 2012 [38] and Mircosoft COCO [76]. We compare the same

baselines including EdgeBoxes [142] and Selective Search [118]. Fig. 3.8, Fig. 3.9 and Fig. 3.10 show that EdgeBoxes variants generally performs the best at the specific IoU setting since their parameters are tuned based on the given IoU threshold, except for EdgeBoxes 90 (at IoU = 0.9). However, if we fix one variant, such as EdgeBoxes70, and compare the performance over all IoU settings (0.5, 0.7 and 0.9), our approach outperforms all baselines.

Chapter 4

Fill-in-the-Blank Image Captioning with Bidirectional Beam Search

4.1 Introduction

Recurrent Neural Networks (RNNs) have emerged as a popular and effective framework for modeling sequential data across varied domains. The application of these powerful models has led to improved performance for many tasks including speech recognition [29, 54], machine translation [6, 26, 59, 62], conversation modeling [124], image captioning [23, 34, 39, 65, 127], and visual question answering (VQA) [3, 97, 134, 137].

Broadly speaking, in these applications RNNs are typically used in two distinct roles – (1) as *encoders* that convert sequential data into vectors, and (2) as *decoders* that convert encoded vectors into sequential output. Models for image caption retrieval and VQA (with classification over answers) [3, 97, 137] consist of encoder RNNs but not decoders. Image caption generation models [127] consist of decoder RNNs but not encoders (image encoding is performed via Convolutional Neural Networks (CNNs)). And machine translation models (*e.g.* [59]) employ RNNs both as encoders to encode sequential inputs (such as a source sentence in French) and as decoders to produce corresponding sequential outputs (like a

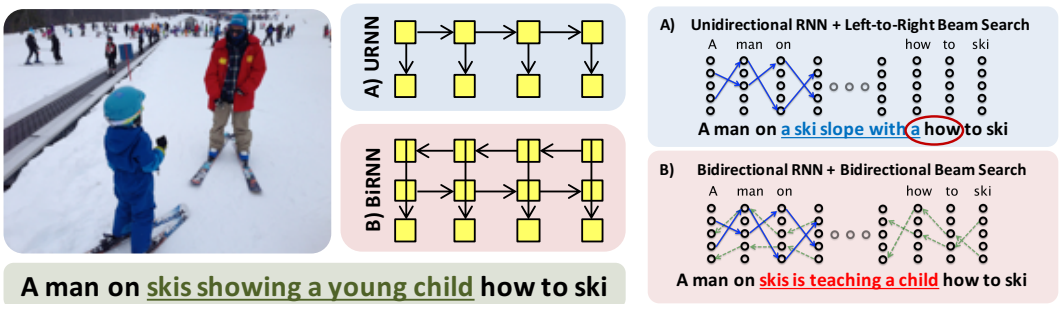


Figure 4.1: We propose a novel Fill-in-the-Blank Image Captioning task as a challenging testbed for neural sequence completion models, such as Unidirectional and Bidirectional Recurrent Neural Network (RNN). Unidirectional RNNs (whether forward or backward) ignore part of the input (either future or past) and produce nonsensical outputs for this task – note the jarring and grammatically incorrect "with a how to" transition produced by classical left-to-right beam search in a Unidirectional RNN (A). We develop a novel approximate inference algorithm for Bidirectional RNNs called Bidirectional Beam Search (BiBS). We find our approach produces significantly better caption completions, *e.g.* "A man on skis is teaching a child how to ski", because it successfully incorporates both the future and past contexts product by a BiDirectional RNN.

target translation in German).

Decoding an RNN model corresponds to finding the most likely sequence $Y = (\mathbf{y}_1, ..., \mathbf{y}_T)$ given some input $\mathbf{x}$, *i.e.* $Y^* = \text{argmax}\, \text{P}(\mathbf{y}_1, ..., \mathbf{y}_T|\mathbf{x})$. Unidirectional RNNs model this probability by estimating the likelihood of outputting a symbol at time t (say $\mathbf{y}_t$) given the history of previous outputs $(\mathbf{y}_1, \ldots, \mathbf{y}_{t-1})$ by "compressing" the history into a hidden state vector $\mathbf{h}_{t-1}$ such that $\text{P}(\mathbf{y}_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_{t-1}, \mathbf{x}) \simeq \text{P}(\mathbf{y}_t \mid \mathbf{h}_{t-1})$. Since each output symbol is conditioned on all previous outputs, the search space of possible sequences is exponential and exact inference is intractable. As a result, approximate inference algorithm are applied, with the Beam Search (BS) being the primary workhorse. BS is a greedy heuristic search that maintains the top-B most likely partial-sequences through the search tree, where B is referred to as the *beam width*. At each time step, BS expands these B partial sequences to all possible beam extensions and then selects the B highest scoring among the expansions. In contrast to Unidirectional RNNs, Bidirectional RNNs model both forward (increasing time) and backward (decreasing time) dependencies $\text{P}(\mathbf{y}_t \mid \mathbf{h}_t^f, \mathbf{h}_t^b)$ via two different hidden state vectors $\mathbf{h}_t^f$ and $\mathbf{h}_t^b$. This enables Bidirectional RNNs to consider both past and future when

making a prediction at time t. Unfortunately, these dependencies also make exact inference in these models more difficult than in Unidirectional RNNs and to the best of our knowledge no efficient approximate algorithms exist.

In this chapter, we study a decoding task that reconstructs missing elements of sequential data. For example, given the blanked image caption *"A man on _____ how to ski"* shown in Figure 4.1, our goal is to generate the missing content *"skis showing a young child"* (or an acceptable paraphrase) to complete the sentence. This fill-in-the-blank image captioning task serves a concrete stand-in for a broad class of other similar 'filling-gaps-in-time-series' tasks, such as estimating missing parts in DNA sequence and reinforcement learning applications where a robot must hit intermediate flag points. Moreover, this fill-in-the-blank task also serves as a testbed for how well existing inference algorithms reason about long-range forward and backward time dependencies during decoding.

On the surface, this task perhaps seems easier than generating an entire sequence from scratch; there is after all, more information in the input. However, the need to condition on the remaining context when generating the missing symbols is challenging for existing greedy approximate inference algorithms. Figure 4.1(a) shows a sample decoding from standard 'left-to-right' BS on a Unidirectional RNN – "A man on a ski slope with a how to ski". Note the jarring and grammatically incorrect "with a how to" transition produced. Using a backward Unidirectional RNN with right-to-left BS simply changes the junction at which the problem occurs. Simply put, the inability to consider both the future and past contexts in BS leads Unidirectional RNNs to fill the blank with words that clash abruptly with the context around the blank.

In addition, there is also a computational challenge in this problem. Consider the following sentence that we know has only a single word missing: *"The _____ was barreling down the tracks."* Filling in this blank feels simple – we just need to find the best single word in the

vocabulary $y_t \in \mathcal{Y}$. However, since all future outputs in a decoder Unidirectional RNN are conditioned on the past, selecting the best word at time t requires evaluating the likelihood *of the entire sequence* once for each possible word $y_t \in \mathcal{Y}$. This amounts to $O(T|\mathcal{Y}|)$ forward passes through an RNN's basic computational unit *to fill in a single blank optimally*! More generally, for an arbitrarily sized blank covering w words, this number grows exponentially $O(T|\mathcal{Y}|^w)$ and quickly becomes intractable. This same argument applies to Bidirectional RNNs, and inference in these models is still inefficient.

To overcome these shortcomings and challenges, we develop an approximate inference algorithm – Bidirectional Beam Search (BiBS) – that performs well on this fill-in-the-blank task by incorporating both forward and backward time information from Bidirectional RNNs. Specifically, we decompose a Bidirectional RNN into two calibrated but independent Unidirectional RNNs (one going forward in time and the other backward). To perform approximate inference with these decomposed models, our method alternatively performs BS on one direction while holding the beams in the opposite direction fixed. The fixed, oppositely-directed beams are used to roughly approximate the conditional probability of all future sequence given the past such that a BS-like update minimizes an approximation of the full joint at each time step. Figure 4.1(b) shows an example result of our algorithm – "A man on skis is teaching a child how to ski" – which smoothly fits within its context while still describing the image content.

We evaluate the effectiveness of our proposed Bidirectional Beam Search (BiBS) algorithm against natural ablations and baselines for fixed length fill-in-the-blank tasks. Our results show that BiBS is an effective and efficient approach for decoding Bidirectional RNNs for fill-in-the-blank sequence generation tasks, consistently outperforming standard BS.

