Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Knowledge-Guided Explanations for Tree Ensemble Models via Levels-Structure Values

Hyunwook Koh

hyunwook.koh@stonybrook.edu

The State University of New York, Korea https://orcid.org/0000-0001-6893-7164

Research Article

Keywords: Levels-Structure Value, Tree Ensemble Models, Game-Theoretic Feature Attribution, Knowledge-Guided Explanation, Tree-Aware Dynamic Programming, Human Microbiome

Posted Date: November 4th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11600957/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

KNOWLEDGE-GUIDED EXPLANATIONS FOR TREE ENSEMBLE MODELS VIA LEVELS-STRUCTURE VALUES

Hyunwook Koh

Department of Applied Mathematics and Statistics
The State University of New York, Korea
Incheon, South Korea
hyunwook.koh@stonybrook.edu

ABSTRACT

Tree ensemble models, such as gradient boosting machines and random forests, have achieved strong predictive performance on tabular data by capturing complex nonlinear relationships and feature interactions. However, their flexibility often comes at the cost of interpretability, especially in scientific applications where features are organized by domain knowledge into *a priori* multi-level structures. Existing game-theoretic explanation methods, including TreeSHAP and TreeOwen, provide principled attribution frameworks based on the Shapley and Owen values; yet, they address either individual features or a single grouping level. In this paper, I introduce Tree Levels-Structure Value (TLSV), a knowledge-guided, principled, and efficient framework for computing the levels-structure value—which generalizes the Owen value from a single *a priori* partition to a chain of nested partitions—for tree ensemble models. TLSV therefore accommodates multi-level coalition structures, such as features within groups and groups within higher-level supergroups, while preserving coherent attributions across all resolutions, with single-level Owen-value attribution recovered as a special case. Algorithmically, TLSV combines hierarchy-aware recursion with tree-aware dynamic programming to avoid exhaustive subset enumeration within terminal feature groups. Its exact algorithm returns levels-structure values for the tree-aware cooperative game induced by the fitted ensemble. For high-branching hierarchies, I also develop a recursive Monte Carlo estimator that provides scalable approximation with unbiasedness and consistency guarantees. Through simulation studies, I assess attribution accuracy, finite-budget Monte Carlo behavior, numerical coherence, and computational scalability. Finally, I illustrate TLSV using microbiome data from an immunotherapy study, showing how taxonomically nested microbial features contribute coherently to predictions of treatment response.

Keywords Levels-Structure Value · Tree Ensemble Models · Game-Theoretic Feature Attribution · Knowledge-Guided Explanation · Tree-Aware Dynamic Programming · Human Microbiome

1 Introduction

Shapley-value-based explanations have become a central framework for interpreting modern machine-learning models. Originating from cooperative game theory (Shapley, 1953) and introduced into model explanation by Štrumbelj and Kononenko (2010, 2014), the Shapley value was later unified and popularized through SHAP (SHapley Additive exPlanations) (Lundberg and Lee, 2017). SHAP provides a principled additive attribution framework for both local explanations at the level of individual observations and global summaries of feature importance (Lundberg and Lee, 2017). A major computational advance came with TreeSHAP (Lundberg et al., 2020), which uses tree-aware dynamic programming over decision-tree paths to compute exact Shapley values efficiently for tree ensemble models, including gradient boosting machines (Friedman, 2001) and random forests (Breiman, 2001). Unlike the interventional SHAP formulation, which integrates absent features over a background marginal distribution and thereby breaks dependence between observed and absent features, the path-dependent TreeSHAP formulation uses the fitted tree structure and

empirical node-cover ratios to construct its coalition-value evaluator, thereby reflecting feature dependence encoded in the tree (Lundberg and Lee, 2017; Lundberg et al., 2020). This combination of predictive flexibility and computationally tractable game-theoretic attribution has made tree ensemble models particularly attractive for scientific tabular data, where nonlinear effects, feature interactions, and structured dependence are common.

In many scientific applications, however, features are not merely an unordered collection of individual variables. Rather, they often possess meaningful *a priori* structure supplied by domain knowledge. In microbiome studies, microbial features are commonly organized according to hierarchical taxonomic annotations, such as species, genera, families, orders, classes, and phyla (Federhen, 2012). In genomics, genes are commonly grouped into pathways or functional modules; in metabolomics, molecular features are often organized into biochemical classes; and in natural language processing, words are naturally nested within phrases, sentences, and documents. In such settings, interpretability is required not only at the individual-feature level but also at multiple structured resolutions. For example, a scientist may want to identify which microbial species drive a prediction, which genera summarize those species, and which higher-level taxonomic clades account for the overall signal. Standard Shapley values do not encode such domain-knowledge structure and therefore can yield attributions that are locally valid but difficult to organize coherently across scientifically meaningful levels.

The Owen value (Owen, 1977) extends the Shapley value to games with a single *a priori* partition of the players. It first allocates total value among groups and then distributes each group contribution among the features within that group. This two-stage allocation preserves the fairness principles of the Shapley value while respecting a predefined group structure. TreeOwen (Koh, 2026) was developed to bring this idea to tree ensemble models (Friedman, 2001; Breiman, 2001), combining single-level Owen-value attribution with tree-aware coalition-value evaluation and enabling structured explanations across a single grouping level, such as genera and species in microbiome applications. Nevertheless, many scientific feature systems are not flat partitions but nested hierarchies. A microbial taxonomy, for example, is not simply feature $\preceq$ group; it is naturally organized as species $\preceq$ genera $\preceq$ families $\preceq$ orders $\preceq$ classes $\preceq$ phyla (Federhen, 2012). Collapsing such a hierarchy into one grouping level discards important intermediate resolutions, whereas computing separate explanations at different levels can produce incoherent summaries that do not add up across the hierarchy.

A natural game-theoretic solution for this setting is the levels-structure value of Winter (1989), which generalizes the Owen value from a single *a priori* partition to a chain of nested partitions. It averages feature marginal contributions over permutations that are consistent with the hierarchy: features within the same lower-level block remain contiguous, lower-level blocks within the same higher-level block remain contiguous, and the allocation proceeds recursively across the hierarchy. This construction reduces to the Shapley value (Shapley, 1953) when no intermediate grouping is specified and to the Owen value (Owen, 1977) when a single grouping level is specified. Importantly, in the tree-ensemble explanation setting considered here, this nested allocation structure yields a useful form of multi-resolution coherence: internal-node attributions can be obtained by aggregating the feature-level allocations within each node, so that each internal-node attribution equals the sum of its descendant feature-level attributions. Thus, feature-, group-, and supergroup-level explanations are not separate *post hoc* summaries but different resolutions of the same underlying allocation.

In this paper, I introduce Tree Levels-Structure Value (TLSV), a knowledge-guided, principled, and efficient framework for computing the levels-structure value (Winter, 1989) in tree ensemble models (Friedman, 2001; Breiman, 2001). The value itself is Winter's; the contribution here is its tree-aware cooperative game together with exact and approximate algorithms that enable scalable computation. TLSV thereby addresses multi-level coalition structures beyond individual-feature attribution (Lundberg et al., 2020) and single-grouping-level attribution (Koh, 2026) for tree ensemble models, computing the levels-structure value directly on the original hierarchy without binary refinement or rebalancing. Algorithmically, TLSV combines hierarchy-aware local Shapley recursion with tree-aware dynamic programming (Lundberg et al., 2020) for context-conditioned local Shapley evaluation, so that the levels-structure allocation is respected across all resolutions of the hierarchy. The exact algorithm returns levels-structure values for the tree-aware cooperative game induced by the fitted ensemble while avoiding exhaustive subset enumeration within terminal feature groups. For high-branching hierarchies, I also develop a recursive Monte Carlo estimator that samples hierarchy-level child orderings while preserving exact tree-aware local Shapley evaluation within terminal feature groups. At any predetermined budget, this estimator is unbiased; it is L^2-consistent as the local budgets increase and converges almost surely under suitable increasing-budget schedules. A hybrid exact–Monte Carlo algorithm further evaluates low-degree hierarchy nodes exactly and applies Monte Carlo sampling only at high-degree nodes, concentrating approximation where exact hierarchy-level computation is most costly.

TLSV is designed to support both theoretical coherence and practical scientific interpretation. The method yields local attributions for individual predictions, global importance summaries obtained by aggregating local explanations across observations, and hierarchy-aware visualizations that display contributions at multiple resolutions. Through simulation experiments, I evaluate attribution accuracy, finite-budget Monte Carlo behavior, numerical coherence, and computational scalability. I then illustrate its practical utility using shotgun metagenomic microbiome data from an immunotherapy study (Limeta et al., 2020), showing how microbial features organized by taxonomic hierarchy contribute to predictions of treatment response to immune checkpoint inhibitor therapy. This application demonstrates

how TLSV can identify important species-level signals while simultaneously summarizing them at higher taxonomic levels, thereby providing coherent explanations that align with the structure used by domain scientists.

The remainder of the paper is organized as follows. Section 2 reviews individual-feature and single-level group-structured attribution methods, including tree-specific formulations, and positions TLSV relative to these existing approaches. Section 3 introduces the cooperative-game formulation, defines the levels-structure value, develops the exact, Monte Carlo, and hybrid TLSV algorithms with their theoretical properties, and describes the associated visualization, importance measures, and software tools. Section 4 presents the simulation studies and real-data microbiome application. Section 5 summarizes the main findings, discusses practical considerations, and outlines future directions.

2 Related Work

For individual-feature attribution without a prespecified grouping structure, Shapley-based explanation methods differ primarily in how absent features are integrated out and the resulting coalition values are evaluated. KernelSHAP is model-agnostic and, in its standard implementation, replaces the conditional distribution of absent features by their marginal distribution, effectively adopting an interventional formulation and an independence approximation between observed and absent features (Lundberg and Lee, 2017; Janzing et al., 2020; Aas et al., 2021). TreeSHAP instead exploits the recursive structure of fitted trees and, in its path-dependent formulation, defines the coalition game from tree paths and node-cover statistics supplied by the fitted model, without separately estimating a joint or conditional feature distribution (Lundberg et al., 2020). Conditional KernelSHAP accounts for feature dependence more directly by estimating the relevant conditional distributions, but requires additional distributional or smoothing assumptions and estimation choices; proposed approaches include multivariate Gaussian and Gaussian-copula models, nonparametric conditional estimation, and combinations thereof (Aas et al., 2021). TLSV adopts the path-dependent tree-aware formulation because it supplies a computationally tractable, model-internal coalition evaluator without requiring a separately estimated conditional feature distribution and is naturally suited to the repeated coalition-value evaluations required for tree ensemble attribution.

For a single grouping level, existing methods can be broadly divided into Shapley-value-based and Owen-value-based approaches. Among the former, the summed Shapley attribution first computes ordinary feature-level Shapley values—using, for example, KernelSHAP (Lundberg and Lee, 2017), TreeSHAP (Lundberg et al., 2020), conditional KernelSHAP (Aas et al., 2021), or another Shapley-value estimator—and then sums them within each group. Although simple and broadly applicable, this remains a *post hoc* aggregation of an unstructured feature game. In the presence of interactions crossing group boundaries, the resulting group totals need not coincide with the Shapley values of the groups in the corresponding quotient game and may fail to satisfy symmetry or null-player requirements defined relative to the prespecified grouping (Koh, 2026). The second Shapley-value-based approach, groupShapley (Jullum et al., 2021), instead treats the groups themselves as players and computes their Shapley values directly. This yields principled group-level attributions but does not further allocate each group contribution among its member features. In contrast, the Owen value provides a linked allocation at both the group and feature levels for a single *a priori* partition (Owen, 1977). TreeOwen adapts this Owen-value formulation to tree ensemble models and reduces the outer computational burden by organizing the groups through an auxiliary balanced binary tree (Koh, 2026). The relationship between TreeOwen and TLSV is worth stating precisely. For a single partition the two target the same value, and TLSV reproduces it; beyond a single partition they differ in target, not merely in implementation. The auxiliary binary tree of TreeOwen is a computational construction chosen to accelerate a one-level Owen computation, not a scientific hierarchy supplied by the user, and imposing it on a multiway or depth-unbalanced taxonomy changes the set of admissible permutations and therefore the value being computed. TreeOwen consequently does not compute the levels-structure value associated with an arbitrary multi-level, multiway, or depth-unbalanced hierarchy. Its exact within-group computation also retains exponential dependence on group size, with Monte Carlo approximation used for larger groups.

Among tree-specific methods most directly related to TLSV, TreeSHAP (Lundberg et al., 2020) and TreeOwen (Koh, 2026) address individual-feature and single-grouping-level attribution, respectively, and therefore do not define a unified allocation across multiple nested resolutions. TLSV addresses this limitation by computing Winter's axiomatically characterized levels-structure value directly on the original user-specified hierarchy, without binary refinement or rebalancing (Winter, 1989). The allocation follows the original child structure at every hierarchy node and remains coherent across all resolutions: feature-level contributions aggregate to the corresponding group, supergroup, and higher-level contributions within the same game-theoretic allocation. This coherence is particularly important when interactions cross group boundaries because, unlike a *post hoc* summation of ordinary Shapley values, the resulting block contributions remain consistent with the quotient games induced by the hierarchy. Building on the path-dependent tree-aware evaluator described above, TLSV also uses dynamic programming within terminal feature groups to replace exponential dependence on within-group cardinality with polynomial computation governed by the size and depth of

the fitted trees (Lundberg et al., 2020). Its Monte Carlo and hybrid algorithms consequently approximate only the costly hierarchy-level ordering calculations while retaining exact tree-aware allocation within terminal feature groups.

3 Methods

This section presents the methodology of TLSV. TLSV decomposes a model prediction, relative to its baseline value, into feature-level contributions that remain coherent when aggregated to any internal node of the hierarchy. The construction combines two ingredients: the levels-structure value of Winter (1989) and a context-conditioned tree-aware path recursion for exact local Shapley evaluation (Lundberg et al., 2020). The section first defines the local cooperative game and reviews the Shapley and Owen values. It then introduces the levels-structure value and develops exact, Monte Carlo, and hybrid algorithms for its computation. Finally, it describes the associated visualization methods, importance measures, and software implementation.

