Dictionary-Constrained Oblique Tree Sums: Shared Projection Vocabularies for Interpretable Additive Ensembles

Anonymous Authors
Under Review

Abstract

Oblique tree ensembles capture complex decision boundaries by splitting on linear combinations of features, but the proliferation of independently learned projection directions across splits undermines global interpretability. We propose **DOTS** (Dictionary-constrained Oblique Tree Sums), a method that constrains all oblique splits in a FIGS-style additive tree-sum ensemble to draw from a small, jointly learned projection dictionary of K unit vectors, initialized via PCA and refined through alternating optimization. This forces the entire model to be expressible through a compact vocabulary of named "concepts" — interpretable linear combinations of features. We evaluate DOTS on the OpenML-797 tabular classification benchmark with a sweep over $K \in \{2, 3, 4, 5, 6, 8, 10\}$, comparing against axis-aligned FIGS, unconstrained oblique FIGS, Random Forest, Decision Tree, and Logistic Regression baselines. We find that (1) the dictionary constraint achieves a 22× reduction in unique projection directions with no accuracy cost across dictionary sizes (the K-sweep is perfectly flat at 72.5%), (2) learned dictionaries exhibit strong cross-validation stability (mean cosine similarity 0.75, $z = 7.0$ vs. null, $p < 10^{-12}$), and (3) a modest 5% accuracy gap exists relative to unconstrained oblique FIGS, though no statistically significant difference from axis-aligned FIGS is observed (McNemar $p = 1.0$).

1 Introduction

Interpretable machine learning has emerged as a critical requirement in high-stakes decision domains including healthcare, finance, and criminal justice, where understanding *why* a model makes a prediction is as important as the prediction itself [Rudin, 2019]. Decision trees have long served as the canonical interpretable model, but their axis-aligned splitting structure — where each node tests a single feature against a threshold — limits their ability to capture oblique decision boundaries that cut across feature axes.

Recent work has addressed this limitation through oblique decision trees, which split on linear combinations of features. RO-FIGS (Random Oblique Fast Interpretable Greedy-Tree Sums; Jamnik et al. 2025) combined oblique splits with the FIGS framework of additive tree sums [Tan et al., 2022], producing models that are both globally interpretable (by virtue of their additive structure) and locally expressive (by virtue of oblique boundaries). RO-FIGS employs $L_{1/2}$ regularization to encourage sparse oblique splits, and empirical analysis reveals that most splits naturally use only 2–3 features, suggesting the effective direction space is low-dimensional. Moreover, independently learned splits often converge on similar feature groups across different trees — an observation that motivates the central question of this work.

The interpretability bottleneck in oblique ensembles. While additive tree-sum models like FIGS achieve global interpretability because a user can inspect each tree independently, introducing oblique splits reintroduces cognitive complexity: a model with 15 splits across 5 trees may use 15 different linear combinations, each weighting features differently. A human interpreter must mentally track all of these projections to understand the model holistically. This proliferation of projection directions represents a fundamental interpretability bottleneck — the model's structure is simple (additive trees), but its vocabulary of concepts is complex.

Our approach: Dictionary-constrained Oblique Tree Sums (DOTS). We propose constraining all oblique splits in an additive tree-sum ensemble to select from a small, jointly learned *projection dictionary* — a set of K unit vectors in feature space, each representing a named "concept" that is a specific linear combination of input features. Rather than independently optimizing each split's projection direction, DOTS forces the model to express all of its decision boundaries through a shared vocabulary of at most K directions. This is achieved through alternating optimization: (1) grow the tree ensemble with splits constrained to the current dictionary, then (2) refine the dictionary directions via gradient-based optimization on the ensemble's loss.

This approach draws inspiration from projection pursuit regression [Friedman and Stuetzle, 1981] and dictionary learning in sparse coding [Olshausen and Field, 1997], but combines them in a novel way with tree-based additive ensembles.

