Submodular meets Structured: Finding Diverse Subsets in Exponentially-Large Structured Item Sets

Adarsh Prasad
UT Austin
adarsh@cs.utexas.edu

Stefanie Jegelka
UC Berkeley
stefje@eecs.berkeley.edu

Dhruv Batra
Virginia Tech
dbatra@vt.edu

Abstract
To cope with the high level of ambiguity faced in domains such as Computer Vision or Natural Language processing, robust prediction methods often search for a diverse set of high-quality candidate solutions or proposals. In structured prediction problems, this becomes a daunting task, as the solution space (image labelings, sentence parses, etc.) is exponentially large. We study greedy algorithms for finding a diverse subset of solutions in structured-output spaces by drawing new connections between submodular functions over combinatorial item sets and High-Order Potentials (HOPs) studied for graphical models. Specifically, we show via examples that when marginal gains of submodular diversity functions allow structured representations, this enables efficient (sub-linear time) approximate maximization by reducing the greedy augmentation step to inference in a factor graph with appropriately constructed HOPs. We discuss benefits, trade-offs, and show that our constructions lead to significantly better proposals.

1 Introduction
Many problems in Computer Vision, Natural Language Processing and Computational Biology involve mappings from an input space $X$ to an exponentially large space $Y$ of structured outputs. For instance, $Y$ may be the space of all segmentations of an image with $n$ pixels, each of which may take $L$ labels, so $|Y| = L^n$. Formulations such as Conditional Random Fields (CRFs) [25], Max-Margin Markov Networks (M3N) [33], and Structured Support Vector Machines (SSVMs) [34] have successfully provided principled ways of scoring all solutions $y \in Y$ and predicting the single highest scoring or maximum a posteriori (MAP) configuration, by exploiting the factorization of a structured output into its constituent “parts”.

In a number of scenarios, the posterior $P(y|x)$ has several modes due to ambiguities, and we seek not only a single best prediction but a set of good predictions:

1. **Interactive Machine Learning.** Systems like Google Translate (for machine translation) or Photoshop (for interactive image segmentation) solve structured prediction problems that are often ambiguous (“what did the user really mean?”). Generating a small set of relevant candidate solutions for the user to select from can greatly improve the results.

2. **M-Best hypotheses in cascades.** Machine learning algorithms are often cascaded, with the output of one model being fed into another [35]. Hence, at the initial stages it is not necessary to make a single perfect prediction. We rather seek a set of plausible predictions that are subsequently re-ranked, combined or processed by a more sophisticated mechanism.

In both scenarios, we ideally want a small set of $M$ plausible (i.e., high scoring) but non-redundant (i.e., diverse) structured-outputs to hedge our bets.

**Submodular Maximization and Diversity.** The task of searching for a diverse high-quality subset of items from a ground set $V$ has been well-studied in information retrieval [5], sensor placement [23], document summarization [27], viral marketing [18], and robotics [10]. Across these domains, submodularity has emerged as an a fundamental and practical concept – a property of functions for measuring diversity of a subset of items. Specifically, a set function $F : 2^V \rightarrow \mathbb{R}$ is submodular if its marginal gains, $F(a|S) \equiv F(S \cup a) - F(S)$ are decreasing, i.e. $F(a|S) \geq F(a|T)$
for all $S \subseteq T$ and $a \notin T$. In addition, if $F$ is monotone, i.e., $F(S) \leq F(T)$, $\forall S \subseteq T$, then a simple greedy algorithm (that in each iteration $t$ adds to the current set $S^t$ the item with the largest marginal gain $F(a|S^t)$) achieves an approximation factor of $\left(1 - \frac{1}{2}\right)$ [28]. This result has had significant practical impact [22]. Unfortunately, if the number of items $|V|$ is exponentially large, then even a single linear scan for greedy augmentation is infeasible.

In this work, we study conditions under which it is feasible to greedily maximize a submodular function over an exponentially large ground set $V = \{v_1, \ldots, v_N\}$ whose elements are combinatorial objects, i.e., labelings of a base set of $n$ variables $y = \{y_1, y_2, \ldots, y_n\}$. For instance, in image segmentation, the base variables $y_i$ are pixel labels, and each item $a \in V$ is a particular labeling of the pixels. Or, if each base variable $y_e$ indicates the presence or absence of an edge $e$ in a graph, then each item may represent a spanning tree or a maximal matching. Our goal is to find a set of $M$ plausible and diverse configurations $\{A_1, \ldots, A_M\}$, whose elements are $\gamma$-best solutions [2, 29] or modes [7, 8], $\forall \gamma \leq M$.

The accuracy of the most accurate segmentation in the set.

**Contributions.** Our principal contribution is a conceptual one. We observe that marginal gains of a number of submodular functions allow structured representations, and this enables efficient greedy maximization over exponentially large ground sets — by reducing the greedy augmentation step to a MAP inference query in a discrete factor graph augmented with a suitably constructed High-Order Potential (HOP). Thus, our work draws new connections between two seemingly disparate but highly related areas in machine learning — submodular maximization and inference in graphical models with structured HOPs. As specific examples, we construct submodular functions for three different, task-dependent definitions of diversity, and provide reductions to three different HOPs for which efficient inference techniques have already been developed. Moreover, we present a generic recipe for constructing such submodular functions, which may be “plugged” with efficient HOPs discovered in future work. Our empirical contribution is an efficient algorithm for producing a set of image segmentations with significantly higher oracle accuracy\(^1\) than previous works. The algorithm is general enough to transfer to other applications. Fig. 1 shows an overview of our approach.

**Related work: generating multiple solutions.** Determinental Point Processes are an elegant probabilistic model over sets of items with a preference for diversity. Its generalization to a structured setting [24] assumes a tree-structured model, an assumption that we do not make. Guzman-Rivera et al. [15, 16] learn a set of $M$ models, each producing one solution, to form the set of solutions. Their approach requires access to the learning sub-routine and repeated re-training of the models, which is not always possible, as it may be expensive or proprietary. We assume to be given a single (pre-trained) model from which we must generate multiple diverse, good solutions. Perhaps the closest to our setting are recent techniques for finding diverse $M$-best solutions [2, 29] or modes [7, 8] in graphical models. While [7] and [8] are inapplicable since they are restricted to chain and tree graphs, we compare to other baselines in Section 3.2 and 4.

