Supplementary Material

A Proofs of Lemmas and Theorems

A.1 Proof of Theorem 1

Theorem. *The extended binary set function f as given by Definition 5 is submodular.*

Recall that we define the extended binary submodular function for the valid states as equal to the original multi-label function and for the invalid states as the following:

$$f(S) = f(\overline{S}) + (|\overline{S}| - |S|)L.$$

Here $\overline{S}$ is the minimum covering state for an invalid state, S, which is defined as the smallest cardinality valid state, $\overline{S} \in Z$, such that $S \subset \overline{S}$. For a valid state $S = \overline{S}$.

Let us factorize $f(S) = g(S) + h(S)$, where $h(S) = f(\overline{S}) + |\overline{S}|L$, and $g(S) = -|S|L$. Since, g is modular, it is sufficient to show that h is submodular. We will need the following result to prove the Theorem.

Lemma 10. *For sets X, Y, and $(X \cap Y) \subseteq \mathcal{V}$ and their minimum covering states $\overline{X}, \overline{Y}$, and $\overline{X \cap Y}$ respectively:*

$$f(\overline{X \cap Y}) \leq f(\overline{X} \cap \overline{Y}))$$

Proof. Recall that for any valid state S, $\overline{S} = S$. Consider, two valid states $A, B \subseteq V$ with $A \subseteq B$. It is easy to see that:

$$h(B) - h(A) = L(|B| - |A|) + f(B) - f(A) \geq 0. \tag{7}$$

Since, $A \subseteq B$, therefore, $|B| - |A| \geq 0$. Also $L \gg f(B) - f(A)$ by definition. Therefore, $h(A) \leq h(B)$. Further, it has been shown in section 6 of [43] that for two $X, Y \in \mathcal{V}$, $\overline{X \cup Y} = (\overline{X} \cup \overline{Y})$ and $\overline{X \cap Y} \subseteq (\overline{X} \cap \overline{Y})$ holds. Therefore, using Eq. (7), $f(\overline{X \cap Y}) \leq f(\overline{X} \cap \overline{Y}))$. □

We can now give the proof of the theorem as follows. For the valid states, the extended function f, has been shown to be submodular in [2]. Therefore, here, we show only for the cases when S is an invalid state. Now, for two arbitrary (valid or invalid) sets, $X, Y \subseteq \mathcal{V}$

$$\begin{aligned}
h(X) + h(Y) &= f(\overline{X}) + f(\overline{Y}) + |\overline{X}|L + |\overline{Y}|L \\
&\geq f(\overline{X} \cup \overline{Y}) + f(\overline{X} \cap \overline{Y}) + |\overline{X}|L + |\overline{Y}|L \\
&\qquad \text{(Using submodularity over } \overline{X}\text{, and } \overline{Y}\text{)} \\
&= f(\overline{X} \cup \overline{Y}) + f(\overline{X} \cap \overline{Y}) + |\overline{X} \cup \overline{Y}|L + |\overline{X} \cap \overline{Y}|L \\
&\qquad \text{(Since } |\overline{X}| + |\overline{Y}| = |\overline{X} \cup \overline{Y}| + |\overline{X} \cap \overline{Y}|\text{)} \\
&= f(\overline{X \cup Y}) + |\overline{X \cup Y}|L + f(\overline{X} \cap \overline{Y}) + |\overline{X} \cap \overline{Y}|L \\
&\qquad \text{(Since } \overline{X \cup Y} = (\overline{X} \cup \overline{Y})\text{)} \\
&\geq f(\overline{X \cup Y}) + |\overline{X \cup Y}|L + f(\overline{X \cap Y}) + |\overline{X \cap Y}|L \\
&\qquad \text{(Using Lemma 10)} \\
&= h(X \cup Y) + h(X \cap Y).
\end{aligned}$$

The above shows that h is submodular. It is easy to see that, g, as defined above is modular. Since addition of a modular function and a submodular function is submodular, therefore, $f = g + h$ is submodular.

A.2 Proof of Lemma 3

Lemma. *For any element, e, of an invalid extreme base, $b^{\prec}$: $b^{\prec}(e) = a_e L + b_e$, where $|a_e|, |b_e| \ll L$ and $a_e \in I$.*

Proof. Let S_2 be the set of all elements smaller than e as per $\prec$. Let $S_1 = S_2 \cup \{e\}$.

$$\begin{aligned}
b^{\prec}(e) &= f(S_1) - f(S_2) && \text{(Definition of extreme base)} \\
&= \big(f(\overline{S}_1) + (|\overline{S}_1| - |S_1|)L\big) - \big(f(\overline{S}_2) + (|\overline{S}_2| - |S_2|)L\big) && \text{(Definition 5)} \\
&= \big(f(\overline{S}_1) - f(\overline{S}_2)\big) + \big(|\overline{S}_1| - |S_1| - |\overline{S}_2| + |S_2|\big)\, L \\
&= a_e L + b && \text{(where } |a_e|, |b_e| \ll L\text{)}
\end{aligned}$$

□

A.3 Proof of Lemma 4

Lemma. *Consider two base vectors x_1 and x_2 such that $\|x_1\|^2, \|x_2\|^2 < |\mathcal{V}|M^2$. If $x_2 = (1-\lambda)x_1 + \lambda b^{\prec}$ and $b^{\prec}$ is an invalid extreme base, then $\lambda \leq |\mathcal{V}|\frac{M}{L}$.*

Proof. Recall that in our algorithm, base vector is represented as the sum of contributions from valid and invalid extreme bases separately: $x = x_v + x_i$, where x_v and x_i are the base vectors collecting contributions of valid and invalid extreme bases respectively. Further, we start from a valid extreme and in each iteration of the algorithm, keep on decreasing the norm of the overall base vector. Note that, all the elements of a valid extreme base are smaller than M. Therefore the squared ℓ_2 norm of the overall base vector is less than $|\mathcal{V}|M^2$ at any point in the algorithm.

We will prove the lemma by contradiction, and show that unless the λ for the invalid extreme base is less than $|\mathcal{V}|M/L$, the squared norm of the overall base vector is more than $|\mathcal{V}|M^2$, which is a contradiction.