3.1 Cooperative Game for Tree Ensemble Models

Let $N := \{1, \dots, p\}$ denote the set of input features. Let $f : \mathbb{R}^p \to \mathbb{R}$ be a trained tree ensemble, such as a gradient boosting machine (Friedman, 2001) or a random forest (Breiman, 2001), written as

$$f(x) = \sum_{m=1}^{B} f^{(m)}(x), \tag{1}$$

where $f^{(m)}$ is the mth tree and B is the number of trees. Let M denote the total number of nodes across all trees. For a target observation $x \in \mathbb{R}^p$, the local explanation problem is represented as a cooperative game (N, v_x), where $v_x(S)$ is the model value associated with the coalition $S \subseteq N$ when the features in S are fixed to their observed values.

A standard formulation uses the conditional-expectation value function

$$v_x(S) = \mathbb{E}[f(X) \mid X_S = x_S], \qquad S \subseteq N, \tag{2}$$

so that $v_x(\emptyset) = \mathbb{E}[f(X)]$ and $v_x(N) = f(x)$ (Štrumbelj and Kononenko, 2010, 2014; Lundberg and Lee, 2017). In tree ensemble models, this conditional expectation is commonly approximated by the path-dependent tree-aware semantics of Lundberg et al. (2020): when a split feature belongs to S, the evaluator follows the child branch determined by the observed value of that feature in x; when the split feature is absent from S, it averages over the child branches using model-native node-cover weights. This yields a computationally efficient, path-dependent tree-aware approximation to (2) whose averaging semantics are determined by the fitted tree representation rather than by a separately estimated conditional feature distribution (Lundberg et al., 2020). The levels-structure construction and the Monte Carlo estimator below are stated for a fixed finite set function v_x, and their guarantees are understood with respect to the chosen evaluator.

3.2 Shapley Value and Owen Value

For a game (N, v_x), the Shapley value of feature $i \in N$ is

$$\phi_i(v_x) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(p - |S| - 1)!}{p!} \left\{ v_x(S \cup \{i\}) - v_x(S) \right\}. \tag{3}$$

It is the unique allocation satisfying efficiency, linearity, symmetry, and the null-player property (Shapley, 1953).

Suppose now that the feature set is partitioned into K disjoint groups $\mathcal{G} = \{G_1, \dots, G_K\}$, with group index set $\mathcal{K} := \{1, \dots, K\}$. The Owen value extends the Shapley value to this single-partition setting (Owen, 1977). For $i \in G_k$, it is

$$\psi_i(v_x, \mathcal{G}) = \sum_{U \subseteq \mathcal{K} \setminus \{k\}} \frac{|U|!\,(K - |U| - 1)!}{K!}\, \phi_i^{G_k}(v_{x, C_U}), \qquad C_U := \bigcup_{\ell \in U} G_\ell, \tag{4}$$

where $v_{x,C}(S) := v_x(C \cup S)$ for $S \subseteq G_k$, and

$$\phi_i^{G_k}(v_{x,C}) = \sum_{S \subseteq G_k \setminus \{i\}} \frac{|S|!\,(|G_k| - |S| - 1)!}{|G_k|!} \left\{ v_x(C \cup S \cup \{i\}) - v_x(C \cup S) \right\} \tag{5}$$

is the Shapley value of feature i within the context-restricted game on G_k. Thus, the Owen value first allocates value among groups and then distributes each group contribution among the features within that group. When every group is a singleton, it reduces to the Shapley value (Shapley, 1953). TreeOwen (Koh, 2026) computes this single-partition attribution problem for tree ensemble models. The levels-structure value defined next generalizes the Owen value to a chain of nested partitions, and TLSV computes it for the tree-aware game; on a single partition the two coincide.

3.3 Hierarchical Coalition Structure

A hierarchical feature structure is represented by a chain of nested partitions

$$\mathcal{G}^{(0)} \preceq \mathcal{G}^{(1)} \preceq \cdots \preceq \mathcal{G}^{(L)} \preceq \mathcal{G}^{(L+1)}, \tag{6}$$

where $\mathcal{G}^{(0)} = \{\{i\} : i \in N\}$ is the singleton partition, $\mathcal{G}^{(L+1)} = \{N\}$ is the grand coalition, and $\mathcal{G}^{(\ell)} \preceq \mathcal{G}^{(\ell+1)}$ means that every block of $\mathcal{G}^{(\ell)}$ lies within a block of $\mathcal{G}^{(\ell+1)}$. The L intermediate partitions are the user-specified levels of domain knowledge. In a microbiome application with species-level features, for example, these levels correspond to genera, families, orders, classes, and phyla.

Equivalently, the nested partitions define a hierarchy tree $\mathcal{H}$ whose root is the grand coalition N, whose leaves are individual features, and whose internal nodes are the blocks appearing in the intermediate partitions. When the coarsest user-specified level contains several blocks—for instance, several phyla—these blocks are not separate roots but children of the single grand-coalition root. For a node u in $\mathcal{H}$, let $A(u) \subseteq N$ be the set of features contained in the subtree rooted at u, and let $\mathrm{ch}(u) = \{u_1, \ldots, u_d\}$ denote its children. The sets $A(u_1), \ldots, A(u_d)$ form a partition of $A(u)$. A node whose children are all individual features is called a feature-parent node. Let $\mathcal{F}(\mathcal{H})$ denote the set of feature-parent nodes of $\mathcal{H}$ and put $K_1 := |\mathcal{F}(\mathcal{H})|$. A bare feature leaf $\{i\}$ is read as a degenerate feature-parent node with $A(\{i\}) = \{i\}$, so that the constructions below are defined at every node of an arbitrary rooted hierarchy tree, including a node whose children mix feature leaves and hierarchy nodes. Equivalently, such a leaf may be wrapped in a singleton feature-parent node; because a degree-one node admits only one child ordering, this normalization changes neither $\Omega(\mathcal{H})$ nor the resulting attribution, and the two representations return identical values. Consequently every feature belongs to exactly one feature-parent node and $\{A(u) : u \in \mathcal{F}(\mathcal{H})\}$ is a partition of N. When $L \geq 1$ and the hierarchy is depth balanced, $\mathcal{F}(\mathcal{H})$ consists of the finest user-specified groups; when $L = 0$, the root itself is the only feature-parent node.

Although (6) is written as a chain of nested partitions, the hierarchy tree $\mathcal{H}$ is the operative representation, and all statements below are phrased in terms of $\mathcal{H}$ and $\mathcal{F}(\mathcal{H})$ rather than in terms of the level index ℓ. In a depth-unbalanced hierarchy the feature-parent nodes need not all lie at a common level, so "the finest level" is not a well-defined object, whereas $\mathcal{F}(\mathcal{H})$ always is. Depth-unbalanced taxonomies with missing or unresolved annotations can be handled by carrying blocks forward across adjacent levels, or equivalently by inserting degree-one nodes. Such nodes do not change $\Omega(\mathcal{H})$ or the resulting attribution, since they introduce no nontrivial child ordering. Together with the singleton wrapping described above, this yields a normalized *feature-parent form* of any rooted hierarchy, in which every leaf of $\mathcal{H}$ is a feature and every feature's parent is a feature-parent node; the algorithms of Sections 3.6–3.7 operate on this form.

3.4 Levels-Structure Value

I now define the levels-structure value used by TLSV. Following the levels-structure formulation of Winter (1989), a permutation π of the features is called $\mathcal{H}$-consistent if, for every node u in the hierarchy, the features in $A(u)$ appear consecutively in π. Equivalently, an $\mathcal{H}$-consistent permutation is generated by independently ordering the children of every internal node and then concatenating the resulting suborderings recursively. Let $\Omega(\mathcal{H})$ denote the set of all such permutations. Since each internal node contributes an independent ordering of its children,

$$|\Omega(\mathcal{H})| = \prod_{u \in \mathrm{int}(\mathcal{H})} |\mathrm{ch}(u)|!, \tag{7}$$

where $\mathrm{int}(\mathcal{H})$ is the set of internal nodes. For a permutation π and feature i, let $P_i^\pi := \{j \in N : \pi(j) < \pi(i)\}$ be the predecessor set of i.

Definition 1 (Levels-structure value)**.** *For a game (N, v_x) and hierarchy $\mathcal{H}$, the levels-structure value of feature $i \in N$ is*

$$\Psi_i^{\mathcal{H}}(v_x) := \frac{1}{|\Omega(\mathcal{H})|} \sum_{\pi \in \Omega(\mathcal{H})} \left\{ v_x(P_i^\pi \cup \{i\}) - v_x(P_i^\pi) \right\}. \tag{8}$$

Definition 1 is the levels-structure value of Winter (1989) specialized to hierarchical feature attribution. It can also be written as a recursive local-Shapley allocation over the hierarchy. Consider an internal node u with children $u_1, \ldots, u_d$. For child u_j and sibling subset $T \subseteq \{1, \ldots, d\} \setminus \{j\}$, define the local Shapley weight

$$w_T^{(d)} := \frac{|T|!\,(d - |T| - 1)!}{d!}. \tag{9}$$

If the current outer context is $C \subseteq N \setminus A(u)$, then the context passed to child u_j when the sibling subset T precedes it is

$$C_T := C \cup \bigcup_{t \in T} A(u_t). \tag{10}$$

Applying this recursion from the root downwards yields, for every node u of $\mathcal{H}$, a multiset $\mathcal{R}_u$ of context–weight pairs (C, w) with which the allocation reaches u: each pair corresponds to one sequence of sibling-subset choices along the root-to-u path, C being the union of the outer context and the accumulated preceding sibling blocks and w the product of the corresponding local weights (9). By construction $\mathcal{R}_{\text{root}} = \{(\emptyset, 1)\}$ and $\sum_{(C,w)\in\mathcal{R}_u} w = 1$ for every u. For a feature-parent node $u \in \mathcal{F}(\mathcal{H})$ and $i \in A(u)$,

$$\Psi_i^{\mathcal{H}}(v_x) = \sum_{(C,w)\in\mathcal{R}_u} w\, \phi_i^{A(u)}(v_{x,C}), \tag{11}$$

where $\phi_i^{A(u)}(v_{x,C})$ is the within-group Shapley value (5) with G_k replaced by $A(u)$, and where the degenerate case $A(u) = \{i\}$ gives $\phi_i^{\{i\}}(v_{x,C}) = v_x(C \cup \{i\}) - v_x(C)$. When $L = 0$ the root is the only feature-parent node, $\mathcal{R}_{\text{root}} = \{(\emptyset, 1)\}$, and (11) reduces to the Shapley value. The weights in (9) are exactly the probabilities that a given subset of siblings precedes a child under a uniform random ordering of the children of u. Therefore, the recursive form (11) is equivalent to the permutation average (8).

The same multiset $\mathcal{R}_u$ defines the attribution that the levels-structure allocation assigns to the *block* $A(u)$, at the resolution of u and for an arbitrary node u. Define

$$\Gamma_u(v_x) := \sum_{(C,w)\in\mathcal{R}_u} w\, \{v_x(C \cup A(u)) - v_x(C)\}, \tag{12}$$

the weighted average of the marginal contribution of the whole block $A(u)$ over the contexts with which the hierarchy recursion reaches u, so that $\Gamma_{\text{root}}(v_x) = v_x(N) - v_x(\emptyset)$. The block quantities Γ_u are induced by context-specific quotient-game Shapley allocations and are therefore not merely bookkeeping totals: for an internal node u with children $u_1, \ldots, u_d$ and a fixed outer context C, define the local quotient game on the children by

$$q_{u,C}(T) := v_x\left(C \cup \bigcup_{t\in T} A(u_t)\right), \qquad T \subseteq \{1, \ldots, d\}. \tag{13}$$

Then the aggregate allocation that the recursion sends into child u_j under context C is exactly the ordinary Shapley value of player j in $q_{u,C}$, and $\Gamma_{u_j}(v_x)$ is the $\mathcal{R}_u$-weighted average of those Shapley values. Proposition 1(v) below states that the feature-level attributions inside $A(u)$ sum exactly to $\Gamma_u(v_x)$.

Proposition 1 (Basic properties)**.** *For any hierarchy $\mathcal{H}$, $\Psi^{\mathcal{H}}$ coincides with the levels-structure value of Winter (1989). Consequently, it satisfies the following properties.*

(i) Efficiency: $\sum_{i\in N} \Psi_i^{\mathcal{H}}(v_x) = v_x(N) - v_x(\emptyset)$, which equals $f(x) - \mathbb{E}[f(X)]$ under (2)*.*

(ii) Linearity: $\Psi^{\mathcal{H}}(\alpha v + \beta w) = \alpha\Psi^{\mathcal{H}}(v) + \beta\Psi^{\mathcal{H}}(w)$.

(iii) Null player property: if $v_x(S \cup \{i\}) = v_x(S)$ for all $S \subseteq N \setminus \{i\}$, then $\Psi_i^{\mathcal{H}}(v_x) = 0$.

(iv) Level symmetry: let u be an internal node with children $u_1, \ldots, u_d$ and let $q_{u,C}$ be the local quotient game (13)*. If two children u_j and $u_{j'}$ are symmetric in $q_{u,C}$ for* every *context C occurring in $\mathcal{R}_u$, that is, if*

$$q_{u,C}(T \cup \{j\}) = q_{u,C}(T \cup \{j'\}) \quad \text{for all such } C \text{ and all } T \subseteq \{1, \ldots, d\} \setminus \{j, j'\},$$

then $\Gamma_{u_j}(v_x) = \Gamma_{u_{j'}}(v_x)$, so the two children receive equal aggregate allocation. A sufficient condition that does not refer to $\mathcal{R}_u$ is that $A(u_j)$ and $A(u_{j'})$ be symmetric blocks of v_x, that is, $v_x(S \cup A(u_j)) = v_x(S \cup A(u_{j'}))$ for every $S \subseteq N \setminus (A(u_j) \cup A(u_{j'}))$.

