Les Cahiers du GERAD ISSN: 0711–2440

Markovian process explainer: A new explainable artificial intelligence method

C. Zhao, E. Parilina, G. Zaccour

G–2026–46

August 2026

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2026 – Bibliothèque et Archives Canada, 2026

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2026 – Library and Archives Canada, 2026

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Markovian process explainer: A new explainable artificial intelligence method

Chi Zhao [a]

Elena Parilina [a]

Georges Zaccour [b, c]

[a] *Saint Petersburg State University, Saint Petersburg, Russia*

[b] *Chair in Game Theory and Management, GERAD, Montréal (Qc), Canada, H3T 1J4*

[c] *HEC Montréal, Montréal (Qc), Canada, H3T 2A7*

vector.zhaochi@gmail.com
e.parilina@spbu.ru
georges.zaccour@gerad.ca

August 2026
Les Cahiers du GERAD
G–2026–46

Abstract : An efficient and accurate evaluation of feature importance method for artificial intelligent and machine learning models without high computational costs is a key research question of this work. Many existing Shapley-value-based explainable artificial intelligence (XAI) methods, while theoretically sound, require evaluating all 2^n feature subsets, making them impractical for high-dimensional data. Additionally, most approaches typically treat feature importance and feature selection as separate tasks, and they often return a single optimal subset of features. The challenge is to design a model-agnostic framework that can deliver Shapley value-competitive importance rankings while simultaneously performing efficient feature selection and uncovering a range of near-optimal subsets of features. To address this challenge, we propose the Markovian Process Explainer (MPE), a novel XAI method grounded in the Markovian coalition process value. Our methodology models feature selection as a stochastic coalition-formation process, where features are sequentially added to or removed from a subset until an absorbing state-corresponding to an optimal or near-optimal feature subset and its associated model is reached. We empirically evaluate the MPE using forward and backward Key Performance Index (KPI) metrics, comparing its performance with established Shapley-value-based XAI methods. The results demonstrate that the MPE produces feature importance rankings that are consistently competitive with, and frequently superior to, those of existing methods. Moreover, the framework performs feature selection as an intrinsic byproduct of its process, rather than as a separate step. This work holds significant practical and theoretical importance. By reducing computational cost of the exact Shapley value computation, the MPE offers a scalable, model-agnostic solution that can be used in applications, from healthcare to finance, without sacrificing explanatory quality. The set of near-optimal subsets of features obtained as byproduct of the MPE enhances robust decision-making and strengthen trust in complex machine-learning and AI models.

Keywords : Explainable artificial intelligence; feature importance; Markovian process value; Shapley value

Résumé : L'évaluation efficace et précise de l'importance des caractéristiques pour les modèles d'intelligence artificielle et d'apprentissage automatique, sans coût de calcul élevé, constitue un enjeu majeur de ce travail. De nombreuses méthodes d'intelligence artificielle explicable (IAE) existantes, basées sur la valeur de Shapley, bien que théoriquement solides, nécessitent l'évaluation de l'ensemble des 2^n sous-ensembles de caractéristiques, ce qui les rend impraticables pour les données de grande dimension. De plus, la plupart des approches traitent généralement l'importance et la sélection des caractéristiques comme des tâches distinctes, et ne renvoient souvent qu'un seul sous-ensemble optimal. Le défi consiste à concevoir un cadre indépendant du modèle, capable de fournir des classements d'importance compétitifs selon la valeur de Shapley, tout en effectuant une sélection efficace des caractéristiques et en identifiant un ensemble de sous-ensembles quasi optimaux. Pour relever ce défi, nous proposons le Markovian Process Explainer (MPE), une nouvelle méthode d'IAE fondée sur la valeur du processus de coalition markovien. Notre méthodologie modélise la sélection de caractéristiques comme un processus stochastique de formation de coalitions, où des caractéristiques sont ajoutées ou retirées séquentiellement d'un sous-ensemble jusqu'à l'obtention d'un état absorbant correspondant à un sous-ensemble de caractéristiques optimal ou quasi optimal et à son modèle associé. Nous évaluons empiriquement le MPE à l'aide d'indicateurs de performance clés (KPI) ascendants et descendants, en comparant ses performances à celles des méthodes d'IA explicable (XAI) établies, basées sur la valeur de Shapley. Les résultats démontrent que le MPE produit des classements d'importance des caractéristiques systématiquement compétitifs, et souvent supérieurs, à ceux des méthodes existantes. De plus, la sélection de caractéristiques est intégrée au processus, et non effectuée comme une étape distincte. Ce travail revêt une importance pratique et théorique considérable. En réduisant le coût de calcul de la valeur de Shapley exacte, le MPE offre une solution évolutive et indépendante du modèle, applicable dans des domaines aussi variés que la santé et la finance, sans compromettre la qualité explicative. L'ensemble des sous-ensembles de caractéristiques quasi optimaux obtenus comme sous-produit de l'MPE améliore la robustesse de la prise de décision et renforce la confiance dans les modèles complexes d'apprentissage automatique et d'IA.

Mots clés : Intelligence artificielle explicable; importance des caractéristiques; valeur du processus markovien; valeur de Shapley

1 Introduction

Machine learning and statistical models are widely employed for regression and classification tasks. Two important challenges arise in this context: selecting a parsimonious yet predictive set of features for model construction, and quantifying the contribution of individual features to the model's predictions. Although both issues are central to model development and interpretation, they have largely been studied independently. In this paper, we propose a unified approach that addresses them simultaneously. The underlying modeling framework is presented in Section 2.

Selecting the features to be included in a model is the classical feature (or variable) selection problem and it is sometimes solved using optimization and game-theoretic techniques [3, 13]. In statistics, this problem is traditionally addressed through *stepwise* procedures—forward selection, backward elimination, or their bidirectional combination—which iteratively add or remove features according to an information criterion such as the AIC [1] or an adjusted goodness-of-fit measure [5, 7]. These procedures are widely implemented in statistical software; for example, R's `step` and `stepAIC` functions, for instance, provide `forward`, `backward`, and `both` directions procedures [21]. A similar principle underlies *wrapper* methods in machine learning, where the search for a feature subset is guided by the predictive performance of the trained model [10, 14]. Despite extensive research in this area and recent methodological developments [4, 20], these approaches ultimately return only a selected subset of features and an order of inclusion or exclusion. Being inherently greedy, they do not provide a principled assessment of the individual contribution of each feature to the final model.

Explaining a trained model is the goal of *Explainable Artificial Intelligence* (XAI) which is also an important task for applications [6]. The dominant tool used in XAI is the Shapley value, which fairly distributes a model's performance among its features [16, 17, 19, 22–24]. As its exact computation requires evaluating all 2^n feature subsets, it is commonly approximated by sampling methods. In all cases, however, these methods presuppose a fixed model, and hence a fixed feature set: they explain a given model but do not perform feature selection.

These two strands of literature are largely disjoint: selection yields a model but no fair importance, whereas XAI yields importance for a model taken as given; neither explains the model-construction process itself. We bridge this gap with the Markovian process explainer, a model-agnostic method based on the Markovian coalition formation process [8, 9]: features are added or removed one at a time until an absorbing state is reached, which identifies the "best" subset and model, while the associated process value ranks feature importance. Our main contributions lie in providing a method that:

- Selects and explains features in a single pass, with no preset subset size, avoiding the 2^n subset evaluations of the exact Shapley value.
- Is flexible in terms of starting point, which can be the empty, full, or a correlation-based feature set. The method proceeds greedily or by using a stochastic rule (Linear, Top-k, Top-p, Min-p). In all cases, it ranks features by their marginal contribution to final model performance.
- Yields robust importance estimates and a set of near-best models rather than a single one. This is obtained by implementing stochastic variants that average the results over many sampled scenarios.
- Performs very well empirically. Indeed, we tested the method on four datasets, with two of them being high-dimensional, and in all cases it achieved competitive or superior importance rankings under forward and backward Key Performance Index (KPI) metrics, with built-in feature selection.

The paper is organized as follows. Section 2 motivates the problem. Section 3 recalls the use of the Shapley value in XAI. Section 4 presents the deterministic explainer, the comparison methodology, and the numerical results. Section 5 develops its stochastic variants. Conclusion and future work are discussed in Section 6.

2 Motivation

In modern data analysis, regression or classification tasks are frequently addressed using a wide variety of models, ranging from interpretable parametric forms to highly complex black-box algorithms such as random forests, gradient boosting machines, support vector machines, and deep neural networks. Regardless of the specific model class, the general setup involves a dataset consisting of k input vectors $x_i = (x_{i1}, \ldots, x_{in})$ and corresponding target values y_i, $i = 1, \ldots, k$. A regression or classification model is then constructed as

$$\hat{y}(x) = f(x),$$

where f is an arbitrary function learned from the data, not necessarily linear or even parametric. In many practical applications, f is a black-box model that provides high predictive accuracy but offers little transparency regarding the internal relationships between inputs and outputs.

Once a model has been trained, researchers are typically interested not only in its predictive accuracy but also in understanding which input features $x_1, \ldots, x_n$ drive its predictions. For linear models, feature importance can often be inferred directly from estimated coefficients or through variance-decomposition techniques [2, 15]. However, when dealing with black-box models, the functional form of f is either unknown or too complex to interpret directly. Assessing the contribution of individual features therefore becomes a challenging task.

This observation raises two fundamental questions in practical statistical modeling:

1. How should an appropriate model be selected for a given dataset? More specifically, what constitutes the best, or nearly best, model among the possible alternatives? If not all available features are necessary, how can one identify the subset of variables that provides the most effective representation of the data? This issue becomes particularly important in high-dimensional settings, where an excessive number of features may increase computational complexity and the risk of overfitting.
2. Once a model, especially a black-box model, has been selected, how can the importance of individual features be reliably quantified? Unlike linear models, black-box regression and classification models do not provide explicit coefficients or direct variance decompositions that reveal the contribution of each input variable. Consequently, traditional feature-importance measures are no longer directly applicable and require alternative, model-agnostic approaches.

Recent developments in XAI have led to the emergence of various approaches based on the Shapley value [16, 19, 22–24] for quantifying feature importance in black-box models. These methods are model-agnostic, meaning that they can be applied to a wide range of models regardless of whether their functional forms are known, and are capable of handling highly complex nonlinear relationships. However, before applying an XAI method to assess feature importance, a fundamental prior question must be addressed: which set of features should be included in the model in the first place? Existing XAI approaches generally assume that the model and its feature set have already been determined, and therefore do not address the model-construction process itself.

The method introduced in this paper aims to address both challenges simultaneously. It provides a unified framework for identifying the best, or a collection of near-best, models for regression and classification tasks while, at the same time, quantifying the contribution of individual features within the selected models. Since our approach is model-agnostic, it can be applied across a broad spectrum of model classes, ranging from simple interpretable models to complex black-box algorithms.

Because our framework is grounded in cooperative game theory, the next section begins with a brief review of game-theoretic solution concepts, with particular emphasis on the Shapley value and its role in XAI for evaluating feature contributions in complex predictive models.

3 Shapley value in XAI

The Shapley value is widely used to evaluate feature importance, a central task in XAI. The underlying idea of Shapley-based methods is to provide a fair allocation, in the game-theoretic sense, of the model's overall performance among its features.

To apply cooperative game-theoretic solution concepts, one must first define a cooperative game in which the features are regarded as the players. In the classical framework, such a game is characterized by a characteristic function that assigns a value to every coalition of players, which, in our setting, corresponds to every subset of features $\mathcal{S}$. The standard approach defines the value of a coalition as the performance of a model built using only the features in $\mathcal{S}$. Model performance may be measured in various ways, depending on the learning task, e.g., by the coefficient of determination R^2, the Akaike Information Criterion (AIC), or the F-statistic in regression problems, and by accuracy, precision, or recall in classification problems. Thus, the characteristic function $v(\mathcal{S})$ is defined for every coalition $\mathcal{S} \subset \mathcal{N}$, where $\mathcal{N}$ is the set of all features.