4.2 Related Work

While Unidirectional RNNs are popular models with widespread adoption [3, 6, 26, 29, 54, 59, 62, 82, 97, 124, 125, 134, 137], Bidirectional RNNs have been utilized in relatively infrequently [58, 103, 132] and even more rarely as decoders [12] – we argue due to the lack of efficient inference approaches for these models.

Huang *et al.* [58] apply a variety of RNN models to the natural language processing task of sequence tagging, including Bidirectional RNNs; however, these models were not used for generation, serving simply as feature extractors. Similarly, Sak *et al.* [103] apply Bidirectional RNNs to predict phonemes for for speech recognition, but do not perform generation.

Wang *et al.* [132] used Bidirectional RNNs (with Long Short Term Memory (LSTM) cells) for image caption generation, but do not perform bidirectional inference, rather simply use BiDirectional RNNs to rescore candidates. Specifically, at inference time they decompose a Bidirectional RNN into two independent Unidirectional RNN, apply standard Beam Search in each direction, and then reranked these two collection of beams based on the max probability of each beam under the forward or backward Unidirectional RNN model. We compare to a similar baseline in our experiments and show that joint optimization via Bidirectional Beam Search leads to better sequence completions for our fill-in-the-blank image captioning task.

Most related to our work is that of Burglund *et al.* [12], which studies generating missing data in time series data in an unsupervised setting using Bidirectional RNNs. They propose three probabilistically justified approaches to fill these gaps by drawing samples from the full joint.

Their first model, Generative Stochastic Networks (GSN), resamples the output y_t at a random time t from the conditional output $\mathrm{P}(\mathbf{y}_t \mid Y_{[1:T]\setminus t})$. For a blank of length w, resampling each output tokens M times requires $O(wMT)$ passes of the RNN. The cost of producing a

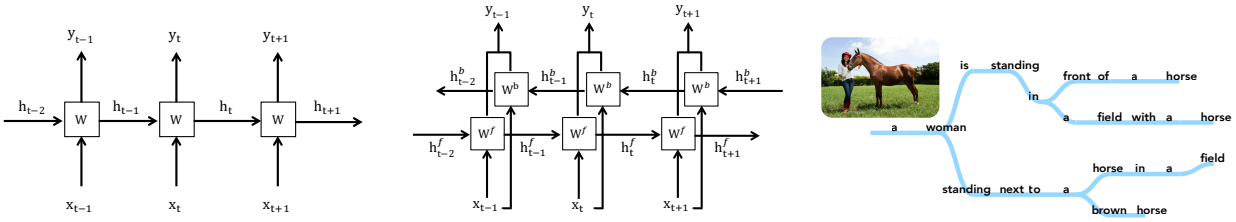
sample with the GSN method scales linearly with the size of the gap and requires a full pass of the Bidirectional RNN, whereas our approach updates all y_t a single pass. Their second approach, NADE, trains a model specifically for filling in blank – some inputs are set to a specific 'missing' token to indicate the content that needs to be generated. At inference time, the inputs from the gap are set to this token and sampled from the resulting conditional. Note that this approach is 'trained to fill in gaps' and as such requires training data of this kind. To contrast, this is a new model for filling in gaps, while we propose a new inference algorithm, which can be broadly applied to any generative bidirectional model. Finally, they propose a third sampling approach based on a Unidirectional RNN which draws from the conditional $\mathrm{P}(\mathbf{y}_t \mid Y_{[1:T]\setminus t})$; however, as the model is a left-to-right Unidirectional RNN, this term requires computing the likelihood of the remaining sequence given each possible token at time t. This costly approach requires $O(w|\mathcal{Y}|MT)$ steps of the RNN and is intractable for large vocabularies.

We also note that these approaches are for generating sample sequences from the Bidirectional RNNs, whereas our BiBS approach is an approximate inference algorithm, seeking the most likely sequences to fill a gap. Moreover, our algorithm does not require training with a specific 'missing' token and requires fewer steps of the RNN model.

4.3 Preliminaries: RNNs and Beam Search

We begin by establishing notation and reviewing RNNs and standard Beam Search for completeness. While our exposition details the classical RNN updates, the techniques developed in this chapter are broadly applicable to any recurrent neural architecture (*e.g.* LSTMs [55] or GRUs [25]).

Notation. Let $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$ denote an input sequence, where $\mathbf{x}_t$ is an input vector at time t. Similarly, let $Y = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T)$ denote an output sequence, where $\mathbf{y}_t$ is an



(a) Unidirectional RNN (b) Bidirectional RNN (c) Example Run of Beam Search

Figure 4.2: **Different architectures of RNNs and left-to-right Beam Search.** (a) The prediction of variable $\mathbf{y}_t$ only depends on the *past* in URNNs. BiRNNs shown in (b) can consider both *past* and *future*. (c) illustrates the search tree for left-to-right beam search in a URNN with a beam width of $B = 4$.

output vector at time t. To avoid notational clutter, our exposition uses the same length for both input and output sequences (T); however, this is not a restriction in theory or practice. Given integers a, b, we use the notation $Y_{[a:b]}$ to denote the sub-sequence $(\mathbf{y}_a, \mathbf{y}_{a+1}, \ldots, \mathbf{y}_b)$; thus, $Y = Y_{[1:T]}$ by convention. Given discrete variables Y, we generalize the classical maximization notation $\text{argmax}_{Y \in \mathcal{Y}} \ f(Y)$ to find the (unique) top B states with highest $f(Y)$ via the notation top-$\text{B}_{Y \in \mathcal{Y}} \ f(Y)$.

Unidirectional RNN (URNNs) model the probability of $\mathbf{y}_t$ given the history of inputs $\mathbf{x}_1, \ldots, \mathbf{x}_t$ by "compressing" the history into a hidden state vector $\mathbf{h}_t$ via a hidden layer such that

$$\text{P}(\mathbf{y}_t \mid X_{[1:t]}) = \phi(W_y \mathbf{h}_t + \mathbf{b}_y) \tag{4.1a}$$

$$\mathbf{h}_t = \tanh(W_x \mathbf{x}_t + W_h \mathbf{h}_{t-1} + \mathbf{b}_h) \tag{4.1b}$$

where $W_x, W_h, W_y, \mathbf{b}_h$, and $\mathbf{b}_y$ are learned parameters defining the transforms from the input $\mathbf{x}_t$ and hidden state $\mathbf{h}_{t-1}$ to the output $\mathbf{y}_t$ and updated hidden state $\mathbf{h}_t$. In applications with symbol sequences as output (such as image captioning), the nonlinear function ϕ is typically the softmax function which produces a distribution over the output vocabulary $\mathcal{Y}$. An

example left-to-right Unidirectional RNN architecture is shown in Figure 4.2a.

Bidirectional RNNs (BiRNNs) (shown in Figure 4.2b) model both forward (positive time) and backward (negative time) dependencies via two different hidden state vectors – forward $\mathbf{h}_t^f$ and backward $\mathbf{h}_t^b$ – each with its own update dynamics and corresponding weights. For a Bidirectional RNN, we can write the probability of the token $\mathbf{y}_t$ given the input sequence as

$$\mathrm{P}(\mathbf{y}_t \mid X_{[1:T]}) = \phi(\underbrace{W_y^{\rightarrow}\mathbf{h}_t^{\rightarrow}}_{\text{Forward score}} + \underbrace{W_y^{\leftarrow}\mathbf{h}_t^{\leftarrow}}_{\text{Backward score}} + \mathbf{b}_y) \tag{4.2a}$$

$$\mathbf{h}_t^{\rightarrow} = \sigma(W_x^{\rightarrow}\mathbf{x}_t + W_h^{\rightarrow}\mathbf{h}_{t-1}^{\rightarrow} + \mathbf{b}_h^{\rightarrow}) \tag{4.2b}$$

$$\mathbf{h}_t^{\leftarrow} = \sigma(W_x^{\leftarrow}\mathbf{x}_t + W_h^{\leftarrow}\mathbf{h}_{t+1}^{\leftarrow} + \mathbf{b}_h^{\leftarrow}) \tag{4.2c}$$

BiRNNs as URNNs. Consider a Bidirectional RNN with the output nonlinearity ϕ defined as the softmax function $p_i = \phi_i(\,) = e^{s_i}/\sum e^{s_j}$ (as is common practice for decoding tasks). It is straightforward to show that the conditional probability of $\mathbf{y}_t$ given all other tokens can be written as