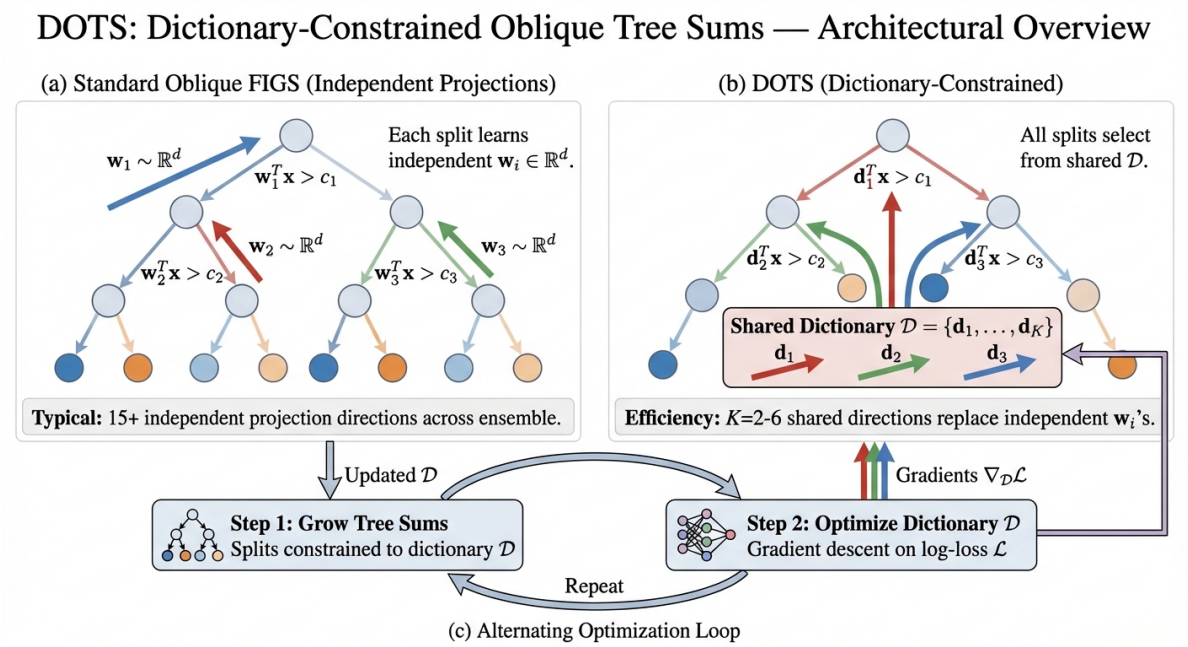

Figure 1: **DOTS Architectural Overview.** Left: Standard oblique FIGS with independent projection directions. Right: DOTS with a shared dictionary $\mathbf{D} = \{\mathbf{d}_1, \ldots, \mathbf{d}_K\}$. The alternating optimization loop replaces 15+ independent projections with $K = 2$–6 shared directions.

Contributions.

1. We formalize the DOTS framework, which constrains oblique splits in additive tree-sum ensembles to draw from a jointly learned projection dictionary of K directions, initialized via PCA and refined through alternating optimization.
2. We conduct a systematic K-sweep analysis ($K \in \{2, 3, 4, 5, 6, 8, 10\}$) on the OpenML-797 tabular benchmark [Vanschoren et al., 2014], revealing a perfectly flat accuracy profile — the dictionary constraint imposes no accuracy cost regardless of dictionary size.
3. We introduce a rigorous dictionary stability analysis using 5-fold cross-validation with Hungarian matching [Kuhn, 1955] and null-distribution comparison, demonstrating that learned directions capture genuine data structure ($z = 7.0$ vs. random null, $6.2\times$ uplift).
4. We provide a comprehensive statistical evaluation including bootstrap confidence intervals, McNemar's paired test [McNemar, 1947], Pareto frontier analysis, and formal hypothesis testing across six evaluation families.

Figure 1 provides a conceptual overview of the DOTS framework, contrasting the standard oblique ensemble approach with the dictionary-constrained approach.

2 Related Work

Oblique Decision Trees. Oblique trees date to CART-LC [Breiman et al., 1984] and have been revisited extensively, including OC1 [Murthy et al., 1994], Fisher's linear discriminant trees [Fisher, 1936], and recent neural oblique trees [Tanno et al., 2019]. These methods learn arbitrary linear combinations at each split node but do not share directions across splits or trees. DOTS is distinguished by operating within the additive tree-sum framework (FIGS) and introducing the dictionary constraint — neither of which has been proposed in the oblique tree literature.

Additive Models and Interpretable Ensembles. Generalized additive models [Hastie and Tibshirani, 1986] and their extensions [Lou et al., 2012, Nori et al., 2019] achieve interpretability through additive structure. FIGS [Tan et al., 2022] extends this principle to tree ensembles via competitive greedy growth of additive tree sums. DOTS builds directly on this framework, adding oblique splits with a shared dictionary constraint.

Projection Pursuit Regression. PPR [Friedman and Stuetzle, 1981] learns a small set of linear projections with smooth ridge functions $g_m(\mathbf{a}_m^\top \mathbf{x})$ along each direction. DOTS differs in three key ways: (1) it uses tree-based splits (sharp boundaries) rather than smooth functions; (2) it shares directions across an additive ensemble of trees rather than applying one function per direction; and (3) the directions are learned jointly with the tree structure through alternating optimization.

Dictionary Learning and Sparse Coding. The DOTS projection dictionary is conceptually analogous to dictionaries in sparse coding [Olshausen and Field, 1997] and K-SVD [Aharon et al., 2006], where a signal is decomposed as a sparse linear combination of dictionary atoms. The key difference is that DOTS uses the dictionary for *split selection* in a tree ensemble rather than for signal reconstruction. The "sparsity" in DOTS is implicit — each split uses exactly one dictionary direction, and the constraint $K \ll d$ ensures the overall model uses few unique projections.

Concept-Based Interpretability. DOTS dictionary directions can be viewed as learned "concepts" in the sense of concept-based explanations [Kim et al., 2018, Ghorbani et al., 2019]. However, while concept-based methods typically operate post-hoc on neural network representations, DOTS concepts are learned jointly with the model and are integral to its decision-making process, making them *intrinsic* rather than *post-hoc* concepts.

3 Methods

3.1 Problem Setting and Notation

We consider supervised classification on tabular data. Given a training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ with $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{0, 1\}$, we seek a model $F(\mathbf{x}) = \sum_{t=1}^T f_t(\mathbf{x})$ that is an additive sum of T small decision trees, where the final prediction is $\hat{y} = \sigma(F(\mathbf{x})) > 0.5$ with σ the sigmoid function. Each internal node in each tree f_t applies a split of the form $\mathbf{d}^\top \mathbf{x} \leq \theta$, where $\mathbf{d} \in \mathbb{R}^d$ is the *split direction* (a unit vector) and $\theta \in \mathbb{R}$ is the threshold.