We will first need to prove the following result:

Lemma 11. *Consider an invalid ordering $\prec$, and its corresponding invalid extreme base $b^{\prec}$. Let e be the smallest element (as per $\prec$), for which validity condition is violated. Then, $\exists\, a_e \in \mathbb{R}$, and $a_e \geq (1 - M/L)$, s.t. $b^{\prec}(e) = a_e L$.*

Proof. Let S_2 be the set of all elements smaller than e as per $\prec$. Let $S_1 = S_2 \cup \{e\}$. Notice that S_2 is a valid and S_1 is an invalid state.

$$\begin{aligned}
b^{\prec}(e) &= f(S_1) - f(S_2) && \text{(Definition of extreme base)}\\
&= \big(f(\overline{S}_1) + (|\overline{S}_1| - |S_1|)L\big) - f(S_2) && \text{(Definition 5)}\\
&\geq \min_{S\in Z} f(S) - f(S_2) + (|\overline{S}_1| - |S_1|)L \\
& && (\overline{S}_1 \text{ is a valid state, therefore } f(\overline{S}_1) \geq \min\nolimits_{S\in Z} f(S))\\
&\geq \min_{S\in Z} f(S) - \max_{S\in Z} f(S) + (|\overline{S}_1| - |S_1|)L \\
& && (S_2 \text{ is a valid state, therefore } f(S_2) \leq \max\nolimits_{S\in Z} f(S))\\
&= (|\overline{S}_1| - |S_1| - M/L)L && \text{(Defintion of } M)
\end{aligned}$$

Note that for any invalid state S_1, $(|\overline{S}_1| - |S_1|) \geq 1$. Therefore there exists $a_e \geq (1 - M/L)$ such that $b^{\prec}(e) = a_e L$. $\square$

To prove our main result by contradiction, assume $\lambda > |\mathcal{V}|M/L$. Let e be the smallest element (as per $\prec$ of invalid extreme base $b^{\prec}$), for which validity condition is violated. Consider:

$$\begin{aligned}
(x_2(e))^2 &= ((1-\lambda)x_1(e) + \lambda b^{\prec}(e))^2 \\
&> ((1-\lambda)x_1(e) + b^{\prec}(e)|\mathcal{V}|M/L)^2 && (\lambda > |\mathcal{V}|M/L) \\
&= ((1-\lambda)x_1(e) + a_e|\mathcal{V}|M))^2. && \text{(Using lemma 11)}
\end{aligned}$$

Two cases are possible:

1. $x_1(e) \geq 0$:

$$\begin{aligned}
(x_2(e))^2 &\geq ((1-\lambda)x_1(e) + a_e|\mathcal{V}|M))^2 \\
&\geq (a_e|\mathcal{V}|M)^2 && (\text{Since } (1-\lambda) \geq 0) \\
&= (a_e|\mathcal{V}|)(|\mathcal{V}|M^2)
\end{aligned}$$

Since $M \ll L$, and $a_e \geq (1 - M/L)$, therefore $a_e \approx 1$. The smallest problem size that we consider is of 3 pixels and 2 labels for which $|\mathcal{V}| = 6$. Hence, for our case, $a_e|\mathcal{V}| > a_e\sqrt{|\mathcal{V}|} > 2$. This implies:

$$(x_2(e))^2 > |\mathcal{V}|M^2.$$

2. $x_1(e) < 0$:

Note that for x_1 and any element $e \in \mathcal{V}$ we have $x_1(e)^2 \leq \|x_1\|^2 \leq |\mathcal{V}|M^2$. This implies that $x_1(e) \geq -\sqrt{|\mathcal{V}|}M$.

$$\begin{aligned}
(x_2(e))^2 &\geq ((1-\lambda)x_1(e) + a_e|\mathcal{V}|M)^2 \\
&\geq (-(1-\lambda)\sqrt{|\mathcal{V}|}M + a_e|\mathcal{V}|M)^2 \\
&\geq (-\sqrt{|\mathcal{V}|}M + a_e|\mathcal{V}|M)^2 && \text{(Since } 1 \geq (1-\lambda) \geq 0\text{)} \\
&= M^2|\mathcal{V}|(a_e\sqrt{|\mathcal{V}|} - 1) && \text{(As described in the first case } a_e\sqrt{|\mathcal{V}|} > 2\text{)} \\
&> |\mathcal{V}|M^2.
\end{aligned}$$

Both the cases imply that if $\lambda > |\mathcal{V}|M/L$ then norm $||x_2||^2 > |\mathcal{V}|M^2$ which is a contradiction. Hence for any invalid extreme base its contribution λ in the overall base vector must be less than $|\mathcal{V}|M/L$. □

A.4 Proof of Lemma 5

Lemma. *Let $\prec$ be an invalid ordering and $\overline{\prec}$ be its canonical ordering. Then, $b^{\prec}(e) - b^{\overline{\prec}}(e) \ll L, \forall e \in \mathcal{V}$.*

Proof. Let S and S' be the set of elements preceding $p^i, p \in \mathcal{P}$, in ordering $\prec$ and $\overline{\prec}$ respectively. Consider the term $b^{\prec}(p^i)$:

$$\begin{aligned}
b^{\prec}(p^i) &= f(S \cup \{p^i\}) - f(S) \\
&= L\sum_{q\in\mathcal{P}}\Big(|(\overline{S \cup \{p^i\}})_q| - |(S \cup \{p^i\})_q|\Big) + f\Big(\overline{S \cup \{p^i\}}\Big) \\
&\quad - L\sum_{q\in\mathcal{P}}\Big(|\overline{S}_q| - |S_q|\Big) - f\Big(\overline{S}\Big) && \text{(Def. 6)} \\
&= L\Big(|\overline{S_p \cup \{p^i\}}| - |S_p \cup \{p^i\}|\Big) - L\Big(|\overline{S}_p| - |S_p|\Big) \\
&\quad + f(\overline{S \cup \{p^i\}}) - f(\overline{S})
\end{aligned}$$

Similarly we obtain:

$$b^{\overline{\prec}}(p^i) = L\Big(|\overline{S'_p \cup \{p^i\}}| - |S'_p \cup \{p^i\}|\Big) - L\Big(|\overline{S'}_p| - |S'_p|\Big)$$
$$+ f\Big(\overline{S' \cup \{p^i\}}\Big) - f\Big(\overline{S'}\Big)$$

Note that a canonical ordering does not change intersay ordering between elements corresponding to a particular pixel. Therefore, $S_p = S'_p$, and:

$$b^{\prec}(p^i) - b^{\overline{\prec}}(p^i) = (f(\overline{S \cup \{p^i\}}) - f(\overline{S}) - f(\overline{S' \cup \{p^i\}}) + f(\overline{S'})$$

Since, all terms in the r.h.s. of the equation above, correspond to valid sets, therefore $b^{\prec}(p^i) - b^{\overline{\prec}}(p^i) \ll L$. $\square$

A.5 Proof of Lemma 6

Lemma. *For a canonical invalid ordering $\overline{\prec}$, let p^i and p^j be two adjacent elements corresponding to a pixel p, s.t. $p^i \overline{\prec} p^j$. Let $\overline{\prec}_p^{i,j}$ be the ordering obtained by swapping p^i and p^j. Then $b^{\overline{\prec}_p^{i,j}} - b^{\overline{\prec}} = (\chi_p^j - \chi_p^i)(aL + b)$, where χ_p^i is an indicator vector for the element p^i, and $a, b \ll L$.*

Proof. Recall that for an extreme base $b^{\prec} : b^{\prec}(k) = f(k_{\prec}) - f((k-1)_{\prec})$, where $k_{\prec}$ is the first k elements in the ordered set $\{v_1, \ldots, v_k, \ldots, v_n\}$. Since the swap between p^i, and p^j leaves the set of preceding elements unchanged for all other elements, therefore, $b^{\overline{\prec}_p^{i,j}} - b^{\overline{\prec}}$ is non-zero corresponding to only p^i, and p^j.