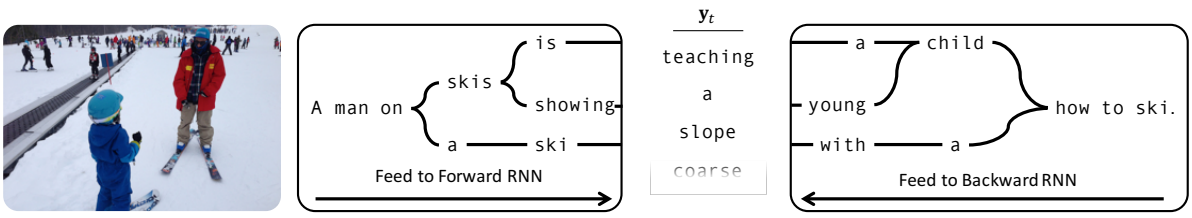
$$\begin{aligned}\mathrm{P}(\mathbf{y}_t \mid X_{[1:T]}) &= \phi(W_y^f\mathbf{h}_t^{\rightarrow} + W_y^b\mathbf{h}_t^{\leftarrow} + \mathbf{b}_y) \\ &\propto \phi\left(W_y^{\rightarrow}\mathbf{h}_t^{\rightarrow} + \tfrac{\mathbf{b}_y}{2}\right)\phi\left(W_y^{\leftarrow}\mathbf{h}_t^{b} + \tfrac{\mathbf{b}_y}{2}\right)\end{aligned}$$

where the resulting terms in the proportionality resemble the URNNs output equation in Eq. 4.1a. That is to say that the output of a Bidirectional RNN with a softmax output layer can be equivalently expressed as the product of the output from two independent but oppositely directed URNNs with specifically constructed weights, renormalized after multiplication. This construction also works inversely such that an equivalent Bidirectional

RNN can be constructed from two independently trained but oppositely directed URNNs. As such, we will consider a Bidirectional RNN as consisting of a forward-time model $\overrightarrow{\text{URNN}}$ and a backward-time model $\overleftarrow{\text{URNN}}$.

RNNs for decoding are trained to produce sequences conditioned on some encoded representation X. For machine translation tasks, X may represent an encoding of some source language sequence to be translated and Y is the translation. For image captioning, X is typically a dense vector embedding of the image produced by a Convolutional Neural Network (CNN) [109], and Y is a sequence of 1-hot encoding of the words of the corresponding image caption. Regardless of its source, this encoded representation is considered the first input $\mathbf{x}_0$ and for all remaining time steps $\mathbf{x}_t = \mathbf{y}_{t-1}$, such that decoder RNNs are learning to model $\mathrm{P}(\mathbf{y}_t|\mathbf{y}_{t-1}, ..., \mathbf{y}_1, \mathbf{x}_0)$. This is the setting of interest in this chapter, but we drop this explicit dependence on the encoding $\mathbf{x}_0$ to reduce notational clutter in later sections.

Beam Search (BS). Maximum a posteriori (MAP) inference in RNNs consists of finding the most likely sequence under the model. The primary difficulty for decoding is that the number of possible T length sequences grows exponentially as $|\mathcal{Y}|^T$, so approximate inference algorithms are employed. Due to this exponential output space and the dependence on previous outputs, exact inference is NP-hard in the general case. Beam Search (BS) is a greedy heuristic search algorithm that traverses the search tree using breadth-first search, while only expanding the most promising nodes at each depth. Specifically, BS in Unidirectional RNNs involves maintaining and expanding the top-B highest-scoring partial hypotheses, called *beams*. Let $Y_{[1:t]} = (\mathbf{y}_1, \ldots, \mathbf{y}_t)$ denote a partial hypothesis (beam) at time t. We use the notation $\mathbf{Y}_{[1:B],[1:t]} = (Y_{1,[1:t]}, Y_{2,[1:t]}, \ldots, Y_{B,[1:t]})$ to denote a collection of B beams. BS begins with empty beams, $Y_{b,0} = (\mathbf{y}_{b,0})$, where $\mathbf{y}_{b,0} = \emptyset,\ \forall b$ and proceeds in a left-to-right manner up to time T or until a special `END` token is generated. At each time t, BS extends the previous beams $\mathbf{Y}_{[1:B],[1:t-1]}$ with each symbol in $\mathcal{Y}$, to find the top-B high-scoring t-length



$$Y_{[1,B],[1,t]} = \underset{b,\mathcal{Y}_T,b'}{\text{top-B}} \quad \log \mathrm{P}(Y_{b,[1,t-1]}) + \log \left(\mathrm{P}(\mathbf{y}_t|Y_{b,[1,t-1]})\mathrm{P}(\mathbf{y}_t \mid Y_{[b',t+1,T]})\right) + \log \mathrm{P}(Y_{b',[t+1,T]}) \tag{4.3}$$

Figure 4.3: **Overview of Bidirectional Beam Search (BiBS).** Starting from a set of B complete sequences $Y_{[1,B],[1,T]}$, BiBS alternately performs left-to-right and right-to-left beam searches to greedily optimize an approximation of the full joint. In the example above, a left-to-right beam search is advancing the beams at time t by considering all possible connections between the current left-to-right beams and the previous right-to-left beams through any token in the dictionary $\mathcal{Y}$. The terms in this joint approximation (written in Eq. 4.3) can be efficiently computed by the forward and backward Unidirectional RNNS and sorted to find the unique top-B extensions.

beams among the expanded hypothesis space $\mathcal{Y}_t = \mathbf{Y}_{[1:B],[1:t-1]} \times \mathcal{Y}$. These updated beams $\mathbf{Y}_{[1:B],[1:t]}$ can be found as the solution to

$$\underset{\mathcal{Y}_t}{\text{top-B}} \quad \log \mathrm{P}\left(Y_{[1:t]}\right) = \sum_{i=1}^{t} \log \mathrm{P}(\mathbf{y}_i \mid \mathbf{y}_{i-1}, \ldots, \mathbf{y}_1).$$

Each log probability term in the above expression can be computed via a forward pass in Unidirectional RNNs such that implementing the top-B operation simply requires sorting $B|\mathcal{Y}_t|$ values. An example run of BS on a left-to-right URNN is shown in Figure 4.2c.

4.4 Bidirectional Beam Search (BiBS)

We begin by analyzing the decision made by classical left-to-right Beam Search at time t. Specifically, at each time t, we can factorize the joint probability $\mathrm{P}(Y_{[1,T]})$ in a particular way:

$$\mathrm{P}(Y_{[1,T]}) = \mathrm{P}(Y_{[1,t-1]})\mathrm{P}(\mathbf{y}_t|Y_{[1,t-1]})\mathrm{P}(Y_{[t+1,T]} \mid \mathbf{y}_t, Y_{[1,t-1]}) \tag{4.4}$$

This left-to-right decomposition of the joint around $\mathbf{y}_t$ is comprised of three terms

(1) the 'marginal' of the sequence prior to term $\mathbf{y}_t$: $\mathrm{P}(Y_{[1,t-1]})$,

(2) the conditional of $\mathbf{y}_t$ given this past: $\mathrm{P}(\mathbf{y}_t|Y_{[1,t-1]})$, and

(3) the conditional of the remaining sequence after $\mathbf{y}_t$ given all prior terms: $\mathrm{P}(Y_{[t+1,T]} \mid \mathbf{y}_t, Y_{[1,t-1]})$.

If we consider choosing $\mathbf{y}_t$ to maximize this joint, the first two terms can be computed exactly via the forward pass of an left-to-right URNN given the existing sequence; however, the third term cannot be exactly computed because it depends on all futures. Even approximating the third term with beams requires re-running beam search for each possible setting of $\mathbf{y}_t$. For long sequences with large vocabularies, this is prohibitively expensive.

One way of interpreting what left-to-right BS does is to view it as approximating the joint in (4.4) with just the first two terms. Specifically, if we assume that $\mathrm{P}(Y_{[t+1,T]} \mid \mathbf{y}_t, Y_{[1,t-1]})$ is uniform, *i.e.* all futures are equally likely conditioned on the past so far, then BS is picking the optimal $\mathbf{y}_t$. Clearly, this approximation does not hold in practice and results in especially poor performance for fill-in-the-blank tasks where all future sequences are not equally likely by construction. In this section, we consider an alternative approximation and derive our BiBS approach.