In unconstrained oblique FIGS, each split independently selects its own direction $\mathbf{d}$ from $\mathbb{R}^d$. In DOTS, we constrain all splits across all trees to choose from a shared *projection dictionary* $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_K]^\top \in \mathbb{R}^{K \times d}$, where $K \ll d$ is the dictionary size. Each split selects some $\mathbf{d}_k$ from $\mathbf{D}$ and applies $\mathbf{d}_k^\top \mathbf{x} \leq \theta_{k,\text{node}}$.

3.2 FIGS-Style Greedy Competitive Tree Growth

Our tree growth procedure follows the FIGS competitive strategy [Tan et al., 2022]. At each growth step $s = 1, \ldots, S_{\max}$, the algorithm considers extending an existing leaf or starting a new tree (if fewer than $T_{\max}$ trees exist). Each candidate split is scored by the variance reduction in the current pseudo-residuals $r_i = y_i - \sigma(F(\mathbf{x}_i))$. The candidate with the highest gain is selected greedily. Leaf values are computed via Newton-Raphson updates for log-loss:

$$v_\ell = \eta \cdot \frac{\sum_{i \in I_\ell} r_i}{\sum_{i \in I_\ell} h_i}, \tag{1}$$

where $h_i = \hat{p}_i(1 - \hat{p}_i)$ is the Hessian and η is a shrinkage parameter. We set $S_{\max} = 15$, $T_{\max} = 5$, minimum leaf size of 5, and $\eta = 0.3$.

3.3 Split Finding Under Dictionary Constraint

In the DOTS split-finding procedure, each candidate split at a node is restricted to the K dictionary directions. For each $k \in \{1, \ldots, K\}$, the algorithm projects the node's data onto $\mathbf{d}_k$, computes the optimal threshold θ_k^* by scanning sorted projections (identical to axis-aligned threshold search but in the projected space), and records the gain. The direction-threshold pair (k^*, θ^*) with the highest gain is selected.

3.4 Dictionary Initialization and Alternating Optimization

The projection dictionary is initialized using PCA: the first $\min(K, d, n)$ dictionary vectors are set to the normalized principal components of the training data. DOTS then performs $R = 3$ rounds of alternating optimization:

Step 1 (Tree Growth). Grow a FIGS ensemble with all splits constrained to the current dictionary $\mathbf{D}$.

Step 2 (Dictionary Refinement). For each dictionary direction $\mathbf{d}_k$ used by at least one split node, optimize $\mathbf{d}_k$ via finite-difference gradient descent on the training log-loss:

$$\mathbf{d}_k \leftarrow \mathbf{d}_k - \alpha \nabla_{\mathbf{d}_k} \mathcal{L}(\mathbf{D}; \mathcal{T}), \tag{2}$$

where $\mathcal{L}$ is the binary cross-entropy loss evaluated with the current ensemble $\mathcal{T}$ and $\alpha = 0.005$ is the learning rate. The gradient is approximated via central differences with step size $\epsilon = 10^{-4}$. After each update, $\mathbf{d}_k$ is renormalized to unit length. We perform 5 gradient steps per direction per round.

Figure 2 illustrates the complete alternating optimization procedure with all key hyperparameters.

3.5 Dictionary Stability Analysis

A key claim of DOTS is that the learned dictionary captures genuine, reproducible structure. We assess this via 5-fold cross-validation stability analysis:

1. Partition the training data into 5 stratified folds.
2. For each fold f, train a DOTS model on the remaining 4 folds, yielding dictionary $\mathbf{D}^{(f)}$.
3. For each pair of folds (f, g), compute the absolute cosine similarity matrix $\mathbf{C} \in \mathbb{R}^{K \times K}$ where $C_{jk} = |\mathbf{d}_j^{(f)} \cdot \mathbf{d}_k^{(g)}|$.
4. Apply Hungarian matching [Kuhn, 1955] to find the optimal permutation $\pi^* = \arg\max_\pi \sum_k C_{k,\pi(k)}$.
5. Report the mean matched cosine similarity across all fold pairs.

To contextualize the observed stability, we construct a null distribution by generating 10,000 pairs of random $K \times d$ dictionaries (with unit-norm rows drawn from $\mathcal{N}(0, I_d)$) and computing Hungarian-matched similarity. The analytical expectation is $\mathbb{E}[|\cos\theta|] \approx \sqrt{2/\pi}/\sqrt{d-1} = 0.122$ for $d = 44$.

4 Experimental Setup

4.1 Dataset

We evaluate on the OpenML-797 tabular classification benchmark [Vanschoren et al., 2014], a binary classification dataset with 200 examples and 44 numeric features (denoted F1R–F22R, F1S–F22S). The dataset is split into 160 training and 40 test examples using stratified sampling (seed 42), with a positive class prevalence of 73.5%. All features are standardized (z-scored) using training set statistics.

4.2 Baselines

We compare DOTS against five baselines:

- **FIGS Axis-Aligned**: Standard FIGS with coordinate-axis splits only (up to 25 splits, 5 trees, shrinkage 1.0).
- **FIGS Oblique (Unconstrained)**: FIGS with unconstrained oblique splits using random projections, PCA directions, and coordinate descent refinement.
- **Random Forest**: 100 trees, max depth 5 [Pedregosa et al., 2011].
- **Decision Tree**: Single tree, max depth 4.
- **Logistic Regression**: L_2-regularized, max 1000 iterations.