Once the characteristic function has been specified, any cooperative game-theoretic solution concept can be used to allocate the value $v(\mathcal{N})$, representing the performance of the model built using the complete set of features. In XAI, the most widely adopted solution concept is the Shapley value, which quantifies the contribution of each feature to the model's overall predictive performance.

Computing the Shapley value requires evaluating the performance of a model for every possible subset of features, since the Shapley value formula depends on all characteristic function values. Consequently, the exact computation of the Shapley value has exponential complexity $O(2^n)$. This motivates the search for methods that reduce the computational burden by avoiding the evaluation of all feature subsets and the repeated retraining of the model, which is particularly costly for complex machine learning models.

In this paper, we propose a dynamic procedure for adding and removing features to construct either a single best model or a collection of near-optimal models, starting from either the empty set or a non-empty subset of features. The procedure is modeled as a specially designed Markovian process, building on the Markovian approach to coalition formation introduced in [8, 9]. As an outcome of this process, we obtain both a high-performing statistical or machine learning model and a corresponding vector of feature importance scores. Our framework captures the dynamic nature of model construction by describing how features are iteratively added to or removed from the model according to their marginal contributions to its predictive performance.

We introduce two Markovian process-based explainers. The first, presented in Section 4, relies on a single trajectory generated by a Markovian process. The underlying feature selection procedure resembles the stepwise algorithms commonly implemented in statistical software for linear and logistic regression models. Starting from the empty set of features, the procedure iteratively adds or removes a single feature from the current model until no further improvement in model performance can be achieved under the prescribed rules. We model this process as a Markovian process and compute the corresponding feature importance vector using the Markovian process value.

The second explainer, presented in Section 5, is more general and provides richer information by generating a graph that represents a stochastic process of feature addition and deletion. The process may start from the empty set, an intermediate subset, or the full set of features, and evolves until it reaches an absorbing state. Each absorbing state corresponds to a statistical or machine learning model whose performance is optimal or near-optimal and cannot be further improved under the transition rules of the process.

4 Markovian process explainer

4.1 Method description

As described in Section 3, we need to define a characteristic function that depends on the task. From now on, we use

$$v(\mathcal{S}) = \bar{R}^2(\mathcal{S}), \tag{1}$$

for regression tasks, where $\bar{R}^2(\mathcal{S})$ is the adjusted coefficient of determination of the model trained on the feature subset $\mathcal{S}$, and

$$v(\mathcal{S}) = \text{Accuracy}(\mathcal{S}), \tag{2}$$

for classification task, where $\text{Accuracy}(\mathcal{S})$ is the accuracy of the model trained on the feature subset $\mathcal{S}$.

The Markovian process explainer (MPE) starts from an initial subset of features $\mathcal{S}_0 \subseteq \mathcal{N}$. We consider three options:

1. $\mathcal{S}_0 = \varnothing$. We start from the empty set of features, which is similar to a forward step-wise procedure. The Markovian explainer is denoted by $\text{MPE}_\emptyset$.
2. $\mathcal{S}_0 \subset \mathcal{N}$, $\mathcal{S}_0 \neq \varnothing$, $\mathcal{S}_0 \neq \mathcal{N}$. We start from a nonempty and non-full set of features. The initial set $\mathcal{S}_0$ can be formed by a correlation-based threshold.[1] The Markovian explainer is denoted by MPE_ρ.
3. $\mathcal{S}_0 = \mathcal{N}$. We start from the full set of features, which is similar to a backward step-wise procedure. The Markovian explainer is denoted by $\text{MPE}_\mathcal{N}$.

The idea behind using a non-empty initial subset of features (option 2) is to reduce the computational burden and reach faster an absorbing state. In the numerical experiments, we test if this idea is indeed attractive.

Given an initial subset $\mathcal{S}_0$, the Markovian process explainer runs a greedy coalition formation process inspired from Faigle and Grabisch [8]. At each step, the algorithm evaluates all possible single-feature additions to the current set and removals from this set, and greedily selects the action with the largest improvement of the characteristic function value, i.e., at stage t with the current set of features $\mathcal{S}_t$ and model performance $v(\mathcal{S}_t)$ the feature $k \notin \mathcal{S}_t$ is added to set $\mathcal{S}_t$ or the feature $k \in \mathcal{S}_t$ is deleted from set $\mathcal{S}_t$ if

$$k \in \arg\max \left\{ 0, \max_{\ell \in \mathcal{N} \setminus \mathcal{S}_t} \Big(v(\mathcal{S}_t \cup \{\ell\}) - v(\mathcal{S}_t) \Big), \max_{\ell \in \mathcal{S}_t} \Big(v(\mathcal{S}_t \setminus \{\ell\}) - v(\mathcal{S}_t) \Big) \right\}.$$

If the maximum of the right-hand side is equal to zero, then there are no improvements of the model performance and the process stops providing the "best" model. Once the process stops we calculate the importance of features using the Shapley I scenario-value proposed in [8]. Suppose that the process ends with the scenario consisting of the sequence of coalitions $\underline{S} = (\mathcal{S}_0, \mathcal{S}_1, \ldots, \mathcal{S}_t)$. The Shapley I scenario-value is defined by:

$$\phi_i^{\underline{S}}(v) = \sum_{k \mid i \in \mathcal{S}_k \Delta \mathcal{S}_{k+1}} \frac{v(\mathcal{S}_{k+1}) - v(\mathcal{S}_k)}{|\mathcal{S}_{k+1} \Delta \mathcal{S}_k|}, \tag{3}$$

where $\mathcal{S}_{k+1}\Delta\mathcal{S}_k = (\mathcal{S}_{k+1} \cup \mathcal{S}_k) \setminus (\mathcal{S}_{k+1} \cap \mathcal{S}_k)$ is the symmetric difference of $\mathcal{S}_{k+1}$ and $\mathcal{S}_k$. The value $\phi_i^{\underline{S}}(v)$ has a natural interpretation in terms of marginal contribution to coalitions (when feature i enters or leaves a coalition). The scenario-value generalizes the classical Shapley value and coincides with it when $|\mathcal{S}_{k+1}\Delta\mathcal{S}_k| = 1$ holds for all $k = 0, \ldots, t-1$ (for comments see [8]). As at each step of the process there is only one feature that is added or deleted, the term $|\mathcal{S}_{k+1}\Delta\mathcal{S}_k| = 1$, so it can be omitted in formula (3).

[1] We add the features to subset $\mathcal{S}$ iff their correlation with the target variable is larger than a threshold, which is chosen empirically for each dataset. For classification tasks we can use mutual information coefficient.

The values given by formula (3) represent feature importance for the final and "best" model with subset of features $\mathcal{S}_t$. We should note that if a feature is not in the final model, then its importance is not defined. The complete procedure is summarized in Algorithm 1. The procedure provides both a feature importance ranking and an implicit feature selection with n_{stop} features defining the "best" model.

Algorithm 1 **Markovian Process Explainer**

Require: Feature set $\mathcal{N} = \{1, \ldots, n\}$, initial subset $\mathcal{S}_0 \subseteq \mathcal{N}$, characteristic function v
Ensure: Feature importance ranking $(\pi_1, \ldots, \pi_{n_{\text{stop}}})$

$\phi_i \leftarrow 0$ for all $i \in \mathcal{N}$
$\mathcal{S} \leftarrow \mathcal{S}_0$
repeat
  for each $i \in \mathcal{N} \setminus \mathcal{S}$ **do** ▷ Evaluate join candidates
    $\delta_i^+ \leftarrow v(\mathcal{S} \cup \{i\}) - v(\mathcal{S})$
  end for
  for each $j \in \mathcal{S}$ **do** ▷ Evaluate leave candidates
    $\delta_j^- \leftarrow v(\mathcal{S} \setminus \{j\}) - v(\mathcal{S})$
  end for
  if $\max_i \delta_i^+ \leq 0$ **and** $\max_j \delta_j^- \leq 0$ **then**
    break ▷ Absorbing state reached
  end if
  if $\max_i \delta_i^+ \geq \max_j \delta_j^-$ **and** $\max_i \delta_i^+ > 0$ **then**
    $i^* \leftarrow \arg\max_i \delta_i^+$
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{i^*\}$ ▷ Join
    $\phi_{i^*} \leftarrow \phi_{i^*} + \delta_{i^*}^+$
  else if $\max_j \delta_j^- > 0$ **then**
    $j^* \leftarrow \arg\max_j \delta_j^-$
    $\mathcal{S} \leftarrow \mathcal{S} \setminus \{j^*\}$ ▷ Leave
    $\phi_{j^*} \leftarrow \phi_{j^*} + \delta_{j^*}^-$
  end if
until no improvement
$\mathcal{S}^* \leftarrow \mathcal{S}, \quad n_{\text{stop}} \leftarrow |\mathcal{S}^*|$
return features in $\mathcal{S}^*$ sorted by decreasing ϕ_i: $\pi_1, \pi_2, \ldots, \pi_{n_{\text{stop}}}$

We compare the MPE with other XAI methods in Section 4.3 following the methodology described in Section 4.2.

4.2 Methodology to compare XAI methods

Three main characteristics/metrics are retained to compare our method with other XAI methods described in Section 4.1.

1. The ratio of the *peak performance* corresponding to the MPE when reaching an absorbing state, which represents the "best model", to the performance of the full model with all features.
2. *Forward or backward key performance index* (KPI):

   Forward KPI. Starting from the empty set, features are added one by one according to the ranking provided by the explainer. Let $\mathcal{S}_k^{\text{fwd}} = \{\pi_1, \ldots, \pi_k\}$ denote the feature set at step k, with $\mathcal{S}_0^{\text{fwd}} = \varnothing$. The marginal gain at step k is $\delta_k^{\text{fwd}} = v(\mathcal{S}_k^{\text{fwd}}) - v(\mathcal{S}_{k-1}^{\text{fwd}})$, that is, the increase in the model's performance after adding a feature at step k. The forward KPI slope is defined by

$$S^{\text{fwd}} = \sum_{k=1}^{K} \beta^{k-1} \, \delta_k^{\text{fwd}}, \tag{4}$$

   where $K \leq n$ is the evaluation horizon and $\beta \in (0, 1)$ is a decay parameter. (We use $\beta = 0.8$ in our experiments.) A higher S^{fwd} indicates that the most predictive features are ranked first, leading to rapid performance recovery.

Backward KPI. Starting from the full set $\mathcal{N}$, features are removed one by one in the same ranking order (most important first). Let $\mathcal{S}_k^{\text{bwd}} = \mathcal{N} \setminus \{\pi_1, \ldots, \pi_k\}$ denote the feature set at step k, with $\mathcal{S}_0^{\text{bwd}} = \mathcal{N}$. The marginal loss at step k is $\delta_k^{\text{bwd}} = v(\mathcal{S}_{k-1}^{\text{bwd}}) - v(\mathcal{S}_k^{\text{bwd}})$. The backward KPI slope is

$$S^{\text{bwd}} = \sum_{k=1}^{K} \beta^{k-1} \, \delta_k^{\text{bwd}}. \tag{5}$$

A higher S^{bwd} indicates that removing the top-ranked features causes steep performance degradation, confirming that the method correctly identifies the most impactful features.

The exponential weighting with $\beta < 1$ emphasises early steps: the first few features added (forward) or removed (backward) carry the most weight, reflecting the intuition that correctly ranking the very top features matters more than the ordering of marginal ones.

3. *Computation time.*

Remark 1. The calculation of both the forward and backward KPIs depends on the feature-importance ranking produced by a given XAI method. In the forward approach, features are added sequentially according to this ranking, whereas in the backward approach they are removed in the same order. This ranking can be compared with the feature ordering obtained from the stepwise procedures implemented in most statistical software packages. Such a procedure is commonly referred to as the Greedy Best (or optimal greedy ordering) method. Starting from the empty set, the Greedy Best algorithm adds one feature at a time, selecting at each step the feature that yields the largest local improvement in the model's performance. The resulting sequence of selected features naturally defines their importance ranking. For example, if feature 1 is selected at the first step, it is regarded as the most important feature for model construction. Unlike the Markovian process explainer defined by equation (3), however, the Greedy Best method does not evaluate feature importance through the features' marginal contributions to the model.