(v) Multi-resolution coherence: for every node u of $\mathcal{H}$,

$$\sum_{i\in A(u)} \Psi_i^{\mathcal{H}}(v_x) = \Gamma_u(v_x),$$

with Γ_u defined in (12)*. The feature-level attributions contained in a block therefore sum to the block-level value that the levels-structure allocation assigns to that block through the quotient games* (13)*.*

(vi) Reductions: if $L = 0$, $\Psi^{\mathcal{H}}$ reduces to the Shapley value (3) *(Shapley, 1953); if $L = 1$, $\Psi^{\mathcal{H}}$ reduces to the Owen value* (4) *for the corresponding partition (Owen, 1977). Consequently* TLSV *applied to a one-level hierarchy targets the same quantity as single-level Owen attribution for tree ensembles (Koh, 2026), and the two agree there; the hierarchy recursion of Section 3.6 is what extends the target beyond a single partition.*

A formal proof of Proposition 1, including the equivalence between the permutation and recursive representations, is provided in Section S1 of the Supplementary Information. Properties (i)–(iii) follow directly from Definition 1, since they hold marginal contribution by marginal contribution along every hierarchy-consistent permutation; properties (iv)–(vi) follow from the recursive representation (11)–(13).

Winter (1989) additionally establishes an axiomatic characterization for games with a levels structure: the levels-structure value is the unique value satisfying levels-structure analogues of efficiency, additivity, the null-player property, and symmetry, where symmetry is imposed within each level of the structure. Properties (i)–(iii) and the level-symmetry statement in (iv) correspond to these axiomatic requirements when expressed in the present notation; the original formulation of the axioms and the precise statement of the uniqueness theorem are given by Winter (1989), and are not restated here. Because $\Psi^{\mathcal{H}}$ is the levels-structure value of $\mathcal{H}$ by the first assertion of Proposition 1, it inherits that uniqueness. This is what separates the levels-structure value from an arbitrary rule that happens to be additive across resolutions. Within Winter's axiomatic framework it is not merely *a* coherent multi-resolution allocation, but the uniquely characterized one for the fixed hierarchy.

The multi-resolution coherence property (v) is central for scientific interpretation. It ensures that feature-level attributions can be summed to obtain group-, supergroup-, or higher-level attributions without recomputing explanations separately at each level and without creating inconsistencies across resolutions. It should be read together with (12). The content of (v) is not that such a summation is well defined—any feature-level vector can be summed over a block—but that the resulting total coincides with the block-level value $\Gamma_u(v_x)$ that the levels-structure allocation assigns to $A(u)$ through the quotient games (13). Accordingly, a numerical check of the identity $\sum_{i \in A(u)} \Psi_i^{\mathcal{H}} = \Gamma_u$ requires computing Γ_u from (12) independently of the feature-level output; simply re-summing the reported feature-level attributions is a bookkeeping identity that holds for any vector and tests nothing.

3.5 Tree-Aware Context-Conditioned Shapley Evaluation

TLSV requires repeated local Shapley computations of the form

$$\phi_i^G(v_{x,C}), \qquad v_{x,C}(S) := v_x(C \cup S), \quad S \subseteq G,$$

where G is a feature-parent group and $C \subseteq N \setminus G$ is the accumulated outer context produced by the hierarchy recursion. A direct implementation enumerates all $2^{|G|}$ subsets of G and evaluates $v_x(C \cup S)$ for each subset. For tree ensemble models with the path-dependent tree-aware value function, however, this restricted Shapley problem can be solved exactly by a context-conditioned tree-aware path recursion (Lundberg et al., 2020).

By ensemble additivity (1), it is sufficient to describe the computation for a single tree. For clarity, the description and Algorithm 2 are written for binary trees, as used by the supported XGBoost (Chen and Guestrin, 2016), LightGBM (Ke et al., 2017), and ranger (Wright and Ziegler, 2017) implementations; multiway splits can be handled by the analogous cover-weighted averaging over all children. Consider a tree node t. If t is a leaf with value μ_t, the leaf contribution is distributed to the active features from G that appear in the current path object, using the corresponding Shapley path weights. If t is an internal node splitting on feature $j(t)$, the traversal distinguishes three cases:

$$j(t) \in C, \qquad j(t) \in G, \qquad j(t) \notin C \cup G.$$

If $j(t) \in C$, the split feature is fixed as present under every coalition in the restricted game $v_{x,C}$, and the recursion follows the child determined by the observed value $x_{j(t)}$. If $j(t) \notin C \cup G$, the split feature is fixed as absent from the restricted game, and the recursion averages over the children using model-native node-cover proportions. If $j(t) \in G$, the split feature is an active player in the local Shapley problem. In this case, TLSV applies the EXTEND/UNWIND recursion (Lundberg et al., 2020): coalitions containing $j(t)$ follow the observed child, whereas coalitions not containing $j(t)$ are propagated through the children according to cover weights. The path object stores the active split features encountered on the current root-to-node path, their one-path and zero-path fractions, and the accumulated Shapley weights for each subset size. EXTEND updates this path object when the traversal moves down the tree. When a repeated split on the same active feature is encountered, UNWIND removes the previous occurrence and returns its stored zero- and one-fractions; the feature is then reinserted with the updated fractions for the current split. This repeated-feature correction ensures that each active feature appears at most once in the path object (Lundberg et al., 2020).

This context-conditioned solver, denoted by

$$\textsc{InnerTreeDP}(G, C),$$

computes the exact Shapley values $\{\phi_i^G(v_{x,C}) : i \in G\}$ for the restricted game on G. Features in C are treated as always present, features outside $C \cup G$ are treated as always absent, and only features in G enter the local Shapley averaging.

The path-dependent coalition-value recursion corresponding to a fixed coalition S is

$$g_t(S) = \begin{cases} \mu_t, & t \text{ is a leaf, or } \mathrm{ch}(t) = \emptyset, \\ g_{c^*(t)}(S), & j(t) \in S \text{ and } c^*(t) \text{ is determined}, \\ \displaystyle\sum_{c \in \mathrm{ch}(t)} \omega_{c|t}\, g_c(S), & \text{otherwise}, \end{cases} \tag{14}$$

where μ_t is the value stored at node t, $c^*(t)$ is the child selected at t by the observed value $x_{j(t)}$, and $\omega_{c|t}$ is the model-native node-cover proportion of child c at node t. Here r_c in (15) denotes the nonnegative node-cover mass supplied by the fitted learner for child c; its concrete representation is learner specific. Two aspects of (14) require explicit conventions. They are stated here because they are part of the definition of the set function v_x, not implementation details: a different choice defines a different game, and every exactness statement below is relative to the choice made here.

Convention (a): selection of $c^*(t)$. The child $c^*(t)$ is determined by comparing $x_{j(t)}$ with the threshold stored at t under the split rule of the fitted implementation or, when $x_{j(t)}$ is missing, by the model's default missing direction. It is *undetermined* in exactly two situations: $x_{j(t)}$ is missing and the model specifies no default direction; or the selected child is absent from the tree dump and no default child is available. In both situations the third line of (14) applies, so that the presence and the absence of $j(t)$ induce the same cover-weighted average at t and $j(t)$ has no local marginal contribution at that node. The split rule and the numerical precision used in the comparison are inherited from the fitted implementation and held fixed; learner-specific parsing, cover provenance, and numerical conventions are documented in Supplementary Section S4.

Convention (b): absent-feature weights. The weights are the children-sum cover proportions

$$\omega_{c|t} := \frac{r_c}{\sum_{a \in \mathrm{ch}(t)} r_a}, \tag{15}$$

with the following fallbacks when that denominator is unavailable, non-finite, or zero: the surviving children are weighted uniformly; if only one child is present it receives the entire mass; and if no child is present the node's own value μ_t is returned, which is the first line of (14). The children-sum normalization is used rather than r_c/r_t so that v_x is well defined for tree dumps in which the stored parent cover is not exactly the sum of the child covers; the two coincide whenever $r_t = \sum_{a \in \mathrm{ch}(t)} r_a$, which holds for regular dumps.

Equations (14)–(15), together with conventions (a) and (b), define the finite set function v_x that is the target of every exactness statement below. Exactly the same conventions are used by the coalition-value evaluator, by INNERTREEDP, and by the exhaustive reference computations of Section 4; agreement of these conventions across the three is what makes the reference comparisons a test of the algorithms rather than of the evaluator. The context-conditioned tree-aware recursion computes the Shapley values of $v_{x,C}$ without explicitly evaluating (14) over all subsets $S \subseteq G$. Thus, it replaces the subset-enumeration cost $O(M2^{|G|})$ by a tree-depth-dependent polynomial cost (Lundberg et al., 2020).

3.6 Exact TLSV Algorithm

TLSV faces two distinct computational bottlenecks, and the algorithms below treat them separately. Within a terminal feature group, a direct local Shapley calculation grows exponentially with the number of features because it enumerates feature subsets. Higher in the hierarchy, a node with many children creates a different exponential cost because the exact recursion averages over many sibling contexts. INNERTREEDP addresses the first bottleneck, whereas hierarchy-level Monte Carlo and hybrid computation address the second.

The exact TLSV algorithm evaluates the recursive representation (11) by traversing the hierarchy from the root to the feature-parent nodes. At each internal node, it applies the local Shapley weights in (9) over the node's children and propagates the corresponding context and weight to the selected child. When the traversal reaches a feature-parent node, the algorithm calls the context-conditioned tree-aware exact solver INNERTREEDP under the accumulated outer context. This computes the exact local Shapley allocation within $A(u)$ without enumerating all subsets of that group.

Proposition 2 (Exactness)**.** *Let v_x be the path-dependent tree-aware cooperative game defined by* (14)–(15) *together with conventions (a) and (b) of Section 3.5, and let $\mathcal{H}$ be in feature-parent form. Then Algorithm 1 with* INNERTREEDP *returns $\Psi_i^{\mathcal{H}}(v_x)$ for every $i \in N$.*

A formal proof of Proposition 2 is provided in Section S2 of the Supplementary Information. The proof combines two facts. First, INNERTREEDP applies a context-conditioned tree-aware path recursion to the restricted game

Algorithm 1 TLSV-EXACT: exact levels-structure value

Require: Ensemble f; target observation x; hierarchy tree $\mathcal{H}$ in feature-parent form (Section 3.3)
Ensure: Exact levels-structure values $\Psi_i^{\mathcal{H}}(v_x)$ for all $i \in N$

$\Psi \leftarrow \mathbf{0}_p$
function AGGREGATE(node u, context C, weight w)
  if $u \in \mathcal{F}(\mathcal{H})$ **then** ▷ feature-parent node: all children of u are features, or u is a bare feature leaf
    $\Psi[A(u)]$ += $w \cdot$ INNERTREEDP$(A(u), C)$
  else
    $(u_1, \ldots, u_d) \leftarrow \mathrm{ch}(u)$
    for $j = 1, \ldots, d$ **do**
      for each $T \subseteq \{1, \ldots, d\} \setminus \{j\}$ **do**
        $w_T \leftarrow |T|!\,(d - |T| - 1)!/d!$
        $C_T \leftarrow C \cup \bigcup_{t \in T} A(u_t)$
        AGGREGATE$(u_j, C_T, w \cdot w_T)$
      end for
    end for
  end if
end function
AGGREGATE$(\mathrm{root}(\mathcal{H}), \emptyset, 1)$
return Ψ

$v_{x,C}(S) = v_x(C \cup S)$ over the active players $S \subseteq G$, with features in C fixed as present and features outside $C \cup G$ fixed as absent. Therefore, under the path-dependent tree-aware evaluator, it returns $\phi_i^G(v_{x,C})$ exactly for all $i \in G$. Second, the outer hierarchy recursion enumerates the local Shapley distributions over child orderings at each internal hierarchy node. Substituting the exact inner values into the recursive representation (11) gives the levels-structure value.

Complexity. Let $d(t)$ be the number of children of a hierarchy node t. For a feature-parent node $u \in \mathcal{F}(\mathcal{H})$, let $\mathrm{path}(u)$ denote the set of internal, non-feature-parent hierarchy nodes on the path from the root to u, excluding u itself. Since a node t of degree $d(t)$ passes $2^{d(t)-1}$ distinct sibling-predecessor subsets to each of its children, the number of outer contexts with which Algorithm 1 reaches u is

$$Q_u := |\mathcal{R}_u| = \prod_{t \in \mathrm{path}(u)} 2^{d(t)-1} = 2^{\sum_{t \in \mathrm{path}(u)} (d(t)-1)}. \tag{16}$$

For each such context, INNERTREEDP computes the exact local Shapley allocation within $A(u)$ by context-conditioned tree-aware dynamic programming. If the ensemble has B trees, at most L_{leaf} leaves per tree, and maximum tree depth $D_{\max}$, the cost of one context-conditioned inner solve is

$$O(BL_{\mathrm{leaf}} D_{\max}^2),$$

or equivalently $O(MD_{\max}^2)$ in terms of the total number of tree nodes M. Therefore the exact TLSV runtime is

$$O\left(BL_{\mathrm{leaf}} D_{\max}^2 \sum_{u \in \mathcal{F}(\mathcal{H})} Q_u\right) = O\left(BL_{\mathrm{leaf}} D_{\max}^2 \sum_{u \in \mathcal{F}(\mathcal{H})} 2^{\sum_{t \in \mathrm{path}(u)} (d(t)-1)}\right). \tag{17}$$

Thus the exponential dependence on the cardinality $|A(u)|$ of each feature-parent group, which a direct subset-enumeration solver would incur, is removed. The bound in (17) suppresses the cost of forming and restoring the outer contexts and of initializing and writing the feature-level output vector, which together contribute $O(p \sum_{u \in \mathcal{F}(\mathcal{H})} Q_u)$ in the worst case. These terms are dominated by the inner solves when $BL_{\mathrm{leaf}} D_{\max}^2$ is of order p or larger. The remaining exponential factor comes from exact integration of the levels-structure allocation over hierarchy children along the path to each feature-parent node in Algorithm 1.

