Table 1. Table of notation.

Symbol	Definition
V	The ground set
n	Size of the ground set: $n = \|V\|$
2^A	The power set of A
m	Number of blocks that we wish to find
ℓ	Upper bound the size of each block
π	Grouping of V s.t. $A_1^\pi, A_2^\pi, \ldots, A_m^\pi$ are the corresponding blocks
A_i^π	ith block in the grouping π of V
$\mathcal{M} = (V, \mathcal{I})$	A matroid with ground set V and independent sets $\mathcal{I} \subseteq 2^V$
$F(\pi)$	Objective function that evaluates the quality of a grouping π
$F_{i,j}\left(A_i^\pi, A_j^\pi\right)$	A cross-block interaction term between blocks i and j
$\lambda_1, \lambda_2, \lambda_3, \lambda_4$	Weights on each of the terms of the objective function
$V^\times$	Expanded ground set consisting of the disjoint union of m copies of V, i.e. $V^\times = \{(v, i) : v \in V, i \in [m]\}$
$V^{(i)}$	The ith "column" of $V^\times$, i.e. $V^{(i)} = \{(v, i) : v \in V\}$
$R^{(v)}$	The vth "row" of $V^\times$, i.e. $R^{(v)} = \{(v, i) : i \in [m]\}$
$\text{abs}\,(S)$	$= \{v \in V : \exists i \in [m]\,.(v, i) \in S\}$
$\text{col}\,(S, i)$	$= \text{abs}\left(S \cap V^{(i)}\right) = \{v \in V : (v, i) \in S\}$. When using an expanded ground set (i.e. $S \in V^\times$), $\text{col}\,(S, i)$ corresponds to A_i^π
$F^\times(S)$	Objective function on an expanded ground set $V^\times$. When using an expanded ground set (i.e. $S \in V^\times$), $F^\times(S)$ corresponds to $F(\pi)$
$G_{i,j}^\cup(S)$	$= \text{col}\,(S, i) \cup \text{col}\,(S, j)$. When using an expanded ground set (i.e. $S \in V^\times$), $f\left(G_{i,j}^\cup(S)\right)$ corresponds to $F_{i,j}^\cup\left(A_i^\pi, A_j^\pi\right)$
$G_{i,j}^\triangle(S)$	$= \text{col}\,(S, i) \triangle \text{col}\,(S, j)$. When using an expanded ground set (i.e. $S \in V^\times$), $f\left(G_{i,j}^\triangle(S)\right)$ corresponds to $F_{i,j}^\triangle\left(A_i^\pi, A_j^\pi\right)$
$G_{i,j}^\cap(S)$	$= \text{col}\,(S, i) \cap \text{col}\,(S, j)$. When using an expanded ground set (i.e. $S \in V^\times$), $f\left(G_{i,j}^\cap(S)\right)$ corresponds to $F_{i,j}^\cap\left(A_i^\pi, A_j^\pi\right)$
m_f^X	Modular subgradient of f at X

A. Ensemble-of-Lattices Models

Lattice models: Lattices (Gupta et al., 2016) are nonlinear models that are particularly easy to regularize or constrain to have desirable properties (e.g. monotonicity or smoothness), but suffer from having large numbers of parameters. A d-dimensional lattice model is a function $f : [0, 1]^d \to \mathbb{R}$ (without loss of generality, the features are assumed to have been scaled to the range $[0, 1]$). Letting $k \in \mathbb{N}^d$ be the "order" of each dimension (these are hyperparameters), the structure of the model is essentially a d-dimensional grid on $[0, 1]^d$, for which there are k_i grid lines, and $k_i - 1$ cells, along the ith dimension. At each "corner", i.e. an intersection of grid lines, the lattice model has a (real-valued) parameter, yielding a total of $\prod_{i=1}^d k_i$ parameters. To evaluate the lattice on a given example $x \in [0, 1]^d$, one first determines which cell contains the example, and then linearly interpolates the value of the model from the parameters on the corners of the cell. It's easy to observe that such a model can approximate any continuous function arbitrarily well (as long as one chooses a fine enough grid).