4.3 Comparison of the MPE with other XAI methods

In the experiments,[2] we use four datasets: two for regression task (*Housing* to predict the median value of owner-occupied homes with 13 features, and *Crime* to predict crime level with 120 features) and two for classification task (*H1N1* to classify the decision on vaccination with 35 features, and *MIC* to classify the patient's lethal outcome (survival vs. death) after myocardial infarction with 111 features). We highlight that two of the four datasets are with a large number of features which complicates application of many existing XAI methods based on the Shapley value.

In all experiments the underlying machine learning model is the gradient boosting decision tree LightGBM [12] (`LGBMRegressor` for regression and `LGBMClassifier` for classification, 100 trees), trained on an 80/20 train–test split.

We compare the proposed MPE in three versions depending on the initial set of features with the following XAI methods:

- **ShapG** [22, 23]: a graph-based approximation of the Shapley value that models feature dependencies by a graph to reduce the number of coalition evaluations. We use its two improved variants, which differ in the similarity measure used to build the feature graph: ShapG$_{\text{cos}}$ (cosine similarity) and ShapG$_{MI}$ (mutual information); both are reported for all datasets. In each variant the complete graph is sparsified by removing edges around the most target-relevant features first.
- **KernelSHAP** [16]: estimates the Shapley value by solving a weighted least squares problem over sampled feature coalitions.

[2] All experiments are performed on a Mac Mini (Apple M4, 16 GB RAM).

- **SamplingSHAP** [19]: approximates the Shapley value by Monte-Carlo sampling of feature permutations and averaging marginal contributions.
- **LeverageSHAP** [17]: a recent estimator based on leverage-score sampling, attaining $O(n \log n)$ evaluations with provable accuracy guarantees.

Tables 1 and 2 report all evaluation metrics for the four datasets: (i) forward KPI S^{fwd} given by formula (4), (ii) *Peak* model performance during feature selection procedure, (iii) *Step* at which this peak is reached, (iv) % *full* defining the share of the peak performance relative to the performance of the full model, (v) backward KPI slope S^{bwd} given by formula (5), (vi) n_{stop} defining the number of features in the final model (MPE only), and (vii) *Time* giving the computation time in seconds. Table 1 covers Housing and H1N1 datasets with small number of features while Table 2 shows the results for the two high-dimensional datasets (Crime and MIC). We note that for the MPE_ρ we use correlation coefficient $|\rho| \geq 0.6$ for Housing and Crime (whose features are continuous), and mutual information $\text{MI} \geq 0.03$ for H1N1 and MIC (whose features are mostly categorical).

Table 1: Consolidated results for Housing and H1N1 datasets. Best (except greedy best method) value per metric is in bold.

	Housing (n=13, $\bar{R}^2_{\text{full}}$=0.869)							**H1N1** (n=35, acc_{full}=0.856)						
Method	S^{fwd}	Peak	Step	% full	S^{bwd}	n_{stop}	Time (s)	S^{fwd}	Peak	Step	% full	S^{bwd}	n_{stop}	Time (s)
Greedy Best	0.794	0.885	9	101.7	—	—	1.2	0.835	0.857	20	100.1	—	—	27.5
KernelSHAP	0.784	0.879	10	101.1	0.274	—	12.8	0.834	0.855	10	**99.9**	0.046	—	13.0
SamplingSHAP	0.784	0.879	10	101.1	0.274	—	13.6	0.834	0.855	10	**99.9**	**0.046**	—	36.5
LeverageSHAP	0.797	0.877	10	100.8	0.335	—	36.7	0.803	0.852	19	99.5	0.012	—	343.3
$\text{ShapG}_{\text{cos}}$	0.769	0.877	10	100.8	0.336	—	2.2	0.824	0.841	20	98.2	0.033	—	23.7
ShapG_{MI}	0.795	0.867	10	99.7	**0.348**	—	2.0	0.822	0.851	20	99.4	0.029	—	19.6
$\text{MPE}_\emptyset$	0.791	0.880	6	101.2	0.284	6	1.1	**0.835**	**0.855**	9	**99.9**	0.044	9	15.8
MPE_ρ	**0.807**	**0.884**	6	**101.7**	0.193	6	**0.9**	0.832	0.855	9	99.8	0.045	9	13.6
$\text{MPE}_{\mathcal{N}}$	0.750	0.881	8	101.3	0.264	9	1.4	0.827	0.853	20	99.6	0.044	35	**2.9**

Table 2: Consolidated results for Crime and MIC datasets. Best (except greedy best method) value per metric is in bold.

	Crime (n=120, $\bar{R}^2_{\text{full}}$=0.943)							**MIC** (n=111, acc_{full}=0.953)						
Method	S^{fwd}	Peak	Step	% full	S^{bwd}	n_{stop}	Time (s)	S^{fwd}	Peak	Step	% full	S^{bwd}	n_{stop}	Time (s)
Greedy Best	0.932	0.986	12	104.5	—	—	79.2	0.951	0.968	8	101.5	—	—	55.8
KernelSHAP	0.917	0.979	3	103.8	0.210	—	60.6	0.938	0.959	19	100.6	0.030	—	49.1
SamplingSHAP	0.917	0.979	3	103.8	0.211	—	152.8	0.939	0.956	3	100.3	0.030	—	132.5
LeverageSHAP	0.903	0.976	9	103.5	**0.338**	—	618.0	0.867	0.921	16	96.6	-0.000	—	282.6
$\text{ShapG}_{\text{cos}}$	0.868	0.968	10	102.6	0.176	—	98.4	0.923	0.944	11	99.1	0.024	—	19.0
ShapG_{MI}	0.861	0.974	16	103.2	0.202	—	99.6	0.921	0.944	11	99.1	0.022	—	17.7
$\text{MPE}_\emptyset$	**0.931**	**0.986**	9	**104.5**	0.083	9	43.1	**0.946**	**0.965**	6	**101.2**	0.025	46	161.7
MPE_ρ	0.924	0.979	3	103.8	0.089	9	**23.5**	0.911	0.953	19	100.0	0.027	45	122.9
$\text{MPE}_{\mathcal{N}}$	0.925	0.981	9	104.0	0.268	111	140.5	0.893	0.947	16	99.4	**0.030**	110	**10.2**

Following our plan of comparison for all datasets we describe the results of comparison by three characteristics/metrics:

1. **Peak performance.** For *Housing* dataset, MPE_ρ reaches peak $\bar{R}^2 = 0.884$ with only 6 features, nearly matching the Greedy Best method ($\bar{R}^2 = 0.885$) and *exceeding* the full 13-feature model ($\bar{R}^2 = 0.869$). $\text{MPE}_\emptyset$ and $\text{MPE}_{\mathcal{N}}$ reach similar peaks ($\bar{R}^2 = 0.880$ at step 6 and $\bar{R}^2 = 0.881$ at step 8), demonstrating that the Markovian stopping criterion correctly identifies when further features become harmful. In contrast, the SHAP baselines KernelSHAP and SamplingSHAP (0.879), LeverageSHAP (0.877) and both ShapG variants (0.867–0.877, reached only at 10 features) all peak below the three Markovian variants.

   For *H1N1*, both $\text{MPE}_\emptyset$ and MPE_ρ reach accuracy 0.855 ($\approx 99.9\%$ of the full model performance) with only 9 out of 35 features. KernelSHAP and SamplingSHAP reach the same peak accuracy as the Markovian explainer (0.855), while LeverageSHAP (0.852) and both ShapG variants (0.841–0.851) fall slightly below.

For *Crime*, $\text{MPE}_\emptyset$ reaches peak $\bar{R}^2 = 0.986$ with only 9 out of 120 features, matching the Greedy Best oracle (104.5% of the full model), above all baselines (KernelSHAP and SamplingSHAP 0.979, LeverageSHAP 0.976, ShapG 0.968–0.974).

For *MIC*, $\text{MPE}_\emptyset$ reaches accuracy 0.965 with only 6 out of 111 features (101.2% of the full model), exceeding full-model performance and the best baselines (KernelSHAP 0.959, SamplingSHAP 0.956, ShapG 0.944, LeverageSHAP 0.921).

2. Forward and backward KPIs. 2.1. Forward KPI. For *Housing*, MPE_ρ achieves the highest forward KPI value ($S^{\text{fwd}} = 0.807$) with only 6 features, exceeding even the Greedy Best oracle ($S^{\text{fwd}} = 0.794$) using 9 features, and above all baselines (LeverageSHAP 0.797, ShapG_{MI} 0.795, KernelSHAP and SamplingSHAP 0.784). For *H1N1*, $\text{MPE}_\emptyset$ matches the Greedy Best oracle with the same value of $S^{\text{fwd}} = 0.835$, a bit better than KernelSHAP and SamplingSHAP (0.834).

For *Crime*, $\text{MPE}_\emptyset$ nearly matches the oracle ($S^{\text{fwd}} = 0.931$ vs. 0.932), clearly above the best baseline (KernelSHAP and SamplingSHAP, 0.917). For *MIC*, $\text{MPE}_\emptyset$ achieves the best forward KPI ($S^{\text{fwd}} = 0.946$), close to the oracle (0.951) and above SamplingSHAP (0.939) and KernelSHAP (0.938).

2.2. Backward KPI. For *Housing*, ShapG_{MI} achieves the highest backward KPI ($S^{\text{bwd}} = 0.348$), followed closely by $\text{ShapG}_{\cos}$ (0.336) and LeverageSHAP (0.335); the best Markovian variant is $\text{MPE}_\emptyset$ (0.284). For *H1N1*, KernelSHAP and SamplingSHAP achieve the highest backward slopes ($S^{\text{bwd}} = 0.046$), with all Markovian variants close behind (0.044–0.045) and ahead of both ShapG variants (0.029–0.033). For *Crime*, LeverageSHAP achieves the highest backward slope ($S^{\text{bwd}} = 0.338$), followed by $\text{MPE}_\mathcal{N}$ (0.268). For *MIC*, $\text{MPE}_\mathcal{N}$, KernelSHAP and SamplingSHAP tie for the highest backward slope ($S^{\text{bwd}} = 0.030$).

We represent the process of adding features to the empty set (forward procedure) in Figure 2 for Housing and H1N1, and in Figure 4 for Crime and MIC. The features are added one by one according to their importance provided by different XAI methods. The vertical red lines with stars mark the absorbing state of the MPE. The corresponding backward KPI curves are presented in Figures 3 and 5.

3. Computation time. Among all compared XAI methods, a variant of the MPE is the fastest on every dataset. Among the Markovian variants, the fastest depends on the dataset: MPE_ρ is fastest on Housing (0.9s) and Crime (23.5s), while $\text{MPE}_\mathcal{N}$ is fastest on H1N1 (2.9s) and MIC (10.2s).

We also summarize the best methods according our metrics in Table 3.

Table 3: Summary of best-performing methods across all metrics

Dataset	% full	Forward KPI	Backward KPI	Time
Housing	MPE_ρ	MPE_ρ	ShapG_{MI}	MPE_ρ
H1N1	$\text{MPE}_\emptyset$, KernelSHAP, SamplingSHAP	$\text{MPE}_\emptyset$	KernelSHAP	$\text{MPE}_\mathcal{N}$
Crime	$\text{MPE}_\emptyset$	$\text{MPE}_\emptyset$	LeverageSHAP	MPE_ρ
MIC	$\text{MPE}_\emptyset$	$\text{MPE}_\emptyset$	$\text{MPE}_\mathcal{N}$, KernelSHAP	$\text{MPE}_\mathcal{N}$

4.4 Summary on MPE

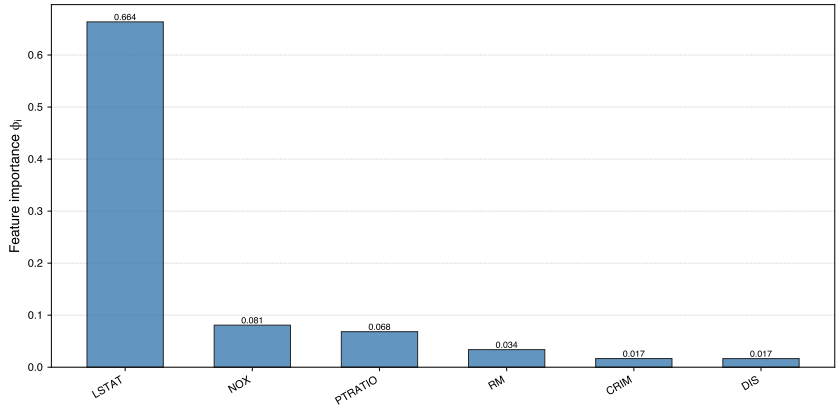
We propose a new XAI method to measure feature importance that provides at the same time the subset of features to build the best model. We demonstrate the results on four datasets (regression and classification tasks, from small to high-dimensional).