Let S be the set of elements preceding p^i in $\overline{\prec}$. Now:

$$\begin{aligned} b^{\overline{\prec}}(p^j) &= f(S \cup \{p^j\} \cup \{p^i\}) - f(S \cup \{p^i\}) \\ &= L(|\overline{S \cup \{p^j\} \cup \{p^i\}}| - |S \cup \{p^j\} \cup \{p^i\}|) + f(\overline{S \cup \{p^j\} \cup \{p^i\}}) \\ &- L(|\overline{S \cup \{p^i\}}| - |S \cup \{p^i\}|) - f(\overline{S \cup \{p^i\}}) \qquad (8a) \end{aligned}$$

Similarly we have,

$$b^{\overline{\prec}_p^{i,j}}(p^j) = L(|\overline{S \cup \{p^j\}}| - |S \cup \{p^j\}|) - L(|\overline{S}| - |S|) + f(\overline{S \cup \{p^j\}}) - f(\overline{S}), \tag{8b}$$

Subtracting Eq. (8a) from Eq. (8b) and using $(|S \cup \{p^i\}| + |S \cup \{p^j\}| - |S \cup \{p^i\} \cup \{p^j\}| - |S|) = 0$ we have

$$\begin{aligned} b^{\overline{\prec}_p^{i,j}}(p^j) - b^{\overline{\prec}}(p^j) &= L(|\overline{S \cup \{p^i\}}| + |\overline{S \cup \{p^j\}}| - |\overline{S \cup \{p^i\} \cup \{p^j\}}| - |\overline{S}|) \\ &\quad + (f(\overline{S \cup \{p^j\}}) - f(\overline{S}) - f(\overline{S \cup \{p^j\} \cup \{p^i\}}) + f(\overline{S \cup \{p^i\}}), \\ &= aL + b, \end{aligned}$$

where:

$$\begin{aligned} a &= |\overline{S \cup \{p^i\}}| + |\overline{S \cup \{p^j\}}| - |\overline{S \cup \{p^i\} \cup \{p^j\}}| - |\overline{S}|, \text{ and} \\ b &= (f(\overline{S \cup \{p^j\}}) - f(\overline{S}) - f(\overline{S \cup \{p^j\} \cup \{p^i\}}) + f(\overline{S \cup \{p^i\}}). \end{aligned}$$

Note that b is sum of function values at valid states and is $\ll L$. Two cases arise for the value of a:

1. $i < j$:
   In this case $|\overline{S_p \cup \{p^i\} \cup \{p^j\}}| = |\overline{S_p \cup \{p^j\}}|$, and $a = |\overline{S \cup \{p^i\}}| - |\overline{S}|$. Therefore, $a \ll L$.
2. $j < i$:
   In this case $|\overline{S_p \cup \{p^i\} \cup \{p^j\}}| = |\overline{S_p \cup \{p^i\}}|$, and $a = |\overline{S \cup \{p^j\}}| - |\overline{S}|$. Therefore, $a \ll L$

Hence $b^{\overline{\prec}_p^{i,j}}(p^j) - b^{\overline{\prec}}(p^j) = aL + b$, such that $a, b \ll L$. Further, since $b^{\overline{\prec}_p^{i,j}}$ and $b^{\overline{\prec}}$ are extreme bases, and the sum of all the elements in them is constant, therefore, the reverse must hold for $b^{\overline{\prec}_p^{i,j}}(p^i) - b^{\overline{\prec}}(p^i)$. Hence $b^{\overline{\prec}_p^{i,j}} - b^{\overline{\prec}} = (\chi_p^j - \chi_p^i)(aL + b)$ $\square$

A.6 Proof of Lemma 7

Lemma. *Consider an elementary invalid extreme base $b^{\tilde{\prec}^i_p}$, obtained by swapping two adjacent elements (p^{i+1}, p^i) in the universal ordering, $\prec_0$ (Def. 2). Then:*

$$b^{\tilde{\prec}^i_p} - b^{\prec_0} = (\chi^i_p - \chi^{i+1}_p)(L + b),$$

where $b^{\prec_0}$ is the valid extreme base corresponding to $\prec_0$.

Proof. Recall:

- The universal ordered sequence, $\prec_0$, which is a valid ordering, and also defines a particular ordering among the pixels.
- The elementary invalid ordering, $\tilde{\prec}$, which is defined as the ordering obtained by making one swap between adjacent elements of a valid ordering. The corresponding extreme base is denoted as $b^{\tilde{\prec}}$.

Further, recall from Section A.5 where while proving Lemma 6, we showed that: $b^{\overline{\prec}^{i,j}_p} - b^{\overline{\prec}} = (\chi^j_p - \chi^i_p)(aL + b)$, such that $a = |\overline{S \cup \{p^i\}}| - |\overline{S}|$ (if $i < j$), or $a = |\overline{S \cup \{p^j\}}| - |\overline{S}|$ (if $j < i$). Now consider an elementary invalid extreme base $b^{\tilde{\prec}^i_p}$, obtained by swapping two adjacent elements (p^{i+1}, p^i) in the universal ordering. The term $(\chi^i_p - \chi^{i+1}_p)$ may be looked upon as corresponding to the creation of the elementary extreme base $b^{\tilde{\prec}^i_p}$ from $b^{\prec_0}$. It is easy to see that for such special elementary invalid extreme bases created from universal ordering, $a = 1$, and we have:

$$b^{\tilde{\prec}^i_p} - b^{\prec_0} = (\chi^i_p - \chi^{i+1}_p)(L + b) \tag{9}$$

Hence, proved. □

A.7 Proof of Lemma 8

Lemma. *An invalid canonical extreme base, $b^{\overline{\prec}}$, can be represented as a linear combination of elementary invalid extreme base vectors such that:*

$$b^{\overline{\prec}} = \sum_{p \in \mathcal{P}} \sum_{i=1}^{m-1} \alpha^i_p b^{\tilde{\prec}^i_p} + \Lambda,$$

where $0 < \alpha_p^i \ll L$*, and* Λ *is a vector with all its elements much smaller than* L*.*

