

# Supplementary Material

Paper Id - 593

## 1 Experiments

We present here further results for both synthetic and real data, which could not be accommodated in the main paper due to space constraints.

### 1.1 Synthetic Data

**Data.** We generate lattices of size  $100 \times 100$ , where each lattice point represents a variable taking one of 20 labels. The cliques are defined as all  $10 \times 10$  subwindows. The unary potentials are uniformly sampled in the range  $[1, 100]$ . The truncation factors for linear case are  $M \in \{5, 10, 15\}$  and for quadratic case  $M \in \{25, 100, 225\}$ . In each energy function, all cliques were assumed to have the same clique weight, being 5 and 10 for linear model, and 3 and 5 for quadratic model. We use  $m = 1$  for the sake of comparison with [?] and [?].

**Method.** For each energy function obtained by a particular setting of the above parameters, we vary the interval length up to  $M + 1$ . We run [?] and [?] only for linear distance function and  $m = 1$ . The experiments are repeated for 50 randomly generated unaries for linear and quadratic cases.

**Results.** The plots in Figure 2 and 1 show the average energy as a function of average time for our algorithm and the baselines using weight as 5 for max-of-linear models, with and without the results of [?] respectively. We see that the energy values of [?] are much higher as compared to our algorithm. Our algorithm gives lower energy solutions than parsimonious labeling method except when the truncation factor is low ( $M = 5$ ). The plots in Figures 4 and 3 shows the results for max-of-linear models using weight as 10 with and without the cooccurrence results respectively. Our algorithm gives better results than the baselines for all  $M$ . Both the baselines converge faster than our method. In practice, intervals smaller than the optimum size give almost equally good results and converge faster. Figure 5 shows the plot for max-of-quadratic cases using weight as 5.

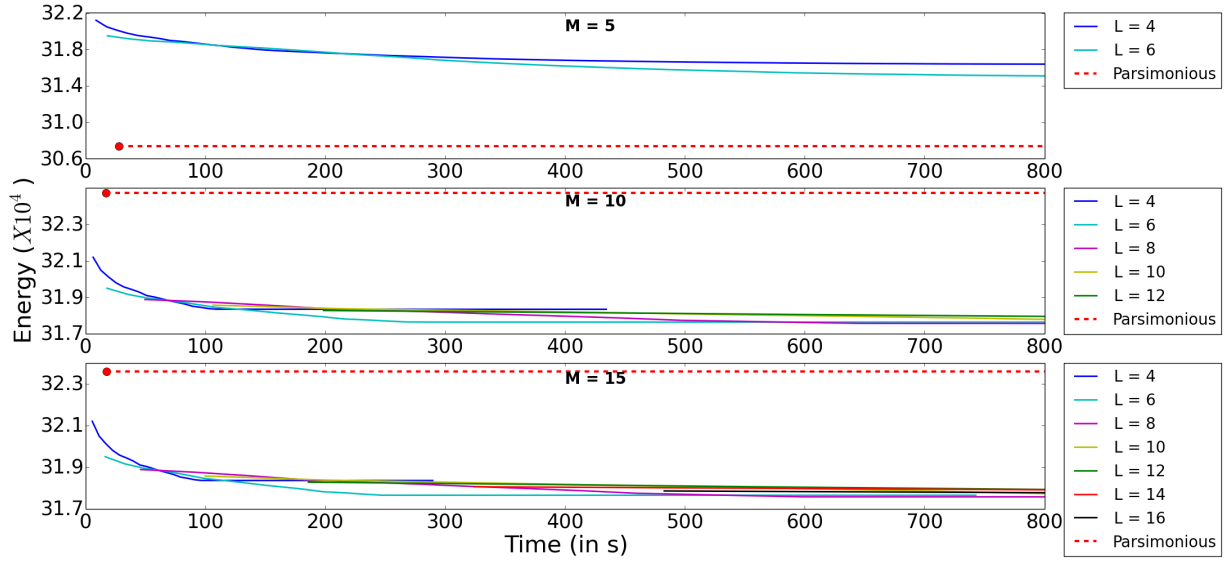


Figure 1: Results for synthetic data using truncated linear distance function. The plots show the variation of energy versus time, averaged over 50 lattices using  $\omega_c = 5$ . We use truncation factors as  $M = 5, 10$  and  $15$  and  $m = 1$ , and for each we vary interval lengths for our algorithm. This plot is the same as shown in the paper, but we include it here for the sake of comparison. Red dot indicates convergence of parsimonious labeling algorithm and dotted line indicates extrapolation.

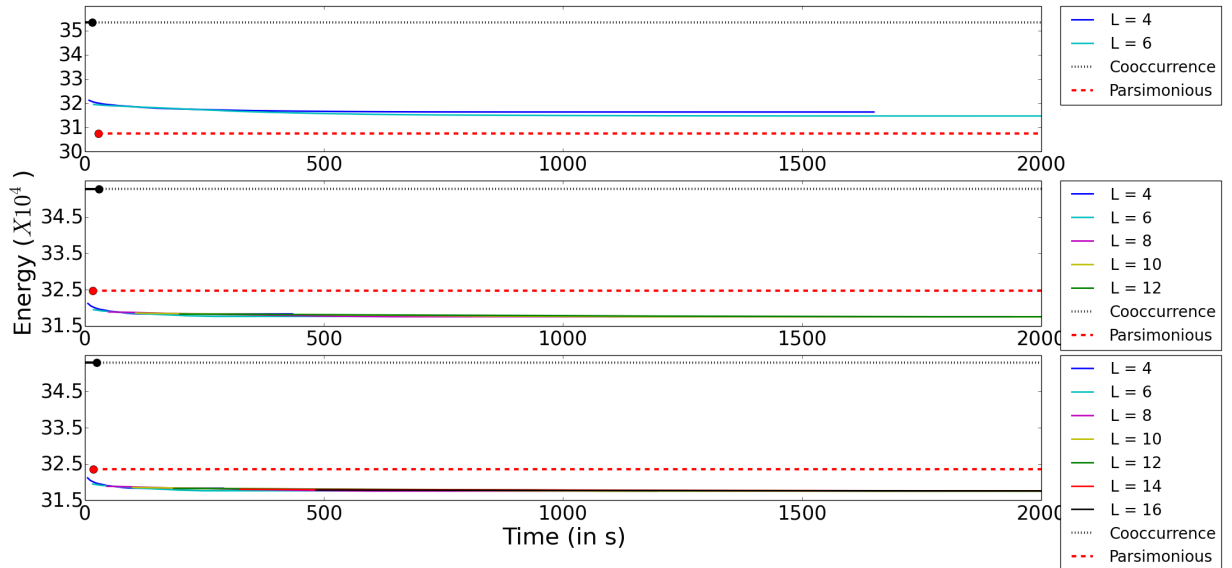


Figure 2: Results for synthetic data using truncated linear distance function. The plots show the variation of energy versus time, averaged over 50 lattices using  $\omega_c = 5$ . We use truncation factors as  $M = 5, 10$  and  $15$  and  $m = 1$ , and for each we vary interval lengths for our algorithm. This plot corresponds to the same experiment as mentioned in the paper, but with results for co-occurrence included. Red and black dots indicate convergence of respective algorithms and dotted line indicates extrapolation.

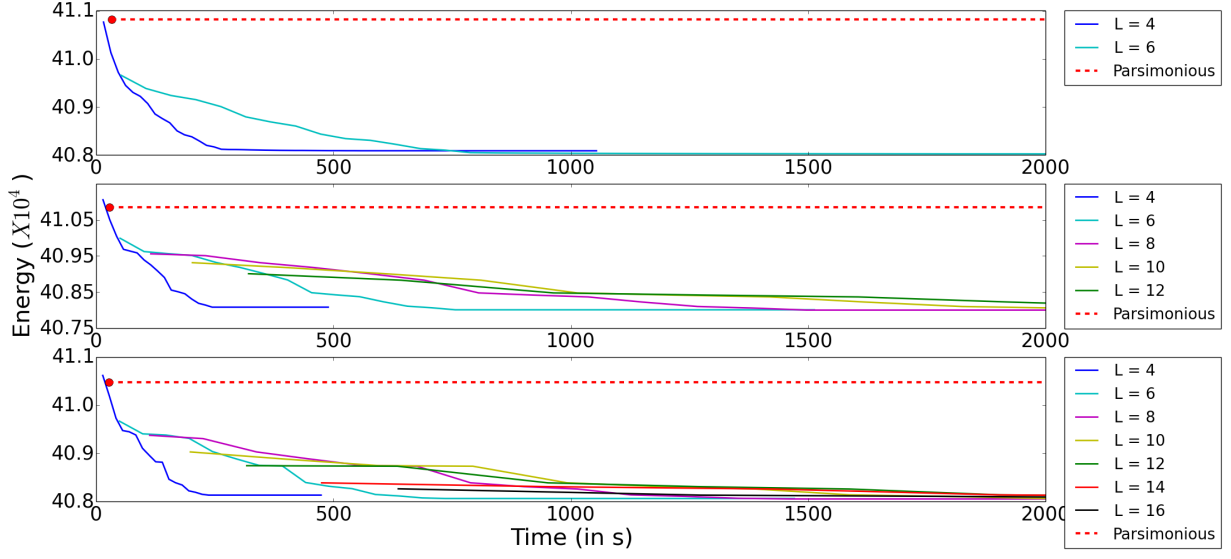


Figure 3: Results for synthetic data using truncated linear distance function. The plots show the variation of energy versus time, averaged over 50 lattices using  $\omega_c = 10$ . We use truncation factors as  $M = 5, 10$  and  $15$  and  $m = 1$ , and for each we vary interval lengths for our algorithm. Parsimonious labeling performs well for  $M = 5$ , but our approach outperforms for higher values of  $M$ . Red dot indicates convergence of parsimonious labeling and dotted line indicates extrapolation.

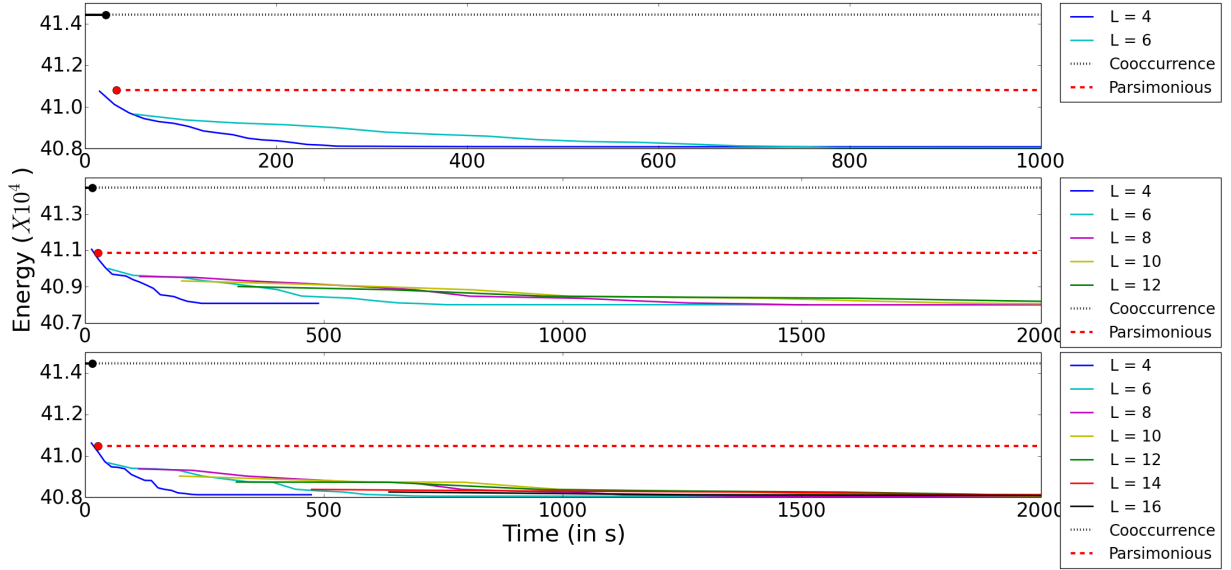


Figure 4: Results for synthetic data using truncated linear distance function. The plots show the variation of energy versus time, averaged over 50 lattices using  $\omega_c = 5$ . We use truncation factors as  $M = 5, 10$  and  $15$  and  $m = 1$ , and for each we vary interval lengths for our algorithm. This plot corresponds to the same experiment as in figure 3, but with results for co-occurrence included. Red and black dots indicate convergence of respective algorithms and dotted line indicates extrapolation.