The MPE demonstrates three key advantages:

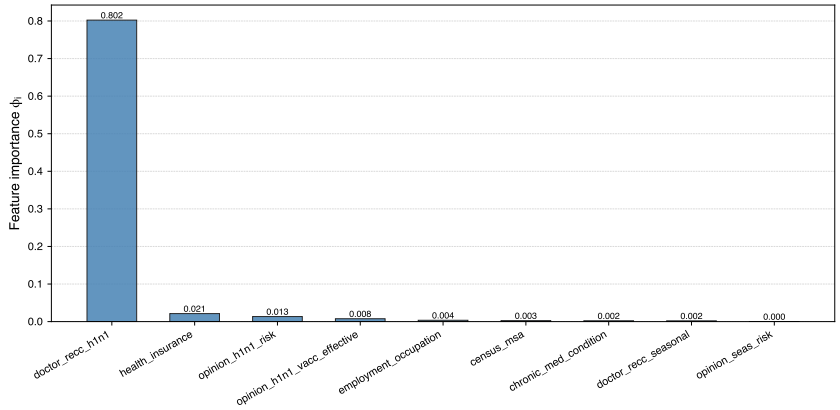
1. *Superior forward KPI*: On all four datasets, a variant of the MPE achieves the best or oracle-matching forward KPI value, consistently outperforming all comparison methods (the two ShapG variants, the SHAP-library baselines KernelSHAP and SamplingSHAP, and LeverageSHAP).

2. *Built-in feature selection*: As a byproduct, and without any preset target size, the absorbing state of the process yields a feature subset whose size n_{stop} is 6/13 for Housing, 9/35 for H1N1, 9/120 for Crime and 46/111 for MIC (Tables 1 and 2). These subsets are compact on three of the four datasets, and the corresponding model reaches or exceeds the full-model performance ($v(\mathcal{S}^*)$ from 99.9% to 104.5% of the full model).
3. *Efficiency*: On the high-dimensional Crime dataset, $\text{MPE}_\emptyset$ dominates the comparison in both quality and time: it reaches nearly oracle-level forward KPI (0.931 vs. 0.932) in 43 s, whereas the strongest baseline, KernelSHAP, reaches only 0.917 in 61 s and the ShapG variants stay below 0.87. On MIC, there is a trade-off between quality and speed: $\text{MPE}_\emptyset$ attains the best forward KPI (0.946, above SamplingSHAP's 0.939 and KernelSHAP's 0.938) but at a higher computational cost (162 s), so on this dataset its advantage lies in quality rather than in speed.

We discuss only the feature importance ranking provided by the Markovian process value but not the values of $\phi_i^S(v)$ given by formula (3). These values for Housing and H1N1 datasets are represented in Figure 1. As we see from this figure, the feature importance of the Top-1 feature is much higher than other importance values. This can be explained by the large marginal contribution of this feature to the model performance when we add it to the empty set, which has a zero performance. This unfair distribution of the performance is caused by considering the unique (non-stochastic) scenario in the feature selection process. In the following section, we extend the model by introducing a stochastic feature selection process, for which the MPE evaluates more carefully the importance of the features used in the final model.

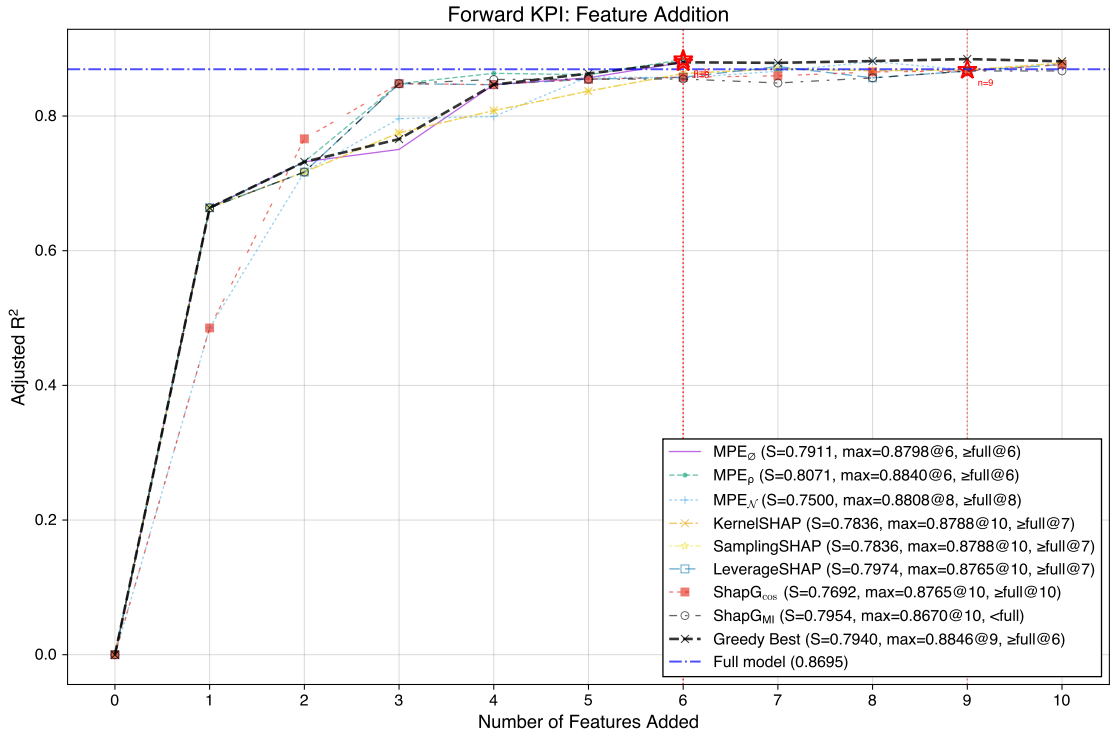


(a) Housing (6 features of the final model)

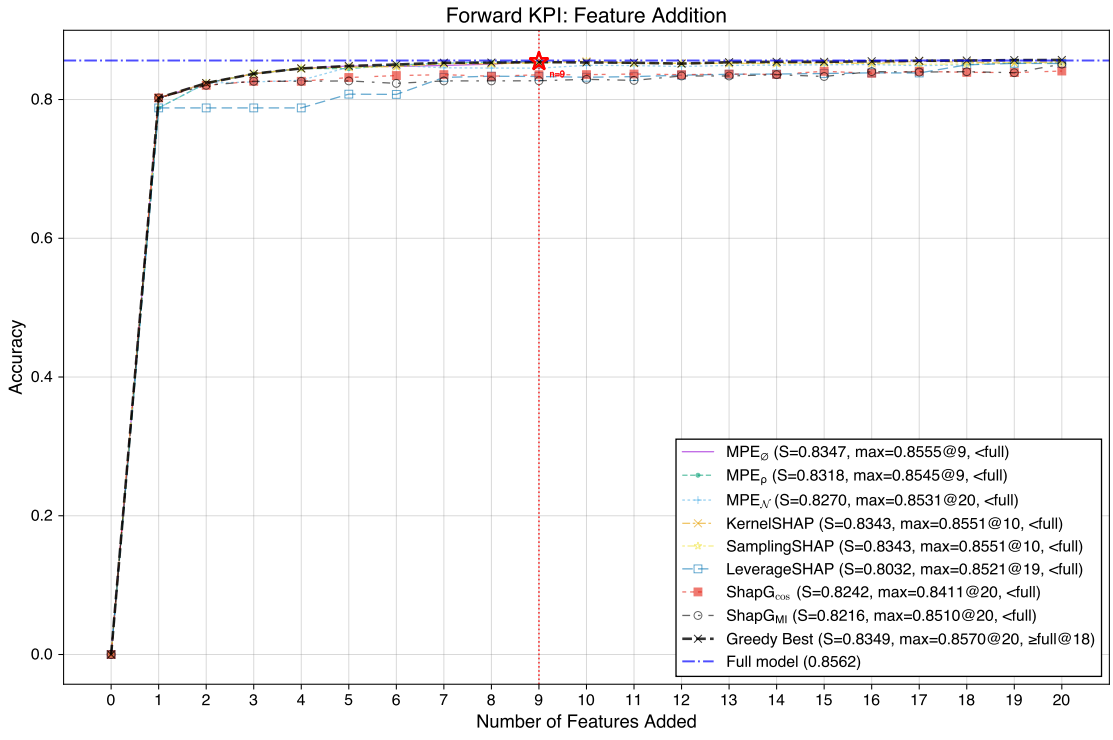


(b) H1N1 (9 features of the final model)

Figure 1: Feature importance values $\phi_i^S(v)$ given by (3) for the final model obtained by $\text{MPE}_\emptyset$, in decreasing order.

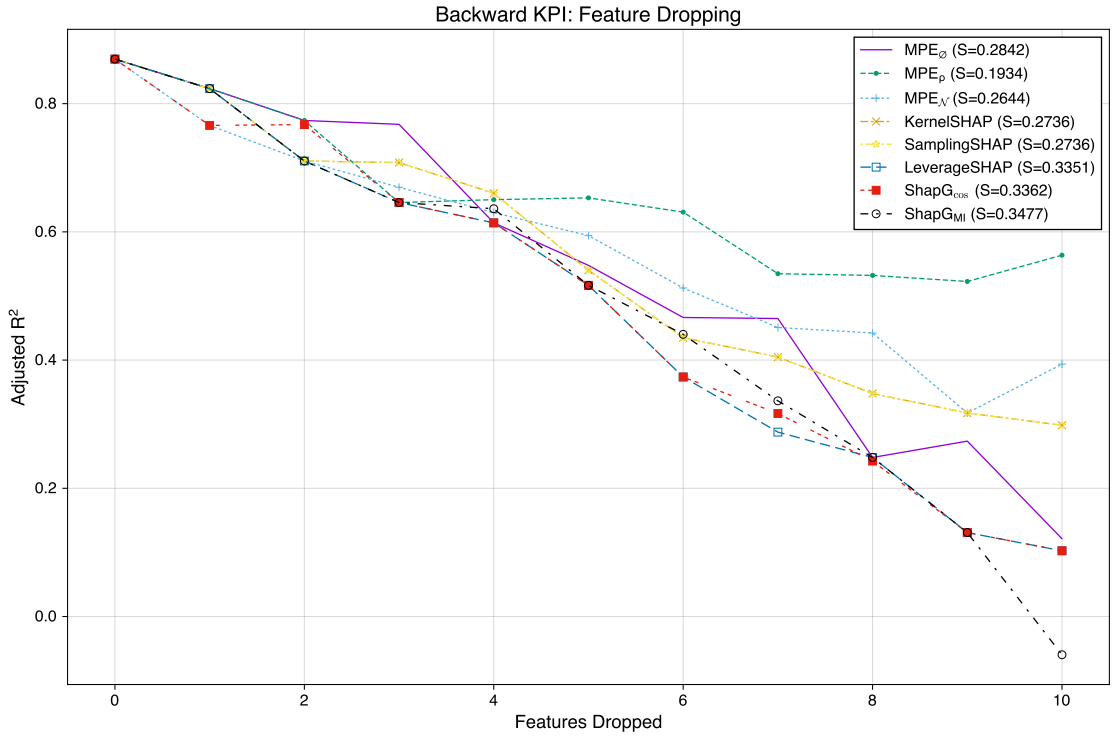


(a) Housing (n=13)