Calibration: In a *calibrated* lattice model, each input feature is "calibrated" by passing it through a one-dimensional function before the transformed feature vector is handed on to the lattice. Gupta et al. (2016) propose using piecewise linear calibration functions, which in their simplest form are nothing but fine-grained one-dimensional lattices. If we take $c_i : [0, 1] \to [0, 1]$ to be the ith such function, and define $c : [0, 1]^d \to [0, 1]^d$ such that $(c(x))_i = c_i(x_i)$, then the overall calibrated lattice model is the composition $f \circ c$. Having finer-grained calibrators, and coarser-grained lattices, generally leads to the best trade-off between performance and model complexity, since increasing the resolution of the one-dimensional calibrators (contrasted with increasing the k_is of a d-dimensional lattice), does not result in an exponential growth in the number of model parameters. For this reason, the lattice grid is typically taken to be as small as possible ($k_i = 2$ for all i).

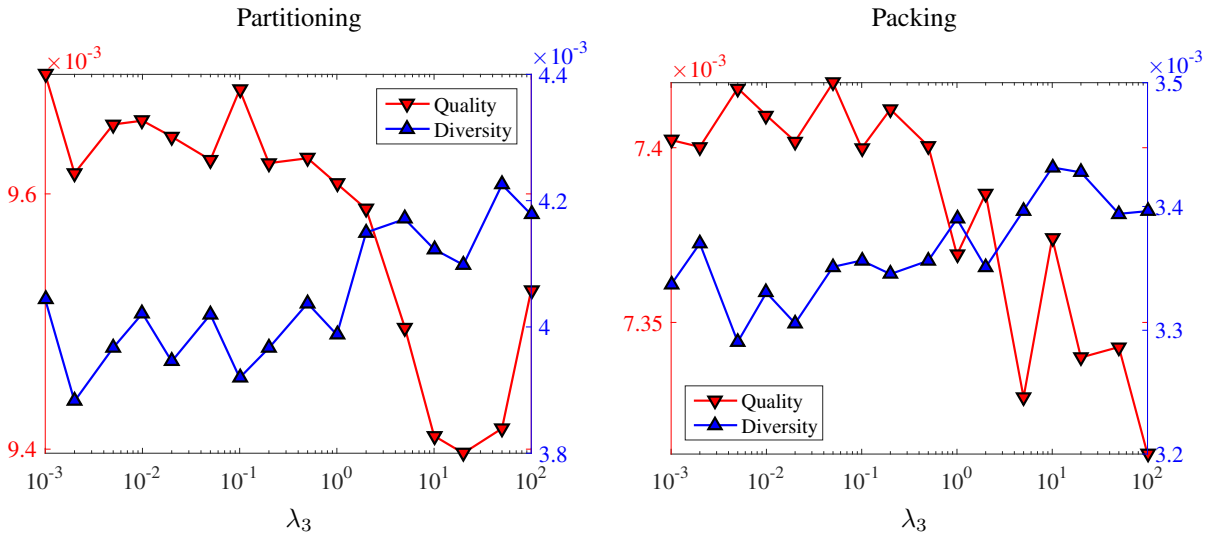

Figure 3. Same as the left-hand plot of Figure 2, but for Eq. (7) on the partitioning (left) and packing (right) problems of Appendix B.

The straightforward structure of calibrated lattice models makes them relatively easy-to-interpret. "Interpretability" is, of course, subjective, but in the case of these particular models it has concrete benefits: it's easy to add constraints or regularizers to control the model's behavior. Imposing monotonicity along a given dimension amounts to simply inserting linear inequality constraints forcing the parameter values of the corresponding calibration function to be monotonic, and likewise for the parameters of the lattice grid along this dimension. Similarly, smoothness regularizers can be imposed by penalizing dissimilarity between the parameters at nearby grid points, or penalizing local deviations from linearity.