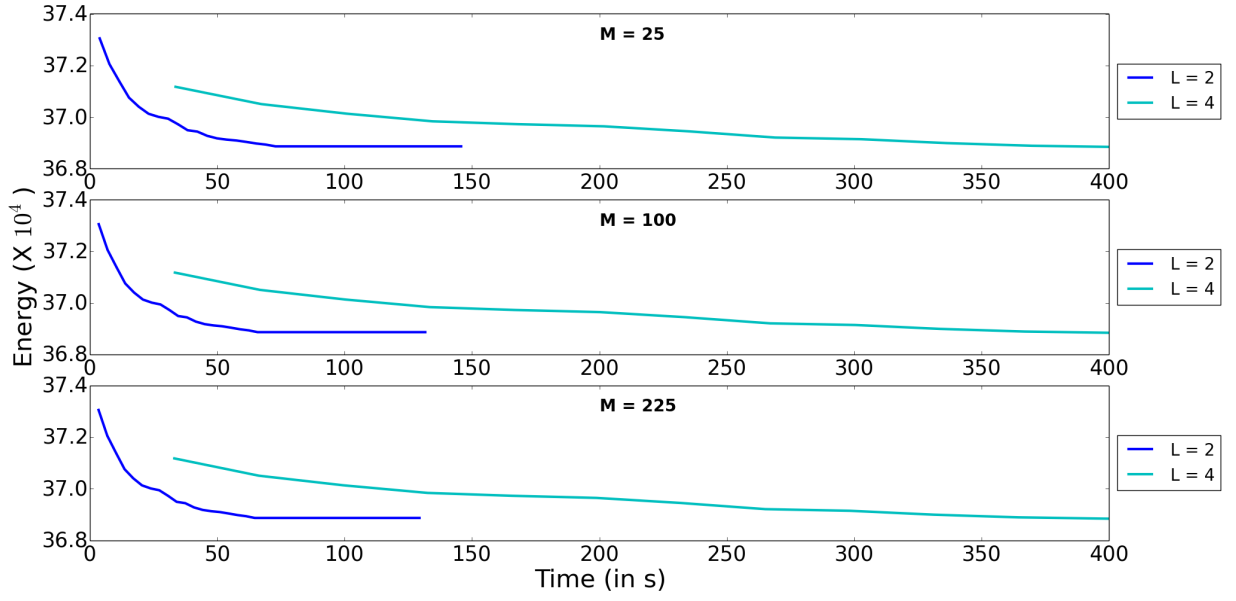


Figure 5: Results for synthetic data using truncated quadratic distance function. The plots show the variation of energy versus time, averaged over 50 lattices using  $\omega_c = 5$ . We use  $M = 25, 100$  and  $225$ , and for each we vary interval lengths for our algorithm.

## 1.2 Real Data

We present results on the two problems of image inpainting and denoising, and stereo matching. These enable us to visually infer the quality of our solutions. We use the energy of final labeling and convergence time as the evaluation criteria.

### 1.2.1 Image Inpainting and Denoising

**Data.** Given an image with noise and obscured/inpainted regions (regions with missing pixels), the task is to denoise it and fill the obscured regions in a way that is consistent with the surrounding regions. The images ‘house’ and ‘penguin’ from the Middlebury data set were used for the experiments. Since the images are grayscale, they have 256 labels in the interval  $[0, 255]$ , each representing an intensity value. The unary potential for each pixel corresponding to a particular label equals the squared difference between the label and the intensity value at the pixel. We use high-order cliques as the super-pixels obtained using the mean-shift method [?]. The parameters  $\omega_c$ ,  $M$  and  $m$  are varied to give different truncated max-of-linear energy functions.

**Method.** For each parameter setting of  $\omega_c$ ,  $M$  and  $m$ , we vary the interval lengths for our algorithm and make a comparison with the baselines.

**Results.** Results for  $\omega_c = 40$ ,  $M = 40$ , and  $m = 1, 3$  and  $5$  for ‘penguin’ and  $\omega_c = 50$ ,  $M = 50$ , and  $m = 1, 3$  and  $5$  for ‘house’ are shown in Figure 7 for varying interval lengths  $L = \{5, 10, 20\}$ . Note that we showed results for only  $L = 20$  in the main paper. Our algorithm consistently gives lower energy labeling as compared to both [?] and [?] even for small  $L$ . For ‘penguin’, our algorithm gives cleaner denoised image, preserving edges and details. On the other hand, both [?] and [?] exhibit significant blocking effect. Moreover, the output is more natural for  $m = 3$  as compared to  $m = 1$ . Even for ‘house’, our output looks more visually appealing as compared to baselines.

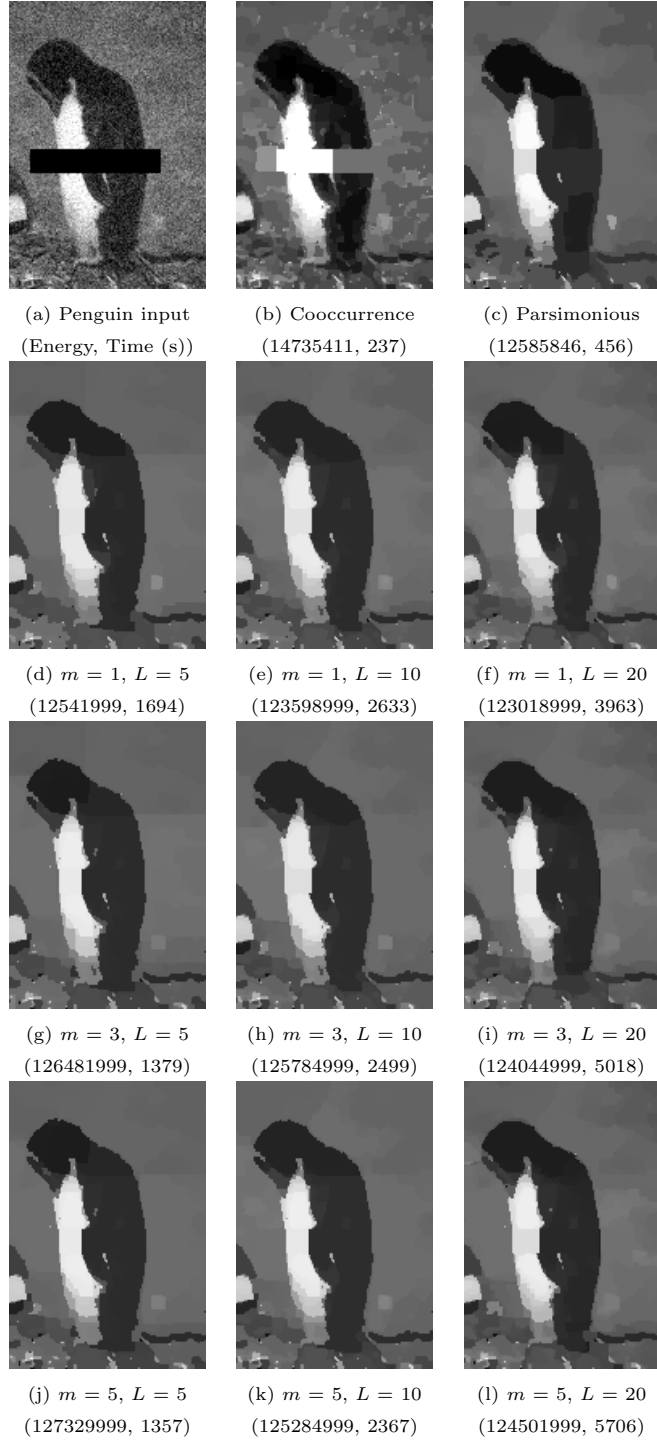


Figure 6: Image inpainting results for ‘penguin’. Note that comparison with (b) and (c) makes sense only for  $m = 1$ . Also, we restricted our experiments to smaller (and suboptimal)  $L$  due to computational issues.

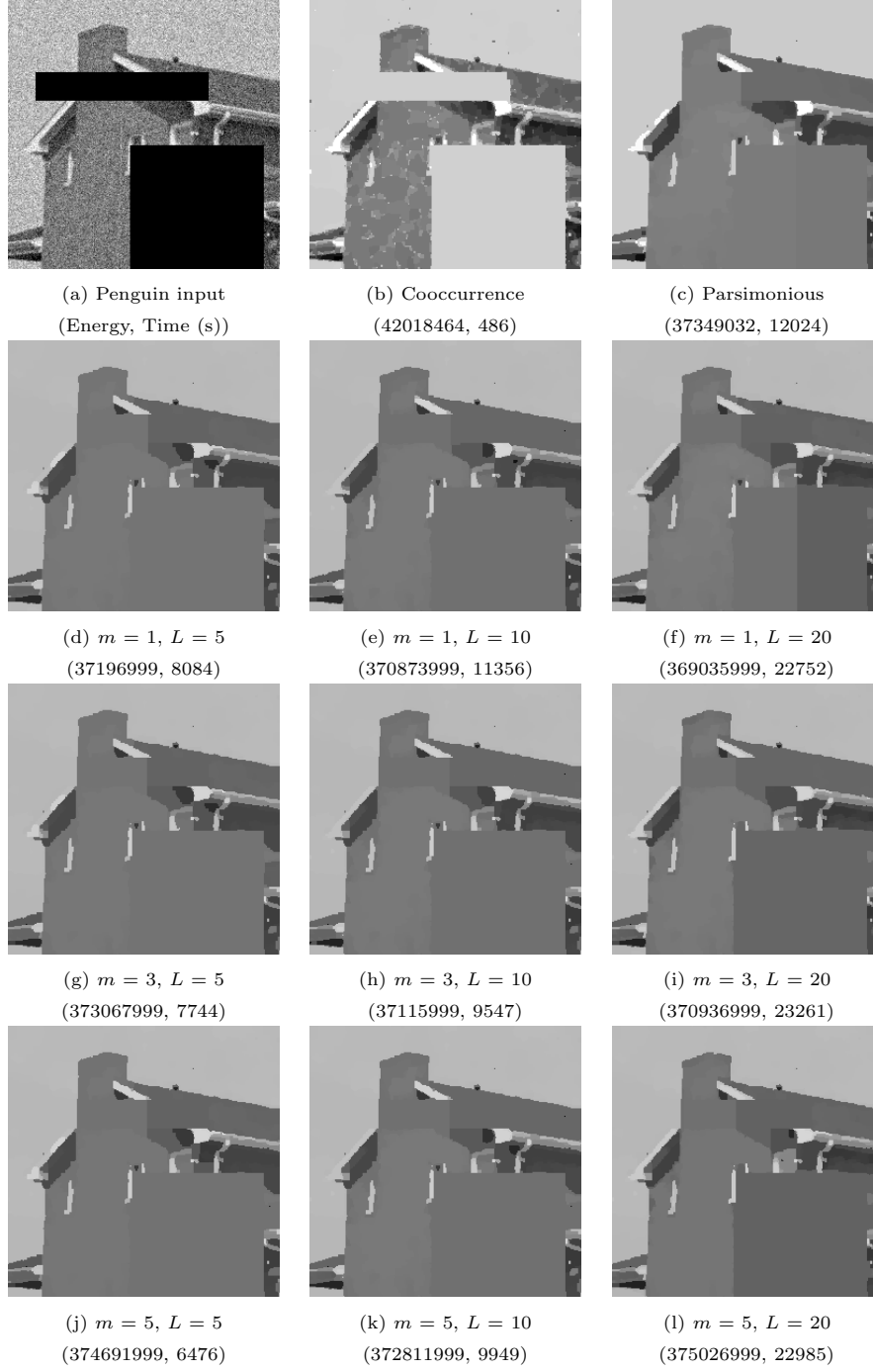


Figure 7: Image inpainting results for ‘house’. Note that comparison with (b) and (c) makes sense only for  $m = 1$ . Also, we restricted our experiments to smaller (and suboptimal)  $L$  due to computational issues.

