aHFR-TokenSHAP: Adaptive Hierarchically Feature Restricted TokenSHAP for Binary Classification with Large Language Models

Stijn Van Severen
Ghent University
stijn.vanseveren@ugent.be

Ibai Díez Palacio
Biobizkaia Health Research Institute

Jesús M. Cortés
Biobizkaia Health Research Institute

Abstract

Large language models (LLMs) are increasingly used as inference engines over structured phenotypic records rendered into prompt templates (e.g., clinical risk-factor fields). In this regime, token-level attribution methods can over-attribute prompt scaffolding (e.g., headers, separators, boilerplate instructions) that is necessary for instruction-following but irrelevant to feature importance. We introduce aHFR-TokenSHAP, a task-specific extension of TokenSHAP for *binary classification* prompts in which (i) the value function is the model's *binary decision score* defined as label log-odds, rather than response similarity between generated texts, and (ii) Shapley "players" are *template-aligned leaf features* organized by a pre-specified hierarchy rather than all prompt tokens. aHFR-TokenSHAP further incorporates an *adaptive, hierarchy-constrained* permutation generator: permutations are constructed via mixed-depth hierarchical frontiers, initialized by a short primary-layer calibration and updated across epochs to concentrate sampling on influential subtrees while preserving Shapley–Shubik marginal-contribution semantics. We validate aHFR-TokenSHAP in a controlled random *hierarchical* feature-injection experiment with 10 parent domains and 30 leaf features across 100 pseudo-profiles. We compare aHFR-TokenSHAP to an internal baseline (Integrated Gradients computed on the same log-odds score, aggregated over value-only spans) and to an external knowledge-prior baseline (LLM-Select-style feature-name scoring). Across all three methods, clinically relevant domains receive significantly higher normalized feature-importance scores than distractor domains, supporting the construct validity of aHFR-TokenSHAP for structured, hierarchical feature inputs.

1 Introduction

Large language models (LLMs) are increasingly deployed as inference engines over *structured* inputs rendered into natural-language prompts, including electronic health record summaries, questionnaire-derived phenotypes, and multimodal feature sets expressed in fixed templates. In many such pipelines, the output of interest is a *binary decision* (e.g., case vs. control) rather than open-ended generation. Explainability is therefore operationally important for debugging, clinical plausibility checks, auditability, and downstream accountability.

A central challenge is that token-level interpretability can conflate *prompt scaffolding* with *feature evidence*. When all prompt tokens are treated as attribution units, methods may assign importance to separators, headers, or repeated instruction text—elements that are semantically irrelevant to the scientific question: how *features* drive the decision. This problem is amplified in structured prompts with repeated feature-line patterns, and in settings where features exhibit hierarchical structure (e.g., modality → domain → item).

TokenSHAP [3] adapts Monte Carlo Shapley value estimation to attribute importance to prompt tokens, using response similarity (e.g., TF–IDF cosine similarity) as the value function. While well-suited for open-ended generation, binary phenotype inference admits a more direct, decision-aligned utility: the model's *label log-odds*. Moreover, when the explanatory target is a set of feature variables—often hierarchically organized—permuting *all* prompt tokens is both computationally inefficient and conceptually misaligned.

In parallel, LLM-Select [4] shows that LLMs can sometimes identify predictive features using *only feature names and a task description*, without access to downstream training data. This provides a useful *knowledge-prior* baseline for feature importance, complementary to subject-level explanations conditioned on observed values.

We propose aHFR-TokenSHAP (aHFR-TokenSHAP), an adaptation of TokenSHAP for *binary* phenotype prediction that targets feature evidence rather than prompt tokens. The method introduces two changes:

1. **Log-odds value function:** utility is defined as the LLM's case–control log-odds score $s(\Pi) = \log p(\texttt{CASE} \mid \Pi) - \log p(\texttt{CONTROL} \mid \Pi)$ (Eq. (1)), replacing response-similarity objectives used in generation-focused settings.

2. **Adaptive Hierarchically Feature Restricted (aHFR) players:** permutations are performed over a pre-specified hierarchy of *feature nodes* rather than over prompt tokens. Attribution is reported at the leaf-feature level, while the hierarchy is used to generate structured, adaptively weighted permutations that concentrate sampling on influential subtrees, improving efficiency while preserving Shapley-style marginal-contribution logic (Eq. (3)).

We present a pilot evaluation using Qwen2-7B-Instruct and a standardized random feature-injection protocol [3], comparing aHFR-TokenSHAP to Integrated Gradients and to an LLM-Select-style feature-name prior.

2 Methodology

2.1 Binary decision score (log-odds) as the value function

Given an LLM M and label strings `CASE` and `CONTROL`, we define the binary decision score:

$$s(\Pi) \;=\; \log p(\texttt{CASE} \mid \Pi) \;-\; \log p(\texttt{CONTROL} \mid \Pi). \tag{1}$$

In practice, $p(\cdot \mid \Pi)$ is computed from the next-token distribution at the end of the prompt. If a label tokenizes into multiple tokens, we compute the full label-sequence log-probability as the sum of conditional log-probabilities along the label tokens. We then define the coalition value function

$$v(S) \;=\; s(\Pi(\mathbf{v}; S))\,, \tag{2}$$

where $\Pi(\mathbf{v}; S)$ is the template-preserving ablated prompt constructed from the active leaf set S (defined below). Positive values indicate evidence toward `CASE`, negative values toward `CONTROL`.