Proof. Consider the canonical invalid ordering $\overline{\prec}$ and let $\prec_s$ be the starting canonical valid ordering from which it can be obtained by a series of swaps between adjacent elements. Note that since in the canonical ordering all the elements of a pixel are already together, therefore all the swaps required are between elements corresponding to same pixels. Let us assume that total number of such swaps required are T. Starting from $\prec_s$, let $\prec_j$ represents the ordering obtained after j such swaps. Hence, $\prec_T = \overline{\prec}$ by definition. Let j^{th} swap happens between elements p^{k_j} and p^{l_j}, where $p \in \mathcal{P}$.

$$\begin{aligned}
b^{\overline{\prec}} - b^{\prec_s} &= b^{\prec_T} - b^{\prec_s} \\
&= \sum_{j=1}^{T} \left(b^{\prec_j} - b^{\prec_{j-1}} \right) \\
&= \sum_{j=1}^{T} \left(\chi_p^{l_j} - \chi_p^{k_j} \right)(a_j L + b_j) && \text{(Using Lemma 6)} \\
&= \sum_{j=1}^{T} (\chi_p^{l_j} - \chi_p^{k_j}) a_j L + \sum_{j=1}^{T} (\chi_p^{l_j} - \chi_p^{k_j}) b_j .
\end{aligned}$$

Since $(\chi_p^{l_j} - \chi_p^{k_j}) = \sum_{i=l_j}^{k_j} (\chi_p^i - \chi_p^{i+1})$ we can write:

$$b^{\overline{\prec}} - b^{\prec_s} = \sum_{j=1}^{T} a_j \sum_{i=l_j}^{k_j} (\chi_p^i - \chi_p^{i+1}) L + \sum_{j=1}^{T} b_j \sum_{i=l_j}^{k_j} (\chi_p^i - \chi_p^{i+1}) \tag{10}$$

Recall from Lemma 7:

$$\begin{aligned}
& b^{\widetilde{\prec}_p^i} - b^{\prec_0} = (\chi_p^j - \chi_p^{i+1})(L + b) \\
\Rightarrow \quad & (\chi_p^i - \chi_p^{i+1}) L = b^{\widetilde{\prec}_p^i} - b^{\prec_0} - (\chi_p^i - \chi_p^{i+1}) b_p^i . && \text{(where } b_p^i \ll L\text{)}
\end{aligned}$$

Substituting the value of $(\chi_p^i - \chi_p^{i+1})L$ in Eq. (10), we get:

$$b^{\overline{\prec}} - b^{\prec_s} = \sum_{j=1}^{T} a_j \sum_{i=l_j}^{k_j} (b^{\widetilde{\prec}_p^i} - b^{\prec_0} - (\chi_p^i - \chi_p^{i+1}) b_p^i) + \sum_{j=1}^{T} b_j \sum_{i=l_j}^{k_j} (\chi_p^i - \chi_p^{i+1})$$

Since both $\prec_s$, and $\prec_0$ are valid orderings, we can write $b^{\prec_s} = b^{\prec_0} + \vec{d}$, where elements of $\vec{d}$ are much smaller than L. Therefore we get

$$\begin{aligned} b^{\overline{\prec}} = &\sum_{j=1}^{T}\sum_{i=l_j}^{k_j} a_j b^{\tilde{\prec}^i_p} + \Big(1 - \sum_{j=1}^{T}\sum_{i=l_j}^{k_j} a_j\Big) b^{\prec_0} - \sum_{j=1}^{T}\sum_{i=l_j}^{k_j} a_j(\chi^i_p - \chi^{i+1}_p) b^i_p \\ &+ \sum_{j=1}^{T}\sum_{i=l_j}^{k_j} (\chi^i_p - \chi^{i+1}_p) b_j + \vec{d} \end{aligned} \tag{11}$$

$$b^{\overline{\prec}} = \sum_{j=1}^{T}\sum_{i=l_j}^{k_j} a_j b^{\tilde{\prec}^i_p} + \Lambda, \tag{12}$$

where Equation (12) has been derived summing the last 4 terms into a vector Λ. Note that all the elements of Λ are $\ll L$. It is easy to see that the first term in the equation essentially is a linear combination of some elementary invalid extreme bases, allowing us to simplify:

$$b^{\overline{\prec}} = \sum_{p\in P}\sum_{i=1}^{m-1} \alpha^i_p b^{\tilde{\prec}^i_p} + \Lambda, \tag{13}$$

where coefficients α^i_p corresponding to elementary extreme bases not present in Equation (12) can be simply set to zero. □

A.8 Proof of Lemma 9

Theorem. *An invalid extreme base can be represented as* $b^{\prec} = \sum_{p\in\mathcal{P}}\sum_{i=1}^{m-1} \alpha^i_p L(\chi^i_p - \chi^{i+1}_p) + \Lambda$*, where* χ^i_p *is an indicator vector corresponding to element* p^i*,* $0 < \alpha^i_p \ll L$*, and* Λ *is some vector whose all elements are* $\ll L$*.*

Proof. Using Equation (13), we have:

$$b^{\overline{\prec}} = \sum_{p\in P}\sum_{i=1}^{m-1} \alpha^i_p b^{\tilde{\prec}^i_p} + \Lambda.$$

Substituting representation of elementary extreme base from Equation (where $b_p^i \ll L$), we have:

$$\begin{aligned} b^{\prec} &= \sum_{p\in P}\sum_{i=1}^{m-1} \alpha_p^i \Big(b^{\prec_0} + (\chi_p^i - \chi_p^{i+1})(L + b_p^i)\Big) + \Lambda. \\ &= \sum_{p\in P}\sum_{i=1}^{m-1} \alpha_p^i b^{\prec_0} + \sum_{p\in P}\sum_{i=1}^{m-1} \alpha_p^i L(\chi_p^i - \chi_p^{i+1}) + \sum_{p\in P}\sum_{i=1}^{m-1} (\chi_p^i - \chi_p^{i+1})\alpha_p^i b_p^i + \Lambda. \\ &= \sum_{p\in P}\sum_{i=1}^{m-1} \alpha_p^i L(\chi_p^i - \chi_p^{i+1}) + \Lambda. \end{aligned}$$

Note that we have replaced Λ with $\sum_{p\in P}\sum_{i=1}^{m-1} \alpha_p^i b^{\prec_0} + \sum_{p\in P}\sum_{i=1}^{m-1} (\chi_p^i - \chi_p^{i+1})\alpha_p^i b_p^i + \Lambda$. □

A.9 Proof of Theorem 2

Theorem (Main Result)**.**

$$\sum_{\forall b^{\prec_i}\in Q} \lambda_i b^{\prec_i} = \sum_{p\in\mathcal{P}}\sum_{k=1}^{m-1} \beta_p^k L(\chi_p^k - \chi_p^{k+1}),$$

where $\lambda_i \geq 0$, $\beta_p^k = \sum_{b_i\in Q} \alpha_p^k \lambda_i$.