### 1.2.2 Stereo Matching

**Data.** In the stereo matching problem, we have two rectified images of the same scene from two cameras set slightly apart. We are required to estimate the horizontal disparity between a pixel in the right camera image from the corresponding pixel in the left camera. We use ‘tsukuba’, ‘teddy’, ‘venus’ and ‘cones’ datasets from the Middlebury stereo collection for our experiments. In each case, we have a pair of RGB images and ground truth disparities. We assume the unary potentials to be the  $L1$ -norm of the difference in RGB values of the left and right image pixels. There are 16 labels and 60 labels for ‘tsukuba’ and ‘teddy’ respectively. The high-order cliques are super-pixels obtained using mean-shift method [?]. The parameters  $\omega_c$ ,  $M$  and  $m$  are varied to give different truncated max-of-linear energy functions.

**Method.** For each parameter setting of  $\omega_c$ ,  $M$  and  $m$ , we vary the interval lengths for our algorithm and make a comparison with the baselines.

**Results.** Results for  $\omega_c = 20$ ,  $M = 5$ , and  $m = 1$  and 3 for ‘tsukuba’, ‘venus’ and ‘cones’, and  $\omega_c = 20$ ,  $M = 1$ , and  $m = 1$  and 3 for ‘teddy’ are shown in Figure 8. We show the results for interval length 3 for ‘teddy’, and 6 for others. Note that in our main paper, we used interval length  $L$  as 4 for ‘tsukuba’ and ‘venus’, and 1 for ‘teddy’ and did not show the results for ‘cones’. Apart from ‘cones’, our algorithm consistently gives lower energy labeling as compared to both [?] and [?].



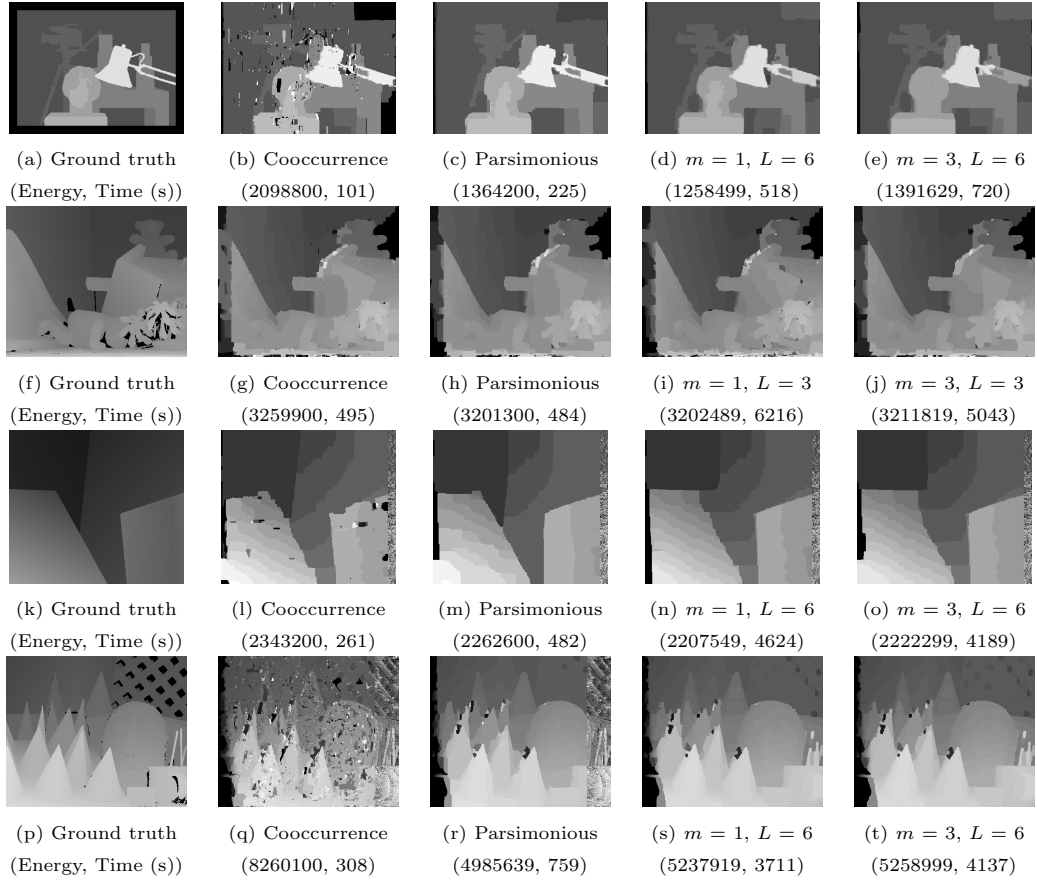


Figure 8: Stereo matching results: Figures (a), (f), (k) and (p) are the ground truth disparity for ‘tsukuba’, ‘teddy’, ‘venus’ and ‘cones’ respectively. Our results for  $m = 1$  (d), (i), (n) and (s) are significantly better than those of [?] (b), (g), (l) and (q), and of [?] (c), (h), (m) and (r) in terms of energy. We also show results for  $m = 3$ . We use super-pixels obtained using mean-shift as cliques. \*Note that  $m = 3$  uses a different energy function from other cases.

## 2 Advantages of using TMCM

Labeling	$m = 1$	$m = 2$	$m = 3$
(a) $\{1,1,1,1,2,2\}$	1	2	2
$\{1,2,3,4,5,6\}$	3	6	7
(b) $\{1,1,1,9,9,9\}$	3	6	9
$\{1,1,1,8,8,9\}$	3	6	9
(c) $\{1,1,1,1,1,7\}$	3	3	3
$\{1,1,1,2,3,4\}$	3	5	6

Table 1: *Clique potential value  $\theta_{\mathbf{c}}(\mathbf{x}_{\mathbf{c}})$  defined by a linear function with  $M = 3$  and  $\omega_{\mathbf{c}} = 1$  for various values of  $m$ . Since clique size is 6,  $0 \leq m \leq 3$ . Pair (a) demonstrates why taking the largest convex distances favors smoothness; (b) demonstrates how truncation prevents overpenalization of discontinuities; (c) demonstrates how using  $m > 1$  can provide some degree of robustness to errors in the definitions of the cliques.*

**Smoothness.** The truncated max-of-convex potentials encourage smooth labelings. In order to illustrate this, let us consider the example of a clique of six random variables  $\mathbf{X}_{\mathbf{c}}$  and a label set  $\mathbf{L}$  of size 10. We can define a truncated convex distance using a linear function  $d(y) = |y|$  and a truncation factor of  $M = 3$ . Consider pair (a) of labelings shown in table 1. If the labeling is  $\{1, 1, 1, 1, 2, 2\}$  and  $M = 3$ ,  $m = 3$ ,  $\omega_{\mathbf{c}} = 1$ ,  $\theta_{\mathbf{c}}(\mathbf{x}_{\mathbf{c}}) = \min(6-1, 3) + \min(5-2, 6) + \min(4-3, 3) = 3 + 3 + 1 = 7$ . Clearly, the first labeling of this pair is significantly smoother than the second, which is reflected in the value of the clique potential for all values of  $m$ . In contrast, if we were to consider the minimum distance among all pairs of labels, both the labelings will provide a clique potential value of 0.

**Discontinuities.** Similar to the pairwise case, the use of a truncation factor helps prevent overpenalization of discontinuities. For example, let us consider the pair (b) of labelings in table 1. In both cases, the six random variables appear to belong to two *groups*, one whose labels are low and one whose labels are high. Without a truncation, such a discontinuity would have been penalized heavily (for example, 8 for  $m = 1$  for both the labelings). This in turn would discourage the clique to be assigned this labeling even though this type of discontinuity is expected to occur in natural images. However, with a truncation, the penalty is significantly less (for example, 3 for  $m = 1$  for both the labelings), which can help preserve the discontinuities in the labeling obtained via energy minimization.

**Robustness.** In order to use a TMCM, we would be required to define the cliques. For example, given an image, we could use a bottom-up oversegmentation approach to obtain superpixels, and then set all the pixels in a superpixel to belong to a clique. However, oversegmentation can introduce errors since it has no notion of the specific vision application we are interested in modeling. To add robustness to errors in the clique definitions, we can set  $m > 1$ . For example, consider pair (c) of labelings in table 1. The first of these labelings contains a single random variable with a very high label, which could be due to the fact that the corresponding pixel has been incorrectly grouped in this superpixel. As can be seen from the values of the potential, the presence of such an erroneous pixel in the superpixel is not heavily penalized when we use  $m > 1$ . For example, when  $m = 3$  the value of the clique potential for the first labeling (with an erroneous pixel) is the same as the second labeling (which is a fairly smooth labeling).

$n$	Number of random variables
$h$	Number of labels
$\mathcal{V}$	$\{1, 2, \dots, n\}$
$\mathbf{L}$	$\{1, 2, \dots, h\}$
$\mathbf{X}$	Set of random variables $\{X_a, a \in \mathcal{V}\}$
$\mathbf{X}_{\mathbf{c}}$	Set of random variable belonging to a clique $\{X_a, a \in \mathbf{c} \subseteq \mathcal{V}\}$
$\mathbf{x}_{\mathbf{c}}$	Labeling of clique $\mathbf{c}$
$\mathbf{p}(\mathbf{x}_{\mathbf{c}})$	Sorted list of the labels present in $\mathbf{x}_{\mathbf{c}}$
$I_n$	Interval of consecutive labels $[i_n + 1, j_n]$
$L$	Length of interval, that is, $L = j_n - i_n$
$\Gamma_r$	Set of intervals $\{[0, r], [r + 1, r + L], \dots, [., h - 1]\}$
$f$	Labeling of the random field ( $v_a$ takes the label $l_{f(a)}$ )
$f^*$	An optimal (MAP) labeling of the random field
$\theta_a(i)$	Unary potential of assigning label $l_i$ to $v_a$
$\omega_{\mathbf{c}}$	Weight for clique $\mathbf{X}_{\mathbf{c}}$
$c$	Size of clique $\mathbf{X}_{\mathbf{c}}$
$d(\cdot)$	Convex function used to define distance between two labels
$M$	Truncation factor
$\theta_{\mathbf{c}}(\mathbf{x}_{\mathbf{c}})$	Clique potential of assigning label $l_i$ to $v_a$
$E(f)$	Energy of the labeling $f$
$\mathcal{A}(f, I_n)$	$\{\mathbf{X}_{\mathbf{c}} \in \mathcal{C}, f(v) \in I_n \forall v \in \mathbf{X}_{\mathbf{c}}\}$
$\mathcal{B}(f, I_n)$	$\{\mathbf{X}_{\mathbf{c}} \in \mathcal{C}, \exists X_v, X_w \in \mathbf{X}_{\mathbf{c}}   f(v) \in I_n \wedge f(w) \notin I_n \}$

Table 2: Definitions of various symbols used in the supplementary document

### 3 Optimization via Range Expansion

As max of truncated convex models are a generalization of truncated convex models, it follows that the corresponding energy minimization problem is NP-hard. If the range expansion algorithm is extended to TCMC, it results in an NP-hard subproblem (specifically problem 1 in algorithm 1). To overcome this problem, we first overestimate the energy function  $E$  by a submodular function  $E'$  which is described in the following subsections. Then we construct a graph which represents  $E'$  up to an additive constant. This enables us to extend the range expansion algorithm to our more general class of energy functions.

#### 3.1 Range Expansion Algorithm for TCMC

---

**Algorithm 1** The range expansion algorithm for TCMC.

---

**input** Energy function  $E(\cdot)$ , initial labeling  $\mathbf{x}^0$ , interval length  $L$ .

1: Initialize the output labeling  $\hat{\mathbf{x}} = \mathbf{x}^0$ .

2: **repeat**

3:   **for all**  $i_m \in [-L + 2, h]$  **do**

4:     Define an interval of labels  $\mathbf{I} = \{f, \dots, l\}$  where  $f = \max\{i_m, 1\}$  and  $l = \min\{i_m + L - 1, h\}$ .

