SUPPLEMENTARY MATERIAL FOR "RO-FIGS: EFFICIENT AND EXPRESSIVE TREE-BASED ENSEMBLES FOR TABULAR DATA"

APPENDIX A
EXTENDED EXPERIMENTAL SETUP

Here, we give more details about the datasets used for evaluation (see Table II), discuss different encodings of nominal features in RO-FIGS (see Table III), and provide details about the search space for hyperparameter tuning of the tree-based baselines (see Table IV) and grid search space along with the best per-dataset configuration for RO-FIGS (see Table V).

A. Datasets

All datasets are publicly available on OpenML and summarised in Table II. We first split the data according to 10 train/test folds provided by OpenML and then split each training fold into training and validation sets, using the same splits as McElfresh et al. [9]. Each dataset is therefore split into training (80%), validation (10%), and test (10%) sets.

Most baselines expect numerical input, therefore we one-hot encode nominal features. The only exceptions are CatBoost and LightGBM, to which we pass the indices of nominal features. For RO-FIGS, we evaluate three encoding approaches of transforming nominal features into numerical: E-OHE, E-COUNT, and E-PROP, as computing linear combinations of nominal features is not trivial. Table III demonstrates that one-hot encoding outperforms the other two, which transform nominal features into ordered ones without increasing the dimensionality of the data. Therefore, we use one-hot encoding to transform nominal features into numerical ones.

The data is min-max scaled to [0, 1]. Specifically, during hyperparameter tuning (with models trained on the training data and evaluated on the validation data), the scaling transformation is learnt from the training data and applied to the validation data. When training the final model on the combined training and validation data (and evaluated on test data), the transformation learnt from the joint dataset is applied to the test data.

B. Hyperparameter tuning

a) Baselines: We employ `hyperopt` [29] with the tree-structured Parzen estimator to fine-tune hyperparameters for the tree-based baselines. Each baseline undergoes 30 tuning iterations, with the exception of the model tree, which is limited to five iterations due to its high computational cost. The details about the search space of hyperparameters are shown in Table IV. TabPFN requires no tuning and we use its default hyperparameter values. For optimal trees, we set the depth to three and trained each model for up to three minutes.

Because we compare RO-FIGS to the baselines w.r.t. the number of trees and splits, we additionally introduce the following penalties to promote the compactness in the tree-based baselines:

- **DT**: 0.001 $\times$ max_depth
- **MT** using `RidgeClassifier` as the base estimator: 0.001 $\times$ max_depth
- **ODT**: 0.001 $\times$ max_depth
- **RF, ETC, CatBoost, LightGBM, XGBoost**: 0.001 $\times$ max_depth + 0.0001 $\times$ n_estimators
- **FIGS**: 0.001 $\times$ max_rules + 0.001 $\times$ max_trees
- **Ens-ODT**: 0.001 $\times$ max_depth + 0.0001 $\times$ num_trees

b) RO-FIGS: To better understand the behaviour of RO-FIGS models across a spectrum of hyperparameter values, we perform a grid search over two hyperparameters: BEAM_SIZE, which limits the number of features considered per split, and MIN_IMP_DEC, defining the stopping condition for training. We use a constant search space for the minimum impurity decrease parameter and tailor the other parameter to the number of features in the dataset (i.e., d):

- MIN_IMP_DEC: 0.05, 0.1, 1, 5, 10, and
- BEAM_SIZE: 1, 2, $\sqrt{d}$, $\frac{d}{4}$, $\frac{d}{2}$, d; where d is the total number of features in a dataset.

The middle two columns in Table V show the optimal configuration, which leads to the best performance on the test data. It should be noted that the BEAM_SIZE value is the maximum number of features that RO-FIGS considers at each iteration. In practice, only a small proportion of features have non-zero weights, leading to a much lower average number of features per split. This is reported in the last column of Table V.

TABLE II: Properties of the benchmark datasets for binary classification tasks. *Nominal* indicates the number of nominal categorical features, while *Final* shows the total number of features after nominal features have been one-hot encoded (datasets are ordered w.r.t. the latter).

Dataset	Original name (OpenML id)	Numerical	Nominal	Final	Samples
blood	blood-transfusion-service-center (1464)	4	0	4	748
diabetes	diabetes (37)	8	0	8	768
breast-w	breast-w (15)	9	0	9	699
ilpd	ilpd (1480)	9	1	11	583
monks2	monks-problems-2 (334)	0	6	17	601
climate	climate-model-simulation-crashes (40994)	18	0	18	540
kc2	kc2 (1063)	21	0	21	522
pc1	pc1 (1068)	21	0	21	1,109
kc1	kc1 (1067)	21	0	21	2,109
heart	heart-c (982)	6	7	26	303
tictactoe	tic-tac-toe (50)	0	9	27	958
wdbc	wdbc (1510)	30	0	30	569
churn	churn (40701)	16	4	33	5,000
pc3	pc3 (1050)	37	0	37	1,563
biodeg	qsar-biodeg (1494)	41	0	41	1,055
credit	credit-approval (29)	6	9	51	690
spambase	spambase (44)	57	0	57	4,601
credit-g	credit-g (31)	7	13	61	1,000
friedman	fri_c4_500_100 (610)	100	0	100	500
usps	USPS (41964)	256	0	256	1,424
bioresponse	Bioresponse (45019)	419	0	419	3,434
speeddating	SpeedDating (40536)	59	61	503	8,378

TABLE III: Classification performance of RO-FIGS on datasets that include nominal features, using three different encodings for nominal features: E-OHE, E-COUNT and E-PROP. We report mean $\pm$ std of the test balanced accuracy across 10 folds. One-hot encoding (E-OHE) leads to the best performance on average.

