Genetic Programming for Explainable Manifold Learning

Ben Cravens, Andrew Lensen, *Member, IEEE,* Paula Maddigan, Bing Xue, *Fellow, IEEE*

***Abstract*—Manifold learning techniques play a pivotal role in machine learning by revealing lower-dimensional embeddings within high-dimensional data, thereby enhancing the efficiency, interpretability, and scalability of data analysis. Despite their utility, current manifold learning methods often lack explicit functional mappings, which are critical for ensuring explainability in regulated and high-stakes applications. This paper introduces Genetic Programming for Explainable Manifold Learning (GP-EMaL), a novel integration of Genetic Programming (GP) and Explainable Artificial Intelligence (XAI). GP-EMaL leverages the inherently interpretable, tree-based structures of GP to generate explicit, functional mappings while directly addressing complexity challenges through innovative penalties for tree size, symmetry, and operator selection. By enabling customisable complexity metrics, GP-EMaL adapts to diverse application needs, achieving high manifold quality and significantly improved explainability. Comprehensive experiments demonstrate that GP-EMaL matches or exceeds the performance of existing approaches, producing simpler and more interpretable models. This work advances the state of explainable manifold learning, paving the way for its adoption in domains such as healthcare, environmental modelling, and financial analysis.**

***Index Terms*—Manifold Learning, Genetic Programming, Dimensionality Reduction, Explainable Artificial Intelligence**

I. Introduction

MANY real-world high-dimensional datasets have highly complex topological structure [1]–[3]. Manifold learning (MaL) methods, a category of Nonlinear Dimensionality Reduction (NLDR) techniques, are crucial for transforming these datasets into reduced embedding spaces, thereby aiding in the comprehension of the data's intrinsic structure. Techniques like functional mapping and graph-based methods facilitate the visualization of these embeddings but often result in significant data loss [4]. In contrast, non-mapping and deep learning methods, while mitigating data loss, compromise interpretability due to their increased complexity. The application of complex, unexplainable NLDR methods, especially in regulated sectors, can have substantial ethical, legal, scientific, and commercial implications, as outlined in legislation such as the EU's General Data Protection Regulation (GDPR) [5].

In high-risk domains such as healthcare, it is essential for practitioners to understand the features driving model outcomes [6]–[9]. This necessity underscores the growing importance of interpretable NLDR techniques. Genetic Programming (GP), an evolutionary computation (EC) method, evolves symbolic functional mappings represented by syntax trees. Prior research [10]–[13] has shown that GP-based methods are effective in creating interpretable models, particularly when the symbolic trees maintain low complexity. Recent work has proposed methods specifically for interpretable GP-NLDR [14], [15]. A recent study [16] further extends the explainability of GP trees from manifold learning models by integrating large language models such as ChatGPT to provide conversational explanations. GP, as a mapping method, provides an explicit function enabling the reproduction of embeddings and the implementation of sensitivity analysis. Techniques like automatic program simplification can further simplify these functions. Hence, GP is highly effective in scenarios where understanding the relative importance of features is crucial for ethical and legal reasons.

Interpretability in machine learning is a subjective concept [17], [18]. Precisely defining what makes a model interpretable is challenging [19], but clear indicators of limited interpretability include having large numbers of parameters or excessively complex operations. By characterising interpretability through a low-complexity tree structure, we can quantify complexity via a measurable metric, thereby reducing subjectivity. However, reducing complexity typically introduces a trade-off, leading to lower-quality embeddings. Existing GP-NLDR approaches, such as GP-MaL-MO [15], balance embedding quality and the number of embedding dimensions but do not explicitly optimise tree complexity. To address this gap, we propose an explainability-focused approach (GP-EMaL), explicitly incorporating a tree complexity metric into the multi-objective optimisation framework alongside embedding quality, thereby significantly enhancing interpretability.

Major Contributions

- We introduce GP-EMaL, a Genetic Programming approach for multi-objective manifold learning, addressing the challenge of model interpretability by explicitly incorporating complexity as an optimization objective.
- A novel complexity metric is proposed, integrating structural, size, and semantic penalties to generate compact, symmetrical, and interpretable trees.
- We demonstrate GP-EMaL's ability to achieve a balance between interpretability and embedding quality, validated through experiments on diverse datasets.
- Open-source code and a web-based application are provided to enable reproducibility and facilitate adoption in real-world applications.

This work was supported by the University Research Fund at Te Herenga Waka–Victoria University of Wellington under grant number 410128/4223.

The authors are with the Centre for Data Science and Artificial Intelligence, and School of Engineering and Computer Science; Victoria University of Wellington; Wellington 6140; New Zealand (e-mail: ben.conor.reese.cravens@gmail.com; andrew.lensen@ecs.vuw.ac.nz; paula.maddigan@ecs.vuw.ac.nz; bing.xue@ecs.vuw.ac.nz).