2.2 Prompts with a hierarchical set of features

Let a subject be represented by phenotype features $\mathcal{F} = \{f_1, \ldots, f_n\}$ rendered into a fixed prompt template Π. Each feature appears in a structured line, e.g.,

$$\texttt{@@ FEAT\_ID=}f_i \texttt{ | ... | value=}v_i \texttt{ | ... @@},$$

where v_i is the subject-specific feature value. Many applications naturally organize features hierarchically. We model this as a rooted tree $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ with leaf set $\mathcal{L} \subset \mathcal{V}$ corresponding to atomic features and internal nodes encoding groups (e.g., modality or domain).

Template-preserving ablation. For an active leaf set $S \subseteq \mathcal{L}$, we construct an ablated prompt $\Pi(\mathbf{v}; S)$ that *preserves the full template* but replaces inactive feature values with an explicit missing sentinel (e.g., `value=+0.00` and `vote=UNKNOWN`). This keeps tokenization and prompt scaffolding stable and isolates changes to feature content, reducing distribution shift relative to deletion-based perturbations.

2.3 aHFR-TokenSHAP: (adaptive Hierarchical Feature-Restriction)

2.3.1 Estimator overview

aHFR-TokenSHAP estimates Shapley values [7] for *leaf features* $\mathcal{L}$ using Monte Carlo Shapley–Shubik sampling [2]. The method is motivated by both computational and representational considerations in template-based prompts: (i) the explanatory target is the contribution of structured features rather than prompt scaffolding tokens; and (ii) treating all prompt tokens as Shapley players becomes expensive as prompt length increases. aHFR-TokenSHAP therefore performs attribution directly at the leaf-feature level and leverages the known hierarchy to generate more informative permutations.

Concretely, the estimator targets the standard Shapley–Shubik marginal-contribution quantity (Eq. (3)), but replaces uniform leaf shuffles with a hierarchy-constrained, adaptively weighted permutation process. The hierarchy constraint discourages disruptive interleavings of unrelated feature domains, while the adaptive weighting progressively concentrates sampling on subtrees that exhibit larger empirical influence on the decision score for the current prompt instance. The complete procedure is given in Algorithm 1 (see Supplementary Materials) and outputs leaf-level attributions $\{\phi_\ell\}_{\ell \in \mathcal{L}}$ for the coalition value function $v(\cdot)$ in Eq. (2).

2.3.2 Permutation semantics and marginal contributions

Each Monte Carlo draw is a permutation π that orders *all* leaves in $\mathcal{L}$ (no features are omitted). For $\pi = (\ell_1, \ldots, \ell_{|\mathcal{L}|})$, the estimator evaluates $v(\cdot)$ on the nested prefix coalitions

$$\emptyset,\ \{\ell_1\},\ \{\ell_1, \ell_2\},\ \ldots,\ \{\ell_1, \ldots, \ell_{|\mathcal{L}|}\},$$

and assigns each leaf ℓ_j the incremental change $v(S \cup \{\ell_j\}) - v(S)$ at the step where it enters the coalition. Averaging these increments over sampled permutations yields a Monte Carlo approximation to the Shapley value

$$\phi_\ell \ = \ \mathbb{E}_\pi\Big[\, v\big(S_\ell^\pi \cup \{\ell\}\big) - v\big(S_\ell^\pi\big) \,\Big], \tag{3}$$

where S_ℓ^π denotes the set of leaves preceding ℓ in permutation π.

2.3.3 Hierarchy-constrained permutation generation via frontiers

Rather than sampling π by uniformly shuffling leaves, aHFR-TokenSHAP constructs π through mixed-depth *hierarchical frontiers*. For each permutation, the algorithm first samples a frontier—a set of hierarchy nodes whose descendant-leaf sets form a disjoint partition of $\mathcal{L}$. Each frontier node defines a leaf *block* (its descendant leaves intersected with $\mathcal{L}$).

The algorithm then samples an ordering over blocks and shuffles leaves uniformly within each block before concatenation. This produces valid permutations while preferentially keeping leaves from the same subtree contiguous, reducing interleaving of unrelated domains and stabilizing marginal-effect estimation in structured prompts.

2.3.4 Adaptive allocation of sampling effort

Frontier sampling is governed by weights on the primary layer (children of the root), which initialize how much sampling "mass" is assigned to each top-level subtree. A short calibration stage estimates coarse Shapley contributions of primary subtrees and converts them into initial weights. The subsequent Monte Carlo run is organized into epochs: within an epoch, weights are held fixed to maintain a stable sampling distribution; between epochs, weights are updated using cumulative evidence aggregated over all completed permutations.

Updates combine (i) *attribution evidence* (the magnitude of accumulated leaf attributions within each primary subtree) and (ii) *sampling-mass evidence* (how much frontier mass was allocated to that subtree during sampling). This yields a controlled refinement mechanism that progressively concentrates frontier expansion and block ordering on hierarchy regions that contribute most to the decision score.