5:     Obtain a new labeling  $\mathbf{x}'$  by solving the following optimization problem:

$$\begin{aligned} \mathbf{x}' = \underset{\mathbf{x}}{\operatorname{argmin}} \quad & E(\mathbf{x}), \\ \text{s.t. } & x_a \in \mathbf{I} \cup \{\hat{x}_a\}, \forall a \in \mathcal{V}. \end{aligned} \tag{1}$$

6:     **if**  $E(\hat{\mathbf{x}}) > E(\mathbf{x}')$  **then**

7:       Update  $\hat{\mathbf{x}} = \mathbf{x}'$ .

8:     **end if**

9:   **end for**

10: **until** The labeling does not change for any value of  $i_m$ .

**output** The labeling  $\hat{\mathbf{x}}$ .

---

Algorithm 1 shows the main steps of the range expansion algorithm. The algorithm starts by assigning the random variables to an initial label (step 1). For example, all the random variables could be assigned to the label 1. Next, it selects an interval of consecutive labels of size at most  $L$  (steps 3-4) where  $L$  is specified as an input to the algorithm. We will see later in the section that the value of  $L$  can be chosen to obtain the optimal worst case bound for specific instances of the max of truncated convex models. Next, it obtains a new labeling that minimizes the energy over all the labelings that either allow a random variable to retain its current label, or choose a new label in the selected interval (step 5). If the energy of the new labeling is lower than that of the current labeling, then the solution is updated as the new labeling (steps 6-8). This process is repeated for all the intervals of consecutive labels of size at most  $L$ . The entire algorithm stops when the energy cannot be reduced further for any choice of the interval.

The crux of the range expansion algorithm is problem (1), which needs to be solved for any given interval  $\mathbf{I}$  and current labeling  $\hat{\mathbf{x}}$ . Unfortunately, this problem may itself be NP-hard. Indeed, when  $L = h$ , problem (1) is equivalent to the original energy minimization problem. In order to operationalize the range expansion algorithm, we need to devise an approximate algorithm for problem (1). We achieve this in two steps that we briefly outline below, and describe in detail in the following two subsections. First, we obtain an overestimate of the energy function  $E(\cdot)$ , which we denote by  $E'(\cdot)$ . The energy function  $E'(\cdot)$  is restricted to the labels in the interval  $\mathbf{I}$  together with labels specified by the current labeling  $\hat{\mathbf{x}}$ . Second, we minimize the

overestimated energy  $E'(\cdot)$  over all of its putative labelings by solving an equivalent *st*-MINCUT problem. We describe our two-step algorithm in the next two subsections in detail. Specifically, subsection 3.2 describes the exact form of the energy function  $E'(\cdot)$ , while subsection ?? describes the construction of the directed graph over which we solve the *st*-MINCUT problem to obtain the labeling  $\mathbf{x}'$ .

### 3.2 Overestimation of the Energy Function

Given an interval  $\mathbf{I} = \{f, \dots, l\}$  of consecutive labels, and the current labeling  $\hat{\mathbf{x}}$ , we define the new energy function  $E'(\cdot)$  over the set of random variables  $\mathbf{X}$ . Unlike the original energy function, the label set corresponding to  $E'(\cdot)$  is equal to  $\mathbf{L}' = \{0, 1, \dots, h'\}$ , where  $h' = l - f + 1$ . Roughly speaking, the label 0 in the set  $\mathbf{L}'$  corresponds to a random variable retaining its current label, while any other label  $i \geq 1$  corresponds to a random variable taking the label  $f + i - 1 \in \mathbf{I}$ . A labeling of the energy function  $E'(\cdot)$  is denoted by  $\mathbf{y} \in (\mathbf{L}')^n$  in order to distinguish it from the labeling corresponding to the original energy function. In order to fully specify the new energy function  $E'(\cdot)$ , we need to define the unary potentials and the clique potentials, which we do as follows.

**Unary Potentials.** The unary potential of a random variable  $X_a$  (where  $a \in \mathcal{V}$ ) being assigned a label  $y_a \in \mathbf{L}'$  is given by the following equation:

$$\theta'_a(y_a) = \begin{cases} \theta_a(\hat{x}_a) + \kappa_a & \text{if } y_a = 0 \\ \theta_a(f + y_a) + \kappa_a & \text{otherwise.} \end{cases} \quad (2)$$

In other words, if  $y_a = 0$  then the unary potential corresponds to the random variable  $X_a$  retaining its current label  $\hat{x}_a$ , and if  $y_a \neq 0$  then the unary potential corresponds to the random variable  $X_a$  being assigned the label  $f + y_a \in \mathbf{I}$ . The constant  $\kappa_a$  is added to the unary potentials to ensure that they are non-negative, which makes the description of the graph construction in the next subsection simpler.

**Clique Potentials.** In order to describe the high-order clique potentials of the new energy function we require a function  $\delta_{a,b} : \mathbf{L}' \times \mathbf{L}' \rightarrow \mathbb{R}$  for each  $(a, b) \in \mathcal{E}$ , which is defined as follows:

$$\delta_{a,b}(y_a, y_b) = \begin{cases} \min\{d(\hat{x}_a - \hat{x}_b), M\} & \text{if } y_a = y_b = 0, \\ M + d(y_b - 1) & \text{if } y_a = 0, y_b \neq 0, \\ M + d(y_a - 1) & \text{if } y_a \neq 0, y_b = 0, \\ d(y_a - y_b) & \text{if } y_a \neq 0, y_b \neq 0. \end{cases} \quad (3)$$

Here,  $d(\cdot)$  is the convex function and  $M$  is the truncation factor associated with the original energy function  $E(\cdot)$ . A reader familiar with this field may recognize that the above function is *submodular*. It is also easy to verify that  $\delta_{a,b}(y_a, y_b)$  is an overestimate of the truncated convex distance between the labels  $x_a \in \mathbf{I} \cup \{\hat{x}_a\}$  and  $x_b \in \mathbf{I} \cup \{\hat{x}_b\}$  corresponding to the labels  $y_a$  and  $y_b$  respectively.

Given a labeling  $\mathbf{y}_{\mathbf{c}} \in \mathbf{L}'^c$  of a clique  $\mathbf{c}$  of size  $c$ , we denote a sorted list of the labels in  $\mathbf{y}_{\mathbf{c}}$  as  $\mathbf{p}(\mathbf{y}_{\mathbf{c}})$ . Furthermore, we denote the indices of the sorted list as  $\mathbf{q}(\mathbf{y}_{\mathbf{c}})$ . In other words, the random variable corresponding to the  $i$ -th smallest label (that is, the  $i$ -th element of the list  $\mathbf{p}(\mathbf{y}_{\mathbf{c}})$ , which is denoted by  $p_i(\mathbf{y}_{\mathbf{c}})$ ) is given by  $X_a$  where  $a = q_i(\mathbf{y}_{\mathbf{c}})$ . In order to avoid clutter in our notation, we will drop the argument  $\mathbf{y}_{\mathbf{c}}$  from  $\mathbf{p}$  and  $\mathbf{q}$  whenever it is clear from context.

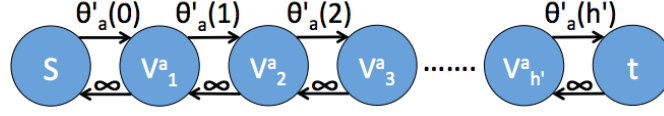


Figure 9: Arcs and their capacities for representing the unary potentials for the random variable  $X_a$ . According to the labeling defined in equation (??), if  $x_a = \hat{x}_a$ , then the arc  $(s, V_1^a)$  will be cut, which will contribute exactly  $\theta_a(\hat{x}_a)$  to the capacity of the cut. If  $x_a = s + i - 1$  where  $i \in \{1, \dots, h' - 1\}$ , then the arc  $(V_i^a, V_{i+1}^a)$  will be cut, which will contribute exactly  $\theta_a(s + i - 1)$  to the capacity of the cut. If  $x_a = l$ , then the arc  $(V_{h'}^a, t)$  will be cut, which will contribute exactly  $\theta_a(l)$  to the capacity of the cut. The arcs with infinite capacity ensure that exactly one of the arcs from the set  $(s, V_1^a) \cup \{(V_i^a, V_{i+1}^a), i = 1, \dots, h' - 1\} \cup (V_{h'}^a, t)$  will be part of an  $st$ -cut with finite capacity, which will guarantee that we are able to obtain a valid labeling.

Using the above definitions, the high-order clique potential for the new energy  $E'(\cdot)$  can be concisely specified as

$$\theta'_c(\mathbf{y}_c) = \omega_c \sum_{i=1}^m \delta_{q_i, q_{c-i+1}}(p_i, p_{c-i+1}). \quad (4)$$

In other words, the clique potentials corresponding to the energy function  $E'(\cdot)$  are the sum of the  $m$  maximum submodular functions over disjoint pairs of random variables in the clique.

### 3.3 Graph Construction

#### 3.3.1 Description

Our problem is to minimize the energy function  $E(\cdot)$  over all possible labelings that allow each random variable  $X_a$  to either retain its current label  $\hat{x}_a$  or choose a label from the interval  $\mathbf{I} = \{s, \dots, l\}$ . To this end, we convert it into an equivalent  $st$ -MINCUT problem over a directed graph, which can be solved efficiently if all arc capacities are non-negative [?].

We construct a directed graph over the set of vertices  $\{s, t\} \cup \mathbf{V} \cup \mathbf{U} \cup \mathbf{W}$ . The set of vertices  $\mathbf{V}$  model the random variables  $\mathbf{X}$ . Specifically, for each random variable  $X_a$  we define  $h' = l - s + 1$  vertices  $V_i^a$  where  $i \in \{1, \dots, h'\}$ . The sets  $\mathbf{U}$  and  $\mathbf{W}$  represent *auxiliary* vertices, whose role in the graph construction will be explained later when we consider representing the high-order clique potentials. We also define a set of arcs over the vertices, where each arc has a non-negative capacity. We would like to assign arc capacities such that the  $st$ -cuts of the directed graph satisfy two properties. First, all the  $st$ -cuts with a finite capacity should include exactly one arc from the set  $(s, V_1^a) \cup \{(V_i^a, V_{i+1}^a), i = 1, \dots, h' - 1\} \cup (V_{h'}^a, t)$  for each random variable  $X_a$ . This property would allow us to define a labeling  $\mathbf{x}$  such that

$$x_a = \begin{cases} \hat{x}_a & \text{if the cut includes the arc } (s, V_1^a) \\ s + i - 1 & \text{if the cut includes the arc } (V_i^a, V_{i+1}^a) \\ l & \text{if the cut includes the arc } (V_{h'}^a, t). \end{cases} \quad (5)$$

Second, we would like the energy of the labeling  $\mathbf{x}$  defined above to be as close as possible to the capacity of the  $st$ -cut. This will allow us to obtain an accurate approximate solution  $\mathbf{x}'$  for problem (1) by finding the  $st$ -MINCUT. We now specify the arcs and their capacities such that they satisfy the above two properties. We consider two cases: (i) arcs that represent the unary potentials; and (ii) arcs that represent the high-order clique potentials.

**Representing Unary Potentials.** We will represent the unary potential of  $X_a$  using the arcs specified in Figure ?? . Since all the unary potentials are non-negative, it follows that the arc capacities in Figure ?? are also non-negative.