Dataset	E-OHE	E-COUNT	E-PROP
ilpd	61.9 $\pm$ 10.4	62.9 $\pm$ 7.7	61.6 $\pm$ 4.2
monks-2	99.5 $\pm$ 0.8	76.5 $\pm$ 6.6	76.2 $\pm$ 4.5
heart	84.8 $\pm$ 6.4	83.5 $\pm$ 6.7	81.8 $\pm$ 5.4
tictactoe	94.4 $\pm$ 3.1	78.1 $\pm$ 4.7	87.5 $\pm$ 3.9
churn	86.3 $\pm$ 2.0	82.3 $\pm$ 3.6	86.7 $\pm$ 2.1
credit	85.7 $\pm$ 5.2	86.7 $\pm$ 3.8	86.2 $\pm$ 3.1
credit-g	65.7 $\pm$ 5.0	65.6 $\pm$ 5.2	66.4 $\pm$ 2.3
Avg. rank	1.7	2.1	2.1

TABLE IV: Search space for hyperparameters of tree-based baselines. We use `hp.quniform` to define the space unless marked otherwise. Symbols $*$, †, and $\star$ denote the use of `hp.uniform`, `hp.loguniform`, and `hp.choice`, respectively. d denotes the total number of features per dataset.

Method	Hyperparameter	Type	Range
DT	max_depth	Discrete	[1, …, 30]
	min_samples_leaf	Discrete	[1, …, 40]
	min_samples_split	Discrete	[2, …, 40]
MT	alpha $*$	Continuous	[0.5, …, 1]
	max_depth	Discrete	[1, …, 20]
	min_samples_leaf	Discrete	[3, …, 40]
	min_samples_split	Discrete	[6, …, 40]
ODT	max_depth	Discrete	[1, …, 30]
	max_features $\star$	Choice	$[1, 2, \sqrt{d}, \frac{d}{4}, \frac{d}{2}, d]$
	min_examples_to_split	Discrete	[2, …, 40]
RF	n_estimators	Discrete	[5, …, 250]
	max_depth	Discrete	[1, …, 30]
	min_samples_leaf	Discrete	[1, …, 40]
	min_samples_split	Discrete	[2, …, 40]
ETC	n_estimators	Discrete	[5, …, 250]
	max_depth	Discrete	[1, …, 30]
	min_samples_leaf	Discrete	[1, …, 40]
	min_samples_split	Discrete	[2, …, 40]
CatBoost	n_estimators	Discrete	[5, …, 250]
	max_depth	Discrete	[1, …, 30]
	learning_rate †	Continuous	[0.01, …, 0.25]
	num_leaves	Discrete	[2, …, 40]
	min_child_samples	Discrete	[1, …, 40]
LightGBM	n_estimators	Discrete	[5, …, 250]
	max_depth	Discrete	[1, …, 30]
	learning_rate †	Continuous	[0.01, …, 0.25]
	num_leaves	Discrete	[2, …, 40]
	min_child_samples	Discrete	[1, …, 40]
XGBoost	n_estimators	Discrete	[5, …, 250]
	max_depth	Discrete	[1, …, 30]
	learning_rate †	Continuous	[0.01, …, 0.25]
	min_split_loss	Discrete	[0, …, 40]
	min_child_weight	Discrete	[1, …, 40]
FIGS	max_rules	Discrete	[1, …, 20]
	max_trees	Discrete	[1, …, 20]
EnsODT	max_depth	Discrete	[1, …, 30]
	max_features $\star$	Choice	$[1, 2, \sqrt{d}, \frac{d}{4}, \frac{d}{2}, d]$
	min_examples_to_split	Discrete	[2, …, 40]
	num_trees	Discrete	[5, …, 50]
MLP	num_epoch	Discrete	[1,000]
	patience	Discrete	[50, …, 300]
	learning_rate †	Continuous	[0.0001, …, 0.01]
	dropout_rate $*$	Continuous	[0, …, 0.25]
	weight_decay $*$	Continuous	[0, …, 0.01]
	L1 $*$	Continuous	[0, …, 0.01]

TABLE V: Optimal configuration, leading to the best performance on test data, is reported in the middle two columns: MIN_IMP_DEC is a minimum impurity decrease value used as a stopping condition, and BEAM_SIZE represents the number of features that are considered per split. Although the optimal BEAM_SIZE value is often large, the actual average number of features per split, reported in the last column, is much lower.