Consider the expansion of the term $x_i = \sum_{b^{\prec_i}\in Q} \lambda_i b^{\prec_i}$ in Eq.(5). Using Theorem (9) we get:

$$x_i = \sum_{b^{\prec_i}\in Q}\sum_{p\in\mathcal{P}}\sum_{k=1}^{m-1} \lambda_i \alpha_p^k L(\chi_p^k - \chi_p^{k+1}) + \sum_{b^{\prec_i}\in Q} \lambda_i \Lambda_i.$$

Recall from Lemma (4) that, for all $b^{\prec_i} \in Q$, the coefficient λ_i can be made arbitrarily small. Therefore, we can drop the term $\sum_{b^{\prec_i}\in Q} \lambda_i \Lambda_i$ and rewrite the above equation as:

$$x_i = \sum_{p\in P}\sum_{k=1}^{m-1}\sum_{b^{\prec_i}\in Q} \lambda_i \alpha_p^k L(\chi_p^k - \chi_p^{k+1}).$$

Replacing by $\beta_p^k = \sum_{b^{\prec_i} \in Q} \lambda_i \alpha_p^k$, we get:

$$x_i = \sum_{p \in P} \sum_{k=1}^{m-1} \beta_p^k L(\chi_p^k - \chi_p^{k+1}).$$

B Example for validating importance of invalid extreme bases

In this section we show that invalid extreme bases contribute to the representation of optimal vector x^*. We consider here small problem with only 2 pixels (p and q) with 3 labels. We consider the unary cost for labeling pixel p as [0, 1, -100]. Similarly assume unary cost for q as [0, -100, 200]. The clique potential is absolute difference between labels and $L = 1000$. It may be noted that in the proposed encoding we showed conceptually that label for a pixel could be encoded using m binary elements. However, notice that the state of the last element corresponding to each encoding is always 1. Therefore, implementation-wise, one can encode label at each pixel using $m - 1$ binary elements only, with the assumption that elements $p^m, \forall p \in \mathcal{P}$ have their labeling set to $1 : p^m = 1, \forall p \in \mathcal{P}$. Hence, in this section we work with the extreme bases of dimension 4, which is corresponding to 2 pixels and 2 binary elements (p^1, and p^2, and no p^3) per pixel only. In the following subsection we first compute the optimal minimizer x^* using all valid and invalid extreme bases.

B.1 Using All Valid and Invalid Extreme Bases

There are $4! = 24$, extreme bases for the example problem. We list below, all valid and invalid extreme base vectors:

The values of corresponding lambda obtained for the optimal minimum norm point (x^*) in the convex hull of all the extreme points are as given below:

$$(0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0.0250,\ 0.0250,\ 0.949,\ 0,\ 0,\ 0).$$

Observe that, as described in the main paper, the contribution of invalid extreme bases in the base vector is still finite and not dependent upon the L. Corresponding to the above convex combination, we get the optimal base vector as $x^* = (-50, -50, 299, -99)$.

$(902, -1000, 1198, -1000)$	$(902, -1000, 299, -101)$	$(902, -1000, 1198, -1000)$
$(902, -1000, 1198, -1000)$	$(902, -1000, 299, -101)$	$(902, -1000, 299, -101)$
$(-100, 2, 1198, -1000)$	$(-100, 2, 299, -101)$	$(-102, 2, 1200, -1000)$
$(-102, 2, 1200, -1000)$	$(-100, 2, 299, -101)$	$(-102, 2, 301, -101)$
$(898, -1000, 1202, -1000)$	$(898, -1000, 1202, -1000)$	$(-102, 0, 1202, -1000)$
$(-102, 0, 1202, -1000)$	$(898, -1000, 1202, -1000)$	$(-102, 0, 1202, -1000)$
$(900, -1000, 299, -99)$	$(900, -1000, 299, -99)$	$(-100, 0, 299, -99)$
$(-102, 0, 301, -99)$	$(898, -1000, 301, -99)$	$(-102, 0, 301, -99)$

Below, we show that $x*$ can not be represented as the convex combination of valid extreme bases only.

B.2 Considering Only Valid Extreme Points

The 6 valid extreme base vectors corresponding to the example problem are given below:

$(-100, 2, 299, -101)$	$(-100, 2, 299, -101)$	$(-102, 2, 301, -101)$
$(-100, 0, 299, -99)$	$(-102, 0, 301, -99)$	$(-102, 0, 301, -99)$

Note that the first two elements of x^* are -50 and -50 which can never be represented as the convex combination of first two elements of only valid extreme points.

C Convergence of SoSMNP [46]

Our focus initially is to show the convergence to the optimal solution by the MNP algorithm running in the block co-ordinate descent mode as in [46]. The problem formally is to minimize the function $f(S) = \sum_{\mathbb{c} \in \mathcal{C}} f_{\mathbb{c}}(S \cap \mathbb{c}) \quad S \subseteq \mathcal{V}$, where $f_{\mathbb{c}} : 2^{|\mathbb{c}|} \to \mathcal{R}$ is a submodular function. It has been shown in [46] that f can be minimized by finding a point $x \in B(f)$ with the minimum ℓ_2-norm $\|x\|^2$. We write x as the sum $x = \sum_{\mathbb{c} \in \mathcal{C}} y_{\mathbb{c}}$ where $y_{\mathbb{c}} \in B(f_{\mathbb{c}})$.