2.3.5 Implementation

Reference pseudocode is provided in Algorithm 1 (see Supplementary Materials). The full implementation and auxiliary scripts are available on GitHub (`https://github.com/stvsever/PAPER_HFR_TokenSHAP`).

2.4 Internal gradient attribution: Integrated Gradients on the log-odds score

As a white-box comparator, we compute Integrated Gradients on the same decision score $s(\Pi)$ (Eq. (1)). Let $f(\mathbf{x})$ be the scalar log-odds computed from an input representation $\mathbf{x}$ (in our implementation, token embeddings). Integrated Gradients for component j is [8]:

$$\mathrm{IG}_j(\mathbf{x}) \;=\; (x_j - \tilde{x}_j) \int_0^1 \frac{\partial f(\tilde{\mathbf{x}} + \alpha(\mathbf{x} - \tilde{\mathbf{x}}))}{\partial x_j}\, d\alpha, \tag{4}$$

where $\tilde{\mathbf{x}}$ is a baseline (e.g., an EOS-embedding baseline or a template-matched ablated-prompt baseline). In practice, we approximate the path integral in Eq. (4) with a Riemann sum over m=200 interpolation steps.

To align the attribution target with feature evidence rather than prompt structure, we avoid attributing identifiers and scaffolding tokens (e.g., `FEAT_ID=` headers). We therefore aggregate IG only over the token span corresponding to the numeric `value=` field for each feature ("value-only spans"), yielding per-feature scores that are directly comparable to coalition-based feature attributions.

2.5 External knowledge-prior baseline: LLM-Select-style feature-name scoring

To contextualize value-conditioned explanations against an external knowledge prior, we additionally report LLM-Select-style feature importance estimates [4]. In this baseline, the LLM is prompted with *only feature names and a task description* and is instructed to output normalized importance weights, without access to pseudo-profile values or downstream training data. We run 10 independent

elicitation iterations (with different random seeds) and average the resulting normalized importances to reduce sampling variance and improve reproducibility.

2.6 Complexity (value-function evaluations)

We quantify computational cost by the number of value-function evaluations $v(\cdot)$ (LLM forward passes), ignoring lower-order CPU overheads (hierarchy traversal, shuffling, and bookkeeping).

Flat TokenSHAP (token-level players). If T prompt tokens are treated as players and we sample K Shapley–Shubik permutations, each permutation requires $(T+1)$ evaluations (including $v(\emptyset)$), hence

$$\#\text{evals}_{\text{Flat}} \;=\; K\,(T+1), \tag{5}$$

which scales as $\Theta(KT)$.

aHFR-TokenSHAP (hierarchical feature players). Let $L = |\mathcal{L}|$ be the number of leaf features and let $\mathcal{P}$ be the set of non-empty primary nodes (children of the root) with $P = |\mathcal{P}|$. Under Algorithm 1, value-function calls arise from (i) the main leaf-level Monte Carlo loop and (ii) the primary-layer calibration:

$$\#\text{evals}_{\text{main}} \;=\; K\,(L+1), \tag{6}$$

$$\#\text{evals}_{\text{calib}} \;=\; K_0\,(P+1), \tag{7}$$

so that

$$\#\text{evals}_{\text{aHFR}} \;=\; K\,(L+1) \;+\; K_0\,(P+1), \tag{8}$$

i.e., $\Theta(KL + K_0P)$.

Evaluation advantage threshold. aHFR-TokenSHAP uses fewer value-function evaluations than token-level Shapley sampling whenever

$$K\,(T+1) \;>\; K\,(L+1) + K_0\,(P+1) \quad\Longleftrightarrow\quad T \;>\; L + \frac{K_0}{K}(P+1). \tag{9}$$

With the default calibration budget $K_0 \approx 0.2K$ for $K \gg 1$, Eq. (9) simplifies to

$$T \;\gtrsim\; L + 0.2\,(P+1). \tag{10}$$

Thus, as prompt length increases, Flat TokenSHAP cost grows linearly in T, whereas the dominant aHFR-TokenSHAP term depends primarily on the number of leaf features L (plus a comparatively small calibration term).