Efficiently Approximating the Joint Distribution. In order to derive a tractable approximation to this third term (and by proxy the full joint), we make two simplifying assumptions (which we know will be violated in practice, but will lead to approximate inference algorithms that work well in applications). First, we assume that future sequence tokens are independent of past sequence tokens given $\mathbf{y}_t$, *i.e.* RNNs are first-order Markov. Second, we assume that $\mathrm{P}(\mathbf{y}_t)$ is uniform, avoiding the need to estimate a distribution over $\mathcal{Y}$ without context for all possible time steps. Under these assumptions, we can write the

conditional probability of the remaining sequence tokens given the past sequence as

$$\begin{aligned} \mathrm{P}(Y_{[t+1,T]} \mid Y_{[1,t]}) &= \mathrm{P}(Y_{[t+1,T]} \mid \mathbf{y}_t) \\ &\propto \mathrm{P}(\mathbf{y}_t \mid Y_{[t+1,T]})\mathrm{P}(Y_{[t+1,T]}) \end{aligned} \tag{4.5}$$

Notice that the resulting terms are exactly the output of a right-to-left Unidirectional RNN. Substituting Eq. 4.5 into Eq. 4.4, we arrive at an expression that is proportional to the full joint, but comprised of terms which can be independently computed from a pair of oppositely-directed Unidirectional RNNs (or equivalently a Bidirectional RNN),

$$\underbrace{\mathrm{P}(Y_{[1,t-1]})\mathrm{P}(\mathbf{y}_t|Y_{[1,t-1]})}_{\text{Compute from } \overrightarrow{\text{URNN}}} \overbrace{\mathrm{P}(\mathbf{y}_t \mid Y_{[t+1,T]})\mathrm{P}(Y_{[t+1,T]})}^{\text{Compute from } \overleftarrow{\text{URNN}}} \tag{4.6}$$

Note that the two central conditional terms are proportional to the output of an equivalent softmax Bidirectional RNN as discussed in the previous section.

Given some initial sequence $Y_{[1,t]}$, a simple coordinate descent algorithm could select a random time t and update $\mathbf{y}_t$ such that this approximate joint is maximized and repeat this until convergence. Computing Eq. 4.6 would require feeding $Y_{[1,t-1]}$ to the forward RNN and $Y_{[t+1,T]}$ to the backward RNN. Therefore, to update each output M times in this approach would require $O(MT^2)$ RNN steps (combined from both the forward and backward models). If we instead follow an alternating left-to-right then right-to-left update order, this can be reduced to $O(MT)$ by reusing cached log probabilities from the previous direction. This algorithm resembles a beam search with $B = 1$ which bases extensions on the value of Eq. 4.6.

Bidirectional Beam Search. Finally, we arrive at our full Bidirectional Beam Search (BiBS) algorithm by generalizing the simple algorithm outlined above to maintain multiple

beams during each update pass. Given some set of initial sequences $Y_{[1,B],[1,T]}$ (perhaps from a left-to-right beams search), we alternate between forward (left-to-right) and backward (right-to-left) beam searches with respect to the approximate joint. We consider a pair of forward and backward updates a single round of BiBS.

Without loss of generality, we will describe a forward update pass of beam width B. At each time t, we have updated the first $t-1$ tokens of each beam such that we have partial forward sequences $Y_{[1,B],[1,t-1]}$ and the values $Y_{[1,B],[t+1,T]}$ have yet to be updated. To update the forward beams, we consider all possible connections between the current left-to-right beams and the right-to-left beams (held fixed from previous round) through any token in the dictionary $\mathcal{Y}$ rather than just update each beam independently to allow for larger changes per update. Our search space is then $\mathcal{Y}_t = Y_{[1,B],[1,t-1]} \times \mathcal{Y} \times Y_{[1,B],[t+1,T]}$ and $|\mathcal{Y}_t| = B \times |\mathcal{Y}| \times B$. Figure 4.3 shows an example left-to-right update step for image captioning as well as the precise update rule based on Eqn. 4.6 for this time step. For each combination of forward beam and backward beam, this objective can be computed easily from stored sum of log probabilities of each beam and conditional output of the forward and backward RNNs. Like standard Beam Search, the optimal extensions can be found exactly by sorting these values for all possible combinations. Our approach requires only $O(2BMT)$ RNN steps to perform M rounds of updates. Our algorithm is summarized below in Alg. 2 with $\theta_{b,i}(\mathbf{y})$ representing $\log \mathrm{P}(\mathbf{y}|Y_{b,[1,i-1]})$ for the sake of compactness.

4.5 Experiments

In this section, we evaluate the effectiveness of our proposed Bidirectional Beam Search (BiBS) algorithm for inference in BiRNNs. To examine the performance of bidirectional inference, we evaluate on tasks that require the generated sequence to fit well with existing structures both in the past and the future. We choose fill-in-the-blank style tasks where

Algorithm 2: Bidirectional Beam Search (BiBS).

Data: Given initial set of sequences $Y_{[1,B],[1,T]}$

$\theta^{f}_{[1,B],[1,T]} = \theta^{b}_{[1,B],[1,T]} = \mathbf{0}$

while *not converged* **do**

  `// Updated beams left-to-right`

  for $t = 1, ..., T$ **do**

    $\theta^{\rightarrow}_{[1,B],t}, h^{\rightarrow}_{[1,B],t} = \overrightarrow{\text{URNN}}\left(h^{\rightarrow}_{[1,B],t-1}, Y_{[1,B],t-1}\right)$

    $Y_{[1,B],t} = \text{top-B} \sum_{i=1}^{t} \theta^{\rightarrow}_{b,i}(\mathbf{y}_{b,i}) + \sum_{j=t}^{T} \theta^{\leftarrow}_{b',j}(\mathbf{y}_{b',j})$

  `// Updated beams right-to-left`

  for $t = T, ..., 1$ **do**

    $\theta^{\leftarrow}_{[1,B],t}, h^{\leftarrow}_{[1,B],t} = \overleftarrow{\text{URNN}}(h^{\leftarrow}_{[1,B],t+1}, Y_{[1,B],t+1})$

    $Y_{[1,B],t} = \text{top-B} \sum_{i=1}^{t} \theta^{\leftarrow}_{b,i}(\mathbf{y}_{b,i}) + \sum_{j=t}^{T} \theta^{\rightarrow}_{b',j}(\mathbf{y}_{b',j})$

a fixed number of tokens have been removed from a sequence and must be reconstructed. Specifically, we evaluate on fill-in-the-blank image captioning on the Common Objects in Context (COCO) [76] dataset and on the Visual Madlibs [139] dataset.

Baselines. We compare our approach, which we denote ***BiRNN-BiBS***, against several sophisticated baselines:

- ***URNN-f:*** that runs BS on a forward LSTM to produce B output beams (ranked by their probabilities under the forward LSTM),
- ***URNN-b:*** that runs BS on a backward LSTM to produce B output beams (ranked by their probabilities under the backward LSTM),
- ***URNN-f+b:*** that runs BS on two LSTMs (forward and backward) to produce 2B output beams (ranked by the maximum of the probabilities assigned by the forward and backward LSTMs). This is what Wang *et al.* [132] experimented with.
- ***BiRNN-f+b:*** that runs BS on two LSTMs (forward and backward) to produce 2B output beams (ranked by the sum of the probabilities assigned by the forward and backward LSTMs). This lacks formal justification but we find it to be a reasonable heuristic for this

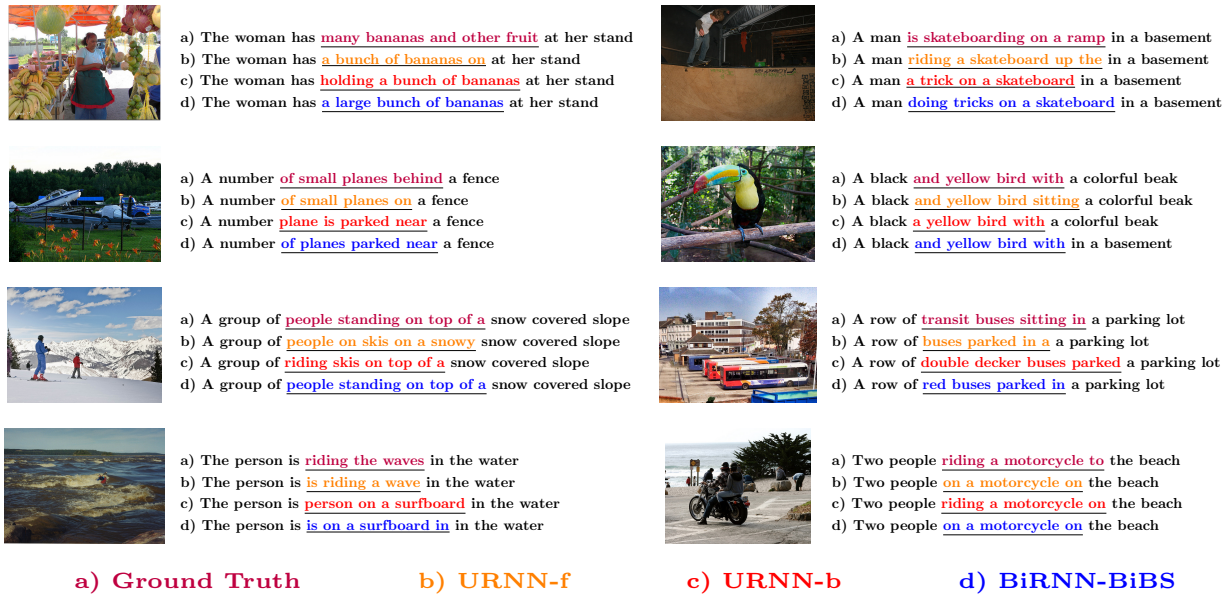


Figure 4.4: **Example fill-in-the-blank image caption completions generated by BS and BiBS.** URNNs decoded with BS often produce blank reconstructions that clash with the remaining context on either side of the blank, while BiBS handles these transitions seamlessly.

task.