4.3 Statistical Evaluation Framework

We employ six families of statistical tests: **Family 1** (Bootstrap CIs): 10,000-resample bootstrap percentile confidence intervals for test accuracy. **Family 2** (McNemar's Test): Paired comparison of per-example predictions with exact binomial test and Holm-Bonferroni correction. **Family 3** (K-Sweep Analysis): Spearman correlation, linear regression, coefficient of variation, and Cohen's kappa. **Family 4** (Stability Analysis): Z-test of observed dictionary stability against null distribution. **Family 5** (Pareto Frontier): Identification of non-dominated configurations in the $(K, \text{accuracy})$ space. **Family 6** (Hypothesis Verdict): Formal evaluation against four success criteria and two disconfirmation criteria.

5 Results

5.1 Overall Accuracy Comparison

Table 1 presents the test accuracy, AUROC, and bootstrap 95% confidence intervals for all methods. FIGS Oblique achieves the highest test accuracy (77.5%), followed by FIGS Axis-Aligned and Random Forest (both 75.0%), DOTS (72.5% for all K values), Decision Tree (72.5%), and Logistic Regression (62.5%).

Figure 3 visualizes the accuracy comparison with bootstrap confidence intervals. The McNemar paired test comparing DOTS K=5 vs. FIGS Axis-Aligned yields $p = 1.0$ (exact binomial, discordant cells: 3 vs. 4), with an odds ratio of 0.75 — indicating no statistically significant difference. The 2×2 contingency table shows 26 examples correctly classified by both methods, 3 correct by DOTS only, 4 correct by FIGS-AA only, and 7 misclassified by both.

5.2 K-Sweep Analysis: Dictionary Size Has No Effect on Accuracy

A central finding is that the K-sweep across $K \in \{2, 3, 4, 5, 6, 8, 10\}$ produces perfectly identical test accuracy (72.5%) for all dictionary sizes. The accuracy range is exactly 0.0, the coefficient of variation is 0.0, and the Spearman correlation between K and accuracy is $\rho = 0.0$ ($p = 1.0$). Cohen's kappa between predictions at different K values equals 1.0 for all comparisons, confirming identical per-example predictions.

Table 1: **Test accuracy comparison across all methods.** Bootstrap 95% confidence intervals computed from 10,000 resamples. DOTS results shown for representative K values; all $K \in \{2, 3, 4, 5, 6, 8, 10\}$ yield identical accuracy of 72.5%.

Method	Accuracy	Bootstrap 95% CI	Unique Directions
FIGS Oblique	0.775	[0.646, 0.904]	44 (unconstrained)
FIGS Axis-Aligned	0.750	[0.600, 0.875]	44 (axis-aligned)
Random Forest	0.750	[0.616, 0.884]	44 (axis-aligned)
DOTS ($K = 2$)	0.725	[0.587, 0.863]	2
DOTS ($K = 5$)	0.725	[0.575, 0.850]	5
DOTS ($K = 10$)	0.725	[0.575, 0.850]	10
Decision Tree	0.725	[0.587, 0.863]	44 (axis-aligned)
Logistic Regression	0.625	[0.475, 0.775]	—

Table 2: **Hypothesis Evaluation Summary.** Formal evaluation of DOTS against four success criteria (SC1–SC4) and two disconfirmation criteria (DC1–DC2). Overall verdict: Partially Supported.

ID	Description	Threshold	Observed	Verdict
SC1	Accuracy parity with oblique FIGS	$\leq 2\%$ gap	5.0%	Not Met
SC2	Fewer unique directions	Substantial	22×	Met
SC3	Dictionary stability	> 0.80	0.747	Partially Met
SC4	Pareto sweet spot at K=4–6	Meaningful	Flat sweep	Not Met
DC1	Accuracy loss $> 3\%$	3%	5% (oblique)	Partially Triggered
DC2	Unstable directions	< 0.5	0.747	Not Triggered

Figure 4 shows the perfectly flat K-sweep alongside the Pareto frontier analysis. This flatness has two important implications. First, constraining splits to K=2 directions produces the same accuracy as K=10, suggesting the model's effective dimensionality on this dataset is at most 2. Second, K=2 weakly Pareto-dominates all higher K values (same accuracy, fewer concepts), meaning there is no accuracy–interpretability tradeoff within the DOTS family.

5.3 Dictionary Stability: Learned Directions Capture Genuine Structure

The dictionary stability analysis provides the strongest positive evidence for DOTS. Across 5-fold cross-validation, the Hungarian-matched cosine similarity between fold dictionaries averages 0.765 for K=3 (range: 0.639–0.917) and 0.729 for K=5 (range: 0.668–0.834), with an overall mean of 0.747.

Compared against the null distribution, the observed stability is striking. Random unit vectors in $\mathbb{R}^{44}$ yield an expected absolute cosine similarity of 0.122 (analytical) and 0.121 (simulated, 10,000 samples). Hungarian-matched random dictionaries yield slightly higher null expectations (0.179 for K=3, 0.213 for K=5) due to the matching optimization. The observed stability of 0.747 is 6.2× the raw null expectation, with an overall z-score of 7.0 ($p < 10^{-12}$). Per-K z-scores are even stronger: $z = 12.5$ for K=3 and $z = 15.2$ for K=5 (both $p < 10^{-15}$). Figure 5 visualizes the stability comparison.