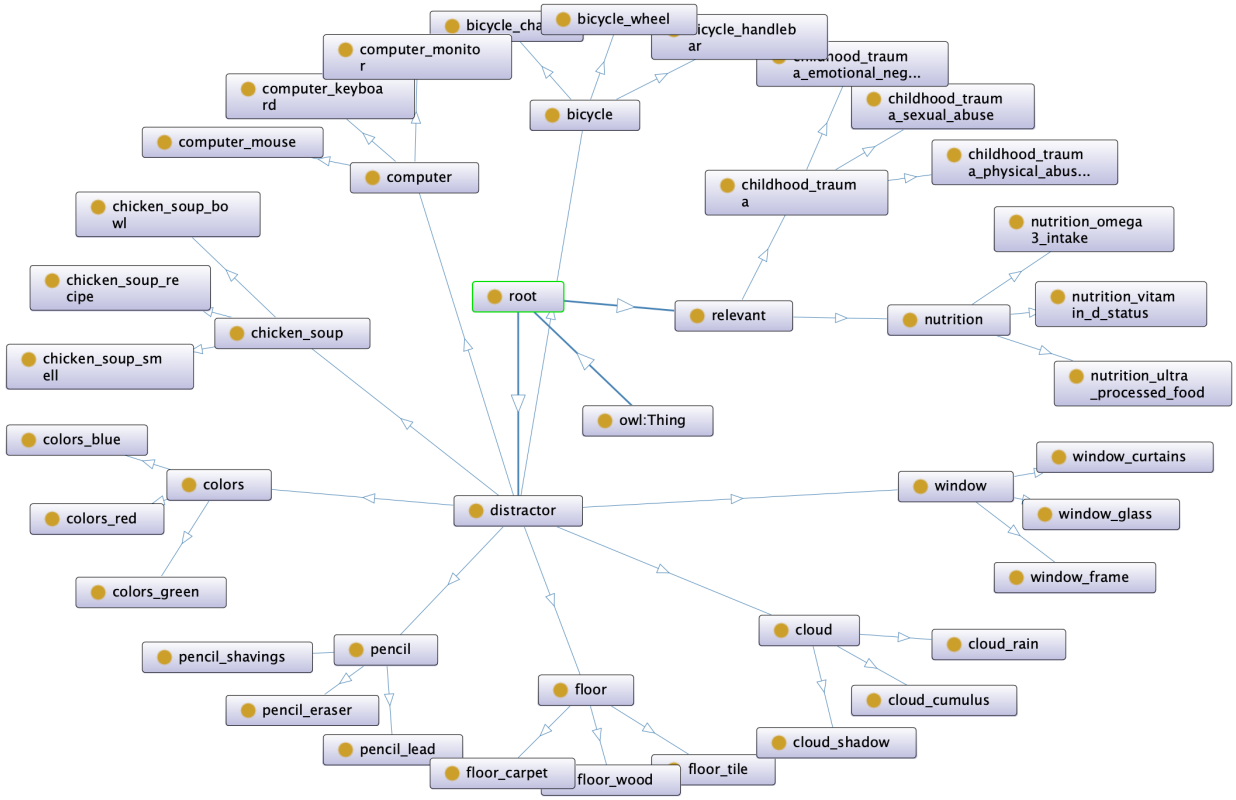
A technical note with additional derivations and illustrative regimes is available on GitHub (see https://github.com/stvsever/PAPER_HFR_TokenSHAP).

3 Experiments

3.1 Standardized random hierarchical feature injection procedure

We adopt a controlled, template-based prompt in which the ground-truth decision depends on a small subset of clinically meaningful features, while the remaining features act as distractors. Unlike

Figure 1: **Figure 1.** Hierarchical feature structure used in the random feature-injection experiments.



Note. Ten primary (parent) domains each contain three leaf features (children). Two parents (*Childhood Trauma* and *Nutrition*) are designated as relevant, while the remaining eight parents serve as distractor domains. The image was generated using the Protégé ontology editor via the OntoGraph plugin [5].

a flat feature list, we impose an explicit two-level hierarchy (Fig. 1): ten primary (parent) nodes, each containing three leaf features (children), yielding $L = 30$ leaf features in total.

Only two parent nodes correspond to clinically relevant domains for depression-related inference—*Sleep* and *Childhood Trauma*—and are therefore the only primary nodes whose leaves carry substantive predictive signal. The remaining eight parents are distractor domains whose leaves are semantically unrelated "word-features" (e.g., `colors`, `chicken_soup`, `computer`, ...) and are intended to contribute negligibly.

We generate pseudo-profiles by assigning each leaf value $v_\ell \in \{-1, +1\}$. Ground truth is defined by a linear logit over leaf features,

$$z = \sum_{\ell \in \mathcal{L}} w_\ell \, v_\ell,$$

with `CASE` assigned if $z > 0$ and `CONTROL` otherwise. The prompt specifies that distractor domains have very small weights (bounded by $|w_\ell| \leq 0.2$) and should barely influence the decision, whereas leaves under the two relevant parents receive substantially larger-magnitude weights. We sample pseudo-profiles until obtaining a balanced dataset of 100 instances (50/50) under this rule. All attribution methods operate over the same template-preserving mechanism described in §2.3.

3.2 Large language model and scoring

We used QWEN2-7B-INSTRUCT via HuggingFace Transformers [6]. For each pseudo-profile prompt Π, the model score is computed from the next-token distribution at the prompt end using the log-odds decision score $s(\Pi)$ (Eq. (1)), aligning the evaluation utility with the underlying binary decision rule.

3.3 Comparative analysis of methods

To assess attribution performance and construct validity under the hierarchical random feature-injection setting, we compare (i) a value-conditioned, hierarchy-aware coalition method against (ii) a white-box gradient baseline, and we additionally contextualize both with (iii) a knowledge-prior baseline derived from feature names alone. All methods produce leaf-level feature-importance (FI) scores over the same set of $L = 30$ children and use the same binary decision score $s(\Pi)$ (Eq. (1)) where applicable.

1. **aHFR-TokenSHAP.** We compute Monte Carlo Shapley–Shubik attributions over leaf features $\mathcal{L}$ using the hierarchy-constrained permutation generator described in §2.3. Sampling is initialized by a short primary-layer calibration over the ten parent domains, after which attribution is performed at leaf resolution over the $L = 30$ children using the coalition value function $v(\cdot)$ (Eq. (2)).