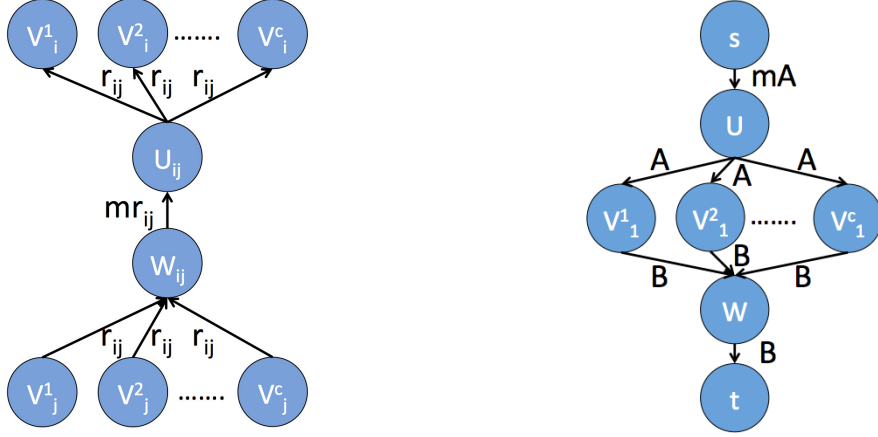


Figure 10: Arcs used to represent the high-order potentials for the clique  $\mathbf{X}_c = \{X_1, X_2, \dots, X_c\}$ . **Left.** The term  $r_{ij}$  is defined in equation (??). The arcs represent the sum of the  $m$  maximum convex distance functions over disjoint pairs of random variables when no random variable retains its old label. These arcs are specified only for  $i \leq j$  and when either one or both of  $i$  and  $j$  are not equal to 1. **Right.** The terms  $A$  and  $B$  are defined in equation (??). The arcs represent an overestimation of the clique potential for the case where some or all the random variables retain their old label.

**Representing Clique Potentials.** Consider a set of random variables  $\mathbf{X}_c$  that are used to define a high-order clique potential. Without loss of generality, we assume  $\mathbf{X}_c = \{X_1, X_2, \dots, X_c\}$ . In order to represent the potential value for a putative labeling  $\mathbf{x}_c$  of the clique, we introduce two types of arcs, which are depicted in Figure ?? . For the arcs shown in Figure ?? (left), the capacities are specified using the term  $r_{ij}$  that is defined as follows:

$$r_{ij} = \begin{cases} \omega_c \frac{\bar{d}(i,j)}{2} & \text{if } i = j \\ \omega_c \bar{d}(i,j) & \text{otherwise.} \end{cases} \quad (6)$$

Here, the term  $\bar{d}(i,j) = d(i-j+1) + d(i-j-1) - 2d(i-j) \geq 0$  since  $d(\cdot)$  is convex, and  $\omega_c \geq 0$  by definition. It follows that  $r_{ij} \geq 0$  for all  $i, j \in \{1, \dots, h'\}$ . For the arcs shown in Figure ?? (right), the capacities are specified using the terms  $A$  and  $B$  that are defined as follows:

$$A = \omega_c M, B = \left( \omega_c M - \frac{\theta_c(\hat{\mathbf{x}}_c)}{m} \right). \quad (7)$$

Since  $M \geq 0$ , and  $\theta_c(\hat{\mathbf{x}}_c) \leq \omega_c m M$  due to truncation, it follows that  $A, B \geq 0$ .

### 3.3.2 Properties of the Graph

This part describes the properties of the above graph construction which will facilitate the analysis of our algorithm for the truncated max-of-linear models.

**Property 1.** *The cost of the  $st$ -cut exactly represents the sum of the unary potentials associated with the corresponding labeling  $f$ , that is,  $\sum_{v_a \in \mathbf{v}} \theta_a(f(a))$ .*

This property follows from the description of the graph construction for unary potentials.

**Property 2.** *For  $c \in \mathcal{C}$ , if for all  $X_a \in \mathbf{X}_c$ ,  $f(a) = f_n(a) \notin I_n$  (all variables in a clique retain their old labels), then the cost of the  $st$ -cut exactly represents the clique potential plus a constant  $\kappa_c = \omega_c m \cdot M - \theta_c(\hat{\mathbf{x}}_c)$ .*

Graph (a) of figure ?? assigns 0 cost. In graph (b), the vertices  $\{V_1^1, \dots, V_1^c\} \in \mathbf{V}_t$  and hence, the  $s - U$  arc of capacity  $mA$  is cut.

$$\begin{aligned} mA &= \omega_c \cdot m \cdot M \\ &= \theta_c(\hat{\mathbf{x}}_c) + \omega_c m \cdot M - \theta_c(\hat{\mathbf{x}}_c) \\ &= \theta_c(\hat{\mathbf{x}}_c) + \kappa_c \end{aligned}$$

**Property 3.** *If all variables in a clique move to a label in the current interval and  $|l_c - l_1| \leq M$ , then the cost of the  $st$ -cut exactly represents the clique potential plus a constant  $\kappa_c = \omega_c m \cdot M - \theta_c(\hat{\mathbf{x}}_c)$ .*

As proved in lemma ??, graph (a) of figure ?? assigns exactly  $\theta_c(\mathbf{x}_c)$  cost. In graph (b), the vertices  $\{V_1^1, \dots, V_1^c\} \in \mathbf{V}_s$  and hence, the  $W - t$  arc of capacity  $B$  is cut, where

$$B = \omega_c \cdot m \cdot M - \theta_c(\hat{\mathbf{x}}_c) = \kappa_c$$

**Property 4.** *If all variables in a clique move to a label in the current interval and  $|l_c - l_1| \geq M$ , then the cost of the  $st$ -cut overestimates the clique potential, being*

$$\omega_c \sum_{i=1}^m d(p_i(\mathbf{x}_c) - p_{c-i+1}(\mathbf{x}_c))$$

*plus a constant  $\kappa_c = \omega_c m \cdot M - \theta_c(\hat{\mathbf{x}}_c)$ .*

**Property 5.** *If  $k(< c)$  variables in a clique retains their old labels, then the cost of the  $st$ -cut overestimates the clique potential, being.*

$$\omega_c \sum_{i=1}^k d(p_{c-i+1}(\mathbf{x}_c) - i_n - 1) + \omega_c \cdot m \cdot M$$

*plus a constant  $\kappa_c = \omega_c m \cdot M - \theta_c(\hat{\mathbf{x}}_c)$ .*

### 3.3.3 Proofs

We first prove the following lemma, which will help us to establish properties of the graph:

**Lemma 1.** *Graph (a) of figure ?? exactly models clique potentials that are proportional to the sum of  $m$  maximum convex distance functions over all disjoint pairs of random variables of the clique.*

Let us arrange the labels assigned to  $\mathbf{X}_c$  in increasing order as  $\{l_{k_1}, \dots, l_{k_c}\}$ , where  $c = |\mathbf{X}_c|$  indicates the size of the clique. The sum of  $m$  maximum convex distance functions over all disjoint pairs of random variables of the clique is given by:

$$\omega_c(d(k_c - k_1) + d(k_{c-1} - k_2) + \dots + d(k_{c-m+1} - k_m)) \quad (8)$$

When  $i \in [k_p + 1, k_{p+1}]$ , exactly  $p$  variables in  $\mathbf{x}_c$  are assigned a label between  $[l_{k_1}, l_{k_p}]$  and the corresponding  $p$  vertices among  $\{V_i^1, V_i^2, \dots, V_i^c\}$  will be in  $\mathbf{V}_t$  (the remaining  $c - p$  vertices will be in  $\mathbf{V}_s$ ). Similarly, when  $j \in [k_{c-p} + 1, k_{c-p+1}]$ , exactly  $p$  variables in  $\mathbf{x}_c$  are assigned a label in the range  $[l_{k_{c-p+1}}, l_{k_c}]$  and the corresponding  $p$  vertices among  $\{V_j^1, V_j^2, \dots, V_j^c\}$  will be in  $\mathbf{V}_s$  (the remaining  $c - p$  vertices will be in  $\mathbf{V}_t$ ).  $st$ -MINCUT includes either the arcs  $(U_{ij}, V_i^a)$  for



all  $V_i^a \in \mathbf{V}_t$  or the arcs  $(V_j^a, W_{ij})$  for all  $V_j^a \in \mathbf{V}_s$  or the arc  $(W_{ij}, U_{ij})$ , whichever contribute the smallest cost to the cut. Let  $C_{ij}$  denote the sum of capacities of arcs cut by *st*-MINCUT in graph (a) of figure ?? for a given pair  $i, j$ . Clearly,  $0 \leq C_{ij} \leq m \cdot r_{ij}$ .

Let us partition the interval  $[k_1 + 1, k_c]$  to  $2m - 1$  intervals  $[k_1 + 1, k_2], [k_2 + 1, k_3], \dots, [k_m + 1, k_{c-m+1}], \dots, [k_{c-2} + 1, k_{c-1}], [k_{c-1} + 1, k_c]$ .  $C_{ij}$  depends on the intervals in which  $i$  and  $j$  lie. Given a labeling  $\mathbf{x}_c$ , the total contribution of all the cut arcs is equal to

$$\begin{aligned} \sum_{i=k_1+1}^{k_c} \sum_{j=i}^{k_c} C_{ij} &= \sum_{i=k_1+1}^{k_2} \sum_{j=i}^{k_c} C_{ij} + \sum_{i=k_2+1}^{k_3} \sum_{j=i}^{k_c} C_{ij} \\ &+ \dots + \sum_{i=k_m+1}^{k_{c-m+1}} \sum_{j=i}^{k_c} C_{ij} + \dots + \sum_{i=k_{c-1}+1}^{k_c} \sum_{j=i}^{k_c} C_{ij} \end{aligned} \quad (9)$$

Consider the second term of the series in (??). When  $i \in [k_2 + 1, k_3]$ , exactly 2 variables in  $\mathbf{X}_c$  take up a label in the range  $[k_1, k_2]$  and hence, two vertices among  $\{V_i^1, V_i^2, \dots, V_i^c\}$  belong to  $\mathbf{V}_t$ . When  $j \in [k_{c-1} + 1, k_c]$ , only one vertex among  $\{V_j^1, V_j^2, \dots, V_j^c\}$  belongs to  $\mathbf{V}_s$ , giving  $C_{ij}$  as  $r_{ij}$ . For  $j$  in all other intervals, at least two vertices among  $\{V_j^1, V_j^2, \dots, V_j^c\}$  belong to  $\mathbf{V}_s$ , and  $C_{ij}$  equals  $2 \cdot r_{ij}$ . Thus, the following holds

$$\begin{aligned} \sum_{i=k_2+1}^{k_3} \sum_{j=i}^{k_c} C_{ij} &= \sum_{i=k_2+1}^{k_3} \sum_{j=k_{c-1}+1}^{k_c} r_{ij} + \sum_{i=k_2+1}^{k_3} \sum_{j=i}^{k_{c-1}} 2r_{ij} \\ &= \sum_{i=k_2+1}^{k_3} \sum_{j=k_{c-1}+1}^{k_c} r_{ij} + \left( \sum_{i=k_2+1}^{k_3} \sum_{j=i}^{k_{c-1}} r_{ij} + \sum_{i=k_2+1}^{k_3} \sum_{j=i}^{k_{c-1}} r_{ij} \right) \\ &= \sum_{i=k_2+1}^{k_3} \sum_{j=i}^{k_c} r_{ij} + \sum_{i=k_2+1}^{k_3} \sum_{j=i}^{k_{c-1}} r_{ij} \end{aligned}$$

where the last step is obtained by combining the first and the third terms in the summation. In general, for  $p \in [1, m - 1]$  we can write

$$\sum_{i=k_p+1}^{k_{p+1}} \sum_{j=i}^{k_c} C_{ij} = \sum_{i=k_p+1}^{k_{p+1}} \sum_{j=i}^{k_c} r_{ij} + \sum_{i=k_p+1}^{k_{p+1}} \sum_{j=i}^{k_{c-1}} r_{ij} + \dots + \sum_{i=k_p+1}^{k_{p+1}} \sum_{j=i}^{k_{c-p+1}} r_{ij} \quad (10)$$