We assume that a block corresponds to a clique in $\mathcal{C}$. Let $x_{\mathbb{c}}$ be the restriction of x to the elements in $\mathbb{c} \in \mathcal{C}$, and let $x_{\not{\mathbb{c}}}$ be the restriction of x on the remaining elements. We can write $\|x\|^2 = \|x_{\mathbb{c}}\|^2 + \|x_{\not{\mathbb{c}}}\|^2$. The block co-ordinate descent algorithm in [46] minimizes $\|x_{\mathbb{c}}\|^2$ using MNP over all the cliques $\mathbb{c} \in \mathcal{C}$ cyclically. This norm minimization step can be viewed as MNP minimizing $f'_{\mathbb{c}}(S) = f_{\mathbb{c}}(S) + a_{\mathbb{c}}(S), \forall S \subseteq \mathbb{c}$ where $a_{\mathbb{c}} = x_c - y_c$, is a denoting the contribution of the other cliques which remains constant while running MNP over this clique/block. Note that $a_{\mathbb{c}}(S) = \sum_{e \in S} a_{\mathbb{c}}(e)$, and we can equivalently treat $a_{\mathbb{c}}$ as a modular function as well. Let $f'_{\mathbb{c}}(S) = f_{\mathbb{c}}(S) + a_{\mathbb{c}}(S), \forall S \subseteq \mathbb{c}$. Note that the f' as shown above is a sum of submodular (f), and a modular function ($a_{\mathbb{c}}$). Therefore, f' is submodular. It is easy to show the following result:

Lemma 12. *Let $q_{\mathbb{c}}$ be a extreme base vector in $B(f_{\mathbb{c}})$ corresponding to an ordering $\prec_{\mathbb{c}}$. Then the vector $q_{\mathbb{c}} + a_{\mathbb{c}}$ is an extreme base of $B(f'_{\mathbb{c}})$ corresponding to the same ordering $\prec_{\mathbb{c}}$.*

Proof. We can calculate the elements in extreme base vector ($q'_{\mathbb{c}} \in B(f')$) corresponding to ordering $\prec_{\mathbb{c}}$ by Edmond's Greedy Algorithm,

$$\begin{aligned}
q'(e) &= f'(S_e \cup e) - f'(S_e), \qquad (S_e \text{ is the set of elements before } e \in \mathbb{c} \text{ in } \prec_{\mathbb{c}}.)\\
&= f(S_e \cup e) + a_{\mathbb{c}}(S_e \cup e) - (f(S_e) + a_{\mathbb{c}}(S_e)),\\
&= f(S_e \cup e) + a_{\mathbb{c}}(S_e) + a_{\mathbb{c}}(e) - (f(S_e) + a_{\mathbb{c}}(S_e)),\\
&\qquad\qquad (a_{\mathbb{c}} \text{ can be seen as a modular function.})\\
&= f(S_e \cup e) - f(S_e) + a_{\mathbb{c}}(e),\\
&= q_{\mathbb{c}}(e) + a_{\mathbb{c}}(e). \qquad (\text{By Edmond's Greedy Algorithm.})
\end{aligned}$$

Hence, $q'_{\mathbb{c}} = q_{\mathbb{c}} + a_{\mathbb{c}}$. □

It is easy to see that:

$$\begin{aligned} x_{\mathbb{c}} = y_{\mathbb{c}} + a_{\mathbb{c}} &= \sum_i \lambda_i q_{\mathbb{c}} + a_{\mathbb{c}} && \text{(where } \textstyle\sum_i \lambda_i = 1\text{, and } \lambda_i \text{ ¿ } 0\text{)} \\ &= \sum_i \lambda_i q_{\mathbb{c}} + \sum_i \lambda_i a_{\mathbb{c}} && \text{(Since } \textstyle\sum_i \lambda_i = 1\text{)} \\ &= \sum_i \lambda_i (q_{\mathbb{c}} + a_{\mathbb{c}}) = \sum_i \lambda_i q'_{\mathbb{c}} \end{aligned}$$

Hence, $x_{\mathbb{c}}$ is a base vector of f'. Therefore, minimizing the minimum norm over a block, the way SoSMNP does it, can be seen as minimizing the norm of $x_{\mathbb{c}}$: the restriction of x over the elements of clique $\mathbb{c}$ (and not $y_{\mathbb{c}}$). Let us suppose, we have reached a situation where the SoSMNP performs minimization over all blocks (cliques), and no change was observed in any of the blocks. The following lemma establishes the relationship between the extreme base of $f_{\mathbb{c}}$, and the one corresponding to f.

Lemma 13. *Let* $q_{\mathbb{c}} = \arg\min_{q \in B(f_{\mathbb{c}})} x_{\mathbb{c}}^T q,\ \forall \mathbb{c} \in \mathcal{C}$. *Then* $b = \sum_{\mathbb{c} \in \mathcal{C}} q_{\mathbb{c}}$ *also satisfies* $b = \arg\min_{b \in B(f)} x^T b$.

Proof. In the SoSMNP algorithm, the extreme base $q_{\mathbb{c}}$ is generated using Edmond's Greedy Algorithm [43] on the order $\prec_{\mathbb{c}}$ of the indices obtained by sorting the elements of $x_{\mathbb{c}}$ in the increasing order. We represent the extreme base so obtained by $q_{\mathbb{c}}^{\prec_{\mathbb{c}}}$. The SoSMNP algorithm for a block terminates when $x_{\mathbb{c}}^T x_{\mathbb{c}} = x_{\mathbb{c}}^T (q_{\mathbb{c}}^{\prec_{\mathbb{c}}} + a_{\mathbb{c}})$

Consider the termination situation of SoSMNP for the overall problem (comprising of all the cliques). In such a case the algorithms tries to minimize for all the blocks/cliques and no change is found on any of the cliques. Therefore, termination condition of each block is met, and $q_{\mathbb{c}}^{\prec_{\mathbb{c}}} = \arg\min_{q \in B(f_{\mathbb{c}})} x_{\mathbb{c}}^T q$.

Let $\prec_f$ be the ordering of elements of x in the increasing order. It is easy to see that the ordering over x and $x_{\mathbb{c}}$ will be consistent with each other, in the sense that $x(e_1) \prec_f x(e_2) \Rightarrow x_{\mathbb{c}}(e_1) \prec_{\mathbb{c}} x_{\mathbb{c}}(e_2)$.

Let us create an extreme base of f, corresponding to the ordering $\prec_f$, and denote as $b^{\prec_f}$. Since $\prec_f$ denotes the ordering over elements of x, therefore, from Edmond's algorithm, we have: $b^{\prec_f} = \arg\min_{b \in B(f)} x^T b$. Further, we also have:

$$b^{\prec_f}(e) = f(S_e \cup e) - f(S_e),$$

(As per Edmond's algorithm. S_e is the set of elements before e in $\prec_f$)

$$= \sum_{\mathbb{c} \in \mathcal{C}} f_{\mathbb{c}}(S_e \cup e \cap \mathbb{c}) - f(S_e \cap \mathbb{c}), \qquad \text{(Since } f(S) = \textstyle\sum_{\mathbb{c} \in \mathcal{C}} f_{\mathbb{c}}(S \cap \mathbb{c})\text{)}$$

$$= \sum_{\mathbb{c} \in \mathcal{C}} q_{\mathbb{c}}^{\prec_{\mathbb{c}}}(e \cap \mathbb{c}). \qquad \text{(Since } \prec_c \text{ is the restriction of } \prec\text{)}$$