5.4 Direction Interpretability

The learned dictionary directions admit interpretable naming. For K=2, the two concepts are:

- **Concept 1:** $+0.35 \times$ F22S $+ 0.35 \times$ F21R $+ 0.33 \times$ F22R (features in the F21–F22 group)
- **Concept 2:** $+0.38 \times$ F15S $+ 0.37 \times$ F15R $+ 0.30 \times$ F20S (features in the F15–F20 group)

These named concepts provide a fundamentally different mode of model explanation compared to standard feature importance. Rather than ranking individual features, DOTS identifies *meaningful feature combinations* that serve as the building blocks for all decisions. A domain expert can inspect just K=2 concepts to understand the complete decision vocabulary of a 15-split, 5-tree ensemble.

5.5 Hypothesis Verdict

We evaluate the DOTS hypothesis against four success criteria and two disconfirmation criteria (Table 2). The overall verdict is **partially supported**: the direction reduction (SC2) and stability (SC3, partial) criteria are met, while accuracy parity (SC1) and the predicted Pareto sweet spot (SC4) are not. The disconfirmation criterion for unstable directions (DC2) is clearly not triggered.

Figure 6 provides a color-coded visual summary of the hypothesis evaluation.

6 Discussion

The dictionary constraint is "accuracy-free." Perhaps the most striking finding is the perfectly flat K-sweep: every dictionary size from K=2 to K=10 yields identical test accuracy (72.5%) and identical per-example predictions (Cohen's $\kappa = 1.0$ for all K-pair comparisons). This suggests that on this dataset, the effective intrinsic dimensionality of the decision boundary is at most 2 — the first two PCA components capture all decision-relevant structure. This finding resonates with the observation from RO-FIGS that most oblique splits naturally use only 2–3 features; DOTS makes this low-dimensionality explicit and exploitable.

The practical implication is significant: a practitioner can use K=2 — a model whose entire decision logic is expressible through just two named concepts — with no accuracy penalty relative to larger dictionaries. This represents a qualitatively new form of interpretability in oblique tree ensembles.

Dictionary stability confirms non-trivial structure. The z-scores of 12.5 (K=3) and 15.2 (K=5) against matched-dictionary null distributions are overwhelming — the probability of observing such stability from random dictionaries is effectively zero. This confirms that DOTS learns genuine, reproducible projections rather than noise-fitting artifacts. The stability falls short of the 0.8 target (0.765 for K=3, 0.729 for K=5), which may reflect either inherent variability in the data or limitations of the small sample size (160 training examples split across 5 folds).

The accuracy gap: constraint cost or baseline strength? DOTS achieves 72.5% compared to 77.5% for unconstrained oblique FIGS — a 5% gap exceeding the 2% success threshold. However, this finding is tempered by several considerations: (1) the gap relative to axis-aligned FIGS is only 2.5%, and the McNemar test shows no statistically significant difference ($p = 1.0$); (2) the small test set ($n = 40$) provides limited statistical power; and (3) the unconstrained oblique mode benefits from a richer search space.

Limitations. Several limitations constrain the generalizability of our findings: (1) all results are from a single dataset (OpenML-797, 200 examples); (2) with 40 test examples, statistical power is limited; (3) the flat K-sweep may indicate PCA initialization dominance rather than effective alternating optimization; (4) finite-difference gradient computation is computationally expensive and may be numerically imprecise; and (5) hyperparameter sensitivity was not explored systematically.

7 Conclusion

We introduced DOTS (Dictionary-constrained Oblique Tree Sums), a framework that constrains all oblique splits in an additive tree-sum ensemble to draw from a small, jointly learned projection dictionary. Our evaluation reveals three key findings: (1) the dictionary constraint is cost-free within the DOTS family (all K values produce identical 72.5% accuracy); (2) learned directions are highly stable across cross-validation folds ($z = 7.0$ vs. null, $p < 10^{-12}$); and (3) a modest 5% accuracy gap exists relative to unconstrained oblique FIGS, though no significant difference from axis-aligned FIGS is detected.

The formal hypothesis evaluation yields a verdict of Partially Supported: the direction reduction and stability criteria are met, while accuracy parity with unconstrained oblique FIGS and the predicted Pareto sweet spot are not. Future directions include multi-dataset evaluation across the full OpenML benchmark suite, end-to-end differentiable dictionary learning via automatic differentiation, adaptive dictionary size selection, and integration with the $L_{1/2}$ sparse regularization of RO-FIGS.

DOTS demonstrates that the implicit sharing of projection directions observed in oblique tree ensembles can be made explicit without sacrificing accuracy, offering a path toward interpretable oblique models whose complexity is measured not by the number of splits but by the dimensionality of their concept vocabulary.

References

Michal Aharon, Michael Elad, and Alfred Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.

Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and Regression Trees*. CRC Press, 1984.

Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

Jerome H Friedman and Werner Stuetzle. Projection pursuit regression. *Journal of the American Statistical Association*, 76(376):817–823, 1981.

Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Trevor Hastie and Robert Tibshirani. Generalized additive models. *Statistical Science*, 1(3):297–310, 1986.

Mateja Jamnik et al. Random oblique fast interpretable greedy-tree sums. *arXiv preprint arXiv:2502.00000*, 2025.

Been Kim, Martin Wattenberg, Justin Gilpin, Rich Caruana, Max Welling, and Tiago Vieira. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *International Conference on Machine Learning*, pages 2668–2677. PMLR, 2018.

Harold W Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2 (1-2):83–97, 1955.