II. Background and Related Work

A. Dimensionality Reduction

Dimensionality reduction, a key technique in machine learning, aims to reduce the number of features in a dataset while preserving as much original information as possible. This process is essential for two main reasons: firstly, it significantly reduces the computational burden associated with large datasets; secondly, it helps overcome the "curse of dimensionality" [20].

Dimensionality reduction methods can be classified into two broad categories: feature selection [21] and feature extraction/construction [22]. Feature selection (FS) methods involve choosing a subset of relevant features directly from the dataset, maintaining their original form and interpretability. Examples of popular feature selection techniques include Mutual Information-based selection and Recursive Feature Elimination (RFE). While inherently interpretable due to their simplicity, feature selection methods are limited in their ability to capture complex, nonlinear relationships among features.

Feature construction, in contrast, transforms the original features into a new set of derived features, potentially capturing complex relationships. Such transformations can be further categorized into mapping and non-mapping methods. Mapping methods, including Principal Component Analysis (PCA), produce a reduced feature space via an explicit functional mapping, offering interpretability by combining the original features. Non-mapping methods like t-SNE [23] and UMAP [24], however, prioritize data compression and efficient representation over interpretability, as they do not provide explicit mappings for reconstructing the original feature space.

GP-based approaches are a distinct category of mapping methods, leveraging evolutionary processes to construct explicit, nonlinear functional mappings. GP-NLDR methods can thus flexibly model complex manifold structures beyond what is achievable by standard FS techniques, while maintaining interpretability through their explicit tree-based representations.

B. Nonlinear Dimensionality Reduction

NLDR is employed when relationships in a dataset are too complex to be captured by linear methods alone [23]. Notable among NLDR techniques are deep neural networks, such as autoencoders, which consist of an encoder compressing data into a lower-dimensional space and a decoder reconstructing it back. This process is assessed using a loss function that measures the discrepancy between the input and output, enabling the autoencoder to efficiently map the data while preserving structural integrity [25].

Another significant category of NLDR methods is graph-based approaches, such as t-distributed stochastic neighbour embedding (t-SNE) [26]. These methods create a nearest neighbour graph to capture the inherent structure of the data. The graph is then embedded into a lower-dimensional space, preserving its structural characteristics. t-SNE specifically constructs a graph by considering pairwise similarities between high-dimensional data points and then subsequently optimising a 2-D or 3-D embedding to align with this graph. This method is particularly effective in preserving local data structures and for visualising high-dimensional data [27].

The integration of GP in NLDR was pioneered by the GP for Manifold Learning (GP-MaL) method [14]. Subsequent studies such as GP-MaL-MO [15] have continued to build on the concept, further advancing the research in this field. GP-Mal-MO applies a multi-objective approach to GP-based NLDR, balancing between preserving the nearest neighbour structure in the embedding and minimising the embedding dimensionality. This paper builds upon these developments to enhance the explainability of GP-based NLDR models.

C. GP and Evolutionary Multi-Objective Optimisation

GP is an evolutionary computation technique that evolves programs, typically represented as tree structures, to solve problems or model data [28]. GP is particularly notable for its ability to evolve interpretable models, making it a valuable tool in domains where understanding the model's decision-making process is crucial [29], [30]. In manifold learning, GP has been effectively used to evolve mappings that reduce data dimensionality while preserving its intrinsic structure [14].

Evolutionary Multi-Objective Optimisation (EMO) extends the EC paradigm to handle multiple, often conflicting, objectives. EMO algorithms evolve a population of solutions, aiming to find a set of Pareto-optimal solutions that represent the best possible trade-offs between the objectives [31]. This approach is particularly beneficial when dealing with complex problems where optimising a single objective could lead to sub-optimal or undesirable solutions.

A critical development in the EMO field was the MOEA/D (Multi-Objective Evolutionary Algorithm based on Decomposition) method [32]. MOEA/D decomposes a multi-objective optimisation problem into a number of scalar optimisation sub-problems and optimises them simultaneously. Each sub-problem focuses on a specific region of the Pareto front, enabling MOEA/D to effectively explore and exploit the search space. This method has gained popularity due to its efficiency and effectiveness, especially in problems with a complex Pareto front landscape.

In the context of GP for manifold learning, MOEA/D offers a robust framework for balancing the trade-offs between manifold quality, embedding complexity, and interpretability. By applying MOEA/D, GP is expected to evolve a diverse set of Pareto-optimal solutions, each representing a different balance between these objectives. This flexibility allows users to select a solution that best fits their specific needs, whether prioritising interpretability for a domain expert's analysis or optimising performance for automated tasks.