	Optimal		Actual
Dataset	MIN_IMP_DEC	BEAM_SIZE	#features per split
blood	5	2	1.6 ± 0.2
diabetes	10	8	3.0 ± 0.0
breast-w	1	9	4.1 ± 0.4
ilpd	0.1	5	2.0 ± 0.0
monks2	0.1	17	7.2 ± 0.5
climate	0.1	18	7.7 ± 0.6
kc2	10	21	10.7 ± 0.8
pc1	0.1	4	1.7 ± 0.1
kc1	0.05	4	2.0 ± 0.1
heart	10	26	9.7 ± 0.7
tictactoe	0.05	13	4.5 ± 0.2
wdbc	1	30	10.4 ± 1.4
churn	5	33	9.6 ± 0.9
pc3	0.1	18	7.4 ± 0.2
biodeg	10	41	14.6 ± 1.7
credit	1	12	3.9 ± 0.5
spambase	0.1	57	31.4 ± 0.5
credit-g	1	2	1.2 ± 0.1
friedman	10	25	9.1 ± 1.3
usps	1	256	46.4 ± 15.2
bioresponse	0.05	419	45.8 ± 13.3
speeddating	10	125	50.1 ± 3.9

APPENDIX B
EXTENDED GENERAL RESULTS

In this section, we provide additional results from Section IV.

A. Performance of additional baselines

We first show additional results of baselines that were excluded in the paper. Table VI compares the performance of three gradient-boosted decision tree methods—CatBoost, LightGBM, and XGBoost—with RO-FIGS. We choose CatBoost as a representative method of the class of gradient-boosted decision trees because it performs the best out of three. Table VII then compares RO-FIGS and TabPFN, which we excluded as a baseline due to potential bias and misleading results; TabPFN was pre-trained on some of the datasets we used for evaluation.

TABLE VI: Classification performance of LightGBM, XGBoost, CatBoost, and RO-FIGS on 22 tabular datasets. We report mean ± std of the test balanced accuracy across 10 folds. The highest accuracy for each dataset across all methods is bolded, and the highest accuracy for each dataset across baselines is underlined. CatBoost is performing best out of the three gradient-boosted decision tree methods with 12 wins.

Dataset	LightGBM	XGBoost	CatBoost	RO-FIGS
blood	60.6 ± 5.5	59.9 ± 5.6	60.3 ± 5.8	**68.5 ± 4.1**
diabetes	73.1 ± 5.4	69.5 ± 5.0	70.5 ± 7.0	**73.6 ± 4.7**
breast-w	95.2 ± 2.6	94.9 ± 2.5	95.9 ± 2.8	**96.5 ± 1.9**
ilpd	59.2 ± 7.7	57.4 ± 5.3	58.9 ± 6.8	**61.9 ± 10.4**
monks2	98.0 ± 2.5	98.0 ± 2.6	78.3 ± 8.1	**99.5 ± 0.8**
climate	71.2 ± 11.0	68.8 ± 12.7	68.1 ± 10.7	**73.1 ± 15.6**
kc2	68.8 ± 6.0	69.0 ± 7.3	69.2 ± 7.5	**77.6 ± 7.9**
pc1	62.8 ± 5.4	62.5 ± 7.9	61.2 ± 9.4	**66.6 ± 7.2**
kc1	63.7 ± 4.0	**66.8 ± 2.5**	65.0 ± 3.3	64.9 ± 4.9
heart	82.2 ± 5.3	82.9 ± 4.8	83.3 ± 5.2	**84.8 ± 6.4**
tictactoe	99.1 ± 1.3	97.9 ± 2.3	**99.9 ± 0.1**	94.4 ± 3.1
wdbc	96.1 ± 3.0	94.5 ± 3.5	**96.3 ± 2.4**	95.6 ± 3.5
churn	87.5 ± 2.3	85.4 ± 2.4	**88.3 ± 1.9**	86.3 ± 2.0
pc3	60.2 ± 5.8	59.0 ± 7.4	60.7 ± 6.5	**62.9 ± 6.6**
biodeg	84.8 ± 3.4	82.4 ± 5.7	**85.3 ± 2.7**	82.7 ± 3.7
credit	86.5 ± 3.9	85.3 ± 4.2	**86.8 ± 4.4**	85.7 ± 5.2
spambase	94.9 ± 1.6	93.8 ± 2.5	**94.9 ± 1.2**	92.9 ± 1.3
credit-g	67.5 ± 3.1	**67.6 ± 5.4**	66.8 ± 4.9	65.7 ± 5.0
friedman	**88.1 ± 3.9**	87.5 ± 3.3	86.4 ± 8.7	80.0 ± 7.4
usps	97.3 ± 1.8	96.6 ± 1.6	**98.0 ± 1.5**	97.5 ± 1.6
bioresponse	**78.2 ± 2.7**	77.1 ± 2.0	77.7 ± 3.3	72.9 ± 2.6
speeddating	68.6 ± 1.4	68.1 ± 1.4	**69.1 ± 1.9**	65.4 ± 1.9

B. Ablation studies on RO-FIGS

We first compare learning processes in FIGS and RO-FIGS, which extends Fig. 1. Specifically, Fig. 5 highlights the main difference between FIGS and RO-FIGS: the use of univariate and oblique splits, respectively.

Table VIII then demonstrates that utilising the total number of splits as a stopping condition (as in FIGS) leads to subpar performance. This implies that employing the minimum impurity decrease parameter (i.e., MIN_IMP_DEC) as a stopping condition for model growth is recommended over simply limiting the number of splits in the model.