We train all models using a modified version of *neuraltalk2* [65]. For all methods and baselines, we compute the joint log probability of the entire sequence (including content before and after the blank), and not just the generated content to ensure the smoothness of transition between generated and contextual tokens is considered.

Evaluation. For all models, we evaluate only the top beam from the sorted list returned by beam search. We compare methods on standard sentence-level metrics – CIDEr[120], Meteor[7], and Bleu[90] – computed between the ground truth captions and the (full) reconstructed sentences. We note that the metrics are computed over the entire sentence (and not just the blank region) in order to capture the quality of the alignment of the generated text with the existing sentence structure. As a side effect, the absolute magnitude of these metrics are inflated due to the correctness of the context words, so we focus on the relative performance.

	r=0.25			r=0.5			r=0.75		
	CIDEr	Bleu-4	Meteor	CIDEr	Bleu-4	Meteor	CIDEr	Bleu-4	Meteor
URNN-f	6.54	0.661	0.488	3.744	0.345	0.350	1.927	0.143	0.238
URNN-b	6.58	0.668	0.491	3.931	0.372	0.356	2.476	0.219	0.259
URNN-f+b[132]	6.98	0.709	0.510	4.15	0.398	0.367	2.40	0.209	0.257
BiRNN-f+b	6.94	0.705	0.508	3.99	0.385	0.361	2.24	0.201	0.252
BiRNN-BiBS	**7.12**	**0.722**	**0.517**	**4.18**	**0.401**	**0.368**	**2.52**	**0.222**	**0.265**

Table 4.1: **Comparison of different approaches on Fill-in-the-Blank Image Captioning on COCO [76].** r is the fraction of removed words from sentence, B=5 by default. BiBS outperforms all baselines on all metrics.

4.5.1 Fill-in-the-Blank Image Captioning on COCO

The COCO [76] dataset contains over 120,000 images, each with a rich set of annotations. This include five captions describing the content of each image, collected from Amazon Mechanical Turk workers. We propose a novel fill-in-the-blank image captioning based on this data. Given an image I and a corresponding ground truth caption $\mathbf{y}_1, ..., \mathbf{y}_T$ from the dataset, we remove a sequential portion of the caption such that we are left with a prefix $\mathbf{y}_1, \ldots, \mathbf{y}_s$ and suffix $\mathbf{y}_e, \ldots, \mathbf{y}_T$ consisting of the remaining words on either side of the blank. Using the image and the context from these remaining words, the goal is to generate the missing tokens $\mathbf{y}_{s+1}, \ldots, \mathbf{y}_{e-1}$. This is a challenging task that explores how well models and inference algorithms reason about the past and future during sequence generation. We consider fixed length blanks (*i.e.* where the number of tokens to be generated is know at inference time) and leave variable length sequence reconstruction as future work.

Fixed Length Blanks. In this experimental, we remove r = 25%, 50%, or 75% of the words from the middle of a caption for each image and task the model with generating the lost content. For example, at r = 50% the caption *"A close up of flowers and plants inside of a bowl"* would appear to the system as the blanked caption *"A close __ __ _____ of a bowl"* and the generation task would then be to reproduce the removed subsequence of words *"up of flowers and plants inside."*

As we are interested in bidirectional inference, we train our models on the original COCO

image captioning task (*i.e.* we do not explicitly train to fill blanked captions). Like [65], we use 5000 images for test, 5000 images for validation, and the rest for training. We evaluate on a single caption per image in the test set.

Table 4.1 reports the performance of our approach (BiBS) on this fill-in-the-blank inference task for differently sized blanks (r% of central words removed per sentence). We find that BiBS outperforms all baselines on all metrics. We make special note that the nearest baselines in performance are reranked from 2B beams. While BiBS operates in an alternating left-to-right and right-to-left fashion, it only ever maintains B beams.

Interestingly, backward time model URNN-b consistently outperforms the forward time model URNN-f on all metrics and across all sizes of blanks. This may be due to the way the dataset was collected. When tasked with describing the content of an image, people often begin by grounding their sentences with respect to specific entities visible in the image (especially when humans are depicted). Given this, we would expect many more sentences to begin with the similar words such that generating the beginning of a sentence from the end would be an easier task.

Fig. 4.6 shows several qualitative examples, comparing completed captions from URNN-f, URNN-b, and our BiRNN-BiBS method with ground truth human annotations. The unidirectional models running standard BS typically generate sentences that abruptly clash with existing words at the edge of the blank. For example in the first example, the forward model produces the grammatically incorrect phrase "bannanas on at her stand" and similarly the backward model outputs "The woman has holding a bunch". This behavior is a natural consequence of the inability for these models to efficiently reason about the future. While these unidirectional models struggle to reason about word transitions on either side of the blank, our BiRNN based BiBS algorithm typically produces reconstructions that smoothly fit with the context, producing a reasonable sentence "The woman has a large bunch of

bananas at her stand."

This example also highlights a possible deficiency in our evaluation metrics; while a human observe can clearly tell which of the three sentences is most natural, the sentence level statistics of each are quite similar, with each sharing only the word banana with the ground truth caption "The woman has many bananas and other fruit at her stand." Evaluating generated language is a difficult and open problem that, like inference, is further complicated by fill-in-the-blank context.

BiBS Convergence. To study the convergence of our approach, we consider the true joint probability of filled-in captions as a function of the number of rounds of BiBS. We compute the average of these joint log probabilities after each meta-iteration of BiBS, where we define a meta-iteration as a pair of full forward and backward update passes. We find that joint log probabilities drop quickly (reducing from -2.47 to -2.11 in a single meta-iteration), indicating high quality solutions are found from unidirectional initializations within only a few meta-iterations of BiBS. In practice we find the beams have converged in typically 1 to 2 rounds of BiBS for fill-in-the-blank image captioning.

4.5.2 Visual Madlibs

In this section, we evaluate our approach on the Visual Madlibs[139] fill-in-the-blank description generation task. The Visual Madlibs dataset contains 10,738 images with 12 types of fill-in-the-blank questions answered by 3 workers on Amazon Mechanical Turk. We use *object's affordance* (type 7) and *pair's relationship* (type 12) fill-in-the-blank questions as these types have blanks in the middle of questions. For instance, *People could relax on the couches.* and *The person is putting food in the bowl.* We use 2000 images for validation, train on the remaining images from the train set, and evaluate on their 2,160 image test set. To the best of our knowledge, ours is the first to explore the performance of CNN+LSTM

	type 7		type 12	
	Bleu-1	Bleu-2	Bleu-1	Bleu-2
URNN-f	0.313	0.138	0.275	0.160
URNN-b	0.460	0.284	0.346	0.213
URNN-f+b[132]	0.447	0.275	0.347	0.214
BiRNN-reranking	0.448	0.275	0.347	0.213
BiRNN-BiBS	**0.470**	**0.389**	**0.353**	**0.216**
nCCA	0.56	0.1	0.46	0.07
nCCA(box)	0.60	0.11	0.48	0.08

Table 4.2: **Comparison of different approaches on the Visual Madlibs task** using BLEU-1 and BLEU-2. B= 5 by default.

text generation for this task.

We compare to two additional baselines for these experiments, nCCA[45] and the nCCA(box) method implemented in the Visual Madlibs paper [139]. nCCA maps image and text to a joint-embedding space and then finds the nearest neighbor from the training set to this embedded point. We note that this is a retrieval and not a description generation technique such that it cannot be directly compared with BiBS. nCCA(box) extracts visual features from the ground-truth bounding box of the relevant person or object refered to in the question and thus is an 'oracle' result that makes use of extra ground truth information.