1.1 Preliminaries and Notation

We select from a ground set $V$ of $N$ items. Each item is a labeling $y = \{y_1, y_2, \ldots, y_n\}$ of $n$ base variables. For clarity, we use non-bold letters $a \in V$ for items, and boldface letters $y$ for base set configurations. Uppercase letters refer to functions over the ground set items $F(a|A)$, $R(a|A)$, $D(a|A)$, and lowercase letters to functions over base variables $f(y)$, $r(y)$, $d(y)$.

\(^1\)The accuracy of the most accurate segmentation in the set.
Formally, there is a bijection \( \phi : V \rightarrow [L]^m \) that maps items \( a \in V \) to their representation as base variable labelings \( y = \phi(a) \). For notational simplicity, we often use \( y \in S \) to mean \( \phi^{-1}(y) \in S \), i.e. the item corresponding to the labeling \( y \) is present in the set \( S \subseteq V \). We write \( \ell \in y \) if the label \( \ell \) is used in \( y \), i.e. \( \exists j \) s.t. \( y_j = \ell \). For a set \( c \subseteq [n] \), we use \( y_c \) to denote the tuple \( \{y_i \mid i \in c \} \).

Our goal is to find an ordered set or list of items \( S \subseteq V \) that maximizes a scoring function \( F \). Lists generalize the notion of sets, and allow for reasoning of item order and repetitions. More details about list vs set prediction can be found in [31, 10].

**Scoring Function.** We trade off the relevance and diversity of list \( S \subseteq V \) via a scoring function \( F : 2^V \rightarrow \mathbb{R} \) of the form

\[
F(S) = R(S) + \lambda D(S),
\]

where \( R(S) = \sum_{a \in S} R(a) \) is a modular nonnegative relevance function that aggregates the quality of all items in the list; \( D(S) \) is a monotone normalized submodular function that measure the diversity of items in \( S \); and \( \lambda \geq 0 \) is a trade-off parameter. Similar objective functions were used e.g. in [27]. They are reminiscent of the general paradigm in machine learning of combining a loss function that measures quality (e.g. training error) and a regularization term that encourages desirable properties (e.g. smoothness, sparsity, or “diversity”).

**Submodular Maximization.** We aim to find a list \( S \) that maximizes \( F(S) \) subject to a cardinality constraint \( |S| \leq M \). For monotone submodular \( F \), this may be done via a greedy algorithm that starts with \( S^0 = \emptyset \), and iteratively adds the next best item:

\[
S^t = S^{t-1} \cup a^t, \quad a^t = \arg \max_{a \in V} F(a \mid S^{t-1}).
\]

The final solution \( S^M \) is within a factor of \((1 - \frac{1}{e})\) of the optimal solution \( S^* \): \( F(S^M) \geq (1 - \frac{1}{e})F(S^*) \) [28]. The computational bottleneck is that in each iteration, we must find the item with the largest marginal gain. Clearly, if \(|V|\) has exponential size, we cannot touch each item even once. Instead, we propose “augmentation sub-routines” that exploit the structure of \( V \) and maximize the marginal gain by solving an optimization problem over the base variables.

2 Marginal Gains in Configuration Space

To solve the greedy augmentation step via optimization over \( y \), we transfer the marginal gain from the world of items to the world of base variables and derive functions on \( y \) from \( F \):

\[
\underbrace{F(\phi^{-1}(y) \mid S)}_{f(y|S)} + \underbrace{\lambda D(\phi^{-1}(y) \mid S)}_{d(y|S)} - \underbrace{R(\phi^{-1}(y))}_{r(y)}.
\]

Maximizing \( F(a\mid S) \) now means maximizing \( f(y\mid S) \) for \( y = \phi(a) \). This can be a hard combinatorial optimization problem in general. However, as we will see, there is a broad class of useful functions \( F \) for which \( f \) inherits exploitable structure, and \( \arg \max_y f(y\mid S) \) can be solved exactly, efficiently or at least approximately.

**Relevance Function.** We use a structured relevance function \( R(a) \) that is the score of a factor graph defined over the base variables \( y \). Let \( G = (\mathcal{V}, \mathcal{E}) \) be a graph defined over \( \{y_1, y_2, \ldots, y_n\} \), i.e. \( \mathcal{V} = [n], \mathcal{E} \subseteq (\mathcal{V}^2) \). Let \( \mathcal{C} = \{C \mid C \subseteq \mathcal{V}\} \) be a set of cliques in the graph, and let \( \theta_C : [L]^{|C|} \rightarrow \mathbb{R} \) be the log-potential functions (or factors) for these cliques. The quality of an item \( a = \phi^{-1}(y) \) is then given by \( R(a) = r(y) = \sum_{C \in \mathcal{C}} \theta_C(y_C) \). For instance, with only node and edge factors, this quality becomes \( r(y) = \sum_{p \in \mathcal{V}} \theta_p(y_p) + \sum_{(p,q) \in \mathcal{E}} \theta_{pq}(y_p, y_q) \). In this model, finding the single highest quality item corresponds to maximum a posteriori (MAP) inference in the factor graph.

Although we refer to terms with probabilistic interpretations such as “MAP”, we treat our relevance function as output of an energy-based model [26] such as a Structured SVM [34]. For instance, \( r(y) = \sum_{C \in \mathcal{C}} \theta_C(y_C) = w^T \psi(y) \) for parameters \( w \) and feature vector \( \psi(y) \). Moreover, we assume that the relevance function \( r(y) \) is nonnegative\(^2\). This assumption ensures that \( F(\cdot) \) is monotone. If \( F \) is non-monotone, algorithms other than the greedy are needed [4, 13]. We leave this generalization for future work. In most application domains the relevance function is learned from data and thus our positivity assumption is not restrictive – one can simply learn a positive relevance function. For instance, in SSVMs, the relevance weights are learnt to maximize the margin between the correct labeling and all incorrect ones. We show in the supplement that SSVM parameters that assign nonnegative scores to all labelings achieve exactly the same hinge loss (and thus the same generalization error) as without the nonnegativity constraint.

\(^2\)Strictly speaking, this condition is sufficient but not necessary. We only need nonnegative marginal gains.
Hamming Ball Groups

Our scheme relies on constructing and for reducing their marginal gains to structured representations over the base variables $d$. We next discuss a general recipe for constructing monotone submodular diversity functions $D(S)$, and for reducing their marginal gains to structured representations over the base variables $d(y | S)$.