Next, we analyse how oblique splits contribute to performance. In Fig. 6, FIGS, shown as a reference point at 0%, and RO-FIGS correspond to values in Table I. We further investigate how the extreme values of the BEAM_SIZE parameter influence the performance of RO-FIGS models. Namely, RO-FIGS-min and RO-FIGS-max consider one or all available features at each split, respectively. We can see a notable increase in performance when using oblique splits (in both RO-FIGS and RO-FIGS-max), although considering all available features at each step can be suboptimal. Finally, we expect RO-FIGS-min to perform the worst as it randomly picks one feature when creating splits. We observe however that it outperforms FIGS on 11 out of 22 datasets. This suggests that using MIN_IMP_DEC as a stopping condition is sometimes better than simply limiting the maximum number of splits in the model. Note that both methods construct univariate splits.

TABLE VII: Classification performance of TabPFN and RO-FIGS on 22 tabular datasets. We report mean ± std of the test balanced accuracy across 10 folds. The highest accuracy for each dataset across all methods is bolded. TabPFN is a competitive and fast baseline, but it is not suitable for datasets with more than 100 features or 10k samples (the last three rows). Moreover, it is a transformer-based method and thus offers very limited interpretability for its decisions.

Dataset	TabPFN	RO-FIGS
blood	63.5 ± 4.6	**68.5 ± 4.1**
diabetes	72.0 ± 6.5	**73.6 ± 4.7**
breast-w	**96.7 ± 2.0**	96.5 ± 1.9
ilpd	60.5 ± 5.0	**61.9 ± 10.4**
monks2	**99.9 ± 0.1**	99.5 ± 0.8
climate	**82.8 ± 8.2**	73.1 ± 15.6
kc2	69.0 ± 7.1	**77.6 ± 7.9**
pc1	52.5 ± 4.9	**66.6 ± 7.2**
kc1	58.2 ± 3.1	**64.9 ± 4.9**
heart	84.5 ± 7.1	**84.8 ± 6.4**
tictactoe	**97.0 ± 2.1**	94.4 ± 3.1
wdbc	**97.7 ± 2.2**	95.6 ± 3.5
churn	83.5 ± 2.6	**86.3 ± 2.0**
pc3	52.3 ± 2.3	**62.9 ± 6.6**
biodeg	**86.8 ± 4.1**	82.7 ± 3.7
credit	**86.2 ± 4.3**	85.7 ± 5.2
spambase	**94.7 ± 1.1**	92.9 ± 1.3
credit-g	**68.5 ± 4.5**	65.7 ± 5.0
friedman	63.4 ± 7.3	**80.0 ± 7.4**
usps	50.0 ± 0.0	**97.5 ± 1.6**
bioresponse	50.0 ± 0.0	**72.9 ± 2.6**
speeddating	50.0 ± 0.0	**65.4 ± 1.9**

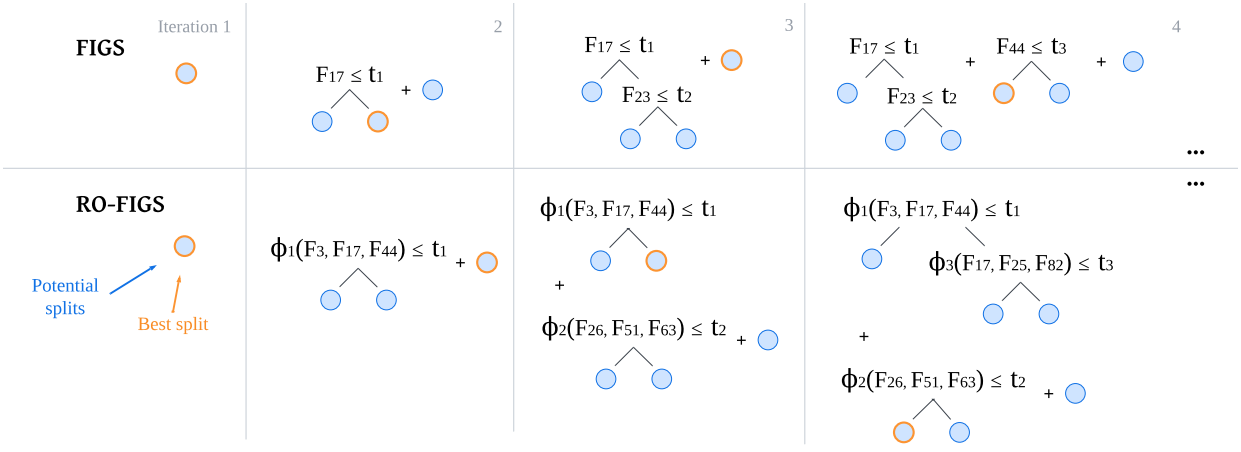


Fig. 5: Comparison of fitting processes of FIGS and RO-FIGS. In each iteration, both methods add one split to the model. However, RO-FIGS computes a linear combination of multiple randomly selected features, while FIGS builds splits with one feature. t, Φ, F denote thresholds, functions, and features, respectively. The figure has been adapted from [10].

TABLE VIII: Performance comparison of RO-FIGS with different stopping conditions: RO-FIGS-splits uses the total number of splits (i.e., MAX_SPLITS, restricted to 20 like FIGS), while RO-FIGS employs the minimum impurity decrease condition (i.e., MIN_IMP_DEC). Both configurations represent the optimal setup. We report the mean $\pm$ std of the test balanced accuracy across 10 folds in the middle two columns, while the last column shows the relative increase in performance of RO-FIGS over RO-FIGS-splits. RO-FIGS, guided by the minimum impurity decrease, exhibits superior accuracy. Notably, the highest performance increase of 9.7% is seen on the *kc2* dataset.