The flexibility and ease of constraining and regularizing such models is offset by their large number of parameters: it generally isn't practical to create a lattice on more than, say, 10 to 15 features. For this reason, on higher-dimensional problems, Canini et al. (2016) proposed using *ensembles* of lattices. Such models are averages of several calibrated lattices, each of which acts on some small subset of the features. This architecture inherits many of the desirable properties of calibrated lattice models, but dramatically decreases the number of parameters. This improvement, however, comes the cost of an additional complication: it isn't clear how to most effectively group the features into lattices. Based on the observation that features that interact nonlinearly should belong the same lattice, and those that interact only linearly need not, Canini et al. (2016) proposed a heuristic for grouping features which, in their experiments and ours, worked very well compared to the natural baseline approach of using a random grouping.

As we discussed in Section 6, this ensemble-construction problem fits naturally into our framework:

1. We cannot accept large lattices, since we don't want the ensemble to have too many parameters. This can be formulated as a matroid constraint.

2. We want each lattice in the ensemble to perform well, individually, by including features that interact strongly and nonlinearly. In other words, we want high *intra-block diversity*.

3. We want the lattices to be different from each other, so that the ensemble *as a whole* performs well. In other words, we want high *inter-block diversity*.

All three of these properties are likely to be present, at least in some form, for ensemble construction *in general*, beyond just lattices.

B. Additional Case Studies

In the same setting as Section 6, we performed two additional case studies (discussed briefly in Section 6.3). There are two cases: (i) partitioning, in which the task is to choose $m = 4$ lattices, each containing up to 8 *distinct* features—again, these

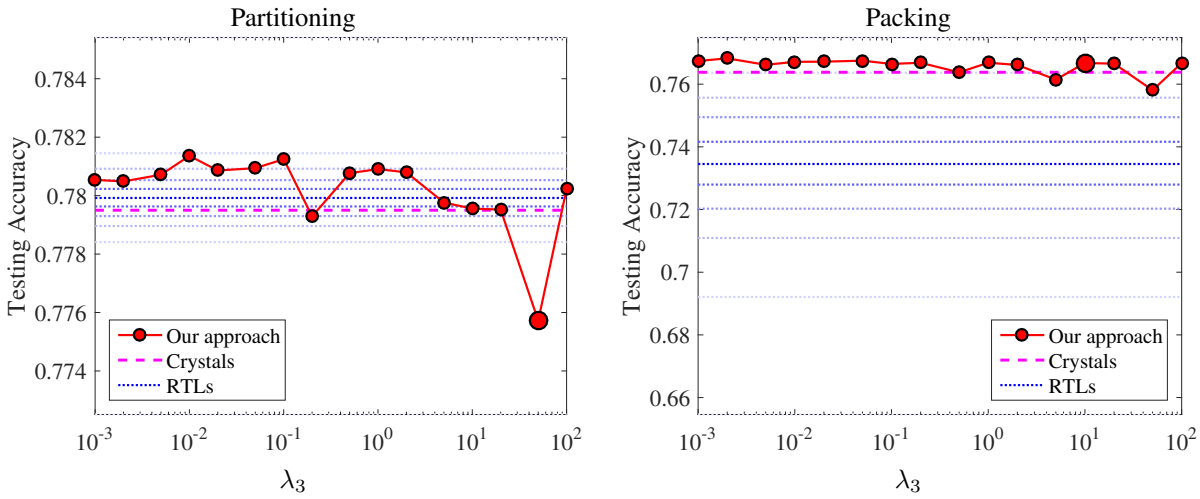

Figure 4. Same as the right-hand plot of Figure 2, but for the partitioning (left) and packing (right) problems of Appendix B.

are the matroid constraints—and (ii) packing, in which the task is to choose $m = 4$ lattices, each containing up to 4 distinct features (recall that there are 29 features in total). In both cases, the goal is to maximize Eq. (7).

We optimized Eq. (7) using Algorithm 1 with the CALLBACK again being the randomized algorithm of Feldman et al. (2017) combined with the procedure of Lemma 4, with $\beta = 0.5$ and δ chosen such that Theorem 1 would hold with probability 0.9.

As in the left-hand plot of Figure 2, we can see from Figure 3 that, as λ_3 increases, the solution's quality term tends to decrease, and diversity term to increase, as expected—this is the good news: the optimization worked.