We train a CNN+LSTM model for both *object's affordance* (type 7) and *pair's relationship* (type 12) question types. We extract visual features using VGGNet [109] which is pretrained on the ILSVRC-2012 dataset [101] to recognize 1000 object classes. We evaluate on the test data using Bleu-1 and Bleu-2 (to be consistent with [139]). Higher order metrics are not meaningful because the average number of words in the blank is less than 2. Table 4.2 shows the results of this experiment. We find that BiBS outperforms the other generation based baselines in both question types and is competitive with the retrieval based nCCa techniques, greatly outperforming the nCCa retrieval and nCCA(box) oracle methods on the Bleu-2 metric.

4.6 Conclusions

In summary, we study a novel fill-in-the-blank image captioning task aimed at evaluating how well sequence generation models and their associated inference algorithms incorporate known information from both the past and future to 'fill in the gaps'. This is a challenging task and we demonstrate that standard Beam Search is poorly suited for this task. We develop a Bidirectional Beam Search (BiBS) algorithm based on an approximation of the full joint distribution over output sequences that is efficient to compute in Bidirectional Recurrent Neural Network models. To the best of our knowledge, this is the first algorithm for top-B MAP inference in Bidirectional RNNs. We have demonstrated that BiBS outperforms natural baselines at both fill-in-the-blank image captioning and Visual Madlibs.

4.7 Appendix

BiBS Convergence. Our proposed algorithm, BiBS, typically converges within in 1 to 2 rounds for the fill-in-the-blank image captioning task. In Figure 4.5, we examine the convergence more closely. We measure convergence by the average joint log probability (denoted log(P) in the figure) achieved by the highest ranked beam at a given round of the algorithm for all examples. We calculate log(P) from either the forward RNN or backward RNN depending on the direction of the most recent pass. Our results show BiBS can generate beams with high probabilities compared with `URNN-b`(-2.4714) and `URNN-f+b` (-2.2077) in very few rounds. In Fig. 4.6, we show additional qualitative results that demonstrate how the highest ranked sentences change as the BiBS algorithm progress through meta-iterations.

Variable length blanks. In the experimental setup, we fix the length of blanks to be generated, *i.e.* the number of words to produce is known at inference time. In this section, we loosen this constraint and analyze the setting where the length of blanks is unknown.

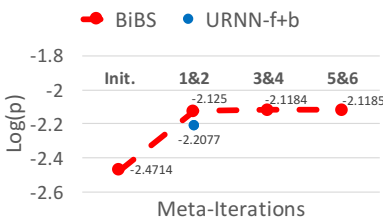


Figure 4.5: log(p) vs. No. Iterations. log(P) is averaged over the entire test dataset. Our model is initialized with right-to-left standard BS (URNN-b, Init). `1&2` indicates to take the maximum probability over left-to-right beams (1st iter) and right-to-left beams (2ed iter), which is consistent with baseline `URNN-f+b`.

	r=0.25			r=0.5			r=0.75		
	CIDEr	Bleu-4	Meteor	CIDEr	Bleu-4	Meteor	CIDEr	Bleu-4	Meteor
URNN-f	5.607	0.569	0.44	4.232	0.432	0.370	2.594	0.268	0.269
URNN-b	5.514	0.561	0.436	4.151	0.424	0.367	2.909	0.303	0.285
URNN-f+b[132]	5.632	0.57	0.44	4.377	0.451	0.376	2,924	**0.306**	0.287
BiRNN-BiBS	**5.935**	**0.614**	**0.460**	**4.4**	**0.454**	**0.38**	**2.936**	0.305	**0.288**

Table 4.3: **Comparison of different approaches on Fill-in-the-Blank Image Captioning on COCO [76] with variable length blanks.** r is the fraction of removed words from sentence, B=5 by default. BiBS outperforms all baselines on all metrics.

We adjust our methodology to perform BiBS over a range of blank lengths. In order to calibrate what lengths we to search over, we first generate the top-1 left-to-right beam Y^f by only feeding words on the left of the blank and right-to-left top-1 beam Y^b by only feeding words on the right side of the blank. Then, we define the range of length of the blank as

$$[\min\{len(Y^f), len(Y^b)\},\ \max\{len(Y^f), len(Y^b)\}] \tag{4.7}$$

where, $len(Y)$ is the length of beam Y. We perform BiBS at each length in this range with all other settings. We select the highest probability completion across all lengths. Table 4.3 reports the performance of our approach (BiBS) on this fill-in-the-blank inference task. We find that BiBS outperforms nearly all baselines on all metrics (narrowly being bested by `URNN-f+b` ar $r = 0.75$ Blue-4).

(GT) Some baseball players are playing a game
(Init) Some baseball baseball bat during a game
(1st) Some baseball baseball players in a game
(2ed) Some baseball players playing in a game

(GT) gourmet pizza cooked and sliced and on a plate
(Init) gourmet pizza pepperoni pizza is sitting on a plate
(1st) gourmet pizza with pizza and cheese on a plate
(2ed) gourmet pizza with pepperoni and cheese on a plate

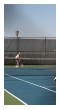
(GT) a woman in a red shirt is holding a blue and orange kite
(Init) a woman in a man is holding a blue and orange kite
(1st) a woman in a a hat is holding a blue and orange kite
(2ed) a woman in a red shirt is holding a blue and orange kite

(GT) the doors of the old stove are opened in the dilapidated room
(Init) the doors of a toilet that is in the dilapidated room
(1st) the doors of is a stove that is in the dilapidated room
(2ed) the doors of the old stove is made in the dilapidated room

(GT) this woman is sitting in front of a restaurant smoking a cigarette
(Init) this woman is is working on a laptop and smoking a cigarette
(1st) this woman is sitting down on a laptop while smoking a cigarette
(2ed) this woman is sitting down in a chair and smoking a cigarette

(GT) a man readies his bat while play a game of baseball
(Init) a man readies of baseball players playing a game of baseball
(1st) a man readies of baseball while playing a game of baseball
(2ed) a man readies a bat while playing a game of baseball

(GT) the skier went down the ramp of fake snow
(Init) the skier street filled with lots of fake snow
(1st) the skier is covered in the of fake snow
(2ed) the skier is sitting in front of fake snow



(GT) a female tennis player in the distance prepares to serve
(Init) a female watch as a tennis player prepares to serve
(1st) a female tennis player a tennis court prepares to serve
(2ed) a female tennis player on the court prepares to serve

(GT) a man drinking some wine using a wineglass
(Init) a man cell phone while holding a wineglass
(1st) a man a smiling while holding a wineglass
(2ed) a man is smiling while holding a wineglass

(GT) a view out of a window that shows a winding river
(Init) a view out a bird flying over a winding river
(1st) a view out of a bird looking over a winding river
(2ed) a view out of a fence looking over a winding river

(GT) an adorable little girl holding two ski poles
(Init) an adorable riding skis while holding ski poles
(1st) an adorable snow skis and two ski poles
(2ed) an adorable young girl holding a ski poles

(GT) a flat screen tv mounted above a fire place
(Init) a flat filled with furniture and a fire place
(1st) a flat room with television and a fire place
(2ed) a flat screen tv sitting on a fire place

Figure 4.6: **Performance vs. Iteration.** Our model is initialized with right-to-left standard BS (Init) and updated alternatively from left-to-right (1st) and right-to-left (2ed).

Chapter 5

Conclusion

To summarize, we studied a bunch of Greedy algorithms in structured (neural) models and applied these techniques in active learning, proposing bounding boxes, and generating image descriptions, etc. The key challenge of structured-output problems is that the search space is exponentially large, which results the exact inference intractable. In above chapters, we proposed 1) a variational "histogram" approach to estimate entropy of Gibbs distribution approximated by a coarsened but effective distribution that may be viewed as a histogram over M bins; 2) formulate the search for a set of diverse bounding boxes as an optimization problem and showed that heuristic approaches used in prior work are special cases in our formulation; 3) a top-B MAP inference in bidirectional RNNs for fill-in-the-blank tasks.

Overall, our proposed approaches are well motivated, computationally efficient, and easy to implement in practice.

Future works involve in applying bidirectional generative models on real tasks. For example, in video generation, objects might be occluded by other objects during a short period of time. Also, in reinforcement learning tasks, the full state of environment is partially observed, in particular, when the environment is non-homogeneous, building a notion of how the environment behaves might boost the performance further.

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Sysstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *PAMI*, (To Appear) 2012.

[2] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *PAMI*, 34(11):2189–2202, Nov 2012.

[3] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *ICCV*, 2015.