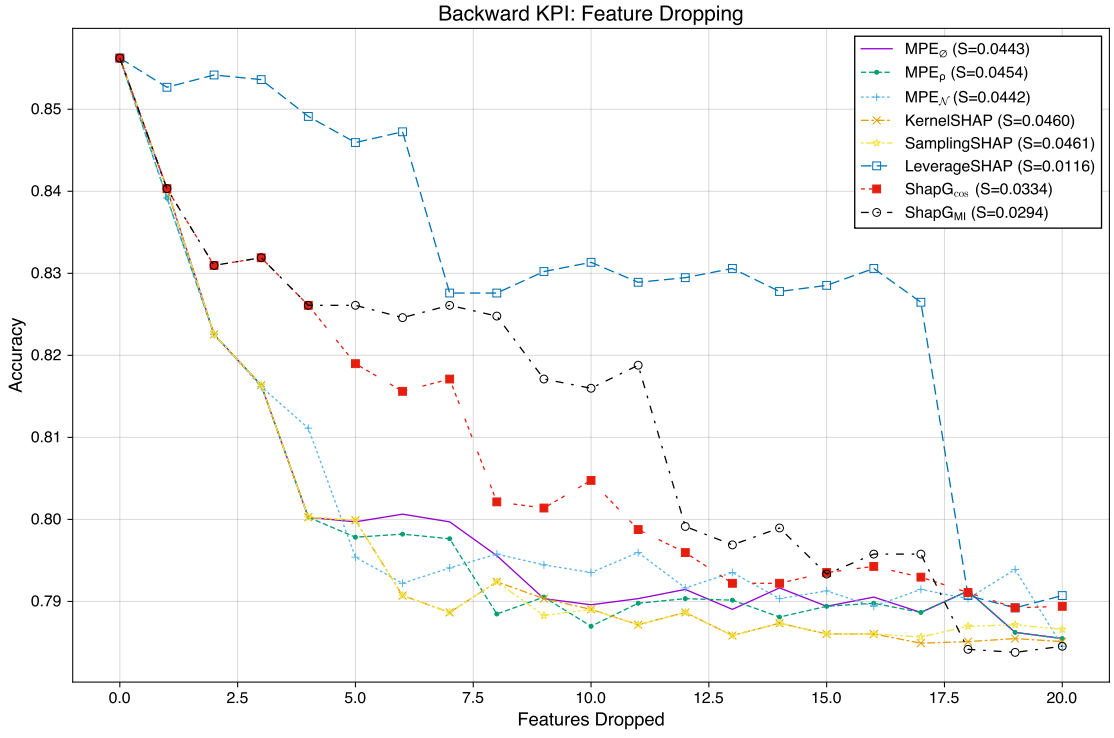


(b) H1N1 (n=35)

Figure 2: Forward KPI curves (retrain mode). Features are added in order of their importance given by different XAI methods. Steeper curves indicate better feature rankings.

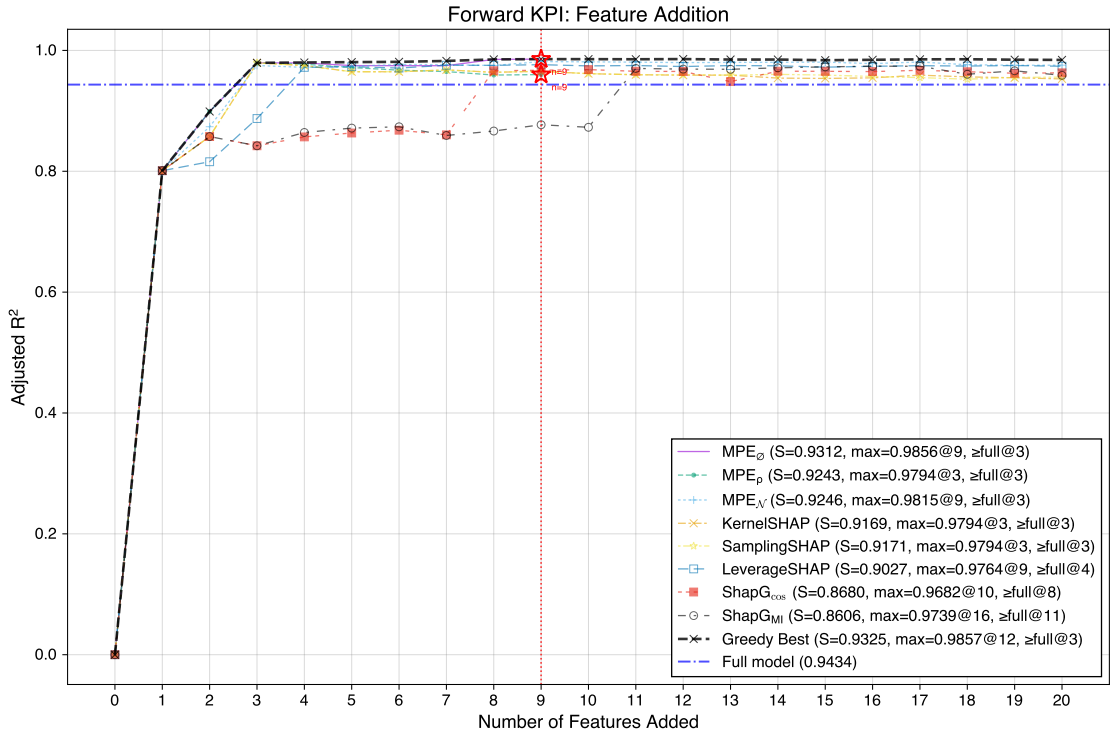


(a) Housing (n=13)

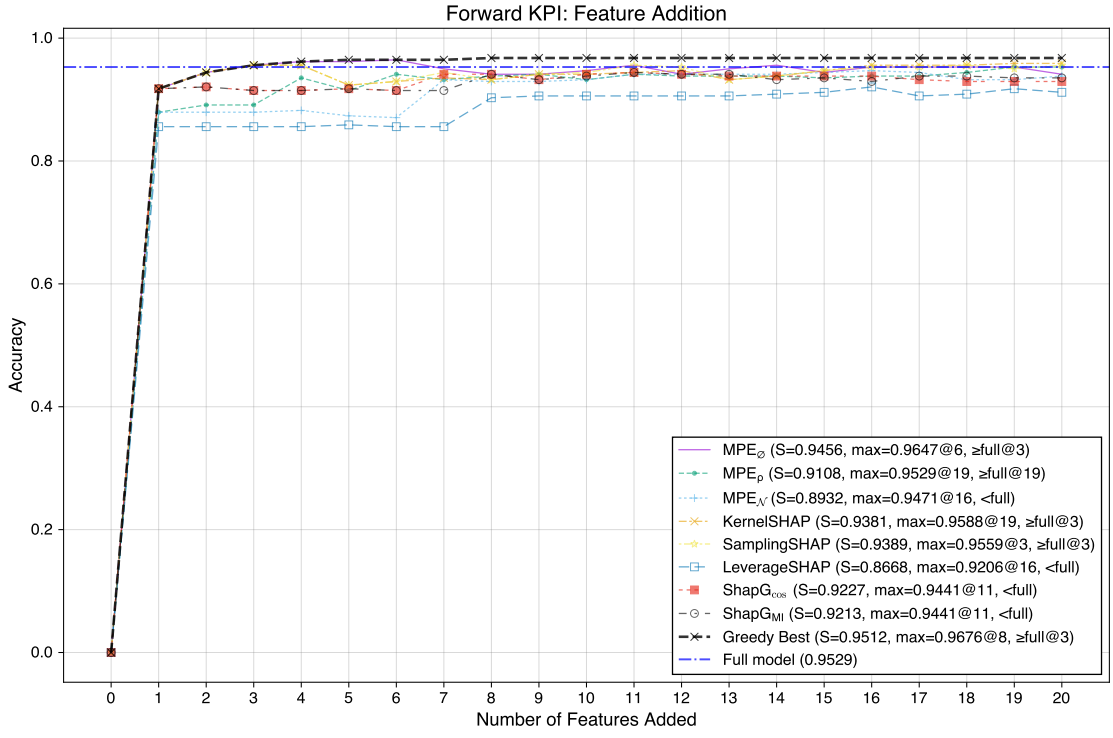


(b) H1N1 (n=35)

Figure 3: Backward KPI curves (retrain mode). Features are removed in order of decreasing importance value given by XAI methods. Steeper decline indicates better identification of important features.

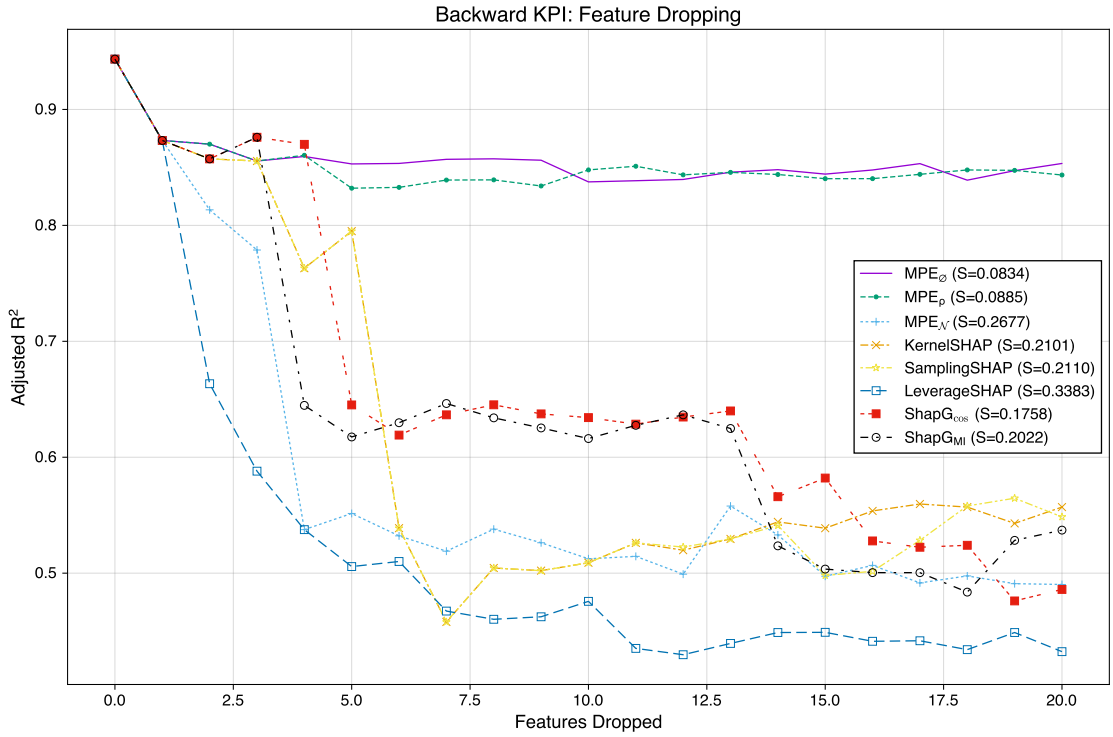


(a) Crime (n=120)

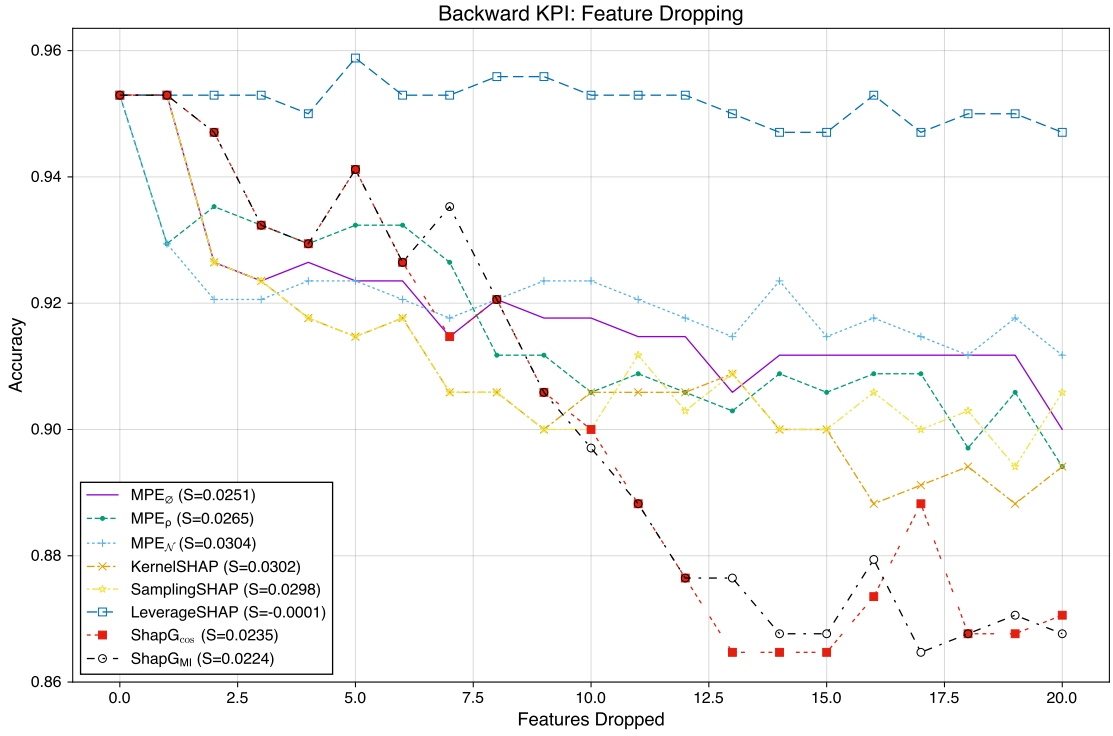


(b) MIC (n=111)

Figure 4: Forward KPI curves for Crime and MIC datasets. Features are added in order of their importance given by different XAI methods. Steeper curves indicate better feature rankings.