Dataset	RO-FIGS-splits	RO-FIGS	Increase (%)
blood	62.7 $\pm$ 6.4	68.5 $\pm$ 4.1	5.8
diabetes	72.6 $\pm$ 6.4	73.6 $\pm$ 4.7	1.0
breast-w	96.0 $\pm$ 2.0	96.5 $\pm$ 1.9	0.5
ilpd	59.1 $\pm$ 5.0	61.9 $\pm$ 10.4	2.8
monks2	99.5 $\pm$ 0.8	99.5 $\pm$ 0.8	0.0
climate	72.0 $\pm$ 15.9	73.1 $\pm$ 15.6	1.1
kc2	67.9 $\pm$ 6.1	77.6 $\pm$ 7.9	9.7
pc1	62.9 $\pm$ 5.6	66.6 $\pm$ 7.2	3.7
kc1	60.7 $\pm$ 4.3	64.9 $\pm$ 4.9	4.2
heart	82.4 $\pm$ 7.6	84.8 $\pm$ 6.4	2.4
tictactoe	91.5 $\pm$ 3.9	94.4 $\pm$ 3.1	2.9
wdbc	95.6 $\pm$ 3.4	95.6 $\pm$ 3.5	0.0
churn	85.5 $\pm$ 1.8	86.3 $\pm$ 2.0	0.8
pc3	59.1 $\pm$ 6.1	62.9 $\pm$ 6.6	3.8
biodeg	82.6 $\pm$ 2.9	82.7 $\pm$ 3.7	0.1
credit	85.3 $\pm$ 3.8	85.7 $\pm$ 5.2	0.4
spambase	92.7 $\pm$ 1.5	92.9 $\pm$ 1.3	0.2
credit-g	64.8 $\pm$ 4.2	65.7 $\pm$ 5.0	0.9
friedman	76.0 $\pm$ 5.9	80.0 $\pm$ 7.4	4.0
usps	97.7 $\pm$ 1.6	97.5 $\pm$ 1.6	-0.2
bioresponse	72.9 $\pm$ 2.6	72.9 $\pm$ 2.6	0.0
speeddating	66.0 $\pm$ 1.6	65.4 $\pm$ 1.9	-0.6

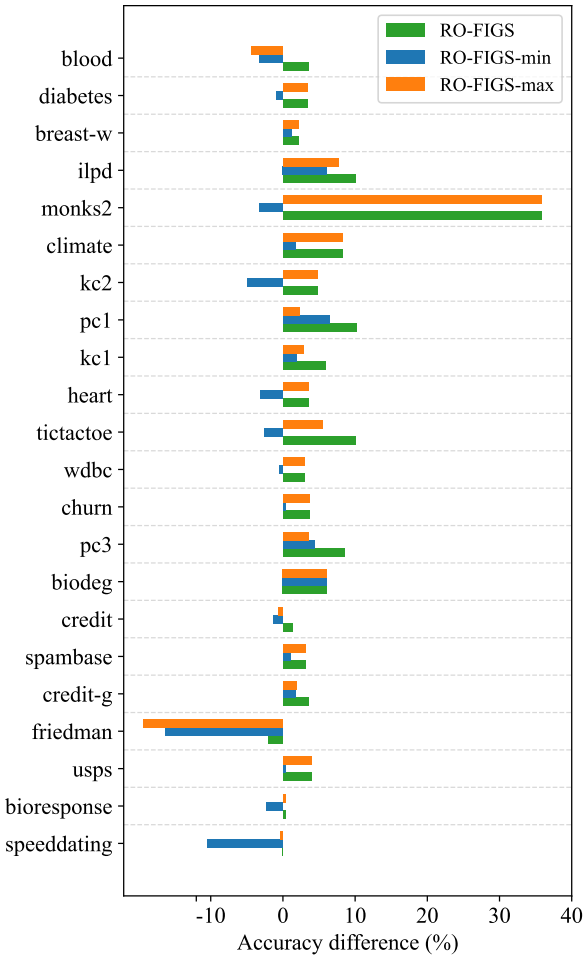


Fig. 6: The effect of oblique splits on the mean balanced accuracy across 10 folds. We use the FIGS baseline with univariate splits as a reference point (with an accuracy difference of 0%). Each bar illustrates the difference in performance between FIGS and the corresponding method, where a positive difference indicates higher accuracy. RO-FIGS represents the optimal configuration of RO-FIGS, while RO-FIGS-min and RO-FIGS-max represent RO-FIGS models with BEAM_SIZE set to extreme values, that is, one and all. There is a clear improvement in accuracy when using oblique splits, but note that using all available features can be suboptimal.

C. *Efficiency*

Tables IX and X show the number of trees and splits for the tree-based methods, respectively. They demonstrate that RO-FIGS models are compact and significantly smaller than the standard tree-based ensembles such as random forest and LightGBM. In terms of size, RO-FIGS models are most comparable to FIGS (limited to 20 splits) but exhibit much better performance (see Table I).

TABLE IX: The number of trees built by tree-based methods. We leave out methods that build exactly one tree and report the mean ± std of the number of trees on the test set across 10 folds. RO-FIGS builds multiple trees when the underlying data has an additive structure present, and this almost always coincides with FIGS finding the additive structure as well. The number of trees in RO-FIGS models is unsurprisingly much lower than the number of trees in ensembles and comparable to that of FIGS.