3 Structured Diversity Functions

We next discuss a general recipe for constructing monotone submodular diversity functions $D(S)$, and for reducing their marginal gains to structured representations over the base variables $d(y | S)$. Our scheme relies on constructing groups $G_i$ that cover the ground set, i.e. $V = \bigcup_i G_i$. These groups will be defined by task-dependent characteristics – for instance, in image segmentation, $G$ contains objects belonging to the same category. The groups can be overlapping. For instance, if a segmentation $y$ contains pixels labeled “grass” and “cow”, then $y \in G_{\text{grass}}$ and $y \in G_{\text{cow}}$.

**Group Coverage: Count Diversity.** Given $V$ and a set of groups $\{G_i\}$, we measure the diversity of a list $S$ in terms of its group coverage, i.e., the number of groups covered jointly by items in $S$:

$$D(S) = \left| \left\{ i \mid G_i \cap S \neq \emptyset \right\} \right|,$$

where we define $G_i \cap S$ as the intersection of $G_i$ with the set of unique items in $S$. It is easy to show that this function is monotone submodular. If $G_\ell$ is the group of all segmentations that contain label $\ell$, then the diversity measure of a list of segmentations $S$ is the number of object labels that appear in any $a \in S$. The marginal gain is the number of new groups covered by $a$:

$$D(a \mid S) = \left| \left\{ i \mid a \in G_i \text{ and } S \cap G_i = \emptyset \right\} \right|.$$

Thus, the greedy algorithm will try to find an item/segmentation that belongs to as many as yet unused groups as possible.

**Group Coverage: General Diversity.** More generally, instead of simply counting the number of groups covered by $S$, we can use a more refined decay

$$D(S) = \sum_i h(\left| G_i \cap S \right|),$$

where $h$ is any nonnegative nondecreasing concave scalar function. This is a sum of submodular functions and hence submodular. Eqn. (4) is a special case of Eqn. (6) with $h(y) = \min\{1, y\}$. Other possibilities are $\sqrt{\cdot}$, or $\log(1 + \cdot)$. For this general definition of diversity, the marginal gain is

$$D(a \mid S) = \sum_{i:G_i \ni a} \left[ h(1 + \left| G_i \cap S \right|) - h(\left| G_i \cap S \right|) \right].$$

Since $h$ is concave, the gain $h(1 + \left| G_i \cap S \right|) - h(\left| G_i \cap S \right|)$ decreases as $S$ becomes larger. Thus, the marginal gain of an item $a$ is proportional to how rare each group $G_i \ni a$ is in the list $S$.

In each step of the greedy algorithm, we maximize $r(y) + \lambda d(y | S)$. We already established a structured representation of $r(y)$ via a factor graph on $y$. In the next few subsections, we specify three example definitions of groups $G_i$ that instantiate three diversity functions $D(S)$. For each $D(S)$, we show how the marginal gains $D(a | S)$ can be expressed as a specific High-Order Potential (HOP) $d(y | S)$ in the factor graph over $y$. These HOPs are known to be efficiently optimizable, and hence we can solve the augmentation step efficiently. Table 1 summarizes these connections.

**Diversity and Parsimony.** If the groups $G_i$ are overlapping, some $y$ can belong to many groups simultaneously. While such a $y$ may offer an immediate large gain in diversity, in many applications it is more natural to seek a small list of complementary labelings rather than having all labels occur in the same $y$. For instance, in image segmentation with groups defined by label presence (Sec. 3.1), natural scenes are unlikely to contain many labels at the same time. Instead, the labels should be spread across the selected labelings $y \in S$. Hence, we include a parsimony factor $p(y)$ that biases towards simpler labelings $y$. This term is a modular function and does not affect the diversity functions directly. We next outline some example instantiations of the functions (4) and (6).
3.1 Diversity of Labels

For the first example, let $G_\ell$ be the set of all labelings $y$ containing the label $\ell$, i.e., $y \in G_\ell$ if and only if $y_j = \ell$ for some $j \in [n]$. Such a diversity function arises in multi-class image segmentation – if the highest scoring segmentation contains “sky” and “grass”, then we would like to add complementary segmentations that contain an unused class label, say “sheep” or “cow”.

**Structured Representation of Marginal Gains.** The marginal gain for this diversity function turns out to be a HOP called *label cost* [9]. It penalizes each label that occurs in a particular segmentation. Let $\text{lcount}_S(\ell)$ be the number of segmentations in $S$ that contain label $\ell$. In the simplest case of coverage diversity (4), the marginal gain provides a constant reward for every as yet unseen label $\ell$:

$$d(y \mid S) = |\{\ell \mid y \in G_\ell, S \cap G_\ell = \emptyset\}| = \sum_{\ell \in y, \text{lcount}_S(\ell)=0} 1.$$  \hspace{1cm} (8)

For the general group coverage diversity (6), the gain becomes

$$d(y \mid S) = \sum_{\ell \in G_L \ni y} \left[h(1 + |G_\ell \cap S|) - h(|G_\ell \cap S|)\right] = \sum_{\ell \in y} \left[h(1 + \text{lcount}_S(\ell)) - h(\text{lcount}_S(\ell))\right].$$

Thus, $d(y \mid S)$ rewards the presence of a label $\ell$ in $y$ by an amount proportional to how rare $\ell$ is in the segmentations already chosen in $S$. The parsimony factor in this setting is $p(y) = \sum_{\ell \in y} c(\ell)$. In the simplest case, $c(\ell) = -1$, i.e. we are charged a constant for every label used in $y$.

With this type of diversity (and parsimony terms), the greedy augmentation step is equivalent to performing MAP inference in a factor graph augmented with label reward HOPs: $\text{argmax}_y r(y) + \lambda(d(y \mid S) + p(y))$. Delong et al. [9] show how to perform approximate MAP inference with such label costs via an extension to the standard $\alpha$-expansion [3] algorithm.

**Label Transitions.** Label Diversity can be extended to reward not just the presence of previously unseen labels, but also the presence of previously unseen label transitions (e.g., a person in front of a car or a person in front of a house). Formally, we define one group $G_{\ell,\ell'}$ per label pair $\ell, \ell'$, and $y \in G_{\ell,\ell'}$ if it contains two adjacent variables $y_i, y_j$ with labels $y_i = \ell, y_j = \ell'$. This diversity function rewards the presence of a label pair $(\ell, \ell')$ by an amount proportional to how rare this pair is in the segmentations that are part of $S$. For such functions, the marginal gain $d(y \mid S)$ becomes a HOP called cooperative cuts [17]. The inference algorithm in [20] gives a fully polynomial-time approximation scheme for any nondecreasing, nonnegative $h$, and the exact gain maximizer for the count function $h(y) = \min\{1, y\}$. Further details may be found in the supplement.