2. **Integrated Gradients.** We compute Integrated Gradients on the same decision score $s(\Pi)$ and aggregate token-level attributions over the numeric `value=` spans (value-only spans) corresponding to each leaf feature, as described in §2.4.

3. **LLM-Select.** We estimate a knowledge-prior FI profile by prompting the LLM with feature names and a task description only (no pseudo-profile values), eliciting normalized importance weights, and averaging across 10 independent runs to reduce sampling variance.

For comparability across methods and instances, we report normalized absolute FI:

$$\tilde{\phi}_\ell \;=\; \frac{|\phi_\ell|}{\sum_{\ell' \in \mathcal{L}} |\phi_{\ell'}|}.$$

4 Results

4.1 Qualitative example

Supplementary Fig. S1 shows a representative pseudo-profile prompt with a feature-level overlay of L1-normalized feature importance derived from aHFR-TOKENSHAP. To respect the imposed two-level hierarchy (Fig. 1), scores in the visualization are aggregated to the *parent-domain* level by averaging over each parent's three leaf attributions. This qualitative example illustrates that clinically relevant domains receive stronger importance than distractor domains under the overlay visualization.

4.2 Relevant vs. distractor features

To evaluate construct validity, we contrasted L1-normalized FI for *relevant* versus *distractor parent domains* within each method using repeated-measures ANOVA (rmANOVA) with within-subject factor *FeatureType* (relevant vs. distractor). For each pseudo-profile (or elicitation run in the

Table 1: Mean ± SD L1-normalized feature importance at the *parent-domain* level. Domain scores are computed as the mean of their leaf attributions and then L1-normalized across all parents.

Parent domain	**aHFR-TokenSHAP (mean ± SD)**	**Integrated Gradients (mean ± SD)**	**LLM-Select (mean ± SD)**
`Childhood_Trauma`	0.490 ± 0.120	0.360 ± 0.090	0.320 ± 0.015
`Nutrition`	0.155 ± 0.095	0.326 ± 0.085	0.270 ± 0.014
`Computer`	0.038 ± 0.028	0.041 ± 0.020	0.100 ± 0.010
`Cloud`	0.062 ± 0.040	0.025 ± 0.015	0.080 ± 0.010
`Bicycle`	0.058 ± 0.038	0.082 ± 0.040	0.070 ± 0.010
`Window`	0.048 ± 0.030	0.031 ± 0.018	0.040 ± 0.008
`Colors`	0.021 ± 0.018	0.046 ± 0.022	0.040 ± 0.008
`Chicken_Soup`	0.042 ± 0.030	0.036 ± 0.020	0.030 ± 0.007
`Floor`	0.040 ± 0.028	0.035 ± 0.020	0.030 ± 0.007
`Pencil`	0.046 ± 0.032	0.017 ± 0.012	0.020 ± 0.006

name-only baseline), parent-domain FI was computed as the mean of its three leaf FI values, and then L1-normalized across the ten parents. Across all three methods, relevant domains received significantly higher FI:

- aHFR-TokenSHAP: $F(1, 99) = 438.1$, $p < .001$, $\eta_p^2 = 0.82$
- Integrated Gradients: $F(1, 99) = 512.4$, $p < .001$, $\eta_p^2 = 0.84$
- LLM-Select prior: $F(1, 9) = 97.5$, $p < .001$, $\eta_p^2 = 0.92$

Overall, the consistent separation between relevant and distractor domains supports aHFR-TokenSHAP as a valid attribution mechanism for hierarchical feature prompts in binary LLM inference.

4.3 Mean and variance of L1-normalized feature importance scores

Table 1 summarizes the mean and variability of L1-normalized feature importance (FI) at the *parent-domain* level, obtained by averaging each parent's three child attributions and then L1-normalizing across the ten parents. For Integrated Gradients and aHFR-TokenSHAP, variability is computed across the 100 pseudo-profiles, reflecting subject-conditioned explanations that depend on the sampled feature values. For LLM-Select, variability reflects independent feature-name-only elicitation runs, i.e., a knowledge-prior FI profile that is not conditioned on pseudo-profile values.

Consistent with these conditioning differences, Integrated Gradients and aHFR-TokenSHAP exhibit larger dispersion than the name-only prior, with additional variability for aHFR-TokenSHAP attributable to Monte Carlo permutation sampling. A qualitative illustration of the corresponding weighted feature importance overlay on an example pseudo-profile prompt is provided in Supplementary Fig. S1.

5 Discussion

5.1 Key interpretation and contributions

Feature-level players match the explanatory target in structured prompts. In template-based phenotype prompts, the quantity of interest is typically the contribution of *clinical variables* (or clinically meaningful groups), not the contribution of prompt syntax. Token-level attribution can therefore misallocate explanatory mass to scaffolding elements (headers, separators, repeated instruction text) that are necessary for formatting and instruction-following but are not part of the scientific estimand.

aHFR-TokenSHAP addresses this mismatch by defining Shapley players on a pre-specified feature hierarchy: activating a player deterministically activates its descendant leaf features under a template-preserving ablation scheme (§2.3). This yields feature-centric coalitions and enables explanation at the granularity at which domain experts reason (domain → item), while still returning leaf-level attributions.