Dataset	RF	ETC	CatBoost	FIGS	EnsODT	RO-FIGS
blood	73.4 ± 51.2	135.4 ± 71.6	78.0 ± 78.0	1.0 ± 0.0	24.8 ± 10.4	1.0 ± 0.0
diabetes	96.1 ± 70.0	143.4 ± 76.7	59.0 ± 39.8	1.6 ± 0.7	29.7 ± 17.7	1.0 ± 0.0
breast-w	41.3 ± 57.7	47.1 ± 58.7	47.8 ± 46.0	1.0 ± 0.0	18.6 ± 12.9	1.0 ± 0.0
ilpd	60.7 ± 58.0	151.5 ± 70.6	169.5 ± 89.0	1.8 ± 0.6	18.7 ± 11.6	3.0 ± 0.8
monks2	184.3 ± 15.2	176.6 ± 18.9	230.6 ± 19.4	1.0 ± 0.0	28.4 ± 10.9	2.3 ± 1.1
climate	71.5 ± 73.5	9.0 ± 0.0	120.9 ± 57.2	1.1 ± 0.3	19.6 ± 14.7	2.0 ± 0.9
kc2	57.5 ± 78.0	84.0 ± 100.8	73.1 ± 65.5	1.0 ± 0.0	23.0 ± 8.8	1.0 ± 0.0
pc1	50.0 ± 58.4	109.9 ± 80.5	91.1 ± 75.5	1.6 ± 0.7	25.2 ± 14.5	2.4 ± 1.0
kc1	64.5 ± 66.6	93.1 ± 85.3	111.7 ± 66.3	1.3 ± 0.5	27.1 ± 17.6	1.4 ± 0.5
heart	49.7 ± 71.8	68.4 ± 46.1	53.8 ± 52.3	3.1 ± 1.5	23.6 ± 13.9	1.0 ± 0.0
tictactoe	173.5 ± 27.0	148.2 ± 60.6	214.0 ± 5.6	4.5 ± 2.7	41.2 ± 5.7	4.9 ± 1.3
wdbc	28.5 ± 38.3	45.9 ± 50.1	68.0 ± 45.1	1.0 ± 0.0	24.1 ± 12.6	1.0 ± 0.0
churn	186.6 ± 21.8	175.4 ± 8.2	92.8 ± 68.3	1.8 ± 0.8	26.8 ± 13.8	1.1 ± 0.3
pc3	94.2 ± 64.1	97.4 ± 93.1	124.0 ± 60.7	1.5 ± 0.5	14.9 ± 11.9	1.8 ± 0.9
biodeg	61.1 ± 46.9	114.3 ± 75.6	105.4 ± 86.0	2.1 ± 0.9	35.0 ± 14.8	1.0 ± 0.0
credit	97.3 ± 55.1	47.9 ± 69.5	49.2 ± 47.1	1.4 ± 0.7	25.0 ± 13.0	1.0 ± 0.0
spambase	37.1 ± 29.6	162.9 ± 56.9	79.1 ± 44.0	2.5 ± 1.4	19.1 ± 11.1	2.9 ± 0.7
credit-g	156.9 ± 60.6	151.1 ± 65.4	131.7 ± 81.1	1.6 ± 1.0	31.3 ± 16.1	4.6 ± 1.2
friedman	118.6 ± 49.4	130.5 ± 76.3	143.8 ± 95.1	1.5 ± 0.5	24.9 ± 8.2	1.2 ± 0.4
usps	27.7 ± 18.7	27.6 ± 17.1	70.8 ± 67.1	1.0 ± 0.0	28.0 ± 8.2	1.0 ± 0.0
bioresponse	188.5 ± 33.7	144.9 ± 64.3	137.9 ± 66.7	2.7 ± 1.1	34.9 ± 11.2	2.0 ± 0.7
speeddating	53.9 ± 72.8	143.0 ± 72.9	178.3 ± 71.5	2.6 ± 1.0	23.7 ± 9.6	2.6 ± 0.5

TABLE X: The number of splits in tree-based models. We report the mean ± std of the number of splits on the test set across 10 folds. RO-FIGS has a surprisingly low number of splits on some datasets (e.g., *diabetes*, *kc2*, *heart*) and is comparable to single decision trees and FIGS models.