When $L = 0$, the root is the only feature-parent node, $\mathrm{path}(\mathrm{root}) = \emptyset$, $Q_{\mathrm{root}} = 1$, and Algorithm 1 reduces to the standard path-dependent tree-aware Shapley computation with complexity $O(BL_{\mathrm{leaf}} D_{\max}^2)$ (Lundberg et al., 2020). For a flat hierarchy with one grouping level and K groups, every feature-parent node has $\mathrm{path}(u) = \{\mathrm{root}\}$ with $d(\mathrm{root}) = K$, so that $Q_u = 2^{K-1}$, $K_1 = K$, and (17) becomes

$$O(BL_{\mathrm{leaf}} D_{\max}^2 K 2^{K-1}),$$

reflecting the remaining exact Owen-value dependence on the number of groups, but not on the within-group feature cardinalities. If every root-to-feature-parent path contains at most L internal non-feature-parent nodes, each of degree at

Algorithm 2 INNERTREEDP: context-conditioned exact Shapley solver

Require: Tree ensemble f; target observation x; active feature set G; fixed-present context $C \subseteq N \setminus G$
Ensure: Exact local Shapley values $\{\phi_i^G(v_{x,C}) : i \in G\}$

$\phi_G \leftarrow \mathbf{0}_{|G|}$
function VISIT(tree node t, path object P, branch weight ρ)
  if t is a leaf, or an internal node with no children, with value μ_t **then**
    add the leaf contribution $\rho\,\mu_t$ to ϕ_G for the active features recorded in P using the corresponding Shapley path weights
  else
    $j \leftarrow j(t)$; let $h(t)$ and $c(t)$ be the observed and complementary children
    $\mathrm{det} \leftarrow$ TRUE if $c^*(t) = h(t)$ is determined in the sense of convention (a) following (14), else FALSE
    if $\neg\,\mathrm{det}$ **then** ▷ presence and absence of j induce the same cover-weighted average, so j has no local marginal here and P is not extended
      for each child $a \in \mathrm{ch}(t)$ **do**
        VISIT$(a, P, \rho\,\omega_{a|t})$
      end for
    else if $j \in C$ **then**
      VISIT$(h(t), P, \rho)$
    else if $j \notin C \cup G$ **then**
      for each child $a \in \mathrm{ch}(t)$ **do**
        VISIT$(a, P, \rho\,\omega_{a|t})$
      end for
    else
      if j already appears in P **then**
        $(P^-, z_j, o_j) \leftarrow \mathrm{UNWIND}(P, j)$ ▷ remove old occurrence and recover old zero/one fractions
      else
        $P^- \leftarrow P$; $z_j \leftarrow 1$; $o_j \leftarrow 1$
      end if
      $P_h \leftarrow \mathrm{EXTEND}(P^-, j, \mathrm{zero} = z_j\,\omega_{h(t)|t}, \mathrm{one} = o_j)$
      $P_c \leftarrow \mathrm{EXTEND}(P^-, j, \mathrm{zero} = z_j\,\omega_{c(t)|t}, \mathrm{one} = 0)$ ▷ only if $c(t)$ exists and $z_j\,\omega_{c(t)|t} > 0$; otherwise $\omega_{h(t)|t} = 1$ and this branch is omitted ▷ if $c(t)$ is absent, $\omega_{h(t)|t} = 1$ and the second branch is skipped
      VISIT$(h(t), P_h, \rho)$
      VISIT$(c(t), P_c, \rho)$ ▷ omitted when $c(t)$ does not exist
    end if
  end if
end function
for each tree in the ensemble **do**
  VISIT(root node of the tree, empty path object, 1)
end for
return ϕ_G

most $d_{\max}$, then $Q_u \le 2^{L(d_{\max}-1)}$ and

$$O\Big(BL_{\mathrm{leaf}}D_{\max}^2 K_1 2^{L(d_{\max}-1)}\Big), \qquad K_1 = |\mathcal{F}(\mathcal{H})|.$$

Consequently, the exact TLSV algorithm is practical when the hierarchy has moderate branching along root-to-feature-parent paths and the fitted trees have bounded depth, even when feature-parent groups contain many features. Because Q_u is a product along the path, the exact cost grows geometrically in hierarchy *depth* as well as in node degree; both are hierarchy-level costs, and they are the bottleneck that the Monte Carlo and hybrid algorithms of Section 3.7 address.

Remark (Direct aggregation over hierarchy nodes)**.** *At a hierarchy node with d children, the levels-structure value is defined by the local Shapley weights in* (9)*. TLSV therefore evaluates each hierarchy node according to the children specified by the user-provided hierarchy. When direct enumeration at a high-degree hierarchy node is expensive, the Monte Carlo algorithm samples child orderings for that same node; it does not coarsen, rebalance, or otherwise modify the target hierarchy.*

Algorithm 3 TLSV-MC: recursive Monte Carlo levels-structure value

Require: Ensemble f; target observation x; hierarchy tree $\mathcal{H}$ in feature-parent form; predetermined local sampling budgets $\{R_u\}$, with independent draws at distinct node calls

Ensure: Monte Carlo estimate $\widehat{\Psi}_i^{\mathcal{H}}(v_x)$ for all $i \in N$

$\widehat{\Psi} \leftarrow \mathbf{0}_p$
function MCAGGREGATE(node u, context C, weight w)
  if $u \in \mathcal{F}(\mathcal{H})$ **then** ▷ feature-parent node, as in Algorithm 1
    $\widehat{\Psi}[A(u)] \mathrel{+}= w \cdot \textsc{InnerTreeDP}(A(u), C)$
  else
    $(u_1, \ldots, u_d) \leftarrow \mathrm{ch}(u)$
    for $r = 1, \ldots, R_u$ **do**
      draw a uniform random ordering $(u_{(1)}, \ldots, u_{(d)})$ of $\mathrm{ch}(u)$
      $C_{\mathrm{prev}} \leftarrow C$
      for $q = 1, \ldots, d$ **do**
        MCAGGREGATE$(u_{(q)}, C_{\mathrm{prev}}, w/R_u)$
        $C_{\mathrm{prev}} \leftarrow C_{\mathrm{prev}} \cup A(u_{(q)})$
      end for
    end for
  end if
end function
MCAGGREGATE$(\mathrm{root}(\mathcal{H}), \emptyset, 1)$
return $\widehat{\Psi}$

3.7 Monte Carlo TLSV Algorithm

For high-branching hierarchies, TLSV can estimate the recursive representation in (11) by Monte Carlo integration over hierarchy-level child orderings. Algorithm 3 estimates these child-ordering expectations recursively: at each non-feature-parent internal node, it samples uniform orderings of the node's children and propagates the resulting sequential contexts to the children. At feature-parent nodes, it continues to use INNERTREEDP. Thus, Monte Carlo sampling is used only for hierarchy-level child-order integration and not for within-group feature-subset enumeration. Algorithm 4 then gives the corresponding hybrid exact–Monte Carlo strategy, which retains exact expansion at low-degree hierarchy nodes, samples child orderings only at high-degree hierarchy nodes, and uses the same INNERTREEDP solver at feature-parent nodes.

Proposition 3 (Statistical guarantees for recursive TLSV-MC)**.** *Consider Algorithm 3 with predetermined local sampling budgets $\{R_u\}$ at all sampled hierarchy-node calls, with mutually independent draws at distinct node calls, and without adaptive stopping. Suppose that, at every feature-parent node,* INNERTREEDP *returns the exact local Shapley allocation for the restricted path-dependent tree-aware game of Section 3.5. Then, for every feature $i \in N$, the following hold.*

(i) Unbiasedness. $\mathbb{E}\{\widehat{\Psi}_i^{\mathcal{H}}(v_x)\} = \Psi_i^{\mathcal{H}}(v_x)$ *at every fixed budget configuration.*

(ii) Exact efficiency. *For every realized set of sampled child orderings,*

$$\sum_{i \in N} \widehat{\Psi}_i^{\mathcal{H}}(v_x) = v_x(N) - v_x(\emptyset).$$

(iii) L^2 consistency. *As* $\min_u R_u \to \infty$,

$$\mathbb{E}\big[\{\widehat{\Psi}_i^{\mathcal{H}}(v_x) - \Psi_i^{\mathcal{H}}(v_x)\}^2\big] \to 0,$$

so the estimator is L^2-consistent and, a fortiori, consistent in probability.

(iv) Rate. *If a common budget $R_u \equiv R$ is used at all sampled node calls, then* $\mathrm{Var}\{\widehat{\Psi}_i^{\mathcal{H}}(v_x)\} = O(R^{-1})$*, so the Monte Carlo standard error is $O(R^{-1/2})$, with constants determined by the finite hierarchy $\mathcal{H}$ and by v_x.*

Corollary 1 (Concentration and almost sure convergence along summable budget schedules)**.** *Adopt the assumptions of Proposition 3, put*

$$\Lambda := 2 \max_{S \subseteq N} |v_x(S)|, \qquad c_\varepsilon := \frac{\varepsilon^2}{2\Lambda^2} \ \ (\varepsilon > 0),$$

and let R_{root} be the sampling budget used at the root call. Then, for every $\varepsilon > 0$ and every $i \in N$,

$$\Pr\left\{\left|\widehat{\Psi}_i^{\mathcal{H}}(v_x) - \Psi_i^{\mathcal{H}}(v_x)\right| > \varepsilon\right\} \leq 2\exp(-c_\varepsilon R_{\mathrm{root}})\,,$$

uniformly in the budgets used at the descendant node calls. Consequently, for any deterministic budget schedule $\{R_u^{(m)}\}_{m\geq 1}$ satisfying $\sum_m \exp(-c_\varepsilon R_{\mathrm{root}}^{(m)}) < \infty$ for every $\varepsilon > 0$—for instance $R_u^{(m)} = m$ for all u—the Borel–Cantelli lemma gives

$$\widehat{\Psi}_i^{\mathcal{H},(m)}(v_x) \to \Psi_i^{\mathcal{H}}(v_x) \qquad \textit{almost surely as } m \to \infty.$$

Some such condition is needed. The displayed inequality controls an upper bound on the deviation probabilities, and divergence of that bound does not by itself imply that the deviation events occur infinitely often; without a summability condition, or without an explicit coupling of the samples across increasing budgets, almost sure convergence may fail under sufficiently slow budget growth with fresh independent draws, whereas L^2 consistency continues to hold.

Formal proofs of Proposition 3 and of Corollary 1 are provided in Section S3 of the Supplementary Information. The concentration bound is obtained by applying Hoeffding's inequality at the root call alone: the R_{root} summands averaged there are independent and identically distributed, have mean $\Psi_i^{\mathcal{H}}(v_x)$ by part (i) of Proposition 3, and take values in $[-\Lambda, \Lambda]$ for every budget configuration, because each realized draw is a convex combination of marginal contributions of v_x. No union bound over the internal node–context states is required.

Monte Carlo complexity. For Algorithm 3 and a feature-parent node $u \in \mathcal{F}(\mathcal{H})$, let

$$\widetilde{Q}_u := \prod_{t\in \mathrm{path}(u)} R_t, \tag{18}$$

with $\mathrm{path}(u)$ as in (16) and R_t the local child-ordering sampling budget used at node t. Then $\widetilde{Q}_u$ is the number of context-conditioned InnerTreeDP calls reaching u under Algorithm 3. Therefore, suppressing lower-order costs for sampling child orderings, maintaining contexts, and writing output vectors, the recursive Monte Carlo TLSV runtime is

$$O\left(BL_{\mathrm{leaf}}D_{\max}^2 \sum_{u\in\mathcal{F}(\mathcal{H})} \widetilde{Q}_u\right) = O\left(BL_{\mathrm{leaf}}D_{\max}^2 \sum_{u\in\mathcal{F}(\mathcal{H})} \prod_{t\in\mathrm{path}(u)} R_t\right). \tag{19}$$

If a common local budget R is used at all sampled hierarchy nodes, then $\widetilde{Q}_u = R^{|\mathrm{path}(u)|}$. Thus TLSV-MC replaces the exact hierarchy-level factor $\prod_{t\in\mathrm{path}(u)} 2^{d(t)-1}$ in (17) by the sampled factor $\prod_{t\in\mathrm{path}(u)} R_t$, while retaining the same context-conditioned InnerTreeDP solver at feature-parent nodes. The comparison is therefore node by node: sampling reduces the per-node factor from $2^{d(t)-1}$ to R_t, so the gain is substantial only at nodes whose degree satisfies $2^{d(t)-1} \gg R_t$, and is negligible at low-degree nodes. Both factors remain multiplicative along the root-to-feature-parent path, so neither algorithm removes the geometric dependence on hierarchy depth. As in (17), the factor B is a worst-case count over all trees; feature-parent-specific tree filtering can reduce the realized cost of each inner solve without changing $\widetilde{Q}_u$ or the Monte Carlo sampling law.