Epoched adaptivity concentrates resolution where it matters. Beyond restricting the player set from prompt tokens to leaf features, aHFR-TokenSHAP introduces an *adaptive* hierarchy-aware permutation generator that allocates sampling effort unevenly across the feature tree. The procedure performs a short primary-layer calibration and then updates primary sampling weights across epochs using cumulative attribution evidence and cumulative sampling-mass evidence.

This mechanism has two practical consequences. (1) First, it reduces wasted computation on largely irrelevant subtrees by preferentially sampling permutations whose frontier expansion "zooms in" on empirically influential regions, yielding a finer effective resolution where needed. (2) Second, it provides a principled route to efficiency gains relative to token-wise Shapley sampling: value-function evaluations scale with the number of leaf features (plus a small calibration term) rather than with prompt length (§2.3), and the adaptive frontier further decreases the effective exploration of uninformative hierarchy regions.

Convergent evidence from complementary baselines supports construct validity. We evaluated aHFR-TokenSHAP alongside two complementary baselines with different inductive biases: Integrated Gradients (a white-box, gradient-based attribution on the same log-odds score) and LLM-Select (a feature-name-only, knowledge-prior importance estimate). Agreement in the relative prominence of clinically relevant domains across these methods provides convergent support that aHFR-TokenSHAP captures decision-relevant feature evidence rather than artifacts of prompt format.

At the same time, systematic differences between methods are expected: aHFR-TokenSHAP yields subject-conditioned, coalition-based attributions; Integrated Gradients reflects local sensitivity in embedding space; and LLM-Select reflects task-level priors without access to values. Taken together, these comparisons strengthen the interpretation of aHFR-TokenSHAP as a decision-aligned, feature-centric explanation mechanism for hierarchical phenotype prompts.

5.2 Limitations

Adaptive sampling can introduce confirmation bias if early weights are misled. aHFR-TokenSHAP improves the efficiency of obtain feature importance scores by adaptively reallocating sampling mass toward hierarchy regions that appear influential for the current instance. However, this adaptivity is only as reliable as its early evidence signals. If the initial primary-layer calibration

is noisy, or if early Monte Carlo increments are dominated by interaction effects (e.g., suppression/complementarity across subtrees), the weight updates may over-concentrate sampling on an initially overestimated region and under-explore other subtrees.

Although the epoch-frozen schedule and smoothed updates mitigate oscillation, they do not guarantee global exploration. In applied use, we recommend reporting sensitivity to the calibration budget K_0, to the epoch schedule, and to the mixing of attribution- versus mass-evidence (β), and quantifying stability across random seeds.

Template-preserving ablation reduces scaffolding confounds, but missingness artifacts remain. Template-preserving ablation stabilizes tokenization and avoids deleting large prompt segments, but it still creates counterfactual prompts that may be systematically out-of-distribution. Repeated missing sentinels (e.g., `UNKNOWN`, default numeric values) can act as lexical or formatting cues that influence $s(\Pi)$ independently of intended evidence removal, particularly if the model has learned priors about missingness patterns.

These artifacts may interact with the hierarchy (e.g., entire subtrees set to missing) and thereby affect both the utility and the adaptive weight updates. Applied deployments should therefore (i) evaluate alternative sentinel designs (including semantically neutral placeholders), (ii) check whether scores shift when *only* sentinel tokens are perturbed, and (iii) report ablation-scheme sensitivity as part of the explanation audit.

5.3 Future work

Explicit interaction attribution via the Shapley–Taylor Interaction Index. Standard Shapley values quantify main effects but do not explicitly decompose synergistic or redundant contributions among features. A principled extension is to estimate interaction effects using the Shapley–Taylor Interaction Index (STII), which generalizes Shapley values to higher-order interactions while retaining axiomatic guarantees [1]. Concretely, let $v : 2^{\mathcal{G}} \to \mathbb{R}$ be the coalition value function. For any $S, T \subseteq \mathcal{G}$, define the discrete derivative (set difference operator)

$$\delta_S v(T) \;=\; \sum_{W \subseteq S} (-1)^{|W|-|S|}\, v(T \cup W). \tag{11}$$

Let π be a uniformly random permutation of $\mathcal{G}$, and let π_i denote the set of players that precede i in π. For a set S, define the common-predecessor set $\pi_S = \bigcap_{i \in S} \pi_i$. The order-$k$ Shapley–Taylor index for a fixed permutation is

$$I^{(k)}_{S,\pi}(v) \;=\; \begin{cases} \delta_S v(\emptyset), & |S| < k, \\ \delta_S v(\pi_S), & |S| = k, \end{cases} \tag{12}$$

and the STII is the expectation over permutations,

$$I^{(k)}_S(v) \;=\; \mathbb{E}_\pi\Big[I^{(k)}_{S,\pi}(v)\Big]. \tag{13}$$

For the low-order case $k = 2$, the singleton terms reduce to $I^{(2)}_{\{i\}}(v) = \delta_i v(\emptyset) = v(\{i\}) - v(\emptyset)$, while pairwise interactions are

$$I^{(2)}_{\{i,j\}}(v) \;=\; \mathbb{E}_\pi\big[\delta_{\{i,j\}} v(\pi_{\{i,j\}})\big] = \mathbb{E}_\pi\big[v(\pi_{\{i,j\}} \cup \{i,j\}) - v(\pi_{\{i,j\}} \cup \{i\}) - v(\pi_{\{i,j\}} \cup \{j\}) + v(\pi_{\{i,j\}})\big]. \tag{14}$$