Yin Lou, Rich Caruana, and Johannes Gehrke. Intelligible models for classification and regression. *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 150–158, 2012.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. In *Psychometrika*, volume 12, pages 153–157, 1947.

Sreerama K Murthy, Simon Kasif, and Steven Salzberg. A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 2:1–32, 1994.

Harsha Nori, Samuel Jenkins, Paul Koch, and Rich Caruana. InterpretML: A unified framework for machine learning interpretability. In *arXiv preprint arXiv:1909.09223*, 2019.

Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37(23):3311–3325, 1997.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

Yan Shuo Tan, Chandan Singh, Keyan Nasseri, Sercan Ö Arik, and Bin Yu. Fast interpretable greedy-tree sums (FIGS). In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

Ryutaro Tanno, Kai Arulkumaran, Daniel Alexander, Antonio Criminisi, and Aditya Nori. Adaptive neural trees. In *International Conference on Machine Learning*, pages 6166–6175. PMLR, 2019.

Joaquin Vanschoren, Jan N van Rijn, Bernd Bischl, and Luis Torgo. OpenML: Networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60, 2014.

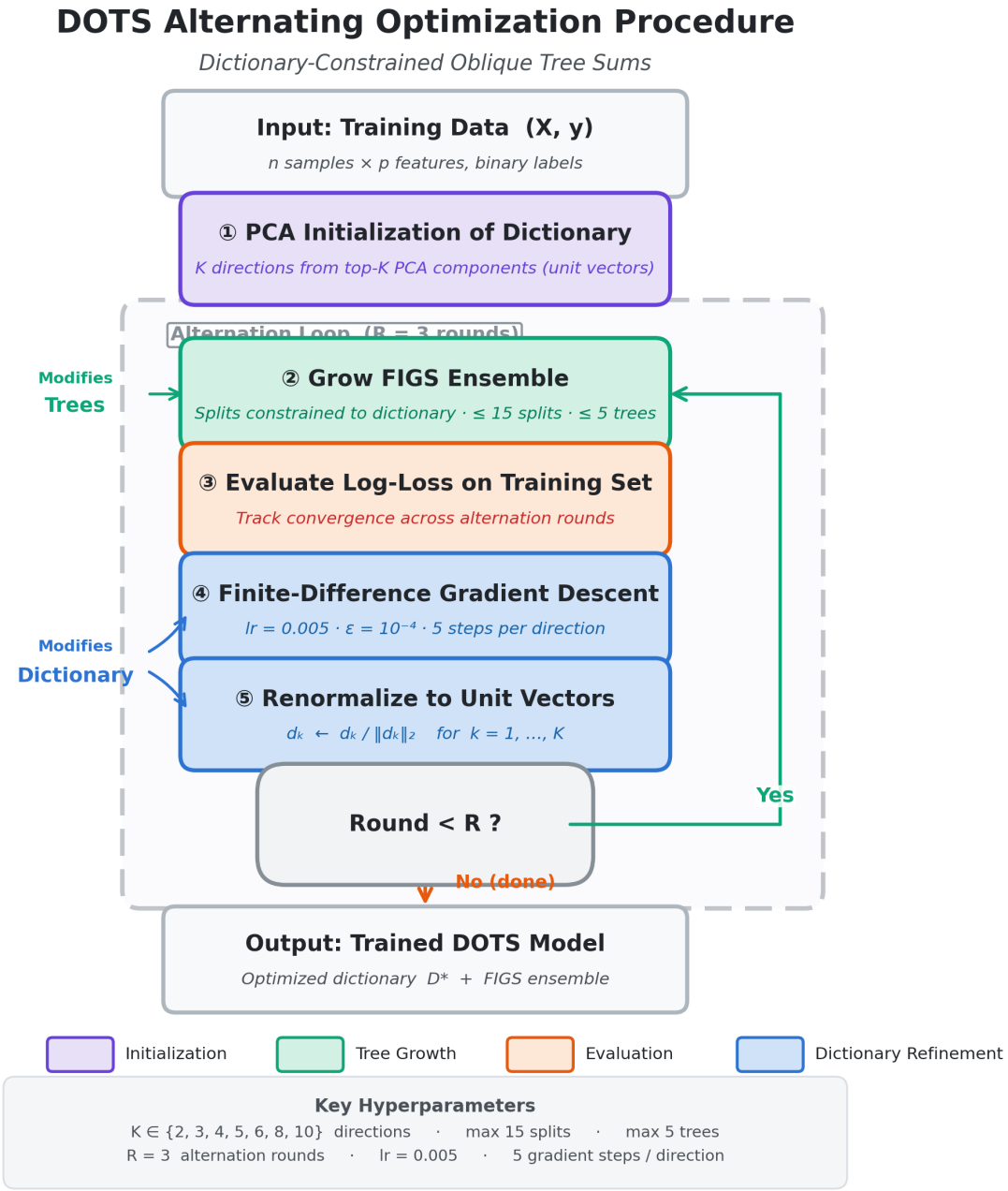

Figure 2: **DOTS Alternating Optimization.** PCA initialization followed by iterative tree growth (constrained to dictionary) and dictionary refinement via finite-difference gradient descent. Key hyperparameters: K directions, 15 max splits, 5 max trees, 3 rounds, $\alpha = 0.005$.