Similar argument can be extended for  $p \in [c - m + 1, c - 1]$

Using (??), each term of (??) can be written as

$$\begin{aligned}
\sum_{i=k_1+1}^{k_2} \sum_{j=i}^{k_c} C_{ij} &= \sum_{i=k_1+1}^{k_2} \sum_{j=i}^{k_c} r_{ij} \\
\sum_{i=k_2+1}^{k_3} \sum_{j=i}^{k_c} C_{ij} &= \sum_{i=k_2+1}^{k_3} \sum_{j=i}^{k_c} r_{ij} + \sum_{i=k_2+1}^{k_3} \sum_{j=i}^{k_{c-1}} r_{ij} \\
\sum_{i=k_3+1}^{k_4} \sum_{j=i}^{k_c} C_{ij} &= \sum_{i=k_3+1}^{k_4} \sum_{j=i}^{k_c} r_{ij} + \sum_{i=k_3+1}^{k_4} \sum_{j=i}^{k_{c-1}} r_{ij} + \sum_{i=k_3+1}^{k_4} \sum_{j=i}^{k_{c-2}} r_{ij} \\
&\vdots \\
\sum_{i=k_m+1}^{k_{c-m+1}} \sum_{j=i}^{k_c} C_{ij} &= \sum_{i=k_m+1}^{k_{c-m+1}} \sum_{j=i}^{k_c} r_{ij} + \sum_{i=k_m+1}^{k_{c-m+1}} \sum_{j=i}^{k_{c-1}} r_{ij} + \cdots + \sum_{i=k_m+1}^{k_{c-m+1}} \sum_{j=i}^{k_{c-m+1}} r_{ij} \\
&\vdots \\
\sum_{i=k_{c-3}+1}^{k_{c-2}} \sum_{j=i}^{k_c} C_{ij} &= \sum_{i=k_{c-3}+1}^{k_{c-2}} \sum_{j=i}^{k_c} r_{ij} + \sum_{i=k_{c-3}+1}^{k_{c-2}} \sum_{j=i}^{k_{c-1}} r_{ij} + \sum_{i=k_{c-3}+1}^{k_{c-2}} \sum_{j=i}^{k_{c-2}} r_{ij} \\
\sum_{i=k_{c-2}+1}^{k_{c-1}} \sum_{j=i}^{k_c} C_{ij} &= \sum_{i=k_{c-2}+1}^{k_{c-1}} \sum_{j=i}^{k_c} r_{ij} + \sum_{i=k_{c-2}+1}^{k_{c-1}} \sum_{j=i}^{k_{c-1}} r_{ij} \\
\sum_{i=k_{c-1}+1}^{k_c} \sum_{j=i}^{k_c} C_{ij} &= \sum_{i=k_{c-1}+1}^{k_c} \sum_{j=i}^{k_c} r_{ij}
\end{aligned}$$

Adding all the above equations by adding all terms vertically, we obtain

$$\begin{aligned}
\sum_{i=k_1+1}^{k_c} \sum_{j=i}^{k_c} C_{ij} &= \sum_{i=k_1+1}^{k_c} \sum_{j=i}^{k_c} r_{ij} + \sum_{i=k_2+1}^{k_{c-1}} \sum_{j=i}^{k_{c-1}} r_{ij} + \cdots + \sum_{i=k_m+1}^{k_{c-m+1}} \sum_{j=i}^{k_{c-m+1}} r_{ij} \\
&= \omega_{\mathbf{c}} d(k_c - k_1) + \omega_{\mathbf{c}} d(k_{c-1} - k_2) + \cdots + \omega_{\mathbf{c}} d(k_{c-m+1} - k_m)
\end{aligned}$$

Hence, graph (a) of figure ?? represents exactly clique potentials that are proportional to the sum of  $m$  maximum convex distance functions over all disjoint pairs of random variables of the clique. ■

The following proposition follows from the properties of the graph described in subsection ??:

**Proposition 1.** *Given a cut that partitions the vertices  $\mathbf{V}$  into two disjoint sets  $\mathbf{V}_s$  and  $\mathbf{V}_t$ , and the corresponding labeling  $\mathbf{y}$ , the capacity of the cut is equal to the energy  $E'(\mathbf{y})$  up to a constant.*

**Energy Minimization.** The above proposition implies that the labeling  $\mathbf{y}'$  that corresponds to the  $st$ -MINCUT minimizes the energy  $E'(\cdot)$  over all possible labelings  $\mathbf{y} \in (\mathbf{L}')^n$ . Since all the arc capacities of the directed graph are non-negative, it follows that the labeling  $\mathbf{y}'$  can be computed efficiently by solving the  $st$ -MINCUT problem on the directed graph defined above.

Once the labeling  $\mathbf{y}'$  is computed, the approximate solution to problem (1) can be computed as follows:

$$x'_a = \begin{cases} \hat{x}_a & \text{if } y'_a = 0, \\ y'_a + f & \text{otherwise.} \end{cases} \quad (11)$$

This completes our description of the range expansion algorithm. In section ??, we establish its theoretical guarantees for some interesting special cases.

## 4 Multiplicative Bounds

### 4.1 Bounds for Truncated Max-of-linear Models

Let  $f_n$  denote the labeling after the  $n$ -th iteration and  $E(f_n)$  denote the corresponding energy. Also,  $f^*$  denotes optimal labeling of the MRF.

Let  $r \in [0, L - 1]$  be a uniformly distributed random integer and  $L$  be the length of the interval. Using  $r$  we define the following set of intervals

$$\Gamma_r = \{[0, r], [r + 1, r + L], [r + L + 1, r + 2L], \dots, [., h - 1]\}$$

where  $h$  is the total number of labels.

Let  $\mathbf{X}(f^*, I_n)$  contain all the random variables that take an optimal labeling in  $I_n$ ,  $\mathcal{A}(f, I_n)$  be the set of all cliques for which all variables take optimum label in the interval  $I_n$  and  $\mathcal{B}(f, I_n)$  be the set of all cliques for which at least one, but not all, variable takes optimum label in the interval  $I_n$  (that is, at least one but not all variables retains old label).

In order to make the analysis more readable, the following shorthand notation is introduced:

- We denote  $\omega_c \max_{a,b \in \mathbf{X}_c} d(f^*(a) - f^*(b))$  as  $t_c^n$
- We denote  $\omega_c \max_{a \in \mathbf{X}_c} d(f^*(a) - (i_n + 1)) + \omega_c M$  as  $s_c^n$

We now define two lemmas which we will make use of later:

**Lemma 2.** *At an iteration of our algorithm, given the current labeling  $f_n$  and an interval  $I_n = [i_n + 1, j_n]$ , the new labeling  $f_{n+1}$  obtained by solving the  $st$ -mincut problem reduces the energy by at least the following:*

$$\begin{aligned} & \sum_{X_a \in \mathbf{X}(f^*, I_n)} \theta_a(f_n(a)) + \sum_{\mathbf{X}_c \in \mathcal{A}(f, I_n) \cup \mathcal{B}(f, I_n)} \theta_c(\mathbf{x}_c) \\ & - \left( \sum_{X_a \in \mathbf{X}(f^*, I_n)} \theta_a(f^*(a)) + \sum_{\mathbf{X}_c \in \mathcal{A}(f, I_n)} t_c^n + \sum_{\mathbf{X}_c \in \mathcal{B}(f, I_n)} s_c^n \right) \end{aligned}$$

It is evident from the properties of the graph discussed in subsection ?? that the energy of the new labeling  $f_{n+1}$  is bounded from above by the cost of the  $st$ -mincut. The cost of the  $st$ -mincut itself is bounded from above by the cost of any other  $st$ -cut in the graph. Consider one such  $st$ -cut that gives the following labeling:

$$f(a) = \begin{cases} f^*(a) & \text{if } X_a \in \mathbf{X}(f^*, I_n) \\ f_n(a) & \text{otherwise.} \end{cases}$$

We can derive the cost of this  $st$ -cut using the properties of subsection ?? . The energy of  $f$  equals the sum of the terms in the properties minus  $\sum_{c \in \mathcal{C}} \kappa_c$ . The energy of  $f_{n+1}$  is less than that of  $f$ . Hence, the difference between the energy of the current labeling  $f_n$  and the new labeling  $f_{n+1}$ , i.e  $E(f_n) - E(f_{n+1})$ , is at least

$$\begin{aligned} & \sum_{X_a \in \mathbf{X}(f^*, I_n)} \theta_a(f_n(a)) + \sum_{\mathbf{X}_c \in \mathcal{A}(f, I_n) \cup \mathcal{B}(f, I_n)} \theta_c(\mathbf{x}_c) \\ & - \left( \sum_{X_a \in \mathbf{X}(f^*, I_n)} \theta_a(f^*(a)) + \sum_{\mathbf{X}_c \in \mathcal{A}(f, I_n)} t_c^n + \sum_{\mathbf{X}_c \in \mathcal{B}(f, I_n)} s_c^n \right) \end{aligned}$$

This proves the lemma.  $\blacksquare$

**Lemma 3.** When  $d(\cdot)$  is linear, that is,  $d(x) = |x|$ , the following inequality holds true:

$$\begin{aligned} & \frac{1}{L} \sum_r \sum_{I_n \in \Gamma_r} \left( \sum_{\mathbf{X}_c \in \mathcal{A}(f, I_n)} t_c^n + \sum_{\mathbf{X}_c \in \mathcal{B}(f, I_n)} s_c^n \right) \\ & \leq \max \left\{ \frac{c}{2} \left( 2 + \frac{L}{M} \right), \left( 2 + \frac{2M}{L} \right) \right\} \sum_{c \in \mathcal{C}} \theta_c(\mathbf{x}_c) \end{aligned} \quad (12)$$

where  $c$  is the largest clique in the random field.

Let us denote the set of optimum labels in a clique  $\mathbf{X}_c$  arranged in an increasing order as  $l_1, l_2, \dots, l_{c-1}, l_c$ . Since we are dealing with the truncated linear metric, the terms  $t_c^n$  and  $s_c^n$  can be simplified as:

$$t_c^n = \omega_c(l_c - l_1), s_c^n = \omega_c(l_c - i_n - 1 + M) \quad (13)$$

The LHS of inequality in lemma ?? can be written as:

$$\frac{1}{L} \sum_{c \in \mathcal{C}} \left( \sum_{\mathcal{A}(f, I_n) \ni c} t_c^n + \sum_{\mathcal{B}(f, I_n) \ni c} s_c^n \right) \quad (14)$$

In order to prove the lemma, we consider the following three cases for each clique  $\mathbf{X}_c$ :

*Case - I :*  $|l_c - l_1| \geq L$  and hence  $\theta_c(\mathbf{x}_c) = \omega_c \cdot \mathbf{M}$

In this case,  $\mathbf{X}_c \notin \mathcal{A}(f, I_n)$  for all intervals  $I_n$  since the length of each interval is  $L$ . Moreover, the conditions for  $\mathbf{X}_c \in \mathcal{B}(f, I_n)$  are given by

$$\mathbf{X}_c \in \mathcal{B}(f, I_n) \iff i_n \in [l_1 - L, l_c - 1]$$

We observe that

$$\begin{aligned}
& \sum_{\mathcal{A}(f, I_n) \ni \mathbf{c}} t_c^n + \sum_{\mathcal{B}(f, I_n) \ni \mathbf{c}} s_c^n \\
& \leq \omega_c \sum_{p=1}^c \sum_{l_p=L}^{l_p-1} (M + l_p - i_n - 1) \\
& = \omega_c \sum_{p=1}^c \left( L \cdot M + \frac{L \cdot (L-1)}{2} \right) \\
& = L \cdot \frac{c}{2} \left( 2 + \frac{L-1}{M} \right) \theta_{\mathbf{c}}(\mathbf{x}_{\mathbf{c}}) \\
& \leq L \cdot \frac{c}{2} \left( 2 + \frac{L}{M} \right) \theta_{\mathbf{c}}(\mathbf{x}_{\mathbf{c}}) \tag{15}
\end{aligned}$$

*Case - II* :  $M \leq |l_c - l_1| < L$  and hence  $\theta_{\mathbf{c}}(\mathbf{x}_{\mathbf{c}}) = \omega_{\mathbf{c}} \cdot \mathbf{M}$   
In this case, the conditions for  $\mathbf{X}_{\mathbf{c}} \in \mathcal{A}(f, I_n)$  and  $X_c \in \mathcal{B}(f, I_n)$  are given by

$$\begin{aligned}
\mathbf{X}_{\mathbf{c}} \in \mathcal{A}(f, I_n) & \iff i_n \in [l_c - L, l_1 - 1] \\
\mathbf{X}_{\mathbf{c}} \in \mathcal{B}(f, I_n) & \iff i_n \in [l_1 - L, l_c - L - 1] \cup [l_1, l_c - 1]
\end{aligned}$$