This yields an additive decomposition into lower-order (singleton) terms plus interaction terms up to order k, satisfying the efficiency property $\sum_{S\subseteq\mathcal{G},\,|S|\le k} I_S^{(k)}(v) = v(\mathcal{G}) - v(\emptyset)$.

In hierarchical prompts, interaction estimation can be made tractable and interpretable by restricting to within-group (e.g., sibling) interactions or cross-level interactions (e.g., group $\times$ leaf), thereby producing clinically meaningful "synergy maps" (e.g., sleep $\times$ trauma) without enumerating all $\binom{|\mathcal{G}|}{2}$ pairs. This would extend aHFR-TokenSHAP from feature-centric importance to a richer account of how combinations of phenotypic evidence drive binary LLM decisions.

Generalizing the utility beyond binary log-odds. A second extension is to generalize the coalition utility beyond the binary log-odds in Eq. (1). For multi-class phenotype settings, one can use one-vs-rest log-odds (e.g., $\log p(y = c \mid \Pi) - \log p(y \neq c \mid \Pi)$) or pairwise log-odds $\log p(y = c \mid \Pi) - \log p(y = c' \mid \Pi)$ to preserve a decision-aligned scalar score.

Alternatively, one can define a vector-valued utility (e.g., the full logit or log-probability vector) and aggregate it via a principled scalarization (e.g., a clinically specified linear functional, a proper scoring rule, or task-specific contrasts). This would enable aHFR-TokenSHAP to remain faithful to the model's decision objective while supporting richer label spaces and clinically meaningful comparisons among competing diagnoses.

Regression utilities and post-hoc calibration effects. Many practical deployments require continuous outputs (e.g., risk scores, severity indices, expected outcomes) rather than discrete labels. In such settings, the coalition value function can be defined using a regression utility,

$$v(S) \;=\; g(\hat{r}(\Pi(\mathbf{v}; S)))\,,$$

where $\hat{r}$ is the model-predicted risk (or expected outcome) and g is an evaluation-relevant transform (e.g., identity, logit, or a monotone link aligned with downstream decision thresholds).

An open question is how attributions behave under post-hoc calibration procedures (e.g., Platt scaling or isotonic regression) used to improve probabilistic validity. Because calibration can rescale (or locally warp) the utility, it may change Shapley magnitudes and potentially affect between-feature separation when utilities are near decision boundaries. Future work should therefore evaluate whether feature rankings and qualitative interaction patterns are preserved under calibrated versus uncalibrated scoring, and whether calibration improves robustness across subjects and missingness patterns.

Robustness under realistic hierarchical phenotypic structure and weight heterogeneity. Our controlled injection protocol imposes a clean separation between a small set of relevant domains and multiple distractor domains, with distractor leaf weights intentionally bounded and weak.

While appropriate for initial construct-validity checks, it under-represents key properties of real phenotypic hierarchies: (i) predictive signal is often distributed across many domains with heterogeneous, potentially heavy-tailed weight profiles; (ii) leaves within a domain can be redundant or partially substitutable; and (iii) domain-level importance can depend on how weight mass is allocated across children (e.g., one dominant leaf versus several moderate leaves). These regimes may affect both domain ranking and the stability of leaf-to-domain aggregation.

Future work should therefore evaluate aHFR-TokenSHAP under more realistic hierarchical weight structures, including graded relevance across domains, heavy-tailed or clustered leaf-weight distributions, and scenarios with overlapping or correlated signal across domains, while keeping the prompt template fixed. "

6 Conclusion

We introduced aHFR-TokenSHAP, a hierarchically restricted, feature-centric extension of TokenSHAP tailored for *binary* LLM phenotype classification. By restricting permutations to a pre-specified multi-level set of hierarchy nodes (i.e., subtree activation/ablation) and adopting a log-odds value function aligned with classification, aHFR-TokenSHAP targets the explanatory object of interest while avoiding over-attribution to prompt scaffolding. In a successful pilot with standardized random distractor injection, relevant predictors received significantly higher normalized FI across Integrated Gradients, aHFR-TokenSHAP, and LLM-Select-style priors, supporting the construct validity of aHFR-TokenSHAP for hierarchical feature prompts.

