To Reviewer #1: *LP-based data selection*: For Fig. 2, we indeed used random sampling to select a subset prior to applying the LP-based data selection on the reduced dataset. This was done because the number of constraints in (7) grows quadratically with the number of samples, N. However, upon further analysis we noticed that the random sampling is unnecessary because the LP in (7) with $f = 0$ and $g = 1$ can in fact be decomposed into N smaller LPs, each with $|\mathcal{N}|$ variables and $d + 2$ constraints. These can be solved in parallel using a LP solver. We implemented this for datasets in Fig. 2 and noticed that the performance of our method is no worse than currently reported, and even better in some cases. We thank the reviewer for pointing out this discrepancy, and will update with new experiments for S1O-DS in Fig. 2 with the LP decomposition-based approach (better results without random subsampling preprocessing).

Cutting planes: Our cutting planes are added once at the beginning before invoking a MILP solver. The cuts can be applied to the other MIP models provided that they use the variables c_i and e_{il} with the same modelling meaning. For example, Fair in [1] can use these cuts, and OCT-H in [4] needs to be modified slightly before they can be applied.

Using data selection for CART, OCT-H (also Reviewer #2): Thank you for your suggestion. Fig. 1 benchmarks the performance of SVM1-ODT on medium-sized datasets and shows that our ODT formulation outperforms CART, OCT-H, Fair and BinOCT in terms of test accuracy *without using data selection*. Fig. 2 addresses scalability by benchmarking our combined approach (SVM1-ODT and data selection) with CART and other methods such as OCT-H that address scalability. While it is certainly possible to apply our data selection procedure prior to using either CART or OCT-H, our intent in Fig. 2 is largely to benchmark the *scalability* of the combined approach.

Clarification on Eq. (6): In Sec. 3, we first present the generalized data selection formulation that maximizes the points in the convex hull ($g^T b$) and minimizes the number of selected data points ($f^T a$). In Sec. 3.1, we present a special case of (6) with $f = 0$ and $g = 1$ because choosing these values allows us to decompose it into N smaller LPs while maximizing the points inside the convex hull. The balanced schema is presented in the Supp. without numerical results.

Optimality gap: For datasets that cannot be solved to optimality within 15 minutes, we observed that the optimality gap could still be large due to the lower bound not improving significantly before the run is terminated. The gap does not indicate time to reach optimality. Thus, we instead use training accuracy to benchmark the performance of our model.

Notation in Section 3.2: J_N is a subset of extreme points that satisfy a specific condition. $K_{\mathcal{N}}$ contains extreme points, except they are never used with convex combination coefficient larger than $\frac{1}{d+1}$ to express some other points. Since points in $K_{\mathcal{N}}$ carry some information about the distribution of the dataset, we select a subset of $K_{\mathcal{N}}$, using Alg. 1, for training. "Boundary hyperplanes" refers to the hyperplanes for each leaf node. dist(a,h) refers to the distance between a to the hyperplane $h^T x = 0$. The term "heuristic" refers to Alg. 1. The LP in (7) is solved in the second step of Alg. 1.

Theorem 2: Our idea is to disperse the numerical issues between parameters by re-scaling. The numerical instability for a very small ϵ should be easier to handle by an MILP solver than a very big M. *Page 2,line 85*: From (1d,1e), it follows that w_{il} equals to some u_l which is integral. Then (1c) implies that at the optimal solution $\hat{y}_i$ is also integral. So we do not explicitly enforce $\hat{y}_i$ to be integral. *CART with deeper trees*: We follow the standard benchmarking used in [1,4] wherein the tree depth for CART and MIPs are the same. Note that a deeper tree for CART is not necessary to outperform the shallow ODT; e.g., for `Dermatology` the ODT with $D = 2$ has a higher test accuracy (80.7%) compared to the CART tree with $D = 3$ (76.1%) As also observed in [11], for many datasets, even we allow CART to choose the tree depth using its default setting (allowing deep trees), a shallow ODT still outperforms CART (see Tab.13 in [11]).

We have incorporated all results in Thm. 1 and 2 into the final model, the final SVM1-ODT model imposes $u_l \in [1, Y]$ and $M = 1$. Model parameters ϵ and α_i are tuned via cross-validation for each dataset. We will fix the terminology issues and revise discussions related to "sparsity" for hyperplane, "linear" for the 1-norm, and other suggestions.

To Reviewer #2: If McCormick linearizations are not used, then there are nonconvex quadratic terms in the constraints. The tractability of the resulting MIQCP is very challenging. CPLEX only supports convex quadratic terms that can be represented as second order cone programs (see shorturl.at/cnDY6). Note that handling logical constraints is not as scalable as the big-M method if we can choose a small value for M. As shown in Theorem 2, we can use $M = 1$.

To Reviewer #3: The requirements for input features and branching rules from [2] and [11] are different from ours. In particular, [2] and [11] only consider binary features $\mathbf{x}_i \in \{0, 1\}^d$, while our formulation takes numerical features $\mathbf{x}_i \in \mathbb{R}^d$ (and it can be extended to the case with mixed numerical and categorical features, see A.2). Branching rules in these references are binary tests (i.e., univariate splits), while we use a multivariate hyperplane. Hence the authors employ special tools for the specific decision tree, they cannot be generalized to our general tree; for example, [2] proposes a max flow-based MIP formulation. Comparing to [4], we impose additional conditions on hyperplanes so that training samples should be far from the boundary of the cluster at the leaf node by using the multi-hyperplane SVM. Moreover, we use lesser number of binary variables to encode the decision tree ($e_{il}, c_i \in \{0, 1\}$ in our model v.s. $z_{it}, s_{jt}, l_t, d_t \in \{0, 1\}$ in [4]), and we can use a small value for the big-M parameter (i.e., $M = 1$).

We did compare with OCT-H from [4] in Figs 1, 2. The model in [2] handles binary input, and 8 data sets were tested. They share 4 datasets with us (Tab. 2 in [2] and Tab. 5 in our Supp.). For same tree depth, we always outperform [2].

We have defined $\overline{y}_i$ in lines 118-119. For L124, we will replace it by "... a small constant ϵe_{il} in (2) instead of e_{il} to ..."

To Reviewer #4: We will give more detailed explanation for cutting planes in Prop. 1 in the revised paper.