We observe that

$$\begin{aligned}
& \sum_{\mathcal{A}(f, I_n) \ni \mathbf{c}} t_c^n + \sum_{\mathcal{B}(f, I_n) \ni \mathbf{c}} s_c^n \\
& = \omega_c \left( \sum_{i_n=l_c-L}^{l_1-1} (l_c - l_1) + \left\{ \sum_{p=1}^{c-1} \sum_{i_n=l_p-L}^{l_{p+1}-L-1} (M + l_p - (i_n + 1)) \right. \right. \\
& \quad \left. \left. + \sum_{i_n=l_1}^{l_c-1} (M + l_c - (i_n + 1)) \right\} \right) \\
& = \omega_c (A + B + C)
\end{aligned}$$

where

$$\begin{aligned}
A &= \sum_{i_n=l_c-L}^{l_1-1} (l_c - l_1) \\
B &= \sum_{p=1}^{c-1} \sum_{i_n=l_p-L}^{l_{p+1}-L-1} (M + l_p - (i_n + 1)) \\
C &= \sum_{i_n=l_1}^{l_c-1} (M + l_c - (i_n + 1))
\end{aligned}$$

Let  $(l_c - l_1) = r$ . Hence, we have:

$$\begin{aligned} A &= (l_c - l_1) \cdot (L - (l_c - l_1)) \\ &= r \cdot (L - r) = L \cdot r - r^2 \end{aligned} \tag{16}$$

$$\begin{aligned} B &= \sum_{p=1}^{c-1} \sum_{i_n=l_p-L}^{l_{(p+1)}-L-1} (M + l_p - (i_n + 1)) \\ &= \sum_{p=1}^{c-1} \sum_{i_n=l_p-L}^{l_{(p+1)}-L-1} M + \\ &\quad \sum_{p=1}^{c-1} \sum_{i_n=l_p-L}^{l_{(p+1)}-L-1} (l_p - (i_n + 1)) \\ &= \sum_{p=1}^{c-1} M \cdot (l_{(p+1)} - l_p) + \\ &\quad \sum_{p=1}^{c-1} \{(L - 1) + \dots + (L - (l_{p+1} - l_p))\} \end{aligned}$$

Let  $y_p = l_{(p+1)} - l_p$ . Clearly

$$\sum_{p=1}^{c-1} y_p = l_c - l_1$$

Hence,

$$\begin{aligned} B &= M \cdot r + L \cdot r - \sum_{p=1}^{c-1} \frac{y_p \cdot (y_p + 1)}{2} \\ &= M \cdot r + L \cdot r - \frac{r}{2} - \sum_{p=1}^{c-1} \frac{(y_p)^2}{2} \end{aligned} \tag{17}$$

$$\begin{aligned}
C &= \sum_{i_n=l_1}^{l_c-1} (M + l_c - (i_n + 1)) \\
&= \sum_{i_n=l_1}^{l_c-1} M + \sum_{i_n=l_1}^{l_c-1} (l_c - (i_n + 1)) \\
&= M \cdot (l_c - l_1) + \{(l_c - (l_1 + 1)) + (l_c - (l_1 + 2)) \dots + 1\} \\
&= M \cdot (l_c - l_1) + \frac{(l_c - l_1 - 1)(l_c - l_1)}{2} \\
&= M \cdot r + \frac{(r-1) \cdot r}{2} \\
&= M \cdot r + \frac{r^2}{2} - \frac{r}{2}
\end{aligned} \tag{18}$$

Using (??), (??), (??), we have

$$A + B + C = (2L + 2M - r)r + \frac{r^2}{2} - r - \sum_{p=1}^{c-1} \frac{y_p^2}{2} \tag{19}$$

We have

$$\sum_{p=1}^{c-1} \frac{y_p^2}{2} \geq \frac{r^2}{2(c-1)}$$

(Taking each  $y_p = r/(c-1)$ )

Hence,

$$A + B + C \leq \left\{ 2L + 2M - 1 - \frac{r}{2} \left( \frac{c}{c-1} \right) \right\} \cdot r$$

Let

$$f(r) = \left\{ 2L + 2M - 1 - \frac{r}{2} \left( \frac{c}{c-1} \right) \right\} \cdot r$$

Taking derivative of  $f(r)$ , we obtain

$$f'(r) = 2L + 2M - 1 - \frac{r \cdot c}{(c-1)}$$

$f(r)$  is a quadratic function which opens downwards (coefficient of  $r^2$  is negative). Also

$$f'(L) = \left( \frac{c-2}{c-1} \right) \cdot L + 2M - 1 > 0$$

Hence, using  $r = L$ ,

$$\begin{aligned}
A + B + C &< L \cdot \left\{ 2L + 2M - 1 - \frac{L}{2} \cdot \left( \frac{c}{c-1} \right) \right\} \\
&< L \cdot \left\{ \frac{2L}{M} + 2 - \frac{L}{2M} \cdot \left( \frac{c}{c-1} \right) \right\} \theta_{\mathbf{c}}(\mathbf{x}_{\mathbf{c}})
\end{aligned} \tag{20}$$

where the last expression is obtained using the fact that  $\theta_{\mathbf{c}}(\mathbf{x}_{\mathbf{c}}) = \omega_{\mathbf{c}} \cdot \mathbf{M}$ .

*Case – III :*  $|l_c - l_1| < M$  and hence  $\theta_{\mathbf{c}}(\mathbf{x}_{\mathbf{c}}) = \omega_{\mathbf{c}}(\mathbf{l}_{\mathbf{c}} - \mathbf{l}_1)$   
In this case, the conditions for  $\mathbf{X}_{\mathbf{c}} \in \mathcal{A}(f, I_n)$  and  $\mathbf{X}_{\mathbf{c}} \in \mathcal{B}(f, I_n)$  are given by

$$\begin{aligned}\mathbf{X}_{\mathbf{c}} \in \mathcal{A}(f, I_n) &\iff i_n \in [l_c - L, l_1 - 1] \\ \mathbf{X}_{\mathbf{c}} \in \mathcal{B}(f, I_n) &\iff i_n \in [l_1 - L, l_c - L - 1] \cup [l_1, l_c - 1]\end{aligned}$$

We observe that

$$\begin{aligned}& \sum_{\mathcal{A}(f, I_n) \ni \mathbf{c}} t_{\mathbf{c}}^n + \sum_{\mathcal{B}(f, I_n) \ni \mathbf{c}} s_{\mathbf{c}}^n \\ &= \omega_c \left( \sum_{i_n=l_c-L}^{l_1-1} (l_c - l_1) + \left\{ \sum_{p=1}^{c-1} \sum_{i_n=l_p-L}^{l_{p+1}-L-1} (M + l_p - (i_n + 1)) \right. \right. \\ & \quad \left. \left. + \sum_{i_n=l_1}^{l_c-1} (M + l_c - (i_n + 1)) \right\} \right) \\ &= \omega_c (A + B + C)\end{aligned}$$

Similar to case II, we can write

$$A + B + C = (2L + 2M - r/2)r - r - \sum_{p=1}^{c-1} \frac{y_p^2}{2}$$

Since  $r < M$

$$\begin{aligned}A + B + C &\leq (2L + 2M)(l_c - l_1) \\ &= L \left( 2 + \frac{2M}{L} \right) \theta_{\mathbf{c}}(\mathbf{x}_{\mathbf{c}})\end{aligned}\tag{21}$$

where the last expression is obtained using the fact that  $\theta_{\mathbf{c}}(\mathbf{x}_{\mathbf{c}}) = \omega_{\mathbf{c}} \cdot \mathbf{M}$ .

Substituting inequalities (??), (??) and (??) in expression (??) and dividing both sides by  $L$  for all  $\mathbf{X}_{\mathbf{c}}$ , we obtain inequality of lemma ??. This proves the lemma.  $\blacksquare$

**Proposition 2.** *The range expansion algorithm with  $h' = M$  has a multiplicative bound of  $O(C)$  for truncated max-of-linear model when  $m = 1$ . The term  $C$  equals the size of the largest clique. Hence, if  $\mathbf{x}^*$  is a labeling with minimum energy and  $\hat{\mathbf{x}}$  is the labeling estimated by range expansion algorithm then*

$$\sum_{a \in \mathcal{V}} \theta_a(\hat{x}_a) + \sum_{\mathbf{c} \in \mathcal{C}} \theta_{\mathbf{c}}(\hat{\mathbf{x}}_{\mathbf{c}}) \leq \sum_{a \in \mathcal{V}} \theta_a(x_a^*) + O(C) \sum_{\mathbf{c} \in \mathcal{C}} \theta_{\mathbf{c}}(\mathbf{x}_{\mathbf{c}}^*).$$

*The above inequality holds for arbitrary set of unary potentials and non-negative clique weights.*



The following equation can be deduced from the above definitions:

$$\sum_{X_a \in \mathbf{X}} \theta_a(f^*(a)) = \sum_{I_n \in \mathcal{I}_r} \sum_{X_a \in \mathbf{X}(f^*, I_n)} \theta_a(f^*(a)) \quad (22)$$

since  $f^*(a)$  belongs to exactly one interval in  $I_r$  for all  $X_a$ .

For the final labeling  $f$  of the range expansion algorithm, the term in lemma ?? should be non-positive for all intervals  $I_n$  because  $f$  is a local optimum. Hence,

$$\begin{aligned} & \sum_{X_a \in \mathbf{X}(f^*, I_n)} \theta_a(f(a)) + \sum_{\mathbf{x}_c \in \mathcal{A}(f, I_n) \cup \mathcal{B}(f, I_n)} \theta_c(\mathbf{x}_c) \\ & \leq \left( \sum_{X_a \in \mathbf{X}(f^*, I_n)} \theta_a(f^*(a)) + \sum_{\mathbf{x}_c \in \mathcal{A}(f, I_n)} t_c^n + \sum_{\mathbf{x}_c \in \mathcal{B}(f, I_n)} s_c^n \right), \forall I_n \end{aligned}$$

We sum the above inequality over all  $I_n \in \Gamma_r$ . The summation of the LHS is at least  $E(f)$ . Also, using (??), the summation of the above inequality can be written as:

$$E(f) \leq \sum_{X_a \in \mathbf{X}} \theta_a(f^*(a)) + \sum_{I_n \in \Gamma_r} \left( \sum_{\mathbf{x}_c \in \mathcal{A}(f, I_n)} t_c^n + \sum_{\mathbf{x}_c \in \mathcal{B}(f, I_n)} s_c^n \right)$$

We now take the expectation of the above inequality over the uniformly distributed random integer  $r \in [0, L - 1]$ . The LHS of the inequality and the first term on the RHS (that is,  $\sum \theta_a(f^*(a))$ ) are constants with respect to  $r$ . Hence, we get

$$E(f) \leq \sum_{X_a \in \mathbf{X}} \theta_a(f^*(a)) + \frac{1}{L} \sum_r \sum_{I_n \in \Gamma_r} \left( \sum_{\mathbf{x}_c \in \mathcal{A}(f, I_n)} t_c^n + \sum_{\mathbf{x}_c \in \mathcal{B}(f, I_n)} s_c^n \right) \quad (23)$$

Lemma ?? allows us to write the above inequality as

$$E(f) \leq \sum_{X_a \in \mathbf{X}} \theta_a(f^*(a)) + \max \left\{ \frac{c}{2} \left( 2 + \frac{L}{M} \right), \left( 2 + \frac{2M}{L} \right) \right\} \sum_{\mathbf{c} \in \mathcal{C}} \theta_c(\mathbf{x}_c) \quad (24)$$

The R.H.S of lemma ?? is minimum under the following condition:

$$\frac{c}{2} \left( 2 + \frac{L}{M} \right) = \left( 2 + \frac{2M}{L} \right) \quad (25)$$

The positive solution of the above quadratic equation gives the optimum value  $L_{opt}$  of interval length:

$$L_{opt} = \left\{ \frac{2 - c + \sqrt{c^2 + 4}}{c} \right\} \cdot M \quad (26)$$

However, we bound the minimum value of L as M, giving:

$$L_{opt} = \max \left\{ \left\{ \frac{2 - c + \sqrt{c^2 + 4}}{c} \right\} \cdot M, M \right\} \quad (27)$$

Note  $L_{opt}$  equals  $\sqrt{2}M$  for  $c = 2$  and  $M$  for  $c > 2$ . Substituting the optimum value of L from equation ?? in inequality of lemma ?? and simplifying inequality ?? gives the multiplicative

bound for truncated max-of-linear model as  $\frac{(c+2)+\sqrt{c^2+4}}{2}$ . Hence, the multiplicative bound is  $O(c)$  where  $c$  is the size of the maximal clique. For  $c = 2$ , this gives a bound of  $2 + \sqrt{2}$  using  $L = \sqrt{2}M$ . ■

## 4.2 Bounds for $m$ -Truncated Max-of-linear Models

We will now give the multiplicative bound of our range expansion algorithm for  $m$ -max-of-linear for any value of  $m$ .