Since above holds for all the elements $e \in \mathcal{V}$, therefore:

$$b^{\prec_f} = \sum_{\mathbb{c} \in \mathcal{C}} q_{\mathbb{c}}^{\prec_{\mathbb{c}}}.$$

Hence, we have proved both the properties of $b^{\prec_f}$ $\square$

We can now give the convergence proof of the SoSMNP with the following lemma:

Lemma 14. *If in a complete cycle of SoSMNP over all the cliques, we can not improve the norm $x_{\mathbb{c}}$ for any $\mathbb{c}$, then we have $x \in B(f)$ such that $\|x\|^2 = x^T x = x^T b$, where $b = \arg\min_{b \in B(f)} x^T b$.*

Proof. Recall that for a clique $\mathbb{c}$, SoSMNP can be seen as minimizing the norm of $x_{\mathbb{c}}$ which is a base vector of $f'_{\mathbb{c}} = f_{\mathbb{c}} + a_{\mathbb{c}}$. Further $q'_{\mathbb{c}} = q_{\mathbb{c}} + a_{\mathbb{c}}$ is an extreme base of f'. Therefore, from the termination of basic MNP algorithm, the following must hold:

$$x_{\mathbb{c}}^T x_{\mathbb{c}} = x_{\mathbb{c}}^T (q_{\mathbb{c}} + a_{\mathbb{c}}). \qquad (q_{\mathbb{c}} = \arg\min_{q \in B(f_{\mathbb{c}})} x_{\mathbb{c}}^T q)$$

Summing over all the cliques we get

$$\sum_{\mathbb{c} \in \mathcal{C}} x_{\mathbb{c}}^T x_{\mathbb{c}} = \sum_{\mathbb{c} \in \mathcal{C}} x_{\mathbb{c}}^T (q_{\mathbb{c}} + a_{\mathbb{c}}),$$

$$\sum_{\mathbb{c} \in \mathcal{C}} x_{\mathbb{c}}^T (y_{\mathbb{c}} + a_{\mathbb{c}}) = \sum_{\mathbb{c} \in \mathcal{C}} x_{\mathbb{c}}^T (q_{\mathbb{c}} + a_{\mathbb{c}}),$$

$$\sum_{\mathbb{c} \in \mathcal{C}} x_{\mathbb{c}}^T y_{\mathbb{c}} = \sum_{\mathbb{c} \in \mathcal{C}} x_{\mathbb{c}}^T q_{\mathbb{c}}. \qquad (\textstyle\sum_{\mathbb{c} \in \mathcal{C}} x_{\mathbb{c}}^T a_{\mathbb{c}} \text{ cancels out})$$

Since vector $y_{\mathbb{c}}$ and $q_{\mathbb{c}}$ have non-zero values only for elements in $\mathbb{c}$. Therefore we can write $x_{\mathbb{c}}^T y_{\mathbb{c}} = x^T y_{\mathbb{c}}$ and $x_{\mathbb{c}}^T q_{\mathbb{c}} = x^T q_{\mathbb{c}}$. Substituting the values, we get:

$$\sum_{\mathbb{c} \in \mathcal{C}} x^T y_{\mathbb{c}} = \sum_{\mathbb{c} \in \mathcal{C}} x^T q_{\mathbb{c}},$$
$$x^T \sum_{\mathbb{c} \in \mathcal{C}} y_{\mathbb{c}} = x^T \sum_{\mathbb{c} \in \mathcal{C}} q_{\mathbb{c}}$$
$$x^T x = x^T b \qquad \text{(where } b = \arg\min_{b \in B(f)} x^T b\text{, by Lemma 13)}$$

The equation above is the termination condition of basic MNP when run over the overall function f [5]. Therefore, the lemma essentially proves that the basic MNP terminating with optimal solutions for all cliques/blocks implies that the the base vector obtained by summing up the base vectors of all the cliques/blocks is the optimal solution for the overall objective function. □

When MNP algorithm is run in the block co-ordinate descent mode it is easy to show that any decrease in the $\|x_c\|^2$ of a clique decreases the over all $\|x\|^2$ by the same amount because $\|x_{\not{\mathbb{c}}}\|$ is untouched when optimizing for $\mathbb{c}$. Since at each cycle there is at least one clique for which $\|x_c\|^2$ decreases, we can say that $\|x\|^2$ decreases monotonically at each cycle. Note that Theorem 4 of [5] gives us a lower bound on the improvement in every MNP iteration. It follows that MNP algorithm running in block co-ordinate descent mode will have a provable rate of convergence.

For the sake of completeness we will also like to point out that the optimal solution obtained when MNP is run globally also corresponds to the individual blocks having reached their local optima.

D Convergence of ML-hybrid Algorithm

Note that in SoSMNP each block is optimized using the MNP algorithm. In MLhybrid, on the other hand, each block is further subdivided. One corresponds to the set of valid extreme bases (the valid block) and the other to the set of invalid extreme bases (the invalid block) whose convex combination defines the base vector $x_{\mathbb{c}}$. MNP is run on the valid block. If at any iteration MNP [13] inserts an invalid extreme base, the flow based

Algorithm 2 is run on the invalid block. We show below that when MNP is run on the valid block now (that is just after a run of the flow based algorithm on the invalid block) the extreme base generated will be valid.

Lemma 15. *Algorithm 2 returns a vector $x_{\mathbb{c}}$ for clique $\mathbb{c}$ such that extreme base q_c given as $q_c = \arg\min_{q \in B(f_{\mathbb{c}})} x_{\mathbb{c}}^T q$ is valid.*

Proof. It is easy to show that when Algorithm 2 terminates, for any pair of indices i, j corresponding to any $p \in \mathcal{P}$ if $i > j$ then $e(p^i) \leq e(p^j)$. Note that by construction the excess vector e is the base vector $x_{\mathbb{c}}$. This implies that the order $\prec_{\mathbb{c}}$ of the indices obtained by sorting the elements of $x_{\mathbb{c}}$ will satisfy $p^i \prec_{\mathbb{c}} p^j, \forall i > j$, and $\forall p \in \mathcal{P}$. This is the condition that has to be satisfied for an ordering to be valid (Cf. Def. 3). Recall that in the MNP algorithm the extreme base is found by computing the ordering of sorted elements of x. Hence, the extreme base $q_c = \arg\min_{q \in B(f_{\mathbb{c}})} x_{\mathbb{c}}^T q$ will be a valid one. $\square$

Lemma 15 implies that an iteration on the invalid block will be followed by the MNP algorithm making progress in the form of generation of a valid extreme base. Also note that the ℓ_2 norm decreases when the flow based algorithm is run on the invalid block. Therefore, termination and convergence of the MLhybrid algorithm running on a clique/block follows along the same lines as that for the standard MNP algorithm [5].