3.2 Diversity via Hamming Balls

The label diversity function simply rewards the presence of a label $\ell$, irrespective of which or how many variables $y_i$ were assigned that label. The next diversity function rewards a large Hamming distance $\text{Ham}(y^1, y^2) = \sum_{i=1}^n [y_i^1 \neq y_i^2]$ between configurations (where $[\cdot]$ is the Iverson bracket.) Let $B_k(y)$ denote the $k$-radius Hamming ball centered at $y$, i.e. $B_k(y) = \{y' \mid \text{Ham}(y', y) \leq k\}$. The previous section constructed one group per label $\ell$. Now, we construct one group $G_y$ for each configuration $y$, which is the $k$-radius Hamming ball centered at $y$, i.e. $G_y = B_k(y)$.

**Structured Representation of Marginal Gains.** For this diversity, the marginal gain $d(y \mid S)$ becomes a HOP called cardinality potential [32]. For count group coverage, this becomes

$$d(y \mid S) = \left|\{y' \mid G_{y'} \cap (S \cup y) \neq \emptyset\}\right| - \left|\{y' \mid G_{y'} \cap S \neq \emptyset\}\right|$$ \hspace{1cm} (9a)

$$= \left|\bigcup_{y' \in S \cup y} B_k(y')\right| - \left|\bigcup_{y' \in S} B_k(y')\right| = |B_k(y)| - |B_k(y) \cap \bigcup_{y' \in S} B_k(y')|, \hspace{1cm} (9b)$$

i.e., the marginal gain of adding $y$ is the number of new configurations $y'$ covered by the Hamming ball centered at $y$. Since the size of the intersection of $B_k(y)$ with a union of Hamming balls does not have a straightforward structured representation, we maximize a lower bound on $d(y \mid S)$ instead:

$$d(y \mid S) \geq d_{lb}(y \mid S) \equiv |B_k(y)| - \sum_{y' \in S} |B_k(y) \cap B_k(y')|$$ \hspace{1cm} (10)
This lower bound $d_{lb}(y|S)$ overcounts the intersection in Eqn. (9b) by summing the intersections with each $B_i(y')$ separately. We can also interpret this lower bound as clipping the series arising from the inclusion-exclusion principle to the first-order terms. Importantly, (10) depends on $y$ only via its Hamming distance to $y'$. This is a cardinality potential that depends only on the number of variables $y_i$ assigned to a particular label. Specifically, ignoring constant terms, the lower bound can be written as a summation of cardinality factors (one for each previous solution $y' \in S$): $d_{lb}(y|S) = \sum_{y' \in S} \theta_{y'}(y)$, where $\theta_{y'}(y) = \frac{b}{|S|} - I_{y'}(y)$, $b$ is a constant (size of a k-radius Hamming ball), and $I_{y'}(y)$ is the number of points in the intersection of k-radius Hamming balls centered at $y'$ and $y$.

With this approximation, the greedy step means performing MAP inference in a factor graph augmented with cardinality potentials: $\arg\max_{\gamma} r(y) + \lambda d_{lb}(y|S)$. This may be solved via message-passing, and all outgoing messages from cardinality factors can be computed in $O(n \log n)$ time [32]. While this algorithm does not offer any approximation guarantees, it performs well in practice. A subtle point to note is that $d_{lb}(y|S)$ is always decreasing w.r.t. $|S|$ but may become negative due to over-counting. We can fix this by clamping $d_{lb}(y|S)$ to be greater than 0, but in our experiments this was unnecessary – the greedy algorithm never chose a set where $d_{lb}(y|S)$ was negative.

Comparison to DivMBest. The greedy algorithm for Hamming diversity is similar in spirit to the recent work of Batra et al. [2], who also proposed a greedy algorithm (DivMBest) for finding diverse MAP solutions in graphical models. They did not provide any justification for greedy, and our formulation sheds some light on their work. Similar to our approach, at each greedy step, DivMBest involves maximizing a diversity-augmented score: $\arg\max_{y} r(y) + \lambda \sum_{y' \in S} \theta_{y'}(y)$. However, their diversity function grows linearly with the Hamming distance, $\theta_{y'}(y) = \text{Ham}(y', y) = \sum_{i=1}^{n} \|y'_i \neq y_i\|$. Linear diversity rewards are not robust, and tend to over-reward diversity. Our formulation uses a robust diversity function $\theta_{y'}(y) = \frac{b}{|S|} - I_{y'}(y)$ that saturates as $y$ moves far away from $y'$.

In our experiments, we make the saturation behavior smoothly tunable via a parameter $\gamma$: $I_{y'}(y) = e^{-\gamma \text{Ham}(y', y)}$. A larger $\gamma$ corresponds to Hamming balls of smaller radius, and can be set to optimize performance on validation data. We found this to work better than directly tuning the radius $k$.

4 Experiments

We apply our greedy maximization algorithms to two image segmentation problems: (1) interactive binary segmentation (object cutout) (Section 4.1); (2) category-level object segmentation on the PASCAL VOC 2012 dataset [11] (Section 4.2). We compare all methods by their respective oracle accuracies, i.e. the accuracy of the most accurate segmentation in the set of $M$ diverse segmentations returned by that method. For a small value of $M = 5$ to 10, a high oracle accuracy indicates that the algorithm has achieved high recall and has identified a good pool of candidate solutions for further processing in a cascaded pipeline. In both experiments, the label “background” is typically expected to appear somewhere in the image, and thus does not play a role in the label cost/transition diversity functions. Furthermore, in binary segmentation there is only one non-background label. Thus, we report results with Hamming diversity only (label cost and label transition diversities are not applicable). For the multi-class segmentation experiments, we report experiments with all three.

Baselines. We compare our proposed methods against DivMBest [2], which greedily produces diverse segmentation by explicitly adding a linear Hamming distance term to the factor graph. Each Hamming term is decomposable along the variables $y_i$ and simply modifies the node potentials $\theta(y_i) = \theta(y_i) + \lambda \sum_{y' \in S} \left[ y_i \neq y'_i \right]$. DivMBest has been shown to outperform techniques such as M-Best-MAP [36, 1], which produce high scoring solutions without a focus on diversity, and sampling-based techniques, which produce diverse solutions without a focus on the relevance term [2]. Hence, we do not include those methods here. We also report results for combining different diversity functions via two operators: $\otimes$, where we generate the top $M$ solutions for each $k$ diversity functions and then concatenate these lists; and $\oplus$, where we linearly combine diversity functions (with coefficients chosen by $k$-D grid search) and generate $M$ solutions using the combined diversity.