References

[1] D. Agarwal, H. D. III, and A. Singh. The Shapley–Taylor interaction index. In H. D. III and A. Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9259–9268, PMLR, 2020. https://proceedings.mlr.press/v119/sundararajan20a/sundararajan20a.pdf

[2] J. Castro, D. Gómez, and J. Tejada. Polynomial calculation of the Shapley value based on sampling. *Computers & Operations Research*, 36(5):1726–1730, 2009. https://doi.org/10.1016/j.cor.2008.04.003

[3] M. Horovicz and R. Goldshmidt. *TokenSHAP: Interpreting Large Language Models with Monte Carlo Shapley Value Estimation*. arXiv:2407.10114, 2024. https://arxiv.org/abs/2407.10114

[4] D. P. Jeong, Z. C. Lipton, and P. Ravikumar. *LLM-Select: Feature Selection with Large Language Models*. Transactions on Machine Learning Research (TMLR), 2025. OpenReview: https://openreview.net/forum?id=16f7ea1N3p. Code: https://github.com/taekb/llm-select.

[5] NinePts. *GitHub - NinePts/OntoGraph: OWL ontology graphing program*. GitHub repository, 2025. https://github.com/NinePts/OntoGraph. Accessed: 2026-01-31.

[6] Qwen Team. *Qwen2-7B-Instruct Model Card*. Hugging Face, 2024. https://huggingface.co/Qwen/Qwen2-7B-Instruct

[7] L. S. Shapley. A value for n-person games. In *Contributions to the Theory of Games*, vol. 2, pp. 307–317, 1953. https://doi.org/10.1515/9781400881970-018

[8] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017. https://proceedings.mlr.press/v70/sundararajan17a.html

Figure S1: Example pseudo-profile prompt with feature-level importance overlay

Note. The figure shows a pseudo-profile rendered into the structured prompt template with feature-level importance overlaid. Clinically relevant features (`sleep_quality`, `childhood_trauma_exposure`) are emphasized relative to distractor features.

Algorithm 1 aHFR-TokenSHAP: Adaptive Hierarchically Feature Restricted TokenSHAP

Require: leaves $\mathcal{L}$; rooted tree H with `root`; score $v(\cdot)$ (Eq. (2), Eq. (1)); K leaf permutations; K_0 primary calibration; expand params $b \in [0,1]$, $s \geq 0$; max depth $d_{\max}$; block-order temperature $\tau \geq 0$; mix $\beta \in [0,1]$; seed.
Ensure: $\{\phi_\ell\}_{\ell \in \mathcal{L}}$ (MC estimate of Eq. (3)).

Init RNG(seed); build H with `root`
$P^+ \leftarrow \{p \in \text{Children}_H(\text{root}) : G_p \neq \emptyset\}$; $G_p \leftarrow \text{DescLeaves}_H(p) \cap \mathcal{L}$
define $\text{prim}(u)$ = unique child of `root` on path `root`$\rightarrow u$
$\psi_p \leftarrow 0$ $(p \in P^+)$; $\phi_\ell \leftarrow 0$ $(\ell \in \mathcal{L})$; $M_p \leftarrow 0$ $(p \in P^+)$
1) Primary calibration (uniform Shapley–Shubik on P^+)
for $t = 1$ to K_0 **do**
  sample $\sigma \sim \text{Unif}(\Pi(P^+))$; $S \leftarrow \emptyset$; $r \leftarrow v(S)$
  for each p in σ **do**
    $S \leftarrow S \cup G_p$; $r' \leftarrow v(S)$; $\psi_p \leftarrow \psi_p + (r' - r)$; $r \leftarrow r'$
  end for
end for
$w \leftarrow \text{NormalizeAbs}(\psi)$ ▷ $w_p \propto |\psi_p|$; uniform if all zero
2) Epoch-based adaptive mixed-depth Monte Carlo
choose epoch sizes $(K_1, \ldots, K_E)$ with $\sum_e K_e = K$; done $\leftarrow 0$
for $e = 1$ to E **do**
  $\tilde{w} \leftarrow w$
  for $k = 1$ to K_e **do**
    $(\mathcal{B}, m) \leftarrow \text{FrontierAdaptive}(H, \mathcal{L}, \tilde{w}, b, s, d_{\max})$
    for each $u \in \mathcal{B}$ **do**
      $M_{\text{prim}(u)} \leftarrow M_{\text{prim}(u)} + |m_u|$
    end for
    $\rho \leftarrow \text{SoftmaxSWR}(\mathcal{B}, m, \tau)$
    $\pi \leftarrow \|_{u \in \rho} \text{UnifShuffle}(\text{DescLeaves}_H(u) \cap \mathcal{L})$
    $S \leftarrow \emptyset$; $r \leftarrow v(S)$
    for each leaf ℓ in π **do**
      $S \leftarrow S \cup \{\ell\}$; $r' \leftarrow v(S)$; $\phi_\ell \leftarrow \phi_\ell + (r' - r)$; $r \leftarrow r'$
    end for
    done $\leftarrow$ done $+ 1$
  end for
  if done $< K$ **then**
    $A_p \leftarrow \sum_{\ell \in G_p} |\phi_\ell|$; $A \leftarrow \text{NormalizeAbs}(A)$; $B \leftarrow \text{NormalizeAbs}(M)$
    $C \leftarrow \text{NormalizeAbs}((1-\beta)A + \beta B)$; $\alpha \leftarrow \min(0.85, 1/\sqrt{e})$
    $w \leftarrow \text{NormalizeAbs}((1-\alpha)w + \alpha C)$
  end if
end for
return $\{\phi_\ell / K\}_{\ell \in \mathcal{L}}$