There is also bad news: in contrast to the results in Section 6.2, the inclusion of a diversity term did not appear to help–and in fact might have *hurt*–the testing accuracies of the ensembles (Figure 4). We believe that the reason for this failure is simply that Eq. (7) rewards the *wrong type of diversity* for this problem. Furthermore, for the partitioning problem, the *highest* validation accuracy corresponds to the *lowest* testing accuracy (although the accuracies cover a very small range).

C. Proofs

Lemma 1. *Let V', V be two ground sets and define a set-to-set mapping function $G : 2^{V'} \to 2^{V}$. Also, let $f : 2^{V} \to \mathbb{R}_+$ be monotone non-decreasing and submodular, and let $g : 2^{V} \to \mathbb{R}_+$ be monotone non-decreasing and supermodular. Then:*

1. *If G is monotone non-decreasing (i.e. $G(S) \subseteq G(T)$ whenever $S \subseteq T$), then $f \circ G$ and $g \circ G$ are both monotone non-decreasing.*
2. *If $\forall S, T \subseteq V'$, $G(S \cup T) = G(S) \cup G(T)$ and $G(S \cap T) \subseteq G(S) \cap G(T)$, then $f \circ G : 2^{V'} \to \mathbb{R}_+$ is submodular.*
3. *If $\forall S, T \subseteq V'$, $G(S \cup T) \supseteq G(S) \cup G(T)$ and $G(S \cap T) = G(S) \cap G(T)$, then $g \circ G : 2^{V'} \to \mathbb{R}_+$ is supermodular.*

Proof. For (1), the monotonicity of $f \circ G$ follows immediately from the monotonicity of f and g.

(2) Let $S, T \subseteq V'$ be given and arbitrary. We have that

$$f(G(S \cup T)) + f(G(S \cap T)) \leq f(G(S) \cup G(T)) + f(G(S) \cap G(T)) \tag{8}$$
$$\leq f(G(S)) + f(G(T)). \tag{9}$$

The first inequality follows from the presumed property of G and the monotonicity of f. The second inequality follows from the submodularity of f. Hence, $f \circ G$ is submodular.

(3) Let $S, T \subseteq V'$ be given and arbitrary. We have that

$$g(G(S \cup T)) + g(G(S \cap T)) \geq g(G(S) \cup G(T)) + g(G(S) \cap G(T)) \tag{10}$$
$$\geq g(G(S)) + g(G(T)). \tag{11}$$

The first inequality follows from the presumed property of G and the monotonicity of g. The second inequality follows from the supermodularity of g. Hence, $g \circ G$ is supermodular. □

Lemma 2. *Let $f : 2^V \to \mathbb{R}$ be monotone non-decreasing submodular, $m : 2^V \to \mathbb{R}$ be non-negative modular, and $g : 2^V \to \mathbb{R}$ be monotone non-decreasing supermodular. Then $f \circ G^{\cup}_{i,j} : 2^{V^\times} \to \mathbb{R}$ is monotone non-decreasing submodular, $g \circ G^{\cap}_{i,j} : 2^{V^\times} \to \mathbb{R}$ is monotone non-decreasing supermodular, and $m \circ G^{\triangle}_{i,j} : 2^{V^\times} \to \mathbb{R}$ is non-negative submodular.*

Proof. From the definition of $G^{\cup}_{i,j}$, for any $S, T \subseteq V^\times$:

$$G^{\cup}_{i,j}(S \cup T) = G^{\cup}_{i,j}(S) \cup G^{\cup}_{i,j}(T) \quad \text{and} \quad G^{\cup}_{i,j}(S \cap T) \subseteq G^{\cup}_{i,j}(S) \cap G^{\cup}_{i,j}(T).$$

We also have that $G^{\cup}_{i,j}$ is monotone non-decreasing. Hence, by Lemma 1, monotone submodularity of $f \circ G^{\cup}_{i,j} : 2^{V^\times} \to \mathbb{R}$ follows. For $G^{\cap}_{i,j}$ we have the inequalities:

$$G^{\cap}_{i,j}(S \cup T) \supseteq G^{\cap}_{i,j}(S) \cup G^{\cap}_{i,j}(T) \quad \text{and} \quad G^{\cap}_{i,j}(S \cap T) = G^{\cap}_{i,j}(S) \cap G^{\cap}_{i,j}(T),$$

and the monotone supermodularity of $g \circ G^{\cap}_{i,j} : 2^{V^\times} \to \mathbb{R}$ follows, again by Lemma 1.