Remark (Fixed-budget estimation and adaptive stopping)**.** *The unbiasedness statement in Proposition 3 applies to the fixed-budget estimator with predetermined local sampling budgets for hierarchy-node calls. Optional adaptive stopping based on running Monte Carlo uncertainty is a data-dependent stopping rule and is therefore not covered by the finite-sample unbiasedness claim. It is used as a practical implementation device and is not covered by the unbiasedness or consistency statements of Proposition 3 and Corollary 1, which assume predetermined budgets; establishing consistency under a data-dependent rule would additionally require the realized stopping counts to diverge, for example by letting the minimum sample size grow as the tolerance decreases.*

Hybrid exact–Monte Carlo computation. The exact and Monte Carlo components can be combined within the same hierarchy to address the remaining hierarchy-level computational bottleneck in (17). In Algorithm 1, a high-degree hierarchy node with d children requires exact local Shapley integration over sibling predecessor subsets under the weights in (9), resulting in $d2^{d-1}$ child-context calls. Algorithm 4 therefore expands low-degree hierarchy nodes exactly, but replaces exact child-order integration at high-degree nodes by uniform sampling of complete child orderings, following the same recursive sampling principle as Algorithm 3. This requires $R_u d$ sampled child-context calls for each call to a high-degree node u. At feature-parent nodes, the default solver remains InnerTreeDP, whose exactness for the path-dependent tree-aware game is stated in Proposition 2. Thus, high-degree hierarchy branching and large feature-parent groups are handled by different mechanisms: child-order sampling addresses the former, whereas context-conditioned tree-aware recursion addresses the latter. Under predetermined local sampling budgets and

Algorithm 4 TLSV-HYBRID: hybrid exact–Monte Carlo levels-structure value

Require: Ensemble f; target observation x; hierarchy tree $\mathcal{H}$ in feature-parent form; degree threshold τ; predetermined local sampling budgets $\{R_u\}$ applied to each high-degree node call, with independent draws at distinct node calls

Ensure: Hybrid estimate $\widetilde{\Psi}_i^{\mathcal{H}}(v_x)$ for all $i \in N$

$\widetilde{\Psi} \leftarrow \mathbf{0}_p$
function HYBRIDAGGREGATE(node u, context C, weight w)
  if $u \in \mathcal{F}(\mathcal{H})$ **then** ▷ feature-parent node, as in Algorithm 1
    $\widetilde{\Psi}[A(u)]$ += $w \cdot$ INNERTREEDP$(A(u), C)$
  else
    $(u_1, \ldots, u_d) \leftarrow \mathrm{ch}(u)$
    if $d \le \tau$ **then**
      for $j = 1, \ldots, d$ **do**
        for each $T \subseteq \{1, \ldots, d\} \setminus \{j\}$ **do**
          $w_T \leftarrow |T|!\,(d - |T| - 1)!/d!$
          $C_T \leftarrow C \cup \bigcup_{t \in T} A(u_t)$
          HYBRIDAGGREGATE$(u_j, C_T, w \cdot w_T)$
        end for
      end for
    else
      for $r = 1, \ldots, R_u$ **do**
        draw a uniform random ordering $(u_{(1)}, \ldots, u_{(d)})$ of $\mathrm{ch}(u)$
        $C_{\mathrm{prev}} \leftarrow C$
        for $q = 1, \ldots, d$ **do**
          HYBRIDAGGREGATE$(u_{(q)}, C_{\mathrm{prev}}, w/R_u)$
          $C_{\mathrm{prev}} \leftarrow C_{\mathrm{prev}} \cup A(u_{(q)})$
        end for
      end for
    end if
  end if
end function
HYBRIDAGGREGATE$(\mathrm{root}(\mathcal{H}), \emptyset, 1)$
return $\widetilde{\Psi}$

independent draws at distinct node calls, the hybrid estimator remains unbiased for the same levels-structure target in (11): exact nodes integrate their local child-order distributions analytically, sampled nodes integrate them by Monte Carlo, and the tower property preserves the overall expectation, paralleling Proposition 3. Because each sampled child ordering yields a telescoping decomposition of the local block contribution, and because the exact and inner solvers satisfy efficiency for their restricted games, the hybrid estimator is exactly efficient for every realized set of sampled orderings. Under increasing local budgets, the hybrid estimator is likewise L^2-consistent for the same target. If the root is sampled, the concentration and almost-sure convergence argument of Corollary 1 also applies. If the root is expanded exactly, no corresponding concentration or almost-sure statement is asserted here; see Supplementary Section S3. This hybrid strategy allows TLSV to retain exact computation where feasible while remaining scalable for larger nested feature systems.

3.8 Visualization and Importance Measures

TLSV provides local visualizations and global importance measures at both the feature level and any internal node of the user-specified hierarchy. These summaries are computed from the local levels-structure values and therefore inherit the multi-resolution coherence of Proposition 1.

Local Explanation: Hierarchy-Aware Beeswarm Visualization. For a dataset of n observations $x^{(1)}, \ldots, x^{(n)}$, TLSV computes levels-structure values $\Psi_i^{\mathcal{H}}(v_{x^{(j)}})$ for each feature $i \in N$ and observation $j = 1, \ldots, n$. Feature-level beeswarm plots summarize the empirical distribution

$$\left\{\Psi_i^{\mathcal{H}}(v_{x^{(j)}})\right\}_{j=1}^{n}$$

across observations. Each point corresponds to an observation j, is positioned horizontally at $\Psi_i^{\mathcal{H}}(v_{x^{(j)}})$, and is colored by the observed feature value $x_i^{(j)}$, following a standard beeswarm-style summary visualization. Features are ordered

by decreasing mean absolute levels-structure value,

$$I_i := \frac{1}{n}\sum_{j=1}^{n}\left|\Psi_i^{\mathcal{H}}(v_{x^{(j)}})\right|, \tag{20}$$

so that the most influential features appear at the top.

For any hierarchy node u, the local node-level attribution for observation j is defined by aggregating the feature-level attributions in its subtree:

$$\Psi_u^{\mathcal{H}}(v_{x^{(j)}}) := \sum_{i\in A(u)} \Psi_i^{\mathcal{H}}(v_{x^{(j)}}). \tag{21}$$

For a feature leaf $u = \{i\}$, this definition reduces to the feature attribution. For an internal node, the additive identity

$$\Psi_u^{\mathcal{H}}(v_{x^{(j)}}) = \sum_{j'=1}^{d} \Psi_{u_{j'}}^{\mathcal{H}}(v_{x^{(j)}})$$

follows directly from (21), since the children's feature sets $\{A(u_{j'}) : u_{j'} \in \mathrm{ch}(u)\}$ partition $A(u)$, with a feature child interpreted as its feature-level attribution. By Proposition 1(v), $\Psi_u^{\mathcal{H}}(v_{x^{(j)}})$ equals the value that the levels-structure allocation assigns to the block $A(u)$ in the corresponding quotient game. In particular, the attributions of the children of the root sum to the prediction gap,

$$\sum_{u'\in \mathrm{ch}(\mathrm{root}(\mathcal{H}))} \Psi_{u'}^{\mathcal{H}}(v_{x^{(j)}}) = v_{x^{(j)}}(N) - v_{x^{(j)}}(\emptyset),$$

which equals $f(x^{(j)}) - \mathbb{E}[f(X)]$ under (2). Thus, node-level visualizations at any chosen resolution partition the same local prediction gap without recomputing explanations separately at that resolution.

A node-level beeswarm plot displays the distribution of $\{\Psi_u^{\mathcal{H}}(v_{x^{(j)}})\}_{j=1}^{n}$ for selected internal nodes. Because an internal hierarchy node does not generally have a unique observed feature value, point colors are based on a node-level display statistic

$$c_u^{(j)} = h_u\left(x_{A(u)}^{(j)}\right),$$

where h_u is chosen according to the application. For abundance-type data, such as microbiome relative abundances, a natural display choice is the total node abundance,

$$c_u^{(j)} = \sum_{i\in A(u)} x_i^{(j)}.$$

Other supported choices include the mean and standardized mean; application-specific summaries over $A(u)$ may also be constructed separately when total abundance is not meaningful. This color statistic is used only for visualization and does not affect the levels-structure attribution. Nodes can be displayed level by level, such as species, genera, families, orders, classes, and phyla, or as a nested hierarchy in which each displayed node is ordered by its global importance.

Global Importance Measures. The global importance of a hierarchy node u is defined as the mean absolute node-level attribution:

$$I_u^{\mathrm{node}} := \frac{1}{n}\sum_{j=1}^{n}\left|\Psi_u^{\mathcal{H}}(v_{x^{(j)}})\right| = \frac{1}{n}\sum_{j=1}^{n}\left|\sum_{i\in A(u)} \Psi_i^{\mathcal{H}}(v_{x^{(j)}})\right|. \tag{22}$$

For a feature leaf $u = \{i\}$, $I_u^{\mathrm{node}} = I_i$. For an internal node, however, $I_u^{\mathrm{node}} \neq \sum_{i\in A(u)} I_i$ in general, because absolute values do not distribute over sums when features within the node have opposing signed effects. Thus, I_u^{node} measures the net predictive contribution of the block $A(u)$ at the chosen hierarchical resolution, whereas $\sum_{i\in A(u)} I_i$ measures the total magnitude of its individual feature-level contributions. Reporting both quantities can be useful: the former summarizes the coherent node-level signal, while the latter indicates whether substantial but canceling within-node effects are present.

3.9 Software: TLSV

The accompanying `tlsv` R package implements TLSV-EXACT, TLSV-MC, and TLSV-HYBRID in Algorithms 1–4, with performance-critical numerical kernels written in `C++` and accessed through `Rcpp`. Version 0.1.0 supports regression and binary classification for validated XGBoost (Chen and Guestrin, 2016), LightGBM (Ke et al., 2017), and ranger (Wright and Ziegler, 2017) tree ensembles with numeric binary splits; native categorical or multiway splits

and multiclass, survival, ranking, or quantile outputs are outside the present software scope. All computation modes use the path-dependent tree-aware coalition-value evaluator in (14)–(15), and feature-parent allocations are computed by INNERTREEDP by default in Algorithm 2. The implementation applies caching and workspace reuse, and skips trees that contain no split feature from the active feature parent, since such a tree is constant in the corresponding restricted local game; these are implementation choices that do not alter the coalition-value evaluator, the hierarchy recursion, or the exact TLSV allocation. Independent observations and repeated explanation tasks can be distributed across processor cores from `R`; for stochastic methods, observation-specific random-number streams derived from a common seed make results reproducible across requested worker counts for a fixed observation order.

For XGBoost and LightGBM, the serialized tree representation can omit a learner-specific additive base or initial score. During model preparation, the software calibrates a single additive constant against native raw predictions, verifies that the remaining discrepancy is constant within numerical tolerance, and represents that constant as a one-leaf tree in the unified evaluator. Adding the same constant to every coalition value leaves all coalition marginal contributions unchanged, but it ensures that the full-coalition evaluator reproduces the fitted model and that the baseline remains fixed across explanation batches. The preparation rows used for this calibration are model-interface validation data, not a SHAP background distribution.

For regression, attributions are returned on the model-output scale. For binary classification with XGBoost or LightGBM, TLSV is computed on the raw-margin scale, the additive ensemble scale on which the tree-aware recursion is exact. Let $F(x)$ denote the fitted raw-margin prediction and F_0 its baseline margin, so that the margin-scale TLSV values sum to $F(x) - F_0$. The package also provides a probability-scale summary obtained from the margin-scale attributions by the observation-specific transformation

$$\Psi_i^{\mathcal{H},\mathrm{prob}} = \Psi_i^{\mathcal{H},\mathrm{margin}} \frac{\sigma\{F(x)\} - \sigma(F_0)}{F(x) - F_0}, \qquad \sigma(z) = \{1 + \exp(-z)\}^{-1},$$

where the ratio is interpreted as $\sigma'(F_0)$ when $F(x) = F_0$. Because the same scalar is applied to every feature for a given observation, these transformed values sum to the corresponding probability difference and preserve the cross-resolution aggregation property in Proposition 1(v). They are interpretability summaries derived from the exact margin-scale TLSV values, rather than levels-structure values for the nonlinear probability-scale coalition game. Ranger probability forests are explained directly on their additive probability-output scale.

Hierarchies may be supplied as nested partitions or as rooted trees and are converted to the feature-parent representation of Section 3.3. This normalization accommodates unequal group sizes and depth-unbalanced structures without changing the target levels-structure value. Users may select exact, Monte Carlo, or hybrid hierarchy-level computation, with feature-parent nodes evaluated exactly by INNERTREEDP by default. Adaptive and alternative solver options are also available; the reported simulations use hierarchy-level Monte Carlo sampling with predetermined local budgets and exact INNERTREEDP evaluation at feature-parent nodes. The theoretical guarantees for fixed budgets and their scope are stated in Section 3.7 and Supplementary Section S3.

The model interfaces preserve the learner-specific split and numerical conventions described in Section 3.5, so the evaluator, INNERTREEDP, and all hierarchy recursions operate on a common set function v_x. As a validation safeguard, the software checks that the full-coalition value $v_x(N)$ reproduces the fitted learner's prediction on the relevant output scale to prespecified relative and absolute tolerances, the latter resolved from the evaluation precision of the fitted learner (Supplementary Section S4), and reports any discrepancy rather than silently accepting a parser mismatch.

The package returns local feature- and node-level attributions, global importance measures in (20) and (22), and the hierarchy-aware visualization tools described in Section 3.8. These include feature- and node-level beeswarm displays, level-specific rankings, and nested summaries that present the same coherent allocation at multiple resolutions.

The `tlsv` package, documentation, and source code are freely available at `https://github.com/hk1785/tlsv`.

4 Results

4.1 Simulation Designs

The simulation study is organized around four practical questions: whether TLSV-EXACT reproduces the defining levels-structure value, how TLSV-MC behaves and how its accuracy changes with the sampling budget, whether the resulting allocations are numerically coherent across hierarchy levels, and how computation scales with the two distinct sources of combinatorial growth—feature-parent size and hierarchy complexity. Each experiment uses the strongest computationally feasible same-target reference. Small hierarchies permit literal averaging over all hierarchy-consistent permutations, denoted by DIRECT-ENUM, whereas larger settings use TLSV-EXACT or a dedicated same-target inner-solver comparator. Detailed grids, sampling conventions, diagnostics, and run-completeness checks are provided in Supplementary Section S4.

Data-generating process and learners. Features are generated from a multilevel latent-factor model that induces stronger correlations among features sharing lower-level hierarchy nodes. For a hierarchy $\mathcal{H}$ and feature i, let $\mathcal{P}(i)$ denote the non-root internal nodes on the path from the root to the feature-parent node containing i. The features are generated as

$$\widetilde{X}_i = a_0 Z_0 + \sum_{u \in \mathcal{P}(i)} a_{\ell(u)} Z_u + E_i, \qquad X_i = \frac{\widetilde{X}_i}{\left(a_0^2 + \sum_{u \in \mathcal{P}(i)} a_{\ell(u)}^2 + 1\right)^{1/2}},$$

where $\ell(u)$ denotes the depth of node u, so that loadings vary with hierarchy level, and all latent factors and idiosyncratic errors are mutually independent standard normal variables. Outcomes contain sparse feature-level, node-level, and interaction signals:

$$\eta(X) = \sum_{i \in S_{\text{feat}}} \beta_i X_i + \sum_{u \in S_{\text{node}}} \gamma_u B_u(X) + \sum_{(u,v) \in S_{\text{int}}} \theta_{uv} B_u(X) B_v(X),$$

where $B_u(X)$ is the standardized aggregate signal of the features in $A(u)$. For regression, $Y = \eta(X) + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0,1)$; for classification, $\Pr(Y = 1 \mid X) = \text{logit}^{-1}\{\alpha + \eta(X)\}$, with α chosen within each replicate to yield approximate class balance. Both regression and binary-classification settings are considered, using XGBoost (Chen and Guestrin, 2016), LightGBM (Ke et al., 2017), and ranger (Wright and Ziegler, 2017). Hyperparameters are fixed across replicates, and classification attribution scales follow Section 3.9.