4.1 Interactive segmentation

In interactive foreground-background segmentation, the user provides partial labels via scribbles. One way to minimize interactions is for the system to provide a set of candidate segmentations for the user to choose from. We replicate the experimental setup of [2], who curated 100 images from the PASCAL VOC 2012 dataset, and manually provided scribbles on objects contained in them. For each image, the relevance model $r(y)$ is a 2-label pairwise CRF, with a node term for each
Table 2: PASCAL VOC 2012 val oracle accuracies for different diversity functions.

| Label Cost (LC) | Hamming Ball (HB) | Label Transition (LT) |
|-----------------|-------------------|-----------------------|
| MAP M=5 M=15    | MAP M=5 M=15      | MAP M=5 M=15          |
| \( \min \{ 1, \cdot \} \) | 42.35 45.43 45.58 | DivMBest              |
| \( \sqrt{\cdot} \) | 42.35 45.72 50.01 | HB                    |
| \( \log(1 + \cdot) \) | 42.35 46.28 50.39 | HB \( \otimes \) LC \( \otimes \) LT |

\( \odot \) Combined Diversity  \( \oplus \) Combined Diversity

\( \otimes \) Combined Diversity  \( \oplus \) Combined Diversity

Table 2: PASCAL VOC 2012 val oracle accuracies for different diversity functions.

Table 3: Interactive segmentation: oracle pixel accuracies averaged over 50 test images.

|               | MAP M=2 | MAP M=6 |
|---------------|---------|---------|
| DivMBest      | 91.57   | 95.02   |
| Hamming Ball  | 91.37   | 94.86   |
| DivMBest \( \otimes \) Hamming Ball | - | 95.16 |
| DivMBest \( \oplus \) Hamming Ball | - | 95.14 |

4.2 Category level Segmentation

In category-level object segmentation, we label each pixel with one of 20 object categories or background. We construct a multi-label pairwise CRF on superpixels. Our node potentials are outputs of category-specific regressors trained by [6], and our edge potentials are multi-label Potts. Inference in the presence of diversity terms is performed with the implementations of Delong et al. [9] for label costs, Tarlow et al. [32] for Hamming ball diversity, and Boykov et al. [3] for label transitions.

Results. For each of the 50 test images in our dataset we generated the single best \( y^1 \) and 5 additional solutions \( \{ y^2, \ldots, y^6 \} \) using each method. Table 3 shows the average oracle accuracies for DivMBest, Hamming ball diversity, and their two combinations. We can see that the combinations slightly outperform both approaches.

Results. We evaluate all methods on the PASCAL VOC 2012 data [11], consisting of train, val and test partitions with about 1450 images each. We train the regressors of [6] on train, and report oracle accuracies of different methods on val (we cannot report oracle results on test since those annotations are not publicly available). Diversity parameters \( (\gamma, \lambda) \) are chosen by performing cross-val on val. The standard PASCAL accuracy is the corpus-level intersection-over-union measure, averaged over all categories. For both label cost and transition, we try 3 different concave
functions $h(\cdot) = \min\{1, \cdot\}$, $\sqrt{\cdot}$ and $\log(1 + \cdot)$. Table 2 shows the results.\footnote{MAP accuracies in Table 2 are different because of two different approximate MAP solvers: LabelCost/-Transition use alpha-expansion and HammingBall/DivMBest use message-passing.} Hamming ball diversity performs the best, followed by DivMBest, and label cost/transitions are worse here. We found that while worst on average, label transition diversity helps in an interesting scenario – when the first best segmentation $y^1$ includes a pair of rare or mutually confusing labels (say dog-cat). Fig. 3 shows an example, and more illustrations are provided in the supplement. In these cases, searching for a different label transition produces a better segmentation. Finally, we note that lists produced with combined diversity significantly outperform any single method (including DivMBest).

5 Discussion and Conclusion

In this paper, we study greedy algorithms for maximizing scoring functions that promote diverse sets of combinatorial configurations. This problem arises naturally in domains such as Computer Vision, Natural Language Processing, or Computational Biology, where we want to search for a set of diverse high-quality solutions in a structured output space.

The diversity functions we propose are monotone submodular functions by construction. Thus, if $r(y) + p(y) \geq 0$ for all $y$, then the entire scoring function $F$ is monotone submodular. We showed that $r(y)$ can simply be learned to be positive. The greedy algorithm for maximizing monotone submodular functions has proved useful in moderately-sized unstructured spaces. To the best of our knowledge, this is the first generalization to exponentially large structured output spaces. In particular, our contribution lies in reducing the greedy augmentation step to inference with structured, efficiently solvable HOPs. This insight makes new connections between submodular optimization and work on inference in graphical models. We now address some questions.

Can we sample? One question that may be posed is how random sampling would perform for large ground sets $V$. Unfortunately, the expected value of a random sample of $M$ elements can be much worse than the optimal value $F(S^*)$, especially if $N$ is large. Lemma 1 is proved in the supplement.

**Lemma 1.** Let $S \subseteq V$ be a sample of size $M$ taken uniformly at random. There exist monotone submodular functions where $\mathbb{E}[F(S)] \leq \frac{M}{N} \max_{|S|=M} F(S)$.

Guarantees? If $F$ is nonnegative, monotone submodular, then using an exact HOP inference algorithm will clearly result in an approximation factor of $1 - 1/e$. But many HOP inference procedures are approximate. Lemma 4 formalizes how approximate inference affects the approximation bounds.

**Lemma 2.** Let $F \geq 0$ be monotone submodular. If each step of the greedy algorithm uses an approximate marginal gain maximizer $b^{t+1}$ with $F(b^{t+1} \mid S^t) \geq \alpha \max_{a \in V} F(a \mid S^t) - \epsilon_{t+1}$, then $F(S^M) \geq (1 - \frac{1}{e\alpha}) \max_{|S|=M} F(S) - \sum_{t=1}^M \epsilon_t$.

Parts of Lemma 4 have been observed in previous work [14, 31]; we show the combination in the supplement. If $F$ is monotone but not nonnegative, then Lemma 4 can be extended to a relative error bound $\frac{F(S^M) - F_{\text{min}}}{F(S^*) - F_{\text{min}}} \geq (1 - \frac{1}{e\alpha}) - \sum_{t=1}^M \epsilon_t$ that refers to $F_{\text{min}} = \min_S F(S)$ and the optimal solution $S^*$. While stating these results, we add that further additive approximation losses occur if the approximation bound for inference is computed on a shifted or reflected function (positive scores vs positive energies). We pose theoretical improvements as an open question for future work. That said, our experiments convincingly show that the algorithms perform very well in practice, even when there are no guarantees (as with Hamming Ball diversity).