Lastly, we note that:

$$m \circ G^{\triangle}_{i,j}(S) = \sum_{v \in V} m(v)\big(\mathbf{1}_{v \in \mathrm{col}(S,i)} \oplus \mathbf{1}_{v \in \mathrm{col}(S,i)}\big) = \sum_{v \in V} m(v)\big(\mathbf{1}_{(v,i) \in S} \oplus \mathbf{1}_{(v,j) \in S}\big)$$

where $\oplus$ is the xor operator. Hence, $m \circ G^{\triangle}_{i,j}(S)$ can be written as a non-negative weighted sum of xor functions, each of which are submodular, and hence the result is submodular. □

Lemma 3. *Given any algorithm that produces a solution $\hat{S}$ having the property that $F(\hat{S}) + m^X_f \circ G^{\triangle}_{i,j}(\hat{S}) \geq \alpha \max_{S \in \mathcal{F}^\times} \Big(F(S) + m^X_f \circ G^{\triangle}_{i,j}(S)\Big)$ for $\alpha > 0$, then $\hat{S}$ also has the property that $F(\hat{S}) + f \circ G^{\triangle}_{i,j}(\hat{S}) \geq \alpha(1 - c) \max_{S \in \mathcal{F}^\times} \Big(F(S) + f \circ G^{\triangle}_{i,j}(S)\Big) = \alpha(1-c)$**OPT**.*

Proof. Define $F_m(S) \triangleq F(S) + m^X_f \circ G^{\triangle}_{i,j}(S)$ and $F_f(S) \triangleq F(S) + f \circ G^{\triangle}_{i,j}(S)$, and let $S_{\text{mopt}} \in \operatorname{argmax}_{S \in \mathcal{F}^\times} F_m(S)$ and $S_{\text{fopt}} \in \operatorname{argmax}_{S \in \mathcal{F}^\times} F_f(S)$. Then for all $S \in \mathcal{F}^\times$, we have:

$$F_m(\hat{S}) \geq \alpha F_m(S_{\text{mopt}}) \geq \alpha F_m(S_{\text{fopt}}) \geq \alpha(1-c) F_f(S_{\text{fopt}}) \geq \alpha(1-c) F_f(S)$$

which completes the proof. □

Lemma 4. *Let A be a randomized algorithm for submodular maximization that has an α-approximation guarantee* in expectation*, i.e. for which $\mathbb{E}[f(S)] \geq \alpha f(S^*)$, where f is the submodular function we wish to maximize, S is the result of algorithm A, and S^* is the maximizer of f. For parameters $\beta, \delta \in (0,1)$, suppose that we run algorithm A k times, where $k = \left\lceil \left(\ln \frac{1}{\delta}\right) / \left(\ln \frac{1-\alpha\beta}{1-\alpha}\right) \right\rceil$, yielding results $S_1, S_2, \ldots, S_k$. Take $S = \operatorname{argmax}_{S_i : i \in [k]} f(S_i)$ to be the best of these results. Then S will have an approximation ratio of $\alpha\beta$, i.e. $f(S) \geq \alpha\beta f(S^*)$, with probability $1 - \delta$.*

Proof. Take S to be the result of algorithm A, and define $q = \Pr\{f(S) < \alpha\beta f(S^*)\}$ as the probability that S fails to achieve an approximation ratio of $\alpha\beta$. Observe that:

$$\mathbb{E}[f(S)] \leq (1-q) f(S^*) + q\alpha\beta f(S^*)$$

Plugging in the approximation guarantee of algorithm A and dividing through by $f(S^*)$:

$$\alpha \leq (1-q) + q\alpha\beta$$
$$q \leq \frac{1-\alpha}{1-\alpha\beta}$$