[4] Pablo Arbelaez, Jordi Pont Tuset, Jonathan T.Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping. In *CVPR*, 2014.

[5] H. Arora, N. Loeff, D.A. Forsyth, and N. Ahuja. Unsupervised segmentation of objects using efficient learning. In *CVPR*, 2007.

[6] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[7] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005.

[8] Dhruv Batra. An Efficient Message-Passing Algorithm for the M-Best MAP Problem. In *UAI*, 2012.

[9] Dhruv Batra, Rahul Sukthankar, and Tsuhan Chen. Semi-supervised clustering via learnt codeword distances. In *BMVC*, 2008.

[10] Dhruv Batra, Adarsh Kowdle, Devi Parikh, Jiebo Luo, and Tsuhan Chen. iCoseg: Interactive Co-segmentation with Intelligent Scribble Guidance. In *CVPR*, 2010.

[11] Dhruv Batra, Payman Yadollahpour, Abner Guzman-Rivera, and Gregory Shakhnarovich. Diverse M-Best Solutions in Markov Random Fields. In *ECCV*, 2012.

[12] Mathias Berglund, Tapani Raiko, Mikko Honkala, Leo Karkkainen, Akos Vetek, and Juha Karhunen. Bidirectional recurrent neural networks as generative models. In *NIPS*, 2015.

[13] Matthew B. Blaschko and Christoph H. Lampert. Learning to localize objects with structured output regression. In *ECCV*, 2008.

[14] M.B. Blaschko. Branch and bound strategies for non-maximal suppression in object detection. In *EMMCVPR*, pages 385–398, 2011.

[15] Yuri Boykov, Olga Veksler, and Ramin Zabih. Efficient approximate energy minimization via graph cuts. *PAMI*, 20(12):1222–1239, 2001.

[16] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23 (11):1222–1239, 2001.

[17] Y.Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. *ICCV*, 2001.

[18] N. Buchbinder, M. Feldman, J. Naor, and R. Schwartz. A tight (1/2) linear-time approximation to unconstrained submodular maximization. In *FOCS*, 2012.

[19] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 335–336, 1998.

[20] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *CVPR*, 2010. doi: 10.1109/CVPR.2010.5540063.

[21] Chao Chen, Vladimir Kolmogorov, Yan Zhu, Dimitris Metaxas, and Christoph H. Lampert. Computing the m most probable modes of a graphical model. In *AISTATS*,

2013.

[22] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016.

[23] Xinlei Chen and C. Lawrence Zitnick. Mind's Eye: A Recurrent Visual Representation for Image Caption Generation. In *CVPR*, 2015.

[24] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr. Bing:binarized normed gradients for objectness estimation at 300fps. In *CVPR*, 2014.

[25] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. In *Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8)*, 2014.

[26] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[27] Brendan Collins, Jia Deng, Kai Li, and Li Fei-Fei. Towards scalable dataset construction: An active learning approach. In *ECCV*, 2008. ISBN 978-3-540-88681-5.

[28] Aron Culotta and Andrew McCallum. Reducing labeling effort for structured prediction tasks. In *AAAI*, 2005.

[29] George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(1):30–42, 2012.

[30] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005. ISBN 0-7695-2372-2. doi: http://dx.doi.org/10.1109/CVPR.2005.177. URL http://dx.doi.org/10.1109/CVPR.2005.177.

[31] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[32] Thomas Deselaers, Bogdan Alexe, and Vittorio Ferrari. Localizing objects while learning their appearance. In *ECCV*, 2010.

[33] D. Dey, T. Liu, M. Hebert, and J. A. Bagnell. Contextual sequence prediction with application to control library optimization. In *Robotics Science and Systems (RSS)*, 2012.

[34] Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term Recurrent Convolutional Networks for Visual Recognition and Description. In *CVPR*, 2015.

[35] E.L.Lawler and D.E.Wood. Branch-and-bound methods: A survey. *Operations Research*, 14(4):699–719, 1966.

[36] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html, 2007.

[37] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, June 2010.

[38] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

[39] Hao Fang, Saurabh Gupta, Forrest N. Iandola, Rupesh Kumar Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C. Platt, C. Lawrence Zitnick, and Geoffrey Zweig. From Captions to Visual Concepts and Back. In *CVPR*, 2015.

[40] Alireza Fathi, Maria Florina Balcan, Xiaofeng Ren, and James M. Rehg. Combining self training and active learning for video segmentation. In *BMVC*, 2011.

http://dx.doi.org/10.5244/C.25.78.

[41] U. Feige, V. Mirrokni, and J. Vondrák. Maximizing non-monotone submodular functions. In *FOCS*, 2007.

[42] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 32(9):1627–1645, 2010.

[43] Yoav Freund, H.Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168, 1997. ISSN 0885-6125. doi: 10.1023/A:1007330508534. URL http://dx.doi.org/10.1023/A%3A1007330508534.

[44] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[45] Yunchao Gong, Qifa Ke, Michael Isard, and Svetlana Lazebnik. A multi-view embedding space for modeling internet images, tags, and their semantics. *International journal of computer vision*, 106(2):210–233, 2014.

[46] Abel Gonzalez-Garcia, Alexander Vezhnevets, and Vittorio Ferrari. An active search strategy for efficient object detection. In *CVPR*, 2015.

[47] Abner Guzman-Rivera, Dhruv Batra, and Pushmeet Kohli. Multiple Choice Learning: Learning to Produce Multiple Structured Outputs. In *Proc. NIPS*, 2012.

[48] Abner Guzman-Rivera, Pushmeet Kohli, and Dhruv Batra. Divmcuts: Faster training of structural svms with diverse m-best cutting-planes. In *AISTATS*, 2013.

[49] Abner Guzman-Rivera, Pushmeet Kohli, Dhruv Batra, and Rob Rutenbar. Efficiently enforcing diversity in multi-output structured prediction. In *AISTATS*, 2014.

[50] James Hays and Alexei A. Efros. im2gps: estimating geographic information from a single image. In *CVPR*, 2008.

[51] T. Hazan and A. Shashua. Norm-product belief propagation: Primal-dual message-

passing for approximate inference. *Information Theory, IEEE Trans. on*, 56(12):6294–6316, Dec 2010.

[52] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014.

[53] Xuming He and Richard S. Zemel. Learning hybrid models for image annotation with partially labeled data. In *NIPS*, 2008. URL http://dblp.uni-trier.de/db/conf/nips/nips2008.html#HeZ08.

[54] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97, 2012.

[55] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. ISSN 0899-7667.

[56] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Recovering surface layout from an image. *IJCV*, 75(1), 2007.

[57] J. Hosang, R. Benenson, and B. Schiele. How good are detection proposals, really? In *BMVC*, 2014.

[58] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *CoRR*, 2015. URL http://arxiv.org/abs/1508.01991.

[59] Sutskever Ilya, Vinyals Oriol, and Quoc V. Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

[60] Prateek Jain and Ashish Kapoor. Active learning for large multi-class problems. In *CVPR*, 2009.

[61] T. Joachims, T. Finley, and Chun-Nam Yu. Cutting-plane training of structural svms. *Machine Learning*, 77(1):27–59, 2009.

[62] Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. In

EMNLP, 2013.

[63] A. Kapoor, K. Grauman, R. Urtasun, and T.J. Darrell. Active learning with gaussian processes for object categorization. In *ICCV*, 2007.

[64] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015.

[65] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015.

[66] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2003.

[67] Pushmeet Kohli and Philip H. S. Torr. Measuring uncertainty in graph cut solutions. *CVIU*, 112(1):30–38, 2008. ISSN 1077-3142.

[68] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[69] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? *PAMI*, 26(2):147–159, 2004.

[70] Philipp Krahenbuhl and Vladlen Koltun. Learning to propose objects. In *CVPR*, 2015.

[71] Andreas Krause and Daniel Golovin. Submodular function maximization. In *Tractability: Practical Approaches to Hard Problems (to appear)*. Cambridge University Press, 2014.

[72] Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *J. Mach. Learn. Res.*, 9:235–284, 2008.

[73] Daniel Küttel, Matthieu Guillaumin, and Vittorio Ferrari. Segmentation propagation in imagenet. In *ECCV*, 2012.

[74] Christoph H Lampert, Matthew B Blaschko, and Thomas Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *TPMAI*, 31(12): 2129–2142, 2009.

[75] H. Lin and J. Bilmes. A class of submodular functions for document summarization. In *ACL*, 2011.

[76] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context, 2014.

[77] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollr, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[78] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[79] W. Luo, A. G. Schwing, and R. Urtasun. Latent Structured Active Learning. In *NIPS*, 2013.