Generalization. In addition to the three specific examples in Section 3, our constructions generalize to the broad HOP class of upper-envelope potentials [19]. The details are provided in the supplement.

Acknowledgements. We thank Xiao Lin for his help. The majority of this work was done while AP was an intern at Virginia Tech. AP and DB were partially supported by the National Science Foundation under Grant No. IIS-1353694 and IIS-1350553, the Army Research Office YIP Award W911NF-14-1-0180, and the Office of Naval Research Award N00014-14-1-0679, awarded to DB. SJ was supported by gifts from Amazon Web Services, Google, SAP, The Thomas and Stacey Siebel Foundation, Apple, C3Energy, Cisco, Cloudera, EMC, Ericsson, Facebook, GameOnTalis, Guavus, HP, Huawei, Intel, Microsoft, NetApp, Pivotal, Splunk, Virdata, VMware, WANdisco, and Yahoo!.}

References

[1] D. Batra. An Efficient Message-Passing Algorithm for the M-Best MAP Problem. In UAI, 2012.
A  Structured SVMs with nonnegativity constraint

In this section, we will show that SSVMs have no natural origin, and that parameters learnt with non-negativity constraints achieve exactly the same hinge loss as achieved without the nonnegativity constraint.

For a set of \( \ell \) training instances \((x^n, y^n) \in \mathcal{X} \times \mathcal{Y}, n = 1, \ldots, \ell \) from a sample space \( \mathcal{X} \) and label space \( \mathcal{Y} \), the structured SVM minimizes the following regularized risk function.

\[
\min_{w, \xi} \|w\|^2 + C \sum_{n=1}^\ell \max_{y \in \mathcal{Y}} (\Delta(y^n, y) + w'\Psi(x^n, y) - w'\Psi(x^n, y^n))
\]

The function \( \Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+ \) measures a distance in label space and is an arbitrary function satisfying \( \Delta(y, y') \geq 0 \) and \( \Delta(y, y) = 0 \) \( \forall y, y' \in \mathcal{Y} \). The function \( \Psi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d \) is a feature function, extracting some feature vector from a given sample and label.

Because the regularized risk function above is non-differentiable, it is often reformulated in terms of a quadratic program by introducing one slack variable \( \xi_n \) for each sample, each representing the value of the maximum. The standard structured SVM primal formulation is given as follows.

\[
\min_{w, \xi} \|w\|^2 + C \sum_{n=1}^\ell \xi_n
\text{ s.t. } w'\Psi(x^n, y^n) - w'\Psi(x^n, y) + \xi_n \geq \Delta(y^n, y), \quad n = 1, \ldots, \ell, \quad \forall y \in \mathcal{Y}
\]

We will refer to the primal objective in Equation 12 as \( P_1 \) and the optimal solution as \( w_1, \xi_1 \).

Next, consider the following augmented formulation, where \( \hat{w} = \begin{bmatrix} w \\ b \end{bmatrix}, \Psi(x^n, y^n) = \begin{bmatrix} \Psi(x^n, y^n) \\ 1 \end{bmatrix} \).

\[
\min_{\hat{w}, \xi} \|w\|^2 + C \sum_{n=1}^\ell \xi_n
\text{ s.t. } \hat{w}'\Psi(x^n, y^n) - \hat{w}'\Psi(x^n, y) + \xi_n \geq \Delta(y^n, y), \quad n = 1, \ldots, \ell, \quad \forall y \in \mathcal{Y}
\]

We will refer to the primal objective in Equation 13 as \( P_2 \) and the optimal solution as \( \hat{w}_2 = \begin{bmatrix} w_2 \\ b_2 \end{bmatrix}, \xi_2 \). Note: We do not regularize \( b \).

**Claim 1.** \( P_1(w_1, \xi_1) = P_2(\hat{w}_2, \xi_2) \)

**Proof.** Adding a bias feature \( b \) doesn’t affect the objective function and every constraint is invariant of it. Hence, the two problems are equivalent.

Next, consider the formulation, where we extend \( P_2 \) to enforce non-negativity of scores.

\[
\min_{\hat{w}, \xi} \|w\|^2 + C \sum_{n=1}^\ell \xi_n
\text{ s.t. } \hat{w}'\Psi(x^n, y^n) - \hat{w}'\Psi(x^n, y) + \xi_n \geq \Delta(y^n, y), \quad n = 1, \ldots, \ell, \quad \forall y \in \mathcal{Y}
\]

We will refer to the primal objective in Equation 14 as \( P_3 \) and the optimal solution as \( \hat{w}_3 = \begin{bmatrix} w_3 \\ b_3 \end{bmatrix}, \xi_3 \).

**Claim 2.** \( P_1(\hat{w}_2, \xi_2) = P_3(\hat{w}_3, \xi_3) \)

**Proof.** It is easy to see that \( P_3(\hat{w}_3, \xi_3) \leq P_3(\hat{w}_2, \xi_2) \) as \( (\hat{w}_2, \xi_2) \) is the optimal solution for \( P_2 \) and \( (\hat{w}_3, \xi_3) \) is also a feasible solution for \( P_2 \).

Consider the vector \( \hat{w}_2^* = \min_n \min_y \hat{w}_2 \Psi(x^n, y) \) since the bias term doesn’t occur in the objective function, hence \( P_1(\hat{w}_2^*, \xi_2) = P_2(\hat{w}_2, \xi_2) \).

Also, \( (\hat{w}_2^*, \xi_2) \) is a feasible solution to \( P_3 \), hence, \( P_3(\hat{w}_3, \xi_3) \leq P_3(\hat{w}_2^*, \xi_2) = P_2(\hat{w}_2, \xi_2) \).

Therefore, \( P_2(\hat{w}_2, \xi_2) = P_3(\hat{w}_3, \xi_3) \).
Therefore, even after adding the nonnegativity constraints, the solutions achieve the same values for the regularised risk function and hence are expected to have the same generalization guarantees. In practice, the non-negativity constraints can be added in a cutting-plane procedure via a MAP call.