Exactness and finite-budget Monte Carlo behavior. The notation $b_1 \times b_2 \times \cdots \times s$ denotes successive hierarchy-level branching factors followed by feature-parent size s, so that $p = s \prod_\ell b_\ell$. The two-child hierarchy $2 \times 2 \times 2$ ($p = 8$) and the multiway hierarchy $2 \times 3 \times 2$ ($p = 12$) are used for exhaustive validation. Their numbers of hierarchy-consistent feature permutations are 128 and $4{,}608$, respectively, so DIRECT-ENUM can evaluate the definition in (8) without binary refinement or rebalancing. These settings are used for the exactness comparison and the primary Monte Carlo experiment. Each design cell uses 100 simulation replicates and 50 held-out target observations. The primary Monte Carlo estimator uses the deliberately small fixed local budget $R_u = 3$; for its finite-budget evaluation, 20 independent Monte Carlo runs are generated for each replicate-level cell.

Numerical coherence. The same two exhaustive-validation hierarchies, 100 simulation replicates, and 50 held-out observations are used to assess allocation efficiency, cross-resolution block agreement, and invariance to degree-one carry-forward nodes. The block-coherence diagnostic compares the sum of descendant feature attributions with the corresponding block value Γ_u computed independently from its context–weight definition, rather than by reaggregating the reported feature-level output. Under TLSV-MC at $R_u = 3$, the same comparison is interpreted as approximation error relative to the exact block target, because the realized feature-level allocation remains exactly efficient while its aggregated Monte Carlo block estimate is compared with the exact block target Γ_u. Complete definitions are given in Supplementary Section S4.

Computational scaling. Three one-factor-at-a-time sweeps separate different sources of computational growth. Hierarchy depth is varied through

$$\underbrace{3 \times 3 \times \cdots \times 3}_{m \text{ factors}}, \qquad m \in \{3,4,5,6,7,8\},$$

while feature-parent size is varied through

$$3 \times 3 \times s, \qquad s \in \{3,6,9,12,15,18\}.$$

The latter keeps the higher-level hierarchy fixed, so the number of context-conditioned calls to the within-group solver is constant in s. A dedicated same-target comparator reruns TLSV-EXACT over the same s grid after replacing INNERTREEDP with direct evaluation of all $2^{|A(u)|}$ feature subsets at every feature-parent node; all other model, hierarchy, context, and coalition-value computations are held fixed. Finally, hierarchy degree is varied using single-level hierarchies

$$d \times 2, \qquad d \in \{3,5,7,9,11,13,15,17\}.$$

For each root child, TLSV-EXACT uses 2^{d-1} sibling contexts, whereas TLSV-MC uses $R_u = 3$ sampled child orderings. DIRECT-ENUM remains feasible for $d \leq 5$ and TLSV-EXACT is the same-target reference thereafter. The production depth, feature-parent-size, and degree sweeps use 20 explained observations and 5 simulation replicates per learner–task cell; the deliberately expensive inner-solver comparator uses 4 explained observations and the same 5 replicates.

Monte Carlo budget sensitivity. To separate the choice of Monte Carlo budget from changes in the fitted model or hierarchy, an additional paired experiment fixes the high-degree hierarchy at $d = 15$ with feature-parent size 2 and varies

$$R \in \{1, 3, 5, 10, 20\}.$$

For every learner–task–replicate cell, all five budgets use the same fitted model, the same 20 explained observations, and the same TLSV-EXACT reference from the matching $d = 15$ degree-sweep cell. Twenty independent Monte Carlo runs are generated at each budget. The primary simulation setting remains $R = 3$; this sensitivity experiment is used only to characterize the empirical accuracy–cost trade-off and to assess whether the observed error decay is consistent with the $O(R^{-1/2})$ Monte Carlo standard-error rate in Proposition 3.

4.2 Simulation Results

The results address the four questions above using same-target comparisons wherever feasible. Runtimes are single-core solver times excluding model fitting, and reported runtime ratios are medians of paired learner–task–replicate ratios. Four coherence diagnostics are used: the normalized residual Δ_{eff} for allocation efficiency, the normalized block residual Δ_{blk} for agreement between a block total and Γ_u in (12), the normalized bookkeeping residual Δ_{agg} for the child-sum identity, and the absolute change Δ_{deg1} under insertion of degree-one nodes. Their definitions and complete summaries are given in Supplementary Section S4.

Exact numerical validation. Table 1 summarizes TLSV-EXACT against DIRECT-ENUM in the two exhaustive-validation hierarchies. Each hierarchy contributes six learner–task cells and 100 replicates per cell, giving 600 replicate-level comparisons per hierarchy; each replicate uses 50 held-out target observations. Detailed learner–task results are retained in Supplementary Section S4.

Agreement is at numerical precision throughout. The feature-level correlation with the reference equals one to all reported digits, every attribution satisfies the relative-tolerance agreement criterion in Table 1, and the largest absolute discrepancy over all replicates is 3.1×10^{-14} for $2 \times 2 \times 2$ and 5.5×10^{-13} for $2 \times 3 \times 2$. Thus, even the 4,608-permutation multiway reference agrees with the recursive exact algorithm to floating-point precision. Allocation efficiency (Proposition 1(i)) holds simultaneously, with $\Delta_{\text{eff}} \leq 1.3 \times 10^{-14}$.

The $2 \times 3 \times 2$ setting contains a degree-three node and therefore directly validates the multiway recursion on the original hierarchy without binary refinement. The same comparison is available for $d \leq 5$ in the degree sweep, where the maximum discrepancy is 2.2×10^{-13}; this anchors the larger degree settings, for which DIRECT-ENUM is no longer feasible, to the definition in (8).

Table 1: Same-target exactness summary for the two exhaustive-validation hierarchies. Each hierarchy contains six learner–task cells and 100 replicates per cell (600 replicate-level cells), with 50 held-out target observations per replicate. Correlation and agreement are minima; the error and efficiency columns are maxima.

Hierarchy	$\lvert\Omega(\mathcal{H})\rvert$	Cells	Corr.	RMSE	Max abs.	Agree (%)	Δ_{eff}
$2 \times 2 \times 2$	128	600	1.000	4.0×10^{-15}	3.1×10^{-14}	100.0	9.1×10^{-15}
$2 \times 3 \times 2$	4,608	600	1.000	5.5×10^{-14}	5.5×10^{-13}	100.0	1.3×10^{-14}

DIRECT-ENUM averages marginal contributions over all $|\Omega(\mathcal{H})|$ hierarchy-consistent permutations and is therefore a same-target reference for the defining levels-structure value. "Agree" is the percentage satisfying $|\widehat{\Psi} - \Psi_{\text{ref}}| \leq 10^{-3} \max(|\Psi_{\text{ref}}|, 10^{-8})$. Detailed learner–task values are reported in Supplementary Section S4.

Finite-budget Monte Carlo behavior. Table 2 summarizes 20 independent runs of TLSV-MC at $R_u = 3$ for each of the 1,200 replicate-level settings, for a total of 24,000 Monte Carlo runs. Exact efficiency is a property of each realized draw by Proposition 3(ii). For the numerical diagnostic in Table 2, the observation-specific normalized efficiency residual is averaged within each Monte Carlo run and the largest such run-level average over the 20 runs is reported; this quantity is at most 1.5×10^{-15} across the simulation cells.

Unbiasedness is established theoretically in Proposition 3(i); the simulation therefore serves as a finite-budget diagnostic rather than a formal test. Comparing the root-mean-square empirical bias with the corresponding root-mean-square Monte Carlo standard error gives a median ratio of 0.99 with interquartile range $[0.90, 1.08]$ across the 1,200 replicate-level settings. The observed finite-run bias is therefore of the same order as the Monte Carlo uncertainty expected under the unbiasedness result. Absolute sampling variability differs across learner–task settings because the learners are explained on different attribution scales, particularly raw-margin versus probability scales in classification (Section 3.9).

Table 2: Finite-budget behavior of TLSV-MC at $R_u = 3$ over 12 hierarchy–learner–task cells and 100 replicates per cell (1200 replicate-level cells), with 20 independent Monte Carlo runs per replicate. Entries are medians except Δ_{eff}, which is a maximum.

Hierarchy	Learner	Task	RMS bias	MCSE	Bias/MCSE	Pooled RMSE	Stability SD	Δ_{eff}
$2 \times 2 \times 2$	XGBoost	Regression	0.0245	0.0245	1.00	0.1099	0.0758	1.2×10^{-15}
	XGBoost	Classification	0.0513	0.0513	0.99	0.2301	0.1843	1.1×10^{-15}
	LightGBM	Regression	0.0267	0.0266	1.00	0.1193	0.0844	1.5×10^{-15}
	LightGBM	Classification	0.0544	0.0546	0.97	0.2433	0.1920	1.4×10^{-15}
	Ranger	Regression	0.0047	0.0048	0.95	0.0215	0.0149	7.8×10^{-16}
	Ranger	Classification	0.0008	0.0008	0.99	0.0037	0.0026	8.9×10^{-16}
$2 \times 3 \times 2$	XGBoost	Regression	0.0182	0.0187	0.99	0.0834	0.0586	1.2×10^{-15}
	XGBoost	Classification	0.0388	0.0394	1.00	0.1769	0.1385	1.2×10^{-15}
	LightGBM	Regression	0.0204	0.0205	1.00	0.0923	0.0653	1.2×10^{-15}
	LightGBM	Classification	0.0425	0.0431	0.99	0.1923	0.1488	1.5×10^{-15}
	Ranger	Regression	0.0034	0.0034	0.99	0.0154	0.0101	7.2×10^{-16}
	Ranger	Classification	0.0006	0.0006	1.00	0.0026	0.0018	6.1×10^{-16}

RMS bias is the root-mean-square empirical bias across feature–observation pairs, relative to DIRECT-ENUM; MCSE is the corresponding RMS Monte Carlo standard error. Values of their ratio near one indicate finite-budget behavior compatible with the theoretical unbiasedness result; the ratio is a diagnostic rather than a formal test. Stability SD is the mean elementwise across-run standard deviation. Absolute variability should not be compared directly across learners that use different attribution scales. The Δ_{eff} column is the maximum over MC runs of the observation-averaged normalized efficiency residual; exact efficiency itself holds observationwise for every realized draw by Proposition 3(ii).

Numerical coherence. The coherence experiment asks whether a hierarchy block receives the same attribution when evaluated directly from the hierarchy recursion as when reconstructed by summing the attributions of its descendant features. Table 3 reports the corresponding diagnostics over 12 hierarchy–learner–task cells, each with 100 replicates and 50 held-out target observations.

The substantive exact test is Δ_{blk}, which compares the aggregated feature-level attributions within each node against Γ_u in (12), evaluated independently over the context–weight multiset $\mathcal{R}_u$ and never reading the feature-level output. Under TLSV-EXACT, Δ_{blk} is at most 1.2×10^{-14}, numerically validating the multi-resolution identity of Proposition 1(v). The exact allocation-efficiency residual in this experiment is likewise at most 1.0×10^{-14}.

Under TLSV-MC at $R_u = 3$, the corresponding block-target quantity is an approximation error relative to the exact Γ_u, not a failure of the identity: efficiency remains exact for each realized draw, whereas the aggregated Monte Carlo feature allocation provides an unbiased finite-budget estimate of the exact block value Γ_u. With the deliberately small budget $R_u = 3$, the cell-wise maximum normalized block-target errors range from 0.059 to 3.721, making the finite-budget approximation cost visible rather than concealing it behind exact aggregation identities.

For contrast, Δ_{agg} is at most 1.7×10^{-16}. Because the children of a node partition its feature set, this vanishes identically for any feature-level vector and is an implementation diagnostic rather than independent evidence for multi-resolution coherence. Finally, Δ_{deg1} is exactly zero in every cell: inserting degree-one carry-forward nodes leaves every attribution unchanged, confirming that the hierarchy normalization of Section 3.3 does not alter the target.

Computational scalability: the within-group solver. Figure 1A and Supplementary Table S4 isolate Algorithm 2 by running Algorithm 1 twice on the same fitted models and hierarchies, replacing INNERTREEDP with direct enumeration of all $2^{|A(u)|}$ subsets and holding everything else fixed. Because the higher-level structure is held at 3×3, the number of context-conditioned inner solves is 144 per explained observation, hence 576 over the four-observation comparator set, and is constant in s. Any systematic divergence in runtime with increasing s therefore isolates the local solver.

The two arms agree throughout the expanded $s \leq 18$ grid, with a maximum absolute difference of 4.8×10^{-13}. At $s = 3$, direct subset enumeration is slightly faster (0.09 versus 0.142 seconds; paired ratio 0.6), reflecting fixed dynamic-programming overhead. The scaling reverses by $s = 6$ and then separates rapidly: the paired median speed-up is 103 at $s = 12$, 830 at $s = 15$, and 6,476 at $s = 18$. At the largest setting, direct subset enumeration requires 962.85 seconds for the four-observation comparator set, whereas INNERTREEDP requires 0.146 seconds. In the production $3 \times 3 \times s$ sweep, the exact hierarchy-level runtime remains nearly flat at approximately 2.2 seconds from $s = 3$ through 18, consistent with the constant outer context count (Supplementary Table S4). These results directly isolate the exponential 2^s behavior avoided by Algorithm 2.