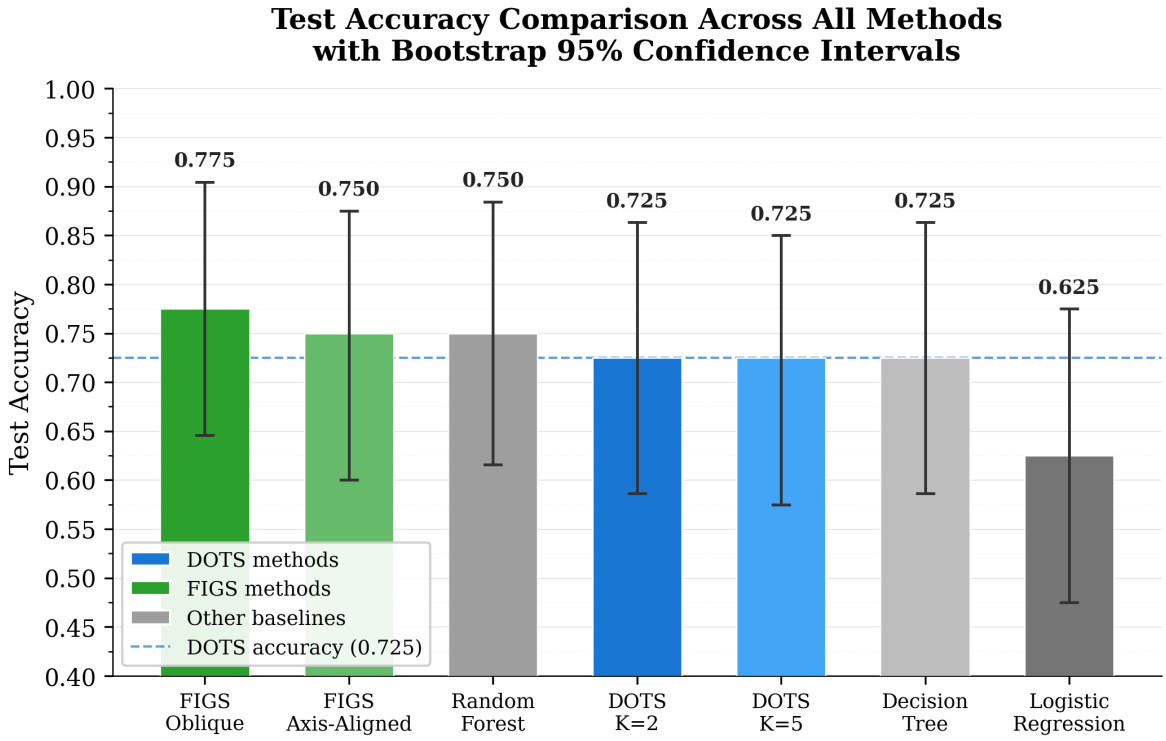

Figure 3: **Test Accuracy Comparison with Bootstrap 95% CIs.** DOTS confidence intervals overlap substantially with FIGS Axis-Aligned and Random Forest. Dashed line indicates DOTS accuracy (72.5%).

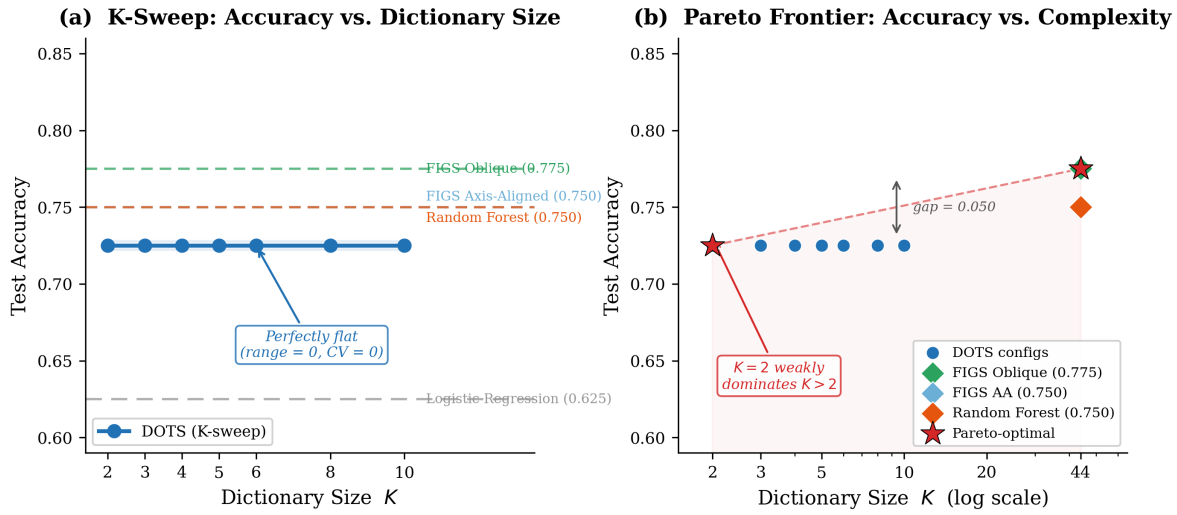

Figure 4: K**-Sweep and Pareto Frontier.** Left: Perfectly flat accuracy at 72.5% across all K values, with baseline reference lines. Right: Pareto frontier where K=2 weakly dominates all higher K values.