## B Label Transitions

### Groups and Motivating Scenario

In this section, we generalize the label cost diversity function to reward not just the presence of certain labels, but the presence of certain label transitions. For instance, if the highest scoring segmentation contains a “cow” on “grass”, this diversity function will reward other segmentations for containing novel label transitions, such as “sheep-grass” or “cow-ground” or “sheep-sky”. Formally, we define one group $G_{\ell,\ell'}$ per label pair $\ell, \ell'$, and an item $a$ belongs to $G_{\ell,\ell'}$ if $y = \phi(a)$ contains two adjacent variables $y_i, y_j$ with labels $y_i = \ell, y_j = \ell'$. 

### Structured Representation of Marginal Gains

For diversity of label transitions, the marginal gain $\ell, y \in E\mid y_i = \ell, y_j = \ell'$ be the cut set for a specific label transition $(\ell, \ell')$. Further, let $\#\text{Cut}_S(\ell, \ell')$ count the number of items $a \in S$ that contain at least one $(\ell, \ell')$ label transition: $\#\text{Cut}_S(\ell, \ell') = |\{a \in S \mid \text{Cut}_{\phi(a)}(\ell, \ell') \neq \emptyset\}| = |S \cap G_{\ell,\ell'}|$. The marginal gains for this diversity function are:

$$ d(y \mid S) = D(\phi^{-1}(y) \mid S) = \sum_{\ell,\ell'} h(\#\text{Cut}_{S \cup \phi^{-1}(y)}(\ell, \ell')) - h(\#\text{Cut}_S(\ell, \ell')) $$

$$ = \sum_{\ell,\ell'} h(1 + \#\text{Cut}_S(\ell, \ell')) - h(\#\text{Cut}_S(\ell, \ell')) $$

Similar to single label groups, the gain for a label pair $(\ell, \ell')$ decreases as $\#\text{Cut}_S(\ell, \ell')$ grows. Thus, $d(y \mid S)$ rewards the presence of pair $(\ell, \ell')$ by an amount proportional to how rare it is in the segmentations in $S$. Analogously to the label costs, the parsimony factor in this setting is $p(y) = \sum_{\text{Cut}_y(\ell, \ell') \neq \emptyset} c(\ell, \ell')$ encouraging each individual $y$ to have a small number of label transitions. Specifically, when using a count coverage and parsimony term with $c(\ell, \ell') = -1$, we eventually maximize

$$ r(y) + p(y) + d(y) = r(y) - \sum_{\ell,\ell' \in G_{\ell,\ell'}} \min\{|\#\text{Cut}_y(\ell, \ell')|, 1\}, $$

which is a supermodular function on the set of cut edges for each label transition. Thus, the multi-label cooperative cut inference algorithm by Kohli et al. [20] applies. The construction for general $h$ looks similar, with a degrading cost in front of the min.

### Experiments

For the sake of completeness and to show the difference in sets of solutions generated by different diversity functions, we show sample sets of solutions generated for a given image (Fig. 5, 6 and 7). These results help in understanding the behavior of different diversity functions.
Figure 5: Sets of solutions generated with different diversity functions.
Figure 6: Sets of solutions generated with different diversity functions.

D Proof of Lemma 1

Lemma 3. Let $S$ be a sample of size $M$ taken uniformly at random. There exist monotone submodular functions where $E[F(S)] \leq (M/N + \epsilon/M) \max_{|S| \leq M} F(S)$ for any $\epsilon \geq 0.$
Figure 7: Sets of solutions generated with different diversity functions.

The bound in the main paper follows with $\epsilon = 0$. 

14
Proof. To prove this statement, we consider a specific worst-case function. Let $R \subseteq V$ be a fixed set of size $M$, and let

$$F(S) = |S \cap R| + \epsilon \min\{|S \setminus R|, 1\}. \quad (17)$$

The function $F$ is obviously nondecreasing and it is also submodular. For this function, the cardinality-constrained optimum is

$$\max_{|S| \leq M} F(S) = F(R) = M. \quad (18)$$

The expected value of an $M$-sized sample is the expectation of a hypergeometric distribution plus a correction taking into account that every set except $R$ will have value at least 1 (using the second part of $F$):

$$\mathbb{E}_S[F(S)] = \binom{N}{M}^{-1} \left( \sum_{S \subseteq V, |S| = M} |S \cap R| + \binom{N}{M} \epsilon - \epsilon \right) \quad (19)$$

$$= \binom{N}{M}^{-1} \left( \sum_{r=1}^{M} \binom{M}{r} \left( \binom{N-M}{M-r} r - \epsilon \right) + \epsilon \right) \quad (20)$$

$$= \frac{M^2}{N} + \epsilon - \binom{N}{M}^{-1} \epsilon \quad (21)$$

$$< (M/N + \epsilon/M)F(R). \quad (22)$$

This is also a fairly tight bound: if we sample each element with probability $M/N$, then, using Lemma 2.2 in [12] and the monotonicity of $F$, it holds that $\mathbb{E}_S[F(S)] \geq \frac{M}{M} F(V) \geq \frac{M}{N} F(S^*)$.

E Proof of Lemma 2

Lemma 4. Let $F \geq 0$ be monotone submodular. If each step of the greedy algorithm uses an approximate gain maximizer $b^{i+1}$ with $F(b^{i+1} | S^i) \geq \alpha \max_{a \in V} F(a | S^i) - \epsilon^{i+1}$, then

$$F(S^M) \geq (1 - \frac{1}{e^\alpha}) \max_{|S| \leq M} F(S) - \sum_{i=1}^{M} \epsilon^i. \quad (23)$$

To prove Lemma 2, we employ a helpful intermediate observation. We will denote the optimal solution by $S^* \in \arg \max_{|S| \leq M} F(S)$.

Lemma 5. If $F(b^{i+1} | S^i) \geq \alpha \max_{a \in V} F(a | S^i) - \epsilon^{i+1}$, then

$$F(b^{i+1} | S^i) \geq \frac{\alpha}{M} (F(S^*) - F(S^i)) - \epsilon^{i+1}. \quad (24)$$

Proof. (Lemma 5). Define $T^i = S^* \setminus S^i$ to be the set of all elements that are in $S^*$ but have not yet been selected. Order the elements in $T^i$ in an arbitrary order as $t^1, \ldots, t^h$ (note that $h \leq M$ because $|S^*| \leq M$. By monotonicity of $F$ and the fact that $S^* \subseteq S^i \cup T^i$, it holds that

$$F(S^*) - F(S^i) \leq F(T^i \cup S^i) - F(S^i) \quad (25)$$

$$= \sum_{j=1}^{h} F(t^j \mid S^i \cup \{t^1, \ldots, t^{j-1}\}) \quad (26)$$

$$\leq \sum_{j=1}^{h} F(t^j \mid S^i) \quad (27)$$

$$\leq M \max_{a \in V \setminus S^i} F(a \mid S^i) \quad (28)$$

$$\leq \frac{M}{\alpha} (F(b^{i+1} \mid S^i) + \epsilon^{i+1}) \quad (29)$$
In Equation (25), we use diminishing marginal returns. In the end, we use the assumption of the lemma, which implies that
\[
\max_a F(a \mid S) \leq \frac{1}{\alpha}(F(b_i^{i+1} \mid S^i) + \epsilon^{i+1}).
\] (28)

Rearranging yields the result of the lemma.