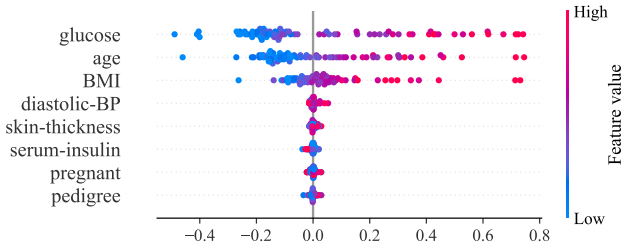
Dataset	DT	MT	OT	ODT	RF	ETC	CatBoost	FIGS	EnsODT	RO-FIGS
blood	29.5 ± 18.1	7.5 ± 3.9	7.0 ± 0.0	12.9 ± 6.5	1710.2 ± 1600.9	11192.1 ± 7989.8	1082.1 ± 976.4	6.7 ± 5.4	348.1 ± 143.6	2.0 ± 0.0
diabetes	31.1 ± 19.1	19.6 ± 5.6	7.0 ± 0.0	21.6 ± 19.0	4461.5 ± 3693.3	13844.3 ± 8236.0	978.0 ± 1003.8	7.1 ± 4.4	805.1 ± 697.5	1.0 ± 0.0
breast-w	9.6 ± 3.9	6.1 ± 1.7	7.0 ± 0.0	7.4 ± 5.2	242.0 ± 346.4	746.5 ± 1181.3	583.2 ± 1080.4	8.4 ± 6.1	92.1 ± 79.5	4.6 ± 1.6
ilpd	16.3 ± 7.8	17.0 ± 6.3	7.0 ± 0.0	13.9 ± 6.2	1356.9 ± 1726.4	13946.2 ± 7690.0	2399.6 ± 2449.8	11.9 ± 6.9	376.0 ± 284.4	75.0 ± 0.0
monks2	43.5 ± 6.4	12.3 ± 2.9	7.0 ± 0.0	16.9 ± 9.7	12997.4 ± 3299.3	17884.9 ± 7510.0	6439.9 ± 895.7	17.6 ± 3.1	428.8 ± 221.3	11.5 ± 5.6
climate	10.4 ± 4.4	7.7 ± 1.8	7.0 ± 0.0	9.3 ± 3.8	1153.5 ± 1368.3	9.0 ± 0.0	970.5 ± 362.3	7.1 ± 3.3	124.3 ± 104.4	14.3 ± 5.3
kc2	9.8 ± 10.1	4.2 ± 2.0	7.0 ± 0.0	13.8 ± 12.5	505.4 ± 1029.1	3520.7 ± 4598.1	1312.7 ± 2038.7	4.3 ± 4.2	305.2 ± 172.9	1.0 ± 0.0
pc1	21.9 ± 10.0	3.5 ± 1.4	7.0 ± 0.0	21.0 ± 12.1	1352.1 ± 2298.3	9208.8 ± 7772.3	1584.7 ± 1249.0	9.6 ± 6.9	530.8 ± 356.5	61.7 ± 10.3
kc1	70.4 ± 39.5	13.1 ± 2.9	7.0 ± 0.0	18.3 ± 14.4	5047.0 ± 8846.3	20019.2 ± 24310.8	2254.7 ± 1343.0	12.1 ± 5.7	1160.6 ± 1103.8	75.0 ± 0.0
heart	8.2 ± 4.6	9.9 ± 1.8	7.0 ± 0.0	9.9 ± 5.2	557.7 ± 1354.8	667.5 ± 455.6	742.7 ± 1659.3	7.0 ± 5.3	232.0 ± 188.2	1.0 ± 0.0
tictactoe	32.8 ± 14.6	0.0 ± 0.0	7.0 ± 0.0	29.4 ± 10.0	17075.7 ± 7980.9	16977.5 ± 8817.3	4787.6 ± 1135.9	18.8 ± 1.2	1072.4 ± 244.4	74.7 ± 0.9
wdbc	5.9 ± 2.9	7.4 ± 3.5	7.0 ± 0.0	4.9 ± 6.5	250.8 ± 309.9	982.7 ± 1657.8	650.7 ± 543.4	5.0 ± 3.0	177.5 ± 180.2	3.3 ± 1.6
churn	89.9 ± 47.5	53.4 ± 23.6	7.0 ± 0.0	54.9 ± 31.3	29896.6 ± 3760.8	50553.5 ± 6254.8	1539.1 ± 680.9	16.5 ± 3.4	1939.1 ± 1195.6	7.6 ± 1.5
pc3	46.5 ± 18.4	22.4 ± 5.9	7.0 ± 0.0	31.9 ± 18.3	4160.6 ± 3864.9	12664.5 ± 13116.7	2374.8 ± 2167.2	12.7 ± 5.8	508.4 ± 488.7	74.6 ± 1.3
biodeg	28.6 ± 13.9	22.1 ± 10.2	7.0 ± 0.0	20.5 ± 18.1	2963.5 ± 2593.0	10230.1 ± 8861.9	1454.1 ± 1161.7	12.4 ± 7.8	1271.8 ± 1085.7	1.5 ± 0.5
credit	17.1 ± 11.9	17.8 ± 3.0	7.0 ± 0.0	16.1 ± 11.7	2166.9 ± 2690.3	472.6 ± 289.9	1446.3 ± 1623.7	4.8 ± 5.2	347.3 ± 226.8	1.0 ± 0.0
spambase	69.8 ± 41.1	34.7 ± 12.8	7.0 ± 0.0	32.5 ± 16.6	3294.5 ± 1900.7	41196.7 ± 19747.7	1057.7 ± 302.1	16.2 ± 2.7	1152.7 ± 770.1	67.1 ± 9.6
credit-g	46.3 ± 23.4	21.5 ± 2.0	7.0 ± 0.0	29.9 ± 25.4	14847.1 ± 10333.6	21764.1 ± 19358.2	1922.3 ± 1344.5	12.3 ± 4.1	934.2 ± 983.3	33.0 ± 5.7
friedman	14.7 ± 8.7	3.2 ± 0.4	7.0 ± 0.0	13.8 ± 7.1	3213.1 ± 2491.3	8503.3 ± 6733.3	1423.5 ± 1036.9	8.1 ± 3.4	407.9 ± 204.5	3.5 ± 1.1
usps	20.0 ± 7.0	2.5 ± 0.5	7.0 ± 0.0	3.3 ± 2.7	563.5 ± 343.2	1635.4 ± 1455.6	651.2 ± 480.3	12.8 ± 6.0	280.6 ± 277.6	2.4 ± 0.8
bioresponse	52.9 ± 18.4	63.7 ± 7.7	7.0 ± 0.0	74.1 ± 84.7	37456.3 ± 11098.1	40008.3 ± 25743.0	2113.2 ± 1808.4	12.0 ± 5.2	3145.8 ± 2438.1	5.1 ± 2.0
speeddating	101.5 ± 67.6	38.0 ± 4.0	7.0 ± 0.0	67.3 ± 42.0	17446.1 ± 26780.3	103295.8 ± 64515.4	4255.5 ± 2865.4	7.2 ± 4.6	2187.7 ± 1208.7	13.1 ± 2.8