(a) Crime (n=120)



(b) MIC (n=111)

Figure 5: Backward KPI curves for Crime and MIC datasets. Features are removed in order of decreasing importance value given by XAI methods. Steeper decline indicates better identification of important features.

5 Stochastic Markovian process explainer

5.1 Method description

Despite the good results obtained by the MPE, the method has an important limitation. According to Algorithm 1, at each step the algorithm always selects the locally optimal action, which does not necessarily lead to the globally optimal absorbing state. By introducing stochasticity into the transition mechanism, the process can explore multiple paths through the state space of size $2^{|\mathcal{N}|}$, corresponding to all subsets of the feature set $\mathcal{N}$, and may therefore discover absorbing states with higher values of $v(\mathcal{S}^*)$ than those obtained by the greedy algorithm. This modification results in a Markovian process similar to the coalition formation process proposed in [9], with the modifications described below.

After constructing the graph describing the stochastic process, we generate M independent scenarios, compute the Shapley I scenario-value for each scenario, and then average these values to obtain more robust estimates of feature importance. We present several approaches for defining the transition probabilities between states. All variants satisfy the same constraint: *only transitions that produce a strictly positive improvement in the characteristic function are assigned positive probability.* Consequently, non-absorbing states cannot have self-loops.

As in the deterministic Markovian process explainer, the characteristic function is defined by formula (1) for regression tasks and by formula (2) for classification tasks.

The Markovian process starts from an initial subset of features $\mathcal{S}_0$. As in the deterministic setting, we consider three possible initial states: (i) $\mathcal{S}_0 = \emptyset$, (ii) $\mathcal{S}_0 \subset \mathcal{N}$, $\mathcal{S}_0 \neq \emptyset$, $\mathcal{S}_0 \neq \mathcal{N}$, and (iii) $\mathcal{S}_0 = \mathcal{N}$.

Given the initial subset $\mathcal{S}_0$, the MPE initiates a coalition formation process. At each step, the algorithm evaluates all possible single-feature additions to and removals from the current subset, retaining only those transitions that yield a positive increase in the value of the characteristic function v. Transition probabilities are then assigned according to one of the following approaches.

Linear. The simplest stochastic variant assigns transition probability proportional from S to $T \in \mathcal{A}(S)$, where $\mathcal{A}(S)$ is the set of all coalitions with positive improvement from state S, to the improvement magnitude:

$$p_{S,T} = \frac{\delta_T}{\sum\limits_{X \in \mathcal{A}(S)} \delta_X}, \tag{6}$$

where $\delta_T = v(T) - v(S)$ denotes the performance improvement by transiting from S to T. This requires no hyperparameters but offers no control over the exploration–exploitation trade-off.

Boltzmann distribution. The Top-k, Top-p, and Min-p strategies described below rely on a Boltzmann (softmax) distribution[3] to convert improvements into probabilities. Given a temperature parameter $\tau > 0$, the Boltzmann probability of an action to transit from state S to T is defined as:

$$\sigma_{S,T} = \frac{\exp(\delta_T/\tau)}{\sum\limits_{X \in \mathcal{A}(S)} \exp(\delta_X/\tau)}. \tag{7}$$

The temperature τ controls the sharpness of the distribution: when $\tau \to 0$, the distribution concentrates on the transition with the largest improvement (recovering the greedy strategy), and when $\tau \to \infty$, it approaches a uniform distribution over all candidates.

Top-k. At each step, only the k candidates with the highest improvement are retained, we denote this set as $\mathrm{Top}_k(\mathcal{A}(S))$. The transition probability within this set is assigned by the Boltzmann

[3]This approach is used in large language models (LLMs) for next-token sampling: Top-p (nucleus) sampling was introduced in [11] and Min-p sampling in [18].

distribution (7) normalized within the retained set:

$$p_{S,T} = \frac{\sigma_{S,T}}{\sum_{X \in \mathrm{Top}_k(\mathcal{A}(S))} \sigma_{S,X}}, \quad T \in \mathrm{Top}_k(\mathcal{A}(S)). \tag{8}$$

When $k = 1$, this recovers the greedy strategy. The parameter k controls the breadth of exploration but it is not adaptive to the distribution of improvements.

Top-p (nucleus sampling). Inspired by nucleus sampling in language models [11], this variant first computes the Boltzmann distribution (7) over all positive candidates, then retains only the smallest set of candidates whose cumulative probability exceeds a threshold p:

$$\mathcal{A}_p(S) = \mathrm{argmin}_{A \subseteq \mathcal{A}(S)} |A| \quad \text{s.t.} \quad \sum_{X \in \mathcal{A}(S)} \sigma_{S,X} \geq p, \tag{9}$$

where $\sigma_{S,X}$ is defined by (7), and followed by normalization within $\mathcal{A}_p(S)$. In other words, the candidates are sorted in decreasing order of their probabilities $\sigma_{S,X}$ and added to $\mathcal{A}_p(S)$ one by one until their cumulative probability reaches the threshold p. The remaining low-probability candidates are discarded. This approach is adaptive: when one action dominates, few candidates survive and the process behaves nearly greedily. When many actions have similar improvements, the candidate set expands to allow broader exploration.

Min-p. Following the Min-p sampling strategy [18], we set a dynamic threshold relative to the most probable candidate under the Boltzmann distribution (7):

$$\mathcal{A}_p(S) = \left\{ X \in \mathcal{A}(S) : \sigma_{S,X} \geq p \cdot \max_{Y \in \mathcal{A}(S)} \sigma_{S,Y} \right\}, \tag{10}$$

again followed by normalization. Min-p is adaptive like Top-p, but it filters by relative quality rather than cumulative mass. The parameter p has a direct interpretation: only candidates whose Boltzmann probability is at least a fraction p of the best candidate's probability are considered.

Example 1. We illustrate how the process is defined according to the four rules given above. Let the current state correspond to coalition S containing features $\{x_1, x_2\}$ with the value of characteristic function $v(S) = 0.5$. We have only four possibilities to increase the value of characteristic function at this step, i.e., $\mathcal{A}(S) = \{T_1, \ldots, T_4\}$. We could transit to the four states by the action of a single-feature addition: $T_1 = \{x_1, x_2, x_3\}$ with $v(T_1) = 0.55$, $T_2 = \{x_1, x_2, x_5\}$ with $v(T_2) = 0.7$, $T_3 = \{x_1, x_2, x_6\}$ with $v(T_3) = 0.75$, and $T_4 = \{x_1, x_2, x_8\}$ with $v(T_4) = 0.75$. The transition process is represented in subgraphs in Figure 6, with the probabilities assigned by each of the four rules.

Once we construct the graph defining the stochastic process of transitions among the states, we can identify the set of absorbing states such that there are no transitions from these states. Each absorbing state corresponds to the final model belonging to the set of nearly best models.

For all stochastic variants, we run M independent scenarios from the same initial state $\mathcal{S}_0$ and compute the process value as the Monte-Carlo averaging:

$$\Psi_i(v) = \frac{1}{M} \sum_{m=1}^{M} \phi_i^{\mathcal{S}_m}(v), \tag{11}$$

where $\phi_i^{\mathcal{S}_m}(v)$ is the Shapley I scenario-value of player i in scenario $\mathcal{S}_m$. This is an unbiased estimator of the process value $\Psi(v) = \sum_{\mathcal{S}} \pi_{\mathcal{S}} \phi^{\mathcal{S}}(v)$ as defined in [8].

The complete procedure is summarized in Algorithm 2. Sampling is necessary because the stochastic process induces a probability distribution over scenarios: the process value $\Psi(v) = \sum_{\underline{\mathcal{S}}} \pi_{\underline{\mathcal{S}}} \phi^{\underline{\mathcal{S}}}(v)$ is an expectation over all paths of the Markov chain, and evaluating it in an exact way is intractable in

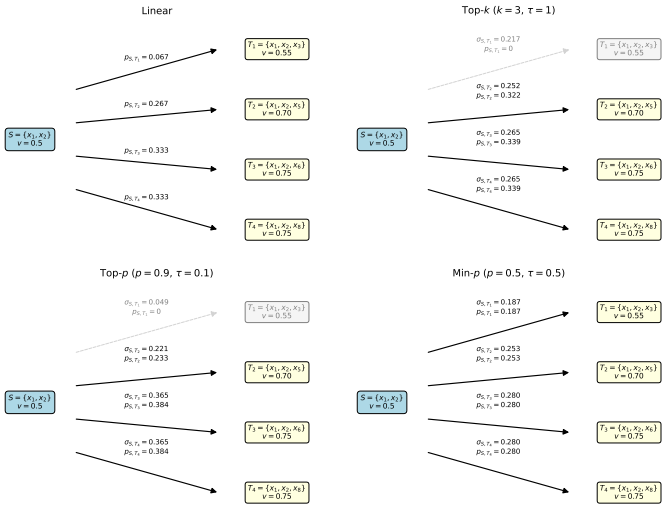


Figure 6: Transition subgraphs for the example: from state $S = \{x_1, x_2\}$ with $v(S) = 0.5$, the probabilities of the four single-feature additions $T_1, \ldots, T_4$ computed by the four rules: Linear, Top-k ($k = 3$, $\tau = 1$), Top-p ($p = 0.9$, $\tau = 0.1$), and Min-p ($p = 0.5$, $\tau = 0.5$), with the same parameter values as in the experiments of Section 5.2. For Top-k, Top-p and Min-p the arrows show both the Boltzmann probability $\sigma_{S,T}$ given by (7) and the final transition probability $p_{S,T}$ after filtering and normalization. Dashed gray arrows denote discarded transitions ($p_{S,T} = 0$).