Computational scalability: the hierarchy-level recursion. Supplementary Table S6 reports the production depth and feature-parent-size sweeps. The measured number of inner solves matches Q_u in (16) and $\widetilde{Q}_u$ in (18) exactly at every

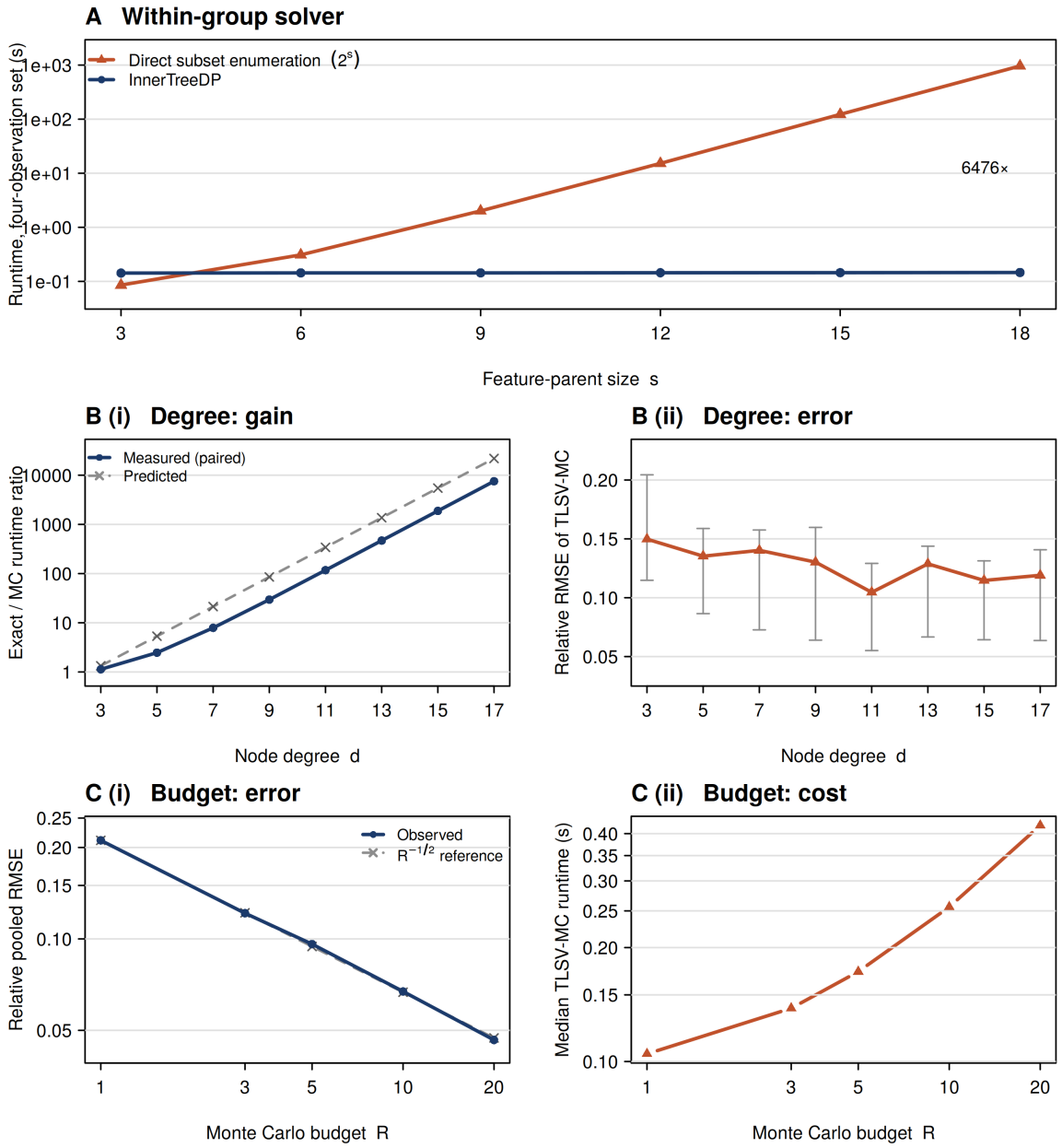


Figure 1: Simulation summary. **A**, Effect of the within-group solver. Algorithm 1 is run twice on the same fitted models and hierarchies, once with InnerTreeDP and once with direct enumeration of all $2^{|A(u)|}$ subsets at feature-parent nodes; the higher-level hierarchy is fixed at 3×3, so the number of context-conditioned inner solves is constant in s. The two arms agree to a maximum absolute discrepancy of 4.8×10^{-13} over the $s \leq 18$ grid, and the runtime curves cross between $s = 3$ and $s = 6$, beyond which the dynamic program is increasingly faster. **B**, Hierarchy-degree scaling in the single-level hierarchies $d \times 2$ at the fixed budget $R_u = 3$: (i) the measured paired speed-up of TLSV-MC over TLSV-Exact against the ratio $2^{d-1}/R_u$ predicted by the hierarchy-context counts, on a logarithmic scale; (ii) the relative root-mean-square error of TLSV-MC, which shows no systematic increase through $d = 17$, with error bars giving the interquartile range across cells. **C**, Monte Carlo budget sensitivity at the fixed high-degree hierarchy $d = 15$, where each cell uses the same fitted model, explained observations and TLSV-Exact reference for all five budgets, with 20 independent runs per budget: (i) median relative pooled RMSE against R, with a dashed reference proportional to $R^{-1/2}$ anchored at $R = 1$; (ii) median runtime of one TLSV-MC call. Throughout, points are medians over six learner–task settings and five simulation replicates.

Table 3: Numerical coherence over 12 hierarchy–learner–task cells and 100 replicates per cell (1200 replicate-level cells), reported as maxima over replicates.

Hierarchy	Learner	Task	Δ_{eff}	Δ_{blk} (exact)	MC block error	Δ_{agg}	Degree-one diff.
$2 \times 2 \times 2$	XGBoost	Regression	1.0×10^{-14}	6.6×10^{-15}	1.928	1.0×10^{-16}	0
	XGBoost	Classification	8.0×10^{-15}	9.2×10^{-15}	2.932	1.2×10^{-16}	0
	LightGBM	Regression	5.1×10^{-15}	1.2×10^{-14}	2.560	1.1×10^{-16}	0
	LightGBM	Classification	1.0×10^{-14}	1.2×10^{-14}	3.721	1.3×10^{-16}	0
	Ranger	Regression	3.4×10^{-15}	3.0×10^{-15}	0.431	8.1×10^{-17}	0
	Ranger	Classification	1.6×10^{-15}	1.2×10^{-15}	0.085	2.0×10^{-17}	0
$2 \times 3 \times 2$	XGBoost	Regression	4.3×10^{-15}	6.9×10^{-15}	1.537	1.4×10^{-16}	0
	XGBoost	Classification	7.4×10^{-15}	6.1×10^{-15}	2.399	1.6×10^{-16}	0
	LightGBM	Regression	7.9×10^{-15}	9.0×10^{-15}	1.482	1.3×10^{-16}	0
	LightGBM	Classification	5.6×10^{-15}	1.0×10^{-14}	2.579	1.7×10^{-16}	0
	Ranger	Regression	2.6×10^{-15}	3.1×10^{-15}	0.241	9.8×10^{-17}	0
	Ranger	Classification	1.7×10^{-15}	1.2×10^{-15}	0.059	2.4×10^{-17}	0

Δ_{blk} compares the aggregated feature-level attributions within each node with Γ_u evaluated independently from its own definition, and is the substantive test of multi-resolution coherence. The MC column reports the finite-budget error relative to the same exact block target at $R_u = 3$; it is an approximation error, not a failure of the exact multi-resolution identity. Δ_{agg} compares a parent sum with the sum of its children's sums; because the children partition the parent's feature set this vanishes identically for any feature-level vector, and it is reported as an implementation diagnostic only. The final column is the maximum change in the attributions after inserting degree-one carry-forward nodes, which the hierarchy normalization requires to be exactly zero.

design point. At the deepest setting, $m = 8$ and $p = 6{,}561$, each explained observation requires $35{,}831{,}808$ context-conditioned inner solves under TLSV-EXACT and $4{,}782{,}969$ under TLSV-MC. Over the 20-observation runtime set, the corresponding median solver runtimes are 509.8 and 74.7 seconds, and the median paired exact-to-Monte-Carlo ratio is 6.85, compared with 7.49 predicted by the context counts. Supplementary Figure S2 shows that the measured ratio moves toward the count-based prediction as depth increases and fixed overhead becomes less dominant. Every non-feature-parent node in these two sweeps has degree three, where exact expansion uses $2^{d-1} = 4$ sibling contexts per child against $R_u = 3$ sampled orderings; the per-node reduction is modest but compounds recursively with depth.

Figure 1B and Supplementary Table S5 vary node degree directly, where the hierarchy-level factors separate much more sharply. The number of hierarchy contexts grows as 2^{d-1} under TLSV-EXACT but only as R_u under TLSV-MC. The measured paired speed-up increases from 1.1 at $d = 3$ to $7{,}533$ at $d = 17$; at $d = 17$, exact computation takes $1{,}065.62$ seconds compared with 0.140 seconds for TLSV-MC. Across $d = 3, 5, \ldots, 17$, the median relative root-mean-square Monte Carlo error remains between 0.105 and 0.150 and shows no systematic increase with degree. DIRECT-ENUM remains feasible for $d \leq 5$ and agrees with TLSV-EXACT to 2.2×10^{-13}.

Monte Carlo budget sensitivity. Figure 1C examines the accuracy–cost trade-off while holding the high-degree hierarchy at $d = 15$ and changing only the local sampling budget. The median relative pooled RMSE decreases from 0.211 at $R = 1$ to 0.122, 0.096, 0.067, and 0.046 at $R = 3, 5, 10$, and 20, respectively. This decrease closely follows the $R^{-1/2}$ rate predicted by Proposition 3: the product of the median relative RMSE and $\sqrt{R}$ stays between 0.208 and 0.215 over the five budgets. The corresponding median TLSV-MC runtime increases from 0.105 to 0.421 seconds, while the paired exact-to-Monte-Carlo speed-up remains large, decreasing from approximately $2{,}430$ at $R = 1$ to 635 at $R = 20$; at the primary $R = 3$ setting it is approximately $1{,}872$. Thus, R provides a direct and predictable accuracy–cost control rather than changing the target estimand. Detailed budget-specific summaries are reported in Supplementary Table S3.

Summary. Taken together, the simulations tell a simple and consistent story. When exact computation is feasible, TLSV reproduces the intended levels-structure allocation to numerical precision, and the resulting explanations remain coherent across hierarchy resolutions (same-target discrepancies and exact coherence residuals are at floating-point scale, with degree-one invariance exactly zero). The two computational components then address different bottlenecks. INNERTREEDP makes large terminal feature groups practical by avoiding exhaustive feature-subset enumeration (a $6{,}476$-fold paired median advantage at $s = 18$), whereas hierarchy-level Monte Carlo avoids exhaustive averaging over sibling contexts at highly branched nodes (a $7{,}533$-fold paired median speed-up at $d = 17$). Finally, the Monte Carlo budget behaves as a transparent tuning parameter: increasing R steadily improves approximation accuracy at essentially the expected $R^{-1/2}$ rate, at the cost of additional computation. Thus, the primary choice $R = 3$ is an intentionally economical operating point rather than a fixed requirement; additional computation can be spent when greater approximation accuracy is needed.

4.3 Real Data Application

This section illustrates the practical utility of TLSV using gut metagenomic data to study response to immune checkpoint inhibitor therapy in a multi-study patient cohort. The data were compiled from five published studies considered by Limeta et al. (2020) (Matson et al., 2018; Frankel et al., 2017; Gopalakrishnan et al., 2018; Routy et al., 2018; Peters et al., 2019). Taxonomic annotations were standardized across studies; insufficiently resolved terminal assignments were excluded, while unresolved intermediate annotations were represented by hierarchy-preserving nodes. As quality-control steps, subjects with fewer than $2{,}000$ total reads and microbial features with mean relative abundance not exceeding 10^{-5} were excluded. Study-specific batch effects were reduced using the nonparametric correction method ConQuR (Ling et al., 2022), and the corrected counts were subsequently normalized to relative abundances for downstream analysis. The final hierarchy contained 379 species nested within 115 genera, 46 families, 29 orders, 19 classes, and 9 phyla. The final dataset contained 257 patients treated with immune checkpoint inhibitors, including 168 non-responders and 89 responders.

Model fitting and TLSV computation. A LightGBM binary classifier was fitted to the 379 species-level relative-abundance features, with a learning rate of 0.05, maximum tree depth 3, row and feature subsampling rates 0.8, and at most $1{,}000$ boosting iterations. Ten-fold stratified cross-validation on binary log loss with early stopping after 50 rounds selected 79 boosting rounds, after which the model was refitted to the complete cohort. Cross-validation was used only to select the boosting-round count and was not interpreted as an estimate of out-of-sample predictive performance.

TLSV-EXACT explanations were then computed for all 257 patients on the six-level taxonomy, with species as terminal features. All reported real-data attributions were obtained without hierarchy-level Monte Carlo sampling. Computational details and numerical validation are reported in Supplementary Section S5.

Attributions across six taxonomic levels. Figure 2 summarizes the allocation across six taxonomic resolutions. Node size encodes the mean absolute node attribution of (22), and node color encodes the fraction of patients receiving a positive attribution, with 0.5 as the midpoint. Size therefore summarizes the magnitude of a taxon's contribution, and color the prevalence of positive relative to negative contributions across patients.

The figure also makes the structure of the allocation visible. Only 116 of the 218 internal nodes and 116 of the 379 species receive a non-zero attribution, because the fitted ensemble never splits on the remaining features; the plotted tree is the subtree these induce. Firmicutes has the largest phylum-level importance, with substantial contributions distributed across several descendant lineages. Within Proteobacteria, by contrast, the similar importances from Gammaproteobacteria through Enterobacteriaceae, *Citrobacter*, and *C. koseri* indicate that the fitted model's signal is highly concentrated in this lineage. These node summaries are coherent aggregations of the same TLSV allocation across taxonomic resolutions, in the sense of Proposition 1(v).