D. Expressiveness

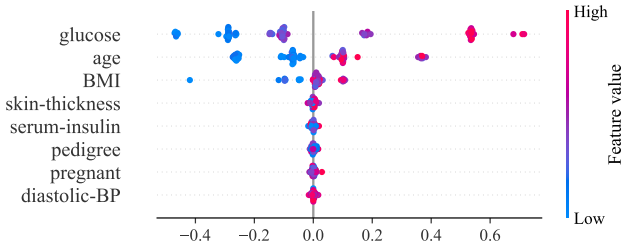
Here, we further discuss the expressiveness of the RO-FIGS models. Due to the robustness of tree-based models to uninformative features, feature importance information is often extracted from them. To extract such information from RO-FIGS models, we first count combinations of features that appear together in the splits. We then analyse this jointly with the SHAP summary plot, which illustrates how each feature contributed to the model. Next, we analyse SHAP plots of baseline methods. In particular, we choose FIGS due to its similarity to RO-FIGS (w.r.t. the framework and size of the models), and CatBoost as the best-performing baseline.

Figs. 7 and 8 present SHAP summary plots for RO-FIGS, FIGS, and CatBoost models on two datasets, *diabetes* and *blood*, respectively. Note that these plots are computed on fold 0, but similar behaviour is observed across all folds. Shap values are estimated using Kernel SHAP, where the nsamples parameter is set to 100. Furthermore, we utilise `shap.sample` to generate a background summary with 100 samples.

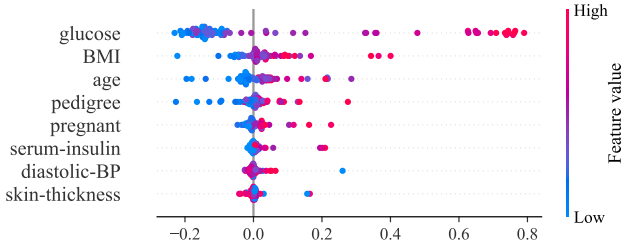
We observe that the oblique nature of splits in RO-FIGS reflect feature interactions, which makes them more expressive than models with univariate trees. Furthermore, such insights cannot be learnt from feature importance tools such as SHAP. For example, on the *diabetes* dataset, RO-FIGS builds models with only one split involving a linear combination of three features. Similarly, on the *blood* dataset, RO-FIGS model consists of two splits, one of which is a linear combination of two features *monetary* and *time*.



(a) RO-FIGS

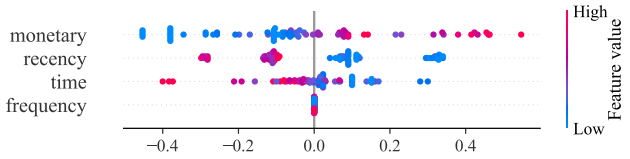


(b) FIGS

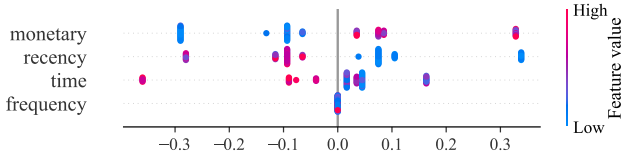


(c) CatBoost

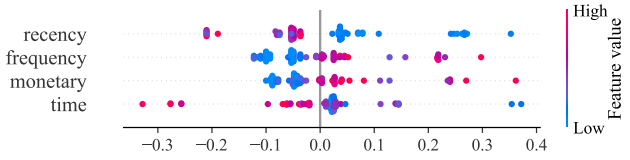
Fig. 7: SHAP summary plots of different models on the first fold of the *diabetes* dataset all implicate that features *glucose*, *age*, and *BMI* contribute to their predictions the most. RO-FIGS with the accuracy of 73.6% (see Table I) outperforms all baselines on this dataset with only one split, demonstrating that a linear combination of features *glucose*, *age*, and *BMI* is optimal.



(a) RO-FIGS



(b) FIGS



(c) CatBoost

Fig. 8: The same as Fig. 7 but on the *blood* dataset. Three methods generally disagree on the importance of features: neither of the FIGS-based models split on *frequency*, but this feature is the second most important for CatBoost. With only 2 splits, one using *recency* and the other a linear combination of *monetary* and *time*), RO-FIGS is able to highlight combinations of features. It also outperforms all baselines, suggesting that this is a highly informative combination of features.