[80] D.J.C. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992. doi: 10.1162/neco.1992.4.4.590.

[81] Subhransu Maji, Tamir Hazan, and Tommi Jaakkola. Active boundary annotation using random map perturbations. In *AISTATS*, 2014.

[82] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan L Yuille. Explain images with multimodal recurrent neural networks. *arXiv preprint arXiv:1410.1090*, 2014.

[83] Franziska Meier, Amir Globerson, and Fei Sha. The More the Merrier: Parameter Learning for Graphical Models with Multiple MAPs. In *ICML Workshop on Inferning: Interactions between Inference and Learning*, 2013.

[84] Thomas P Minka. Expectation propagation for approximate bayesian inference. In *Pro-*

ceedings of the Seventeenth conference on Uncertainty in artificial intelligence, pages 362–369. Morgan Kaufmann Publishers Inc., 2001.

[85] M Minoux. Accelerated greedy algorithms for maximizing submodular set functions. *Optimization Techniques*, pages 234–243, 1978.

[86] G.L. Nemhauser, L.A. Wolsey, and M.L. Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294, 1978.

[87] Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *ICML*, 2005.

[88] D. Nilsson. An efficient algorithm for finding the m most probable configurations in probabilistic expert systems. *Statistics and Computing*, 8:159–173, 1998. ISSN 0960-3174. URL http://dx.doi.org/10.1023/A:1008990218483. 10.1023/A:1008990218483.

[89] George Papandreou and Alan L. Yuille. Perturb-and-map random fields: Using discrete optimization to learn and sample from energy models. In *ICCV*, 2011.

[90] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL http://dx.doi.org/10.3115/1073083.1073135.

[91] D. Parikh and C.L. Zitnick. The role of features, algorithms and data in visual recognition. In *CVPR*, 2010. doi: 10.1109/CVPR.2010.5539920.

[92] Amar Parkash and Devi Parikh. Attributes for classifier feedback. In *ECCV*, 2012.

[93] John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, 1999.

[94] Adarsh Prasad, Stefanie Jegelka, and Dhruv Batra. Submodular meets structured:

Finding diverse subsets in exponentially-large structured item sets. In *NIPS*, 2014.

[95] G.-J. Qi, Xian-Sheng Hua, Yong Rui, Jinhui Tang, and Hong-Jiang Zhang. Two-dimensional active learning for image classification. In *CVPR*, pages 1–8, 2008.

[96] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[97] Mengye Ren, Ryan Kiros, and Richard Zemel. Exploring models and data for image question answering. In *NIPS*, 2015.

[98] Stephane Ross, Jiaji Zhou, Yisong Yue, Debadeepta Dey, and J. Andrew Bagnell. Learning policies for contextual submodular prediction. In *ICML*, 2013.

[99] Dan Roth and Kevin Small. Margin-based active learning for structured output spaces. In *ECML*, 2006.

[100] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "Grabcut": interactive foreground extraction using iterated graph cuts. *SIGGRAPH*, 2004.

[101] Olga Russakovsky, Jia Deng, Jonathan Krause, Alex Berg, and Li Fei-Fei. The ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012). http://www.image-net.org/challenges/LSVRC/2012/, 2012.

[102] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Labelme: A database and web-based tool for image annotation. *IJCV*, 77(1-3):157–173, May 2008. ISSN 0920-5691. URL http://dx.doi.org/10.1007/s11263-007-0090-8.

[103] Haşim Sak, Andrew Senior, Kanishka Rao, and Françoise Beaufays. Fast and accurate recurrent neural network acoustic models for speech recognition. *arXiv preprint arXiv:1507.06947*, 2015.

[104] M. Schmidt. http://www.cs.ubc.ca/~schmidtm/Software/UGM.html. UGM: A Matlab toolbox for probabilistic undirected graphical models.

[105] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and

Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *ICLR*, 2014. URL http://openreview.net/document/d332e77d-459a-4af8-b3ed-55ba.

[106] B. Settles. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool, 2012.

[107] Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *EMNLP*, 2008.

[108] Behjat Siddiquie and Abhinav Gupta. Beyond active noun tagging: Modeling contextual interactions for multi-class active learning. In *CVPR*, 2010. URL http://dblp.uni-trier.de/db/conf/cvpr/cvpr2010.html#SiddiquieG10.

[109] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[110] A. Sorokin and D. Forsyth. Utility data annotation with amazon mechanical turk. In *Workshop on Internet Vision, CVPR.*, pages 1–8, 2008. doi: 10.1109/CVPRW.2008.4562953.

[111] Matthew Streeter and Daniel Golovin. An online algorithm for maximizing submodular functions. In *NIPS*, 2008.

[112] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep neural networks for object detection. In *NIPS*, pages 2553–2561, 2013. URL http://papers.nips.cc/paper/5207-deep-neural-networks-for-object-detection.pdf.

[113] Christian Szegedy, Scott Reed, and Dumitru Erhan. Scalable, high-quality object detection. In *CVPR*, 2014.

[114] Daniel Tarlow, Ryan Prescott Adams, and Richard S. Zemel. Randomized optimum models for structured prediction. In *Fifteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2012.

[115] Ben Taskar, Carlos Guestrin, and Daphne Koller. Max-margin markov networks. In

NIPS, 2003.

[116] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *JMLR*, 2:45–66, March 2002. ISSN 1532-4435.

[117] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 6:1453–1484, 2005.

[118] J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, and A.W.M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 2013.

[119] Leslie G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2), 1979.

[120] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4566–4575, 2015.

[121] Jakob Verbeek and William Triggs. Scene Segmentation with CRFs Learned from Partially Labeled Images. In *NIPS*, 2008. URL http://hal.inria.fr/inria-00321051.

[122] S. Vijayanarasimhan and K. Grauman. What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations. In *CVPR*, 2009.

[123] Sudheendra Vijayanarasimhan and Kristen Grauman. Active frame selection for label propagation in videos. In *ECCV*, 2012.

[124] Oriol Vinyals and Quoc V. Le. A neural conversational model. *http://arxiv.org/pdf/1506.05869v3.pdf*, 2015.

[125] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. *CoRR*, abs/1411.4555, 2014. URL http://arxiv.org/abs/1411.4555.

[126] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

[127] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015.

[128] Paul Viola and Michael J. Jones. Robust real-time face detection. *Int. J. Comput. Vision*, 57(2):137–154, May 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000013087.49260.fb. URL http://dx.doi.org/10.1023/B:VISI.0000013087.49260.fb.

[129] Luis von Ahn and Laura Dabbish. Labeling images with a computer game. In *CHI*, CHI '04, 2004. ISBN 1-58113-702-8.

[130] Carl Vondrick, Donald Patterson, and Deva Ramanan. Efficiently scaling up crowdsourced video annotation. *IJCV*, 101(1):184–204, January 2013. ISSN 0920-5691. doi: 10.1007/s11263-012-0564-1. URL http://dx.doi.org/10.1007/s11263-012-0564-1.

[131] Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008. doi: http:/dx.doi.org/10.1561/2200000001.

[132] Cheng Wang, Haojin Yang, Christian Bartz, and Christoph Meinel. Image captioning with deep bidirectional lstms. *CoRR*, 2016. URL http://arxiv.org/abs/1604.00790.

[133] J. Winn and N. Jojic. LOCUS: learning object classes with unsupervised segmentation. In *CVPR*, 2005. doi: 10.1109/ICCV.2005.148.

[134] Huijuan Xu and Kate Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. *arXiv preprint arXiv:1511.05234*, 2015.

[135] Jia Xu, Alexander G. Schwing, and Raquel Urtasun. Tell me what you see and i will show you where it is. In *CVPR*, 2014.

[136] Rong Yan, Jie Yang, and Alexander Hauptmann. Automatically labeling video data using multi-class active learning. In *ICCV*, 2003. ISBN 0-7695-1950-4.

[137] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention

networks for image question answering. *arXiv preprint arXiv:1511.02274*, 2015.

[138] Chen Yanover and Yair Weiss. Finding the m most probable configurations using loopy belief propagation. In *NIPS*, 2003.

[139] Licheng Yu, Eunbyung Park, Alexander C. Berg, and Tamara L. Berg. Visual Madlibs: Fill in the blank Description Generation and Question Answering. *ICCV*, 2015.

[140] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537, 2015.

[141] Xiangxin Zhu, Carl Vondrick, Deva Ramanan, and Charless Fowlkes. Do we need more training data or better models for object detection? In *BMVC*, 2012. doi: http://dx.doi.org/10.5244/C.26.80.

[142] C.Lawrence Zitnick and Piotr Dollar. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014.