Now we show that termination over a clique/block results in $x_{\mathbb{c}}$ using which minimizer obtained comes on a valid state.

Note that generation of an invalid extreme base can always be followed by generation of a valid extreme base (by running the flow based algorithm on the invalid block). Therefore, at termination it is guaranteed that the order $\prec_{\mathbb{c}}$ of the indices obtained by sorting the elements of $x_{\mathbb{c}}$ is valid. That is the optimal solution corresponds to a valid primal state. Hence, it follows, using Lemma 14, that the MLHybrid algorithm run in the block coordinate descent manner converges to the optimal.

Algorithm 2 ComputeInvalidContribution

```
Input: Vector $x_{\mathbb{c}}$ the output of the max flow algorithm.
Output: The transformed vector $x_{\mathbb{c}}$ with minimum $\ell_2$ norm .
1: for $\forall p \in \mathbb{c}$ do
2:     for $i = 2 : m$ do
3:         repeat
4:             find smallest $k$, $i \geq k \geq 1$, such that
               $x_{\mathbb{c}}(p^i) > x_{\mathbb{c}}(p^{i-1}) = x_{\mathbb{c}}(p^{i-2}) \cdots = x_{\mathbb{c}}(p^k)$ or $x_{\mathbb{c}}(p^i) = x_{\mathbb{c}}(p^{i-1}) = x_{\mathbb{c}}(p^{i-2}) \cdots = x_{\mathbb{c}}(p^{k+1}) > x_{\mathbb{c}}(p^k)$;
5:             let $av_k$ be the average of $x_{\mathbb{c}}(p^i), x_{\mathbb{c}}(p^{i-1}), \ldots, x_{\mathbb{c}}(p^k)$;
6:             set $x_{\mathbb{c}}(p^i), x_{\mathbb{c}}(p^{i-1}), \ldots, x_{\mathbb{c}}(p^k)$ equal to $av_k$;
7:         until $x_{\mathbb{c}}(p^{k+1}) \leq x_{\mathbb{c}}(p^k)$
8:     end for
9: end for
```

E Proposed Complete Algorithm

In this section we give the complete proposed method in Algorithm 3. The algorithm takes tranformed 2-label submodular clique potentials $f_{\mathbb{c}}$'s and computes minimum ℓ_2 norm of $x \in B(f)$ s.t. $f = \sum_{\mathbb{c} \in \mathcal{C}} f_{\mathbb{c}}$. The overall algorithm solves valid block with the SoS-MNP algorithm given in [46] and uses Algorithm 1 to solve invalid block. Let $x_{\mathbb{c}}$ be the restriction of x over clique $\mathbb{c}$, the norm $\|x\|^2$ is optimized by computing minimum norm $\|x_{\mathbb{c}}\|^2$ over each clique cyclically. Algorithm 4 minimizes $\|x_{\mathbb{c}}\|^2$ in a very similar way as MNP Algorithm [46] described in Background section. The only difference lies in handling the invalid extreme base at step 4 of Algorithm 4.

Algorithm 3 HybridML: Algorithm for minimizing a sum of multilabel submodular functions

```
Input: $\{f_{\mathbb{c}}\}$ such that $f = \sum f_{\mathbb{c}}$.
Output: $x = \arg\min \|x\|^2$ subject to $x \in B(f)$.
 # Initialize
 1: for all ($\mathbb{c} \in \mathcal{C}$) do
 2:    $q_{\mathbb{c}} \leftarrow$ Take any extreme base of $f_{\mathbb{c}}$;
 3:    $S_{\mathbb{c}} := \{q_{\mathbb{c}}\}$;
 4:    $y_{\mathbb{c}} := q_{\mathbb{c}}$;
 5: end for
 6: $x := \sum_{\mathbb{c}} y_{\mathbb{c}}$;
 # Perform Block Coordinate Descent with blocks specified by Cliques
 7: while ($\|x\|$ decreases by more than $\delta$) do
 8:    for all ($\mathbb{c} \in \mathcal{C}$) do
 9:       MLHybridOverAClique($f_{\mathbb{c}}$,$S_{\mathbb{c}}$,$x_{\mathbb{c}}$,$y_{\mathbb{c}}$);
10:    end for
11: end while
```

Algorithm 4 MLHybridOverAClique

```
Input: Clique function: $f_{\mathbb{c}}$
Input: Set of valid extreme bases selected in last iteration: $S_{\mathbb{c}}$
Input: Restriction of current solution vector $x$ on $\mathbb{c}$: $x_{\mathbb{c}}$
Input: Current clique vector: $y_{\mathbb{c}}$
Output: Clique vector $y^*_{\mathbb{c}} \in B(f_c)$ minimizing $\|x_{\mathbb{c}}\|^2$
Output: Updated set $S^*_{\mathbb{c}}$ of valid extreme bases

 1: while (TRUE) do
 2:    Find new translation $a_{\mathbb{c}} := x_{\mathbb{c}} - y_{\mathbb{c}}$;
 3:    Find extreme base $\hat{q}_{\mathbb{c}} := \arg\min_{q_{\mathbb{c}} \in B_{f_{\mathbb{c}}}} \langle x_{\mathbb{c}}, q_{\mathbb{c}} \rangle$ using Edmond's algorithm.
 4:    if Extreme base $\hat{q}_{\mathbb{c}}$ is invalid according to Definition 3 then
 5:       ComputeInvalidContribution($x_{\mathbb{c}}$);
 6:       continue;
 7:    end if
 8:    Find translated extreme base $\hat{p}_{\mathbb{c}} = \hat{q}_{\mathbb{c}} + a_{\mathbb{c}}$;
 9:    if ($\|x_{\mathbb{c}}\|^2 \leq \langle x_{\mathbb{c}}, \hat{p} \rangle + \epsilon$) then
10:       break;
11:    end if
12:    $S_{\mathbb{c}} := S_{\mathbb{c}} \cup \hat{q}_{\mathbb{c}}$;
13:    $P_{\mathbb{c}} = \{\hat{q_{\mathbb{c}}} + a_{\mathbb{c}} | q_{\mathbb{c}} \in S_{\mathbb{c}}\}$;
14:    Find $x_{\mathbb{c}}$ in affine hull of $P_{\mathbb{c}}$;
15:    If $x_{\mathbb{c}}$ is not in convex hull $P_{\mathbb{c}}$, translate to nearest point in convex hull and update
       $S_{\mathbb{c}}$;
16: end while
```