We need some definitions before we analyse the bound. Let  $\mathcal{A}(f, I_n)$  be the set of all cliques for which all variables take optimum label in the interval  $I_n$  and  $\mathcal{B}_k(f, I_n)$  where  $k \in [1, \dots, m-1]$  be the set of all cliques for which exactly  $k$  variables retain their old label. Let  $\mathcal{B}_m(f, I_n)$  be the set of all cliques for which  $m$  or more variables retain their old label.

We also introduce the following shorthand notation:

- We denote  $\omega_{\mathbf{c}}\{\sum_{i=1}^m d(p_i(\mathbf{x}_c) - p_{c-i+1}(\mathbf{x}_c))\}$  as  $t_{m,c}^n$
- We denote  $\omega_{\mathbf{c}}\{\sum_{i=2}^{m-1} d(p_i(\mathbf{x}_c) - p_{c-i+1}(\mathbf{x}_c) + d(p_c(\mathbf{x}_c) - i_m - 1))\} + \omega_{\mathbf{c}} \cdot M$  as  $s_{1,c}^n$
- In general, we denote  $\omega_{\mathbf{c}}\{\sum_{i=k+1}^{m-k} d(p_i(\mathbf{x}_c) - p_{c-i+1}(\mathbf{x}_c) + \sum_{i=1}^k d(p_{c-i+1}(\mathbf{x}_c) - i_m - 1))\} + k \cdot \omega_{\mathbf{c}} \cdot M$  as  $s_{k,c}^n$

We state the following lemma which is a generalization of lemma ??.

**Lemma 4.** *At an iteration of our algorithm, given the current labeling  $f_n$  and an interval  $I_n = [i_n + 1, j_n]$ , the new labeling  $f_{n+1}$  obtained by solving the st-mincut problem reduces the energy by at least the following:*

$$\begin{aligned} & \sum_{X_a \in \mathbf{X}(f^*, I_n)} \theta_a(f_n(a)) + \sum_{\mathbf{X}_c \in \mathcal{A}(f, I_n) \cup \mathcal{B}_1(f, I_n) \cup \dots \cup \mathcal{B}_m(f, I_n)} \theta_{\mathbf{c}}(\mathbf{x}_c) \\ & - \left( \sum_{X_a \in \mathbf{X}(f^*, I_n)} \theta_a(f^*(a)) + \sum_{\mathbf{X}_c \in \mathcal{A}(f, I_n)} t_{m,c}^n + \sum_{\mathbf{X}_c \in \mathcal{B}_1(f, I_n)} s_{1,c}^n + \dots + \sum_{\mathbf{X}_c \in \mathcal{B}_m(f, I_n)} s_{m,c}^n \right) \end{aligned}$$

We also make use of the following lemma which generalizes lemma ??:

**Lemma 5.** *When  $d(\cdot)$  is linear, that is,  $d(x) = |x|$ , the following inequality holds true:*

$$\begin{aligned} & \frac{1}{L} \sum_r \sum_{I_n \in \Gamma_r} \left( \sum_{\mathbf{X}_c \in \mathcal{A}(f, I_n)} t_{m,c}^n + \sum_{\mathbf{X}_c \in \mathcal{B}_1(f, I_n)} s_{1,c}^n + \dots + \sum_{\mathbf{X}_c \in \mathcal{B}_m(f, I_n)} s_{m,c}^n \right) \\ & \leq \max \left\{ m \cdot \frac{c}{2} \left( 2 + \frac{L}{M} \right), \left( 2 + \frac{2M}{L} \right) \right\} \sum_{\mathbf{c} \in \mathcal{C}} \theta_{\mathbf{c}}(\mathbf{x}_c) \end{aligned} \quad (28)$$

where  $c$  is the largest clique in the random field.

Let us denote the set of optimum labels in a clique  $\mathbf{X}_c$  arranged in an increasing order as  $l_1, l_2, \dots, l_{c-1}, l_c$ . Since we are dealing with the truncated linear metric, the terms  $t_{m,c}^n$  and  $s_{k,c}^n$  can be simplified as:

$$\begin{aligned}
t_{m,c}^n &= \omega_c \{ (l_c - l_1) + (l_{c-1} - l_2) + \cdots + (l_{c-m+1} - l_m) \} \\
s_{1,c}^n &= \omega_c \{ (l_c - i_n - 1 + M) + (l_{c-1} - l_2) + \cdots + (l_{c-n+1} - l_n) \} \\
&\vdots \\
s_{m,c}^n &= \omega_c \{ (l_c - i_n - 1 + M) + (l_{c-1} - i_n - 1 + M) + \cdots + (l_{c-n+1} - i_n - 1 + M) \}
\end{aligned}$$

Since the labels are sorted in ascending order  $\{l_1, \dots, l_c\}$ , it follows that

$$\begin{aligned}
t_{m,c}^n &\leq \omega_c \cdot m \cdot \{l_c - l_1\} \\
s_{1,c}^n &\leq \omega_c \cdot m \cdot \{l_c - i_n - 1 + M\} \\
&\vdots \\
s_{m,c}^n &\leq \omega_c \cdot m \cdot \{l_c - i_n - 1 + M\}
\end{aligned}$$

The LHS of inequality in lemma ?? can be written as:

$$\begin{aligned}
&\frac{1}{L} \sum_{\mathbf{c} \in \mathcal{C}} \left( \sum_{\mathcal{A}(f, I_n) \ni \mathbf{c}} t_{m,c}^n + \sum_{\mathcal{B}_1(f, I_n) \ni \mathbf{c}} s_{1,c}^n + \cdots + \sum_{\mathcal{B}_m(f, I_n) \ni \mathbf{c}} s_{m,c}^n \right) \\
&\leq \frac{1}{L} \sum_{\mathbf{c} \in \mathcal{C}} \left( \sum_{\mathcal{A}(f, I_n) \ni \mathbf{c}} \omega_c \cdot m \cdot \{l_c - l_1\} + \sum_{\mathcal{B}_1(f, I_n) \ni \mathbf{c}} \omega_c \cdot m \cdot \{l_c - i_n - 1 + M\} + \dots \right. \\
&\quad \left. + \sum_{\mathcal{B}_m(f, I_n) \ni \mathbf{c}} \omega_c \cdot m \cdot \{l_c - i_n - 1 + M\} \right) \tag{29} \\
&\leq \frac{1}{L} \sum_{\mathbf{c} \in \mathcal{C}} \left( \sum_{\mathcal{A}(f, I_n) \ni \mathbf{c}} \omega_c \cdot m \cdot \{l_c - l_1\} + \sum_{\mathcal{B}_1(f, I_n) \cup \dots \cup \mathcal{B}_m(f, I_n) \ni \mathbf{c}} \omega_c \cdot m \cdot \{l_c - i_n - 1 + M\} \right) \\
&= m \cdot \frac{1}{L} \sum_{\mathbf{c} \in \mathcal{C}} \left( \sum_{\mathcal{A}(f, I_n) \ni \mathbf{c}} \omega_c \cdot t_c^n + \sum_{\mathcal{B}(f, I_n) \ni \mathbf{c}} \omega_c \cdot s_c^n \right)
\end{aligned}$$

The last expression in equation ?? is  $m$  times expression ?. The analysis below follows the same steps as in lemma ??.

In order to prove the lemma, we consider the following three cases for each clique  $\mathbf{X}_{\mathbf{c}}$ :

*Case - I* :  $|l_c - l_1| \geq L$  and hence  $\omega_c \cdot M \leq \theta_{\mathbf{c}}(\mathbf{x}_{\mathbf{c}}) \leq \omega_c \cdot m \cdot M$

This results in

$$\begin{aligned}
m \cdot \frac{1}{L} \sum_{\mathbf{c} \in \mathcal{C}} \left( \sum_{\mathcal{A}(f, I_n) \ni \mathbf{c}} \omega_c \cdot t_c^n + \sum_{\mathcal{B}(f, I_n) \ni \mathbf{c}} \omega_c \cdot s_c^n \right) &\leq m \cdot \frac{c}{2} \left( 2 + \frac{L}{M} \right) \omega_c \cdot M \\
&\leq m \cdot \frac{c}{2} \left( 2 + \frac{L}{M} \right) \theta_{\mathbf{c}}(\mathbf{x}_{\mathbf{c}}) \quad (30)
\end{aligned}$$

*Case – II* :  $M \leq |l_c - l_1| < L$  and hence  $\omega_c \cdot M \leq \theta_{\mathbf{c}}(\mathbf{x}_{\mathbf{c}}) \leq \omega_c \cdot m \cdot M$   
This gives

$$\begin{aligned}
m \cdot \frac{1}{L} \sum_{\mathbf{c} \in \mathcal{C}} \left( \sum_{\mathcal{A}(f, I_n) \ni \mathbf{c}} \omega_c \cdot t_c^n + \sum_{\mathcal{B}(f, I_n) \ni \mathbf{c}} \omega_c \cdot s_c^n \right) &\leq m \cdot \left\{ \frac{2L}{M} + 2 - \frac{L}{2M} \cdot \left( \frac{c}{c-1} \right) \right\} \omega_c \cdot M \\
&\leq m \cdot \left\{ \frac{2L}{M} + 2 - \frac{L}{2M} \cdot \left( \frac{c}{c-1} \right) \right\} \theta_{\mathbf{c}}(\mathbf{x}_{\mathbf{c}}) \quad (31)
\end{aligned}$$

*Case – III* :  $|l_c - l_1| < M$  and hence  $\omega_c(l_c - l_1) \leq \theta_{\mathbf{c}}(\mathbf{x}_{\mathbf{c}}) \leq m \cdot \omega_c(l_c - l_1)$   
This leads to

$$\begin{aligned}
m \cdot \frac{1}{L} \sum_{\mathbf{c} \in \mathcal{C}} \left( \sum_{\mathcal{A}(f, I_n) \ni \mathbf{c}} \omega_c \cdot t_c^n + \sum_{\mathcal{B}(f, I_n) \ni \mathbf{c}} \omega_c \cdot s_c^n \right) &\leq m \cdot (2L + 2M)(l_c - l_1) \\
&\leq m \cdot L \left( 2 + \frac{2M}{L} \right) \theta_{\mathbf{c}}(\mathbf{x}_{\mathbf{c}}) \quad (32)
\end{aligned}$$

Substituting inequalities (??), (??) and (??) in expression (??), we obtain inequality of lemma ???. This proves the lemma. ■

**Proposition 3.** *The range expansion algorithm with  $h' = M$  has a multiplicative bound of  $O(m \cdot C)$  for truncated max-of-linear model for any general value of  $m$ . The term  $C$  equals the size of the largest clique. Hence, if  $\mathbf{x}^*$  is a labeling with minimum energy and  $\hat{\mathbf{x}}$  is the labeling estimated by range expansion algorithm then*

$$\sum_{a \in \mathcal{V}} \theta_a(\hat{x}_a) + \sum_{\mathbf{c} \in \mathcal{C}} \theta_{\mathbf{c}}(\hat{\mathbf{x}}_{\mathbf{c}}) \leq \sum_{a \in \mathcal{V}} \theta_a(x_a^*) + O(m \cdot C) \sum_{\mathbf{c} \in \mathcal{C}} \theta_{\mathbf{c}}(\mathbf{x}_{\mathbf{c}}^*).$$

*The above inequality holds for arbitrary set of unary potentials and non-negative clique weights.*

The proof of proposition ?? follows exactly the same steps as that of proposition ??.