Now we are equipped to prove Lemma 4.

Proof. Lemma 5 implies that
\[
F(b_i^{i+1} \mid S^i) = F(S^{i+1}) - F(S^i) \geq \frac{\alpha}{M}(F(S^*) - F(S^0)) - \epsilon^{i+1}.
\] (29)

We rearrange this to
\[
F(S^*) - F(S^{i+1}) \leq (1 - \frac{\alpha}{M})F(S^*) - (1 - \frac{\alpha}{M})F(S^i) + \epsilon^{i+1}
\] (30)

\[
= (1 - \frac{\alpha}{M})(F(S^*) - F(S^i)) + \epsilon^{i+1}
\] (31)

\[
\leq (1 - \frac{\alpha}{M})^{i+1}(F(S^*) - F(S^0)) + \sum_{j=1}^{i} (1 - \frac{\alpha}{M})^{i-j} \epsilon^j
\] (32)

\[
\leq (1 - \frac{\alpha}{M})^{i+1}(F(S^*) - F(S^0)) + \sum_{j=1}^{i} \epsilon^j
\] (33)

With \( F(S^0) = F(\emptyset) = 0 \) we rearrange to
\[
F(S^M) \geq (1 - (1 - \frac{\alpha}{M})^{M})F(S^*) - \sum_{j=1}^{M} \epsilon^j
\] (34)

\[
\geq (1 - e^{-\alpha})F(S^*) - \sum_{j=1}^{M} \epsilon^j.
\] (35)

F Relative Error

If we have a monotone but not nonnegative function, we may shift the function and obtain a relative bound:

**Lemma 6.** Let \( F \) be an arbitrary monotone submodular function, and let \( F_{\min} = \min_{S \subseteq V} F(S) \).

We can define a new, shifted monotone non-negative version of \( F \) as \( F^+(S) = F(S) - F_{\min} \). If we apply the greedy algorithm to \( F^+ \), we obtain a solution \( \hat{S} \) that satisfies \( F^+(\hat{S}) \geq \alpha F^+(S^*) \), then the solution \( \hat{S} \) has a bounded relative approximation error:

\[
\frac{F(\hat{S}) - F_{\min}}{F(S^*) - F_{\min}} \geq \alpha.
\]

Proof. By a proof analogous to that of Lemma 2, we get, for \( \alpha = (1 - \frac{1}{e}) \),
\[
F^+(\hat{S}) \geq \alpha F^+(S^*) \quad \Leftrightarrow \quad F(\hat{S}) - F_{\min} \geq \alpha(F(S^*) - F_{\min})
\] (38)

\[
\Leftrightarrow \quad \frac{F(\hat{S}) - F_{\min}}{F(S^*) - F_{\min}} \geq \alpha = 1 - \frac{1}{e}.
\] (39)
G Generalization: Upper Envelope Potentials

In addition to the three specific examples given in the main document (Section 4.1, 4.1, 4.2), we can also generalize these constructions to a broad class of HOPs called upper envelope potentials [19]. Let $G_1, G_2, \ldots, G_q, \ldots, G_b$ be disjoint groups where $b$ is polynomial in the size of the number of base variables ($n$). We consider the group count diversity. At iteration $t$ in the greedy algorithm, assume without loss of generalization that we have covered $G_1, G_2, \ldots, G_k$. Then for the $(t + 1)^{th}$ iteration, the marginal gain of $y$ is:

$$d(y | S^t) = \begin{cases} 0 & \text{if } y \in \{G_1, \ldots, G_k\} \\ 1 & \text{otherwise.} \end{cases} \quad (41)$$

We can now express $d(y | S^t)$ as an upper-envelope potential, i.e.

$$\nu^q(y) = \mu^q + \sum_{i \in \mathbb{N}} \sum_{\ell \in L} w^q_i \delta_i(\ell) \quad (42)$$

where $\delta_i(\ell)$ returns 1 if $y_i = \ell$.

Each linear function $\nu^q(y)$ encodes two pieces of information:

1. $\mu^q$ encodes if $G_q$ is uncovered at the end of the $t^{th}$ step, i.e.

$$\mu^q = \begin{cases} 0 & \text{if } q \in \{1, \ldots, k\} \\ 1 & \text{otherwise.} \end{cases} \quad (43)$$

2. The second part of (42) indicates whether or not $y$ lies in the group $G_q$ (defined below).

For general groups, it may not be possible to have linear encodings for membership. By construction, we describe one practical example:

Region Consistency Diversity. Consider a region $C$ in the image whose superpixels $y_C$ we want to bias towards a homogenous or uniform labeling. Thus, when we search for diverse labelings, we want to encourage the entire set of superpixels $y_C$ to change together. In this case, each group $G_\ell$ corresponds to the set of segmentations that assign label $\ell$ to the region $y_C$. For such a diversity function, we define $w^q_i$ as:

$$w^q_i = \begin{cases} 0 & \text{if } i \in |C| \text{ and } \ell = q \\ -\infty & \text{otherwise.} \end{cases} \quad (44)$$

With this definition of $w^q_i$, the second part of (42) indicates whether or not $y$ lies in the group $G_q$.

It is known [30, 19] that maximizing any upper-envelope HOP can be reduced to the maximization of a pairwise function with the addition of an auxiliary switching variable $x$ that takes values from the index set of $q$ (in this case $|L|$).

$$\max_{y_e} d(y | S^t) = \max_{y_e, x} \phi_x(q) + \sum_{i \in V} \phi_{xi}(q, y_i) \quad (45)$$

where $\phi_x(q) = \mu^q$ and $\phi_{xi}(q, y_i) = w^q_i$. This pairwise function can be maximised using standard message passing algorithms such as TRW and BP. However, for some cases, such as the region consistency diversity defined above, the pairwise function is supermodular, and graph cuts can be used.