Sign prevalence also differs across lineages. The Enterobacteriaceae–*Citrobacter* chain and the Bacteroidetes lineage have positive shares near 0.27 and 0.35, respectively, whereas Selenomonadales, Selenomonadaceae, and *Mitsuokella* have positive shares near 0.71, as do Ruminococcaceae and *Ruminococcus* at about 0.70. The positive orientation of Ruminococcaceae is consistent with the higher abundance reported among anti-PD-1 responders by Gopalakrishnan et al. (2018). *Akkermansia muciniphila*, which has been associated with ICI benefit in prior cohorts (Routy et al., 2018; Derosa et al., 2022), also appears in the hierarchy, although with a smaller and less consistently positive contribution in this fitted model.

These are descriptive explanations of how the fitted model uses the microbiome in this cohort rather than causal microbial effects. The hierarchical representation makes the patterns explicit at the taxonomic resolutions at which they arise.

Supplementary Table S8 lists every taxon with a non-zero attribution, arranged so that the nesting of the taxonomy is preserved, together with its mean absolute attribution and positive share.

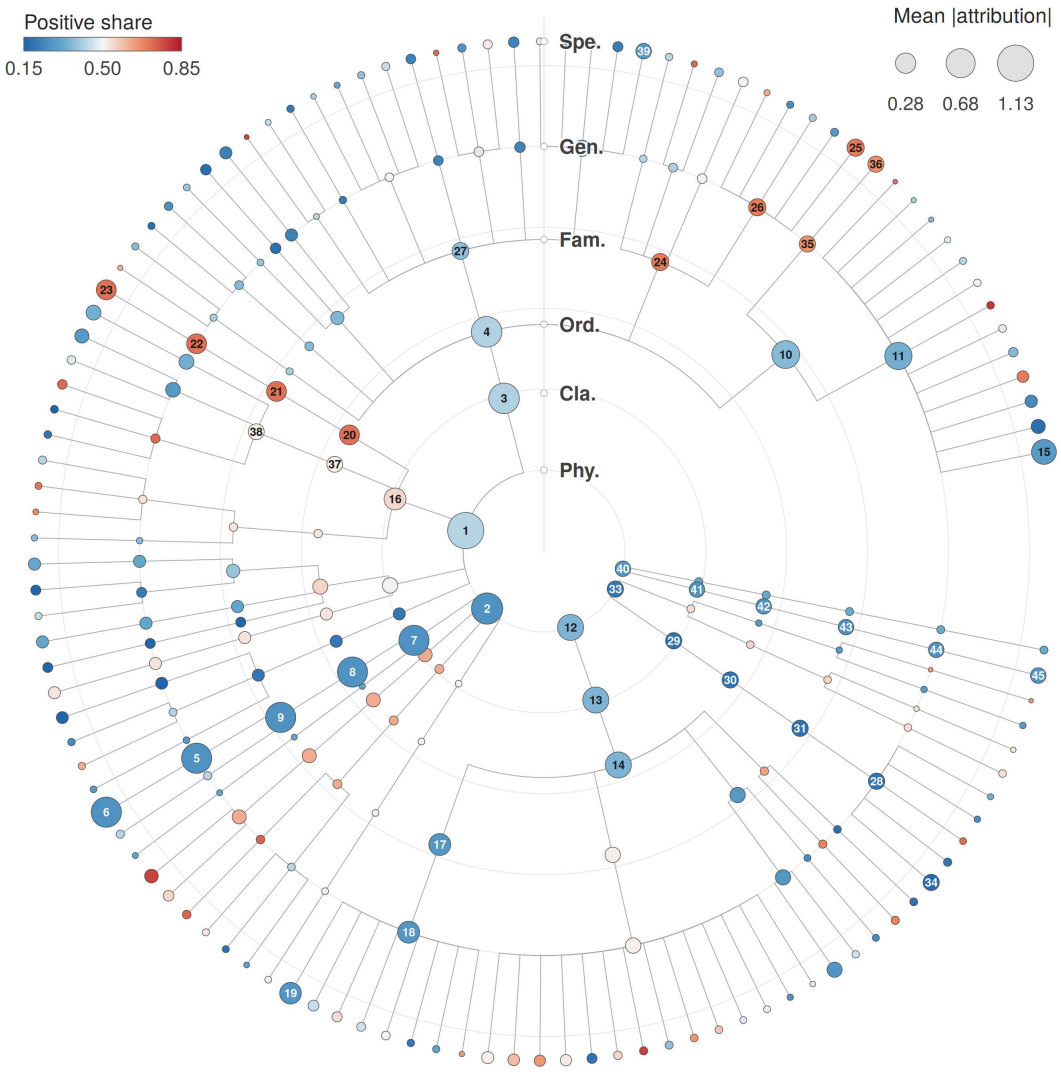


Figure 2: Levels-structure attributions across the six-level taxonomy for 257 immune checkpoint inhibitor patients. Rings denote taxonomic levels from phylum at the center to species at the rim. Node size encodes mean absolute node attribution I_u^{node} of (22), and node color encodes the fraction of patients with a positive attribution, diverging about 0.5 (blue, predominantly negative; red, predominantly positive). Only taxa with non-zero attribution and their ancestors are shown. The 45 most important taxa are numbered and identified, with numerical values, in Supplementary Table S8.

5 Discussion

In this paper, I developed TLSV as a knowledge-guided framework for multi-resolution attribution in tree ensemble models. TLSV applies the levels-structure value (Winter, 1989) to a prespecified feature hierarchy and combines hierarchy-aware allocation with tree-aware dynamic programming (Lundberg et al., 2020). This construction extends tree-based attribution from individual features and a single grouping level to multiple nested resolutions while maintaining a single coherent allocation across the hierarchy. The exact algorithm avoids exhaustive feature-subset enumeration within feature-parent groups, while the Monte Carlo and hybrid algorithms provide scalable alternatives when hierarchy-level branching makes exact computation costly.

The simulation studies supported the numerical validity, computational efficiency, and scalability of TLSV. TLSV-EXACT and INNERTREEDP agreed with their exhaustive reference computations to numerical precision, while the Monte Carlo estimator exhibited the expected accuracy–cost trade-off and preserved exact allocation efficiency and hierarchical aggregation. The immunotherapy application further illustrated the practical utility of TLSV by identifying both distributed and concentrated predictive signals across the six-level microbial taxonomy, including a broad Firmicutes signal and a more concentrated Proteobacteria–Enterobacteriaceae–*Citrobacter* lineage. Together, these results demonstrated that TLSV can provide computationally scalable and biologically interpretable explanations across multiple hierarchical resolutions.

Several considerations are worth noting. As a *post hoc* explanation method, TLSV describes the behavior of a fitted model and therefore inherits associations arising from the data, preprocessing, and model-fitting procedure; its attributions should not in themselves be interpreted causally. The hierarchy is also supplied by the analyst rather than learned from the data, so its scientific usefulness depends on the appropriateness of the chosen structure. Recent advances in efficient tree attribution and permutation sampling (Yang, 2021; Yu et al., 2022; Amoukou et al., 2022; Zern et al., 2023; Mitchell et al., 2022) suggest further computational refinements that may be adapted to the levels-structure setting. Other directions include broader model outputs and split types, overlapping or uncertain hierarchical structures, and hierarchy-aware interaction measures (Muschalik et al., 2024; Fumagalli et al., 2024; Kolpaczki et al., 2024).

Overall, TLSV provides a principled and computationally practical way to incorporate multi-level domain knowledge into explanations of tree ensemble models. By linking feature-, group-, and higher-level attributions within a single levels-structure allocation, it offers a useful framework for scientific applications in which predictors have meaningful nested organization.

Supplementary Materials

The Supplementary Information contains the proofs of Proposition 1, Proposition 2, Proposition 3 and Corollary 1 in Sections S1–S3, including the extension of the Monte Carlo results to the hybrid estimator. Section S4 gives the complete simulation designs, reference computations, parameter grids, implementation settings and diagnostics, together with the tables and figures reporting the inner-solver, hierarchy-degree, depth and feature-parent-size experiments and the detailed same-target agreement. Section S5 reports the exact computation, runtime and validation residuals for the real-data application and the full hierarchical attribution table.

Data and Code Availability

The `tlsv` R package, including the source code, documentation, and tutorials, is freely available at `https://github.com/hk1785/tlsv`. All processed datasets used in the real-data analyses are included in the package as example data.

Acknowledgements

I gratefully acknowledge the foundational contributions of Lloyd S. Shapley, Guillermo Owen, and Eyal Winter. Their seminal work laid the theoretical foundation for this study.

Funding

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. RS-2026-25477838).

Competing Interests

The author declares no competing interests.

Ethics Approval

This study is a secondary analysis of publicly available, de-identified human gut metagenomic data from previously published studies (Limeta et al., 2020). Ethical approval and informed consent were obtained in those original studies, and no additional institutional review board approval was required for the present work.

References

Aas, K., Jullum, M., & Løland, A. (2021). Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. *Artificial Intelligence*, 298, 103502.

Amoukou, S. I., Salaün, T., & Brunel, N. (2022). Accurate Shapley values for explaining tree-based models. *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, 151, 2448–2465.

Breiman, L. (2001). Random forests. *Machine Learning*, 45, 5–32.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794).

Derosa, L., Routy, B., Thomas, A. M., et al. (2022). Intestinal *Akkermansia muciniphila* predicts clinical response to PD-1 blockade in patients with advanced non-small-cell lung cancer. *Nature Medicine*, 28, 315–324.

Federhen, S. (2012). The NCBI Taxonomy database. *Nucleic Acids Research*, 40(D1), D136–D143.

Frankel, A. E., Coughlin, L. A., Kim, J., Froehlich, T. W., Xie, Y., Frenkel, E. P., & Koh, A. Y. (2017). Metagenomic shotgun sequencing and unbiased metabolomic profiling identify specific human gut microbiota and metabolites associated with immune checkpoint therapy efficacy in melanoma patients. *Neoplasia*, 19, 848–855.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29, 1189–1232.

Fumagalli, F., Muschalik, M., Kolpaczki, P., Hüllermeier, E., & Hammer, B. (2024). KernelSHAP-IQ: Weighted least square optimization for Shapley interactions. *Proceedings of the 41st International Conference on Machine Learning*, 235, 14308–14342.

Gopalakrishnan, V., Spencer, C. N., Nezi, L., et al. (2018). Gut microbiome modulates response to anti-PD-1 immunotherapy in melanoma patients. *Science*, 359, 97–103.

Janzing, D., Minorics, L., & Blöbaum, P. (2020). Feature relevance quantification in explainable AI: A causal problem. In *Proceedings of the Twenty-Third International Conference on Artificial Intelligence and Statistics* (PMLR 108, pp. 2907–2916).

Jullum, M., Redelmeier, A., & Aas, K. (2021). groupShapley: Efficient prediction explanation with Shapley values for feature groups. *arXiv preprint*, arXiv:2106.12228, `https://doi.org/10.48550/arXiv.2106.12228`.

Ke, G., Meng, Q., Finley, T., et al. (2017). LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems* (Vol. 30, pp. 3146–3154).

Koh, H. (2026). Efficient game-theoretic explanations for tree-based ensembles via Owen values. *bioRxiv preprint*, `https://doi.org/10.64898/2026.08.12.744440`.

Kolpaczki, P., Muschalik, M., Fumagalli, F., Hammer, B., & Hüllermeier, E. (2024). SVARM-IQ: Efficient approximation of any-order Shapley interactions through stratification. *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, 238, 3520–3528.

Limeta, A., Ji, B., Levin, M., et al. (2020). Meta-analysis of the gut microbiota in predicting response to cancer immunotherapy in metastatic melanoma. *JCI Insight*, 5, e140940.

Ling, W., Lu, J., Zhao, N., et al. (2022). Batch effects removal for microbiome data via conditional quantile regression. *Nature Communications*, 13, 5418.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems* (Vol. 30, pp. 4765–4774).

Lundberg, S. M., Erion, G., Chen, H., et al. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2, 56–67.

Matson, V., Fessler, J., Bao, R., et al. (2018). The commensal microbiome is associated with anti-PD-1 efficacy in metastatic melanoma patients. *Science*, 359, 104–108.

Mitchell, R., Cooper, J., Frank, E., & Holmes, G. (2022). Sampling permutations for Shapley value estimation. *Journal of Machine Learning Research*, 23(43), 1–46.

Muschalik, M., Fumagalli, F., Hammer, B., & Hüllermeier, E. (2024). Beyond TreeSHAP: Efficient computation of any-order Shapley interactions for tree ensembles. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(13), 14388–14396.

Owen, G. (1977). Values of games with a priori unions. In R. Henn & O. Moeschlin (Eds.), *Mathematical economics and game theory* (pp. 76–88). Springer.

Peters, B. A., Wilson, M., Moran, U., et al. (2019). Relating the gut metagenome and metatranscriptome to immunotherapy responses in melanoma patients. *Genome Medicine*, 11, 61.

Routy, B., Le Chatelier, E., Derosa, L., et al. (2018). Gut microbiome influences efficacy of PD-1-based immunotherapy against epithelial tumors. *Science*, 359, 91–97.

Shapley, L. S. (1953). A value for n-person games. In H. W. Kuhn & A. W. Tucker (Eds.), *Contributions to the theory of games II* (pp. 307–317). Princeton University Press.

Štrumbelj, E., & Kononenko, I. (2010). An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research*, 11, 1–18.

Štrumbelj, E., & Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, 41, 647–665.

Winter, E. (1989). A value for cooperative games with levels structure of cooperation. *International Journal of Game Theory*, 18, 227–240.

Wright, M. N., & Ziegler, A. (2017). ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*, 77, 1–17.

Yang, J. (2021). Fast TreeSHAP: Accelerating SHAP value computation for trees. *XAI 4 Debugging Workshop at NeurIPS 2021*.

Yu, P., Bifet, A., Read, J., & Xu, C. (2022). Linear tree Shap. In *Advances in Neural Information Processing Systems* (Vol. 35, pp. 25818–25828).

Zern, A., Broelemann, K., & Kasneci, G. (2023). Interventional SHAP values and interaction values for piecewise linear regression trees. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 37, pp. 11164–11173).

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Supplementary.pdf