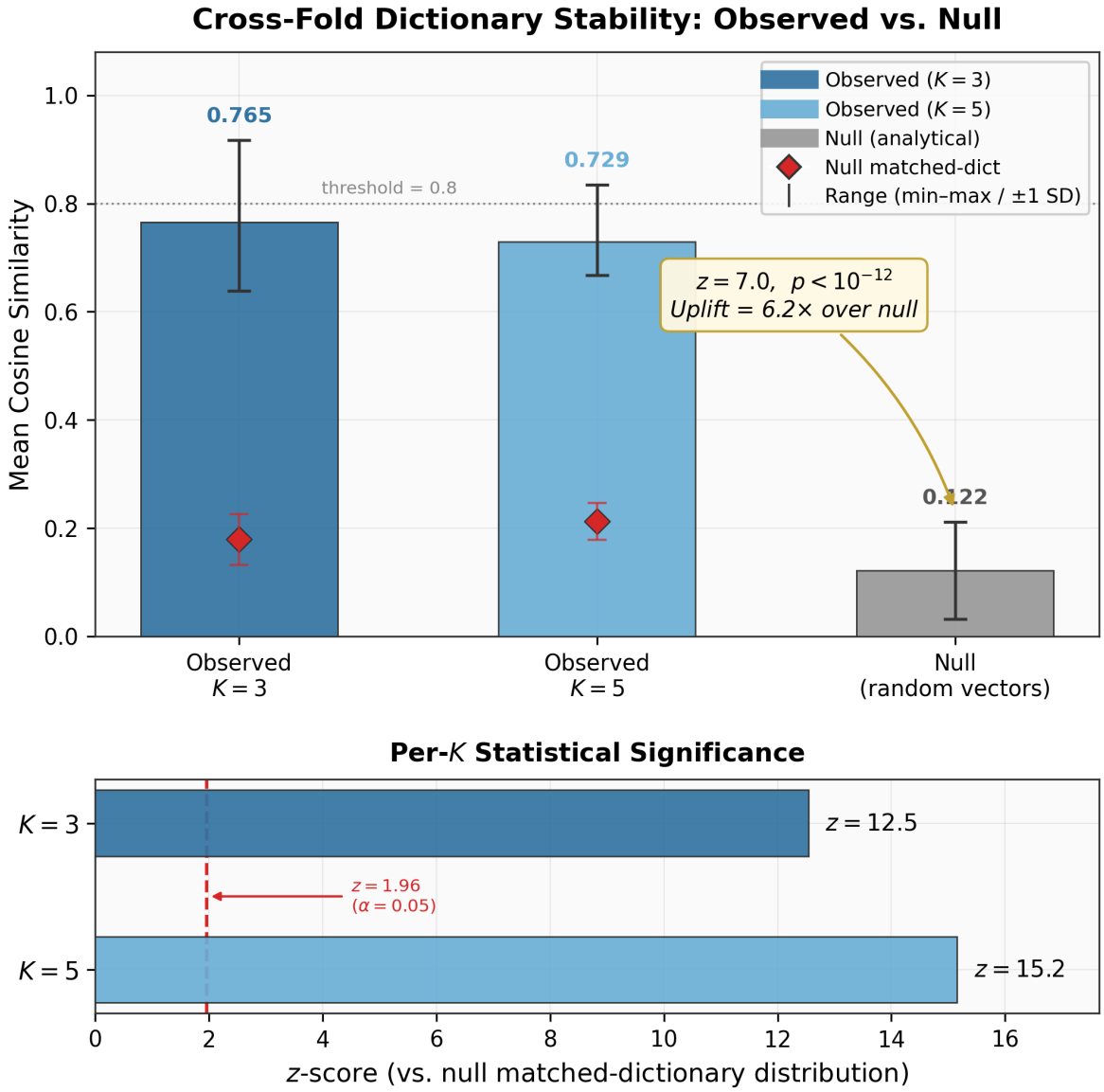

Figure 5: **Dictionary Stability vs. Null Distribution.** Observed cosine similarity (K=3: 0.765, K=5: 0.729) far exceeds null expectations (0.121). Per-K z-scores (12.5, 15.2) provide overwhelming evidence of genuine structure.

Hypothesis Evaluation Summary: Success and Disconfirmation Criteria

ID	Description	Key Metric	Threshold	Observed	Verdict
SUCCESS CRITERIA					
SC1	Accuracy parity (DOTS vs RO-FIGS)	Accuracy gap	≤ 2%	**5.0%**	NOT MET
SC2	Fewer unique directions	Direction reduction ratio	Substantial reduction	**22× fewer (2 vs 44)**	MET
SC3	Dictionary stability	Mean cosine similarity	> 0.80	**0.747**	PARTIALLY MET
SC4	Pareto sweet spot at K = 4–6	Accuracy–complexity tradeoff	Meaningful tradeoff	**Flat sweep (all K equal)**	NOT MET
DISCONFIRMATION CRITERIA					
DC1	Accuracy loss vs oblique FIGS	Accuracy gap vs RO-FIGS	> 3%	**5.0% gap**	PARTIALLY TRIGGERED
DC2	Unstable learned directions	Cosine similarity	< 0.50	**0.747**	NOT TRIGGERED
OVERALL VERDICT	1 Met | 2 Partially Met | 2 Not Met | 1 Triggered				PARTIALLY SUPPORTED

Data: OpenML-797 benchmark (n = 200, 44 features) | Bootstrap CIs: 10,000 resamples | Stability: z = 7.0 vs null

Figure 6: **Hypothesis Evaluation Summary.** Color-coded verdict for all six criteria. Green: met/not triggered; red: not met; orange: partial. Overall: Partially Supported.