Algorithm 2 Stochastic Markovian Process Explainer

Require: Feature set $\mathcal{N} = \{1, \ldots, n\}$, initial subset $\mathcal{S}_0 \subseteq \mathcal{N}$, characteristic function v, transition rule R (Linear, Top-k, Top-p or Min-p, with parameters k, p, τ), number of scenarios M
Ensure: Averaged feature importance $(\Psi_i)_{i \in \mathcal{N}}$, absorbing states $\mathcal{S}_1^*, \ldots, \mathcal{S}_M^*$

for $m = 1, \ldots, M$ **do** ▷ Independent scenarios
  $\phi_i^m \leftarrow 0$ for all $i \in \mathcal{N}$; $\mathcal{S} \leftarrow \mathcal{S}_0$
  for each $i \in \mathcal{S}_0$ **do** ▷ Importance of initial members
    $\phi_i^m \leftarrow v(\mathcal{S}_0) - v(\mathcal{S}_0 \setminus \{i\})$
  end for
  repeat
    for each $i \in \mathcal{N} \setminus \mathcal{S}$ **do** ▷ Evaluate join candidates
      $\delta_i^+ \leftarrow v(\mathcal{S} \cup \{i\}) - v(\mathcal{S})$
    end for
    for each $j \in \mathcal{S}$ not added at the previous step **do** ▷ Evaluate leave candidates
      $\delta_j^- \leftarrow v(\mathcal{S} \setminus \{j\}) - v(\mathcal{S})$
    end for
    $\mathcal{A}(\mathcal{S}) \leftarrow$ states reachable by a single join or leave with strictly positive improvement
    if $\mathcal{A}(\mathcal{S}) = \varnothing$ **then**
      break ▷ Absorbing state reached
    end if
    compute $p_{\mathcal{S},\mathcal{T}}$ for all $\mathcal{T} \in \mathcal{A}(\mathcal{S})$ by rule R ▷ Formulas (6)–(10)
    sample $\mathcal{T} \sim p_{\mathcal{S},\cdot}$ and let ℓ be the feature with $\mathcal{S} \,\Delta\, \mathcal{T} = \{\ell\}$
    $\phi_\ell^m \leftarrow \phi_\ell^m + v(\mathcal{T}) - v(\mathcal{S})$; $\mathcal{S} \leftarrow \mathcal{T}$
  until $\mathcal{S}$ was already visited in this scenario **or** $3n$ iterations ▷ Cycle guard
  $\mathcal{S}_m^* \leftarrow \mathcal{S}$ ▷ Absorbing state of scenario m
end for
$\Psi_i \leftarrow \frac{1}{M} \sum_{m=1}^{M} \phi_i^m$ for all $i \in \mathcal{N}$ ▷ Monte-Carlo average (11)
return $(\Psi_i)_{i \in \mathcal{N}}$ sorted decreasingly, $\{\mathcal{S}_1^*, \ldots, \mathcal{S}_M^*\}$

general (see Remark 2). We therefore sample M independent scenarios and use the Monte-Carlo average (11), an unbiased estimator whose standard error decreases as $1/\sqrt{M}$. In our experiments, we set $M = 50$ as a compromise between the variance of the estimate (see the small standard deviations in Table 4) and the computation cost: every step of every scenario evaluates all candidate features once but then follows a single sampled transition, so the total cost is of order $M \cdot L \cdot n$ model retrainings for paths of length L,linear in n and independent of the size of the reachable state space. The transition graph (Figure 7) is the union of the M sampled paths: its nodes are the visited coalitions, and each

edge (S, T) is labeled with its empirical frequency (the number of scenarios in which the transition was sampled) and the exact probability $p_{S,T}$ computed by the chosen rule.

Remark 2. In principle, $\Psi(v)$ under the exact distribution over the absorbing states can be computed without sampling by propagating the probability mass from $\mathcal{S}_0$ through the state graph, requiring $O(n)$ model retrainings for each reachable coalition. We verified this approach on the Housing dataset, where propagation under the Top-k rule visits only a few hundred coalitions and reproduces the corresponding Monte Carlo estimates reported in Table 4. In general, however, the number of reachable coalitions grows exponentially with both n and the depth of the absorbing states, making sampling unavoidable.

Table 4: Stochastic Markovian process results ($M = 50$ scenarios). $v(\mathcal{S}^*)$: absorbing state performance (mean $\pm$ std); $|\mathcal{S}^*|$: absorbing state size ($[\min, \max]$ over the M scenarios).

Strategy	Init	Housing (n=13)		H1N1 (n=35)		Crime (n=120)		MIC (n=111)	
		$v(\mathcal{S}^*)$	$\lvert\mathcal{S}^*\rvert$	$v(\mathcal{S}^*)$	$\lvert\mathcal{S}^*\rvert$	$v(\mathcal{S}^*)$	$\lvert\mathcal{S}^*\rvert$	$v(\mathcal{S}^*)$	$\lvert\mathcal{S}^*\rvert$
Full model		0.869	13	0.857	35	0.943	120	0.953	111
Greedy Best		0.885	9	0.863	20	0.986	12	0.968	8
Greedy	$\emptyset$	0.880	6	0.857	8	0.986	9	0.968	51
	ρ	0.884	6	0.855	8	0.979	9	0.971	57
	$\mathcal{N}$	0.885	9	0.858	34	0.968	111	0.962	110
Linear	$\emptyset$	$0.882_{\pm 0.006}$	[5, 10]	$0.858_{\pm 0.002}$	[8, 18]	$0.981_{\pm 0.002}$	[8, 26]	$0.966_{\pm 0.004}$	[27, 66]
	ρ	$0.882_{\pm 0.005}$	[5, 10]	$0.859_{\pm 0.002}$	[8, 18]	$0.980_{\pm 0.002}$	[12, 27]	$0.958_{\pm 0.004}$	[18, 53]
	$\mathcal{N}$	$0.885_{\pm 0.002}$	[9, 10]	0.858	34	$0.970_{\pm 0.003}$	[92, 112]	$0.961_{\pm 0.002}$	[107, 110]
Top-p	$\emptyset$	$0.881_{\pm 0.005}$	[6, 10]	$0.858_{\pm 0.001}$	[12, 22]	$0.979_{\pm 0.003}$	[10, 44]	$0.965_{\pm 0.006}$	[27, 62]
	ρ	$0.881_{\pm 0.007}$	[5, 10]	$0.858_{\pm 0.002}$	[10, 18]	$0.978_{\pm 0.004}$	[16, 36]	$0.958_{\pm 0.003}$	[18, 73]
	$\mathcal{N}$	$0.885_{\pm 0.002}$	[9, 10]	0.858	34	$0.971_{\pm 0.003}$	[70, 106]	$0.961_{\pm 0.002}$	[106, 110]
Top-k	$\emptyset$	$0.882_{\pm 0.003}$	[6, 10]	$0.858_{\pm 0.002}$	[8, 17]	$0.983_{\pm 0.002}$	[6, 17]	$0.968_{\pm 0.003}$	[27, 58]
	ρ	$0.881_{\pm 0.003}$	[6, 10]	$0.859_{\pm 0.002}$	[8, 19]	$0.981_{\pm 0.003}$	[9, 24]	$0.958_{\pm 0.003}$	[18, 71]
	$\mathcal{N}$	$0.885_{\pm 0.002}$	[9, 10]	0.858	34	$0.970_{\pm 0.002}$	[96, 113]	0.962	[109, 110]
Min-p	$\emptyset$	$0.883_{\pm 0.005}$	[5, 10]	$0.858_{\pm 0.002}$	[11, 24]	$0.978_{\pm 0.005}$	[15, 37]	$0.965_{\pm 0.006}$	[25, 61]
	ρ	$0.882_{\pm 0.006}$	[5, 10]	$0.859_{\pm 0.002}$	[8, 19]	$0.978_{\pm 0.003}$	[18, 39]	$0.958_{\pm 0.003}$	[18, 73]
	$\mathcal{N}$	$0.885_{\pm 0.002}$	[9, 10]	0.858	34	$0.970_{\pm 0.003}$	[78, 103]	$0.961_{\pm 0.002}$	[106, 110]

5.2 Stochastic MPE results

We compare five strategies: Greedy, Linear, Top-p ($p = 0.9$, $\tau = 0.1$), Top-k ($k = 3$, $\tau = 1$), and Min-p ($p = 0.5$, $\tau = 0.5$), each with three initial states ($\emptyset$, ρ, $\mathcal{N}$) across the four datasets. Stochastic variants use $M = 50$ scenarios. Results are summarized in Table 4.

Several observations emerge from Table 4. First, absorbing states broadly match or exceed the full-model performance ($v(\mathcal{S}^*) \gtrsim v(\mathcal{N})$), confirming that the MPE performs effective feature selection regardless of the strategy. Second, on Housing, all variants starting from $\mathcal{N}$ converge to $v(\mathcal{S}^*) = 0.885$, matching the Greedy Best oracle. Third, on H1N1, Min-p with ρ-initialization reaches 0.859 ± 0.002, improving over Greedy$_\emptyset$ (0.857). Among stochastic variants, Top-k ($\emptyset$) tends to produce the most compact absorbing states closest to the greedy solution (e.g., $|\mathcal{S}^*| \in [6, 17]$ on Crime vs. greedy's 9), while achieving competitive $v(\mathcal{S}^*)$. Notably, on MIC, Greedy$_\rho$ achieves $v(\mathcal{S}^*) = 0.971$, surpassing even the Greedy Best oracle (0.968). However, all strategies starting from $\mathcal{N}$ on H1N1 converge to the same absorbing state of 34 features, suggesting a strong attractor in the coalition landscape.

Regarding the initial state, the forward procedure ($\emptyset$) generally yields the best or comparable $v(\mathcal{S}^*)$ with the most compact absorbing states. The backward procedure ($\mathcal{N}$) tends to retain many features, particularly on high-dimensional datasets (Crime: $|\mathcal{S}^*| \approx 100$; MIC: $|\mathcal{S}^*| \approx 109$). Stochastic variants partially mitigate this but do not fully overcome the tendency of backward procedures to get trapped near the full set. The ρ-initialization offers a middle ground but does not consistently outperform the forward procedure.

Overall, the stochastic variants provide comparable or slightly improved absorbing state performance with low variance across scenarios, while additionally revealing the multiplicity of locally optimal feature subsets. Figure 7 illustrates the transition graph for the Housing dataset using the Top-k strategy ($k = 3$) with empty initial state and $M = 50$ scenarios. Each node represents a visited coalition state with its characteristic function value, and edges show observed transitions with empirical frequencies and exact transition probabilities. The absorbing state is highlighted in orange.

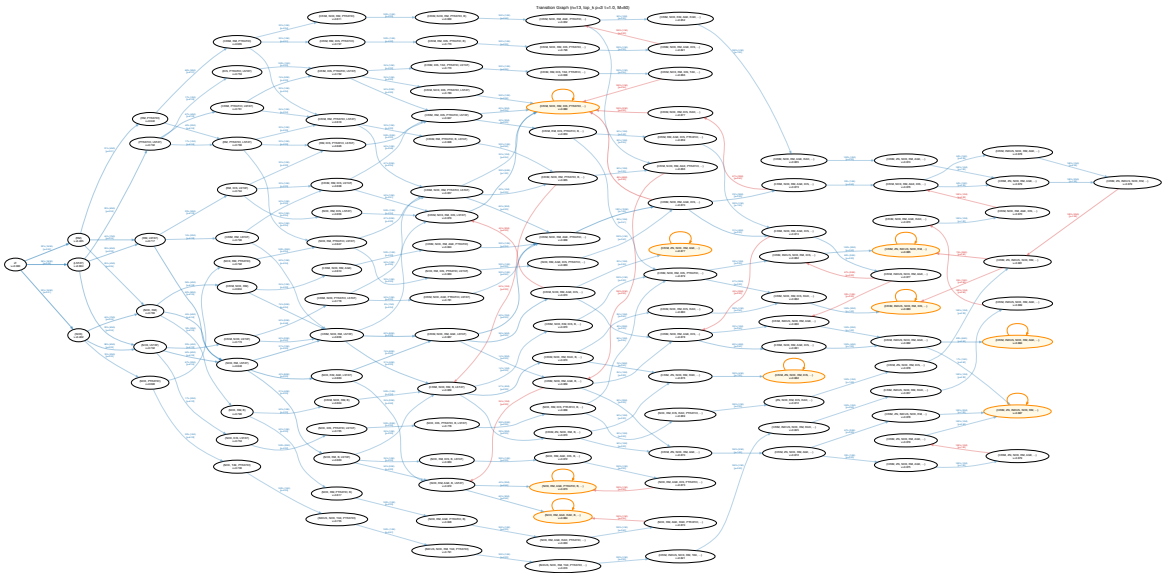


Figure 7: Transition graph for Housing dataset (Top-k, $k = 3$, $\mathcal{S}_0 = \emptyset$, $M = 50$). Nodes represent visited coalition states; edge labels show empirical frequency and exact transition probability [$p = \cdot$]. The absorbing state is marked in orange. Only transitions realized in at least one of the M scenarios are drawn, so the exact probabilities on the outgoing edges of a node may sum to less than one; the remaining probability mass belongs to transitions that never materialized in the samples.

Table 5 reports the computation time. Aggregating $M = 50$ scenarios makes the stochastic variants much more expensive than the single-scenario Greedy procedure, and the cost grows with the dimension and the path length. It stays within a minute on Housing (1–43 s), but reaches minutes to hours on the high-dimensional datasets, especially for the wider rules (Linear, Top-p and Min-p). The stochastic explainer is thus best suited to small and medium datasets, or to settings where a family of near-optimal models justifies the cost.

Table 6 lists all absorbing states reached on Housing by the Top-k strategy (the setting of Figure 7). Over the $M = 50$ scenarios the procedure reaches nine distinct states, all exceeding the full-model performance (100.4%–102.1%): each has 6 to 10 features and always includes the dominant predictors LSTAT, NOX and RM, differing only in the less essential ones. The most frequent state (19/50) is a compact 6-feature model ($v(\mathcal{S}^*) = 0.880$), while the best-performing one (0.887, 10 features) is sampled less often. The explainer thus returns a family of near-optimal models trading off size against performance, rather than a single subset.

To compare these states we use three complementary criteria. Their *intersection* gives the features every near-optimal model retains (on Housing, the core is {LSTAT, NOX, RM}), a more robust importance conclusion than any single model yields. Their *Pareto front* in the (size, performance) plane (Figure 8) collects the states that no other dominates (being both smaller and at least as accurate), from which the analyst picks a compactness–accuracy trade-off. Their *sampling frequency* (marker size in Figure 8) measures robustness: the most frequent state (6 features, 0.880, 19/50) has the largest basin of attraction and is a sensible default. When a single ranking is needed, the adjusted R^2 already penalizes size, so the highest-$v(\mathcal{S}^*)$ state is the "best" model.

Table 5: Computation time (in seconds) of the stochastic Markovian process explainer ($M = 50$ scenarios), by strategy and initial state. Greedy is the single-scenario deterministic procedure, included for reference.

Strategy	Init	Housing	H1N1	Crime	MIC
		(n=13)	(n=35)	(n=120)	(n=111)
Greedy	$\emptyset$	1.3	15.4	50.6	229.9
	ρ	1.0	8.4	24.6	223.9
	$\mathcal{N}$	1.1	6.3	150.0	13.7
Linear	$\emptyset$	28.1	1439.8	4703.8	9007.3
	ρ	17.2	694.0	4100.0	2819.6
	$\mathcal{N}$	3.1	7.0	13892.7	575.3
Top-p	$\emptyset$	27.4	2306.2	8244.1	8329.4
	ρ	25.3	912.0	8438.0	3082.2
	$\mathcal{N}$	3.1	7.0	23647.6	655.0
Top-k	$\emptyset$	16.8	919.7	1939.0	7251.0
	ρ	9.9	488.8	2310.8	3759.0
	$\mathcal{N}$	2.6	6.3	9320.2	61.2
Min-p	$\emptyset$	43.1	2523.4	10173.3	9227.2
	ρ	35.2	1105.0	9476.4	3099.8
	$\mathcal{N}$	4.4	6.4	27289.0	585.3

Table 6: The distinct absorbing states reached on the Housing dataset by the Top-k strategy ($k = 3$, $\mathcal{S}_0 = \varnothing$, $M = 50$), ordered by sampling frequency. Every state exceeds the full-model performance ($R^2_{\text{adj}} = 0.869$) and contains the three dominant features LSTAT, NOX and RM; the last column lists the remaining features.

Freq.	$\lvert\mathcal{S}^*\rvert$	$v(\mathcal{S}^*)$	% full	Other features
19/50	6	0.880	101.2	CRIM, PTRATIO, DIS
6/50	9	0.885	101.7	CRIM, ZN, PTRATIO, B, INDUS, DIS
5/50	10	0.884	101.6	CRIM, B, INDUS, AGE, DIS, RAD, TAX
4/50	10	0.887	102.1	CRIM, ZN, B, INDUS, AGE, DIS, TAX
4/50	6	0.884	101.7	B, AGE, RAD
4/50	9	0.883	101.6	CRIM, PTRATIO, B, INDUS, DIS, RAD
4/50	8	0.882	101.4	CRIM, ZN, B, DIS, RAD
3/50	6	0.873	100.4	PTRATIO, B, AGE
1/50	7	0.877	100.8	CRIM, ZN, AGE, DIS

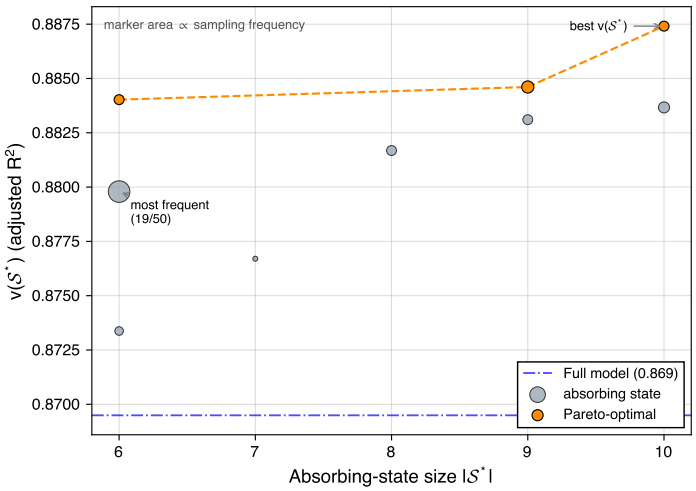


Figure 8: The nine absorbing states on Housing (Top-k, $k = 3$, $\mathcal{S}_0 = \emptyset$, $M = 50$) in the (size, performance) plane; marker area is proportional to sampling frequency. Pareto-optimal states (orange, dashed line) are those no other state dominates (being both smaller and at least as accurate). All exceed the full-model performance (dash-dotted line).

This structure varies across datasets. H1N1 and MIC also exhibit a stable core (respectively 4 and 19 features shared by all absorbing states), whereas on Crime, whose many continuous features are highly correlated, the absorbing states are far more diverse and share only a single common feature, reflecting a large number of nearly equivalent feature subsets that the stochastic explainer is able to reveal.

6 Conclusion and future work

In this paper, we introduced the Markovian Process Explainer (MPE), a unified XAI framework that simultaneously performs feature selection and quantifies feature importance within a model-agnostic procedure. By modeling feature selection as a Markovian coalition formation process, MPE bridges the gap between conventional stepwise selection of features and Shapley-value-based method of measuring their importance in the final model. The method offers flexible initialization, allowing it to start from the empty set, the full feature set, or a correlation-based subset of features, and accommodates both greedy and stochastic transition rules. Empirical evaluation across four datasets, including high-dimensional cases, demonstrates that MPE achieves importance rankings competitive with, and often superior to, established approaches, while delivering built-in feature selection as a byproduct. Crucially, its stochastic variants return a set of near-optimal models rather than a single one, offering significant practical flexibility when non-predictive constraints (e.g., measurement costs or regulatory preferences) influence feature subset choice.

While the current results are promising, we see several directions to extend the method's applicability and deepen its theoretical foundations:

- **Richer transition rules.** The current process adds or removes a single feature at each step. Generalizing it to compound moves, such as adding or removing a subset of features at once, could help the process escape local absorbing states, at the cost of a larger action set per step.
- **Theoretical analysis.** A formal study of the induced Markov chain would strengthen the method, for instance by explaining why the absorbing states range from highly consistent (sharing a large common core of features) to highly diverse (many near-equivalent subsets with little overlap), and how the transition temperature controls the trade-off between exploring them and quickly converging to one.
- **Scalability of the characteristic function.** The main cost of MPE comes from repeatedly evaluating $v(\mathcal{S})$. Using approximate evaluations instead of retraining a model for every feature subset could make the method applicable to higher-dimensional data and larger models.
- **Beyond feature selection: model compression.** The same coalition process can be applied to the structural components of a large language model (individual weight matrices, attention heads, or entire layers), with $v(\mathcal{S})$ obtained by masking the excluded components instead of retraining. MPE would then return not a single pruned network but a Pareto front of size-versus-quality configurations, a promising but challenging direction for structured pruning under different deployment budgets.

References

[1] Hirotugu Akaike. A new look at the statistical model identification. IEEE Transactions on Automatic Control, 19(6):716–723, 1974.

[2] R. Azen and D. V. Budescu. The dominance analysis approach for comparing predictors in multiple regression. Psychological Methods, 8:129–148, 2003.

[3] Matt Baucum and Meysam Rabiee. Supervised clustered interpretability: Explainable subgroup discovery via cluster separation and partial dependence disparity. INFORMS Journal on Computing, 38(4):1253–1268, 2026.

[4] Giorgos Borboudakis and Ioannis Tsamardinos. Forward-backward selection with early dropping. Journal of Machine Learning Research, 20(8):1–39, 2019.

[5] Norman R. Draper and Harry Smith. Applied Regression Analysis. Wiley, New York, 3rd edition, 1998.

[6] Kaushik Dutta, Ajay Kumar, Zhiling Guo, Martin Bichler, Ramesh Ram Delen, Dursun, and J. Paul Brooks. Responsible ai and data science for social good. INFORMS Journal on Computing, 38(4):1033–1044, 2026.

[7] Michael Alin Efroymson. Multiple regression analysis. Mathematical Methods for Digital Computers, 191–203, 1960.

[8] U. Faigle and M. Grabisch. Values for Markovian coalition processes. Economic Theory, 51:505–538, 2012.

[9] U. Faigle and M. Grabisch. A note on values for Markovian coalition processes. Economic Theory Bulletin, 1:111–122, 2013.

[10] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. Journal of Machine Learning Research, 3:1157–1182, 2003.

[11] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In International Conference on Learning Representations (ICLR), 2020.

[12] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A highly efficient gradient boosting decision tree. In Advances in Neural Information Processing Systems (NeurIPS), volume 30:3149–3157, 2017.

[13] Kimia Keshanian, Daniel Zantedeschi, and Kaushik Dutta. Features selection as a nash-bargaining solution: Applications in online advertising and information systems. INFORMS Journal on Computing, 34(5):2485–2501, 2022.

[14] Ron Kohavi and George H. John. Wrappers for feature subset selection. Artificial Intelligence, 97(1–2):273–324, 1997.

[15] Jing Liu, Chi Zhao, and Elena Parilina. Feature importance methods for linear regression model. Applied Intelligence, 2026.

[16] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems, 30, 2017.

[17] Christopher Musco and R. Teal Witter. Provably accurate shapley value estimation via leverage score sampling. In International Conference on Learning Representations (ICLR), 2025. Spotlight.

[18] Minh Nguyen. Turning up the heat: Min-p sampling for creative and coherent llm outputs. arXiv preprint arXiv:2310.06022, 2024.

[19] Erik Strumbelj and Igor Kononenko. An efficient explanation of individual classifications using game theory. The Journal of Machine Learning Research, 11:1–18, 2010.

[20] Dipti Theng and Kishor K Bhoyar. Feature selection techniques for machine learning: a survey of more than two decades of research. Knowledge and Information Systems, 66(3):1575–1637, 2024.

[21] William N Venables and Brian D Ripley. Modern applied statistics with S. Springer Science & Business Media, 2013.

[22] C. Zhao, J. Liu, and E. Parilina. Complete-to-sparse: A novel graph construction strategy for efficient shapg. In International Conference on Mathematical Optimization Theory and Operations Research. 180–194. Springer, 2025.

[23] C. Zhao, J. Liu, and E. Parilina. ShapG: new feature importance method based on the shapley value. Engineering Applications of Artificial Intelligence, 148:110409, 2025.

[24] C. Zhao, J. Liu, and E. Parilina. The shapley value contribution to explainable artificial intelligence: A comprehensive survey. Dynamic Games and Applications, 16:1013–1050, 2026.