D. Tree Complexity

In GP, the complexity of the syntax trees is a crucial factor affecting both the performance and interpretability of the generated models. One method for calculating tree complexity starts at the root node and progresses recursively [33]. This method, outlined in Table I, aggregates the complexity of each node based on specific rules, facilitating efficient ($\mathcal{O}(t)$ for a tree containing t nodes) and consistent calculations (e.g. as

TABLE I: Complexity Values [33]

Complexity(n)	Symbol of node(n)
1	constant
2	variable
$\sum_{c \in c_n}$ Complexity(c)	$+, -$
$\prod_{c \in c_n}$ Complexity(c) + 1	$*, /$
Complexity$(n_1)^2$	square
Complexity$(n_1)^3$	square root
$2^{\text{Complexity}(n_1)}$	sin, cos, tan, exp, log

where c is a child of node n, and n_1 is the first child of n.

opposed to time-based complexity measures [34]). While this approach generates simpler trees, it has its limitations: it does not constrain the shape of the embedding tree, potentially resulting in asymmetrical and visually confusing/uninterpretable structures. Asymmetrical trees are harder to understand than the equivalent symmetrical tree with the same number of nodes, as they contain more layers of nested functions. Additionally, while this existing approach penalises complex operations (function nodes), it does not explicitly minimise the tree size, often leading to large trees using many simpler operators. The fixed function set also does not consider the varying interpretability of functions across different contexts.

E. Tree Complexity Metrics

Various tree complexity metrics have been proposed in the literature, falling into two main categories: structural complexity and functional (semantic) complexity metrics [35]–[38]. Structural complexity assesses the tree at a node level, while functional complexity is based on the complexity of the tree's overall behaviour.

Research on structural complexity has explored factors like penalizing large trees through parsimony pressure [39] and reducing nested functions [40]. However, the integration of other metrics, such as assessing the asymmetry of trees, has been less explored. Our work aims to fill this gap.

In terms of functional complexity, the prevalence of many nonlinear operators, especially when nested, has often been identified as a factor of complexity [33]. The use of Tikhonov regularization to apply a global smoothing function to the tree has also been studied [41]. This method penalizes the function based on the norm of its higher-order derivatives, resulting in a smoother function.

The distinction between structural and functional complexity is not absolute, as these aspects can overlap. For example, a tree represented by a nested function like $sin(cos(exp(x)))$ may be considered both functionally complex (due to its nonlinear nature) and structurally complex (due to its nested structure). Thus, these complexity measures should be viewed as complementary heuristics rather than distinct categories. For measuring explainability, structural complexity approaches are more appropriate as they directly consider the complexity of the tree structure that the user will be trying to interpret.

III. Proposed Method: GP-EMaL

In our proposed method, GP-EMaL, we build upon the framework established by GP-MaL-MO [15], targeting a key limitation: the tendency towards complex tree structures which can impede interpretability. Our GP-EMaL approach introduces a novel complexity metric for expression trees, replacing the manifold dimensionality objective in GP-MaL-MO that focuses on minimizing the complexity while retaining the essential manifold quality cost metric based on neighbourhood ordering [14], [15]. This change aims to enhance the tree interpretability without compromising the effectiveness of the manifold learning process. The core algorithm for multi-objective optimisation is similar to GP-MaL-MO, as illustrated in Fig. 1, which outlines the architecture of GP-EMaL.

A. GP Design

GP-EMaL is a filter-based approach that evaluates functional mappings independent of specific supervised tasks. This significantly reduces computational cost compared to wrapper methods, making GP-EMaL more scalable to larger datasets.

The principle of GP-EMaL's dimensionality reduction lies in evolving symbolic functional mappings that optimize embedding quality and tree complexity simultaneously. Each GP individual consists of multiple trees, where each tree maps one dimension of the embedding space. These trees are constructed from functional building blocks, including arithmetic operations (e.g. addition, multiplication), mathematical functions (e.g. sine, exponential), and terminals such as input features and constants. For instance, a tree could represent the mapping `y = (f1 + f2) × sin(f3)`, where `f1`, `f2`, and `f3` are input features, and +, ×, and `sin` are functional nodes.

GP-EMaL optimises two conflicting objectives: (1) preserving neighbourhood relationships in the embedding space and (2) minimizing tree complexity. Complexity is measured using metrics such as the number of nodes, symmetry of tree structures, and use of expensive operators. The MOEA/D algorithm is used to balance these objectives, producing a Pareto-optimal front of solutions that represent trade-offs between embedding quality and interpretability. The optimization process involves: (1) initializing a diverse population of GP individuals with random tree structures; (2) evaluating individuals using the multi-objective fitness function (described in Section III-D); and (3) applying genetic programming operators such as crossover and mutation to evolve the population over generations.

The multi-tree structure in GP-EMaL allows each individual to contain several trees, initialized with a number of trees randomly chosen within the range $[2, m \div 2]$, where m is the number of features in the dataset. Evolutionary operators include a specialized Add/Remove Tree mutation, which enables individuals to dynamically adjust their tree count, balancing manifold quality and complexity. For example, a new tree can be added to improve embedding quality or an existing tree can be removed to simplify the individual. A Standard mutation operator modifies a random subtree within any tree, while crossover swaps subtrees between two individuals. This flexibility allows GP-EMaL to adaptively evolve interpretable solutions tailored to different datasets.

B. Complexity Metric

GP-EMaL introduces an enhanced complexity metric that explicitly targets tree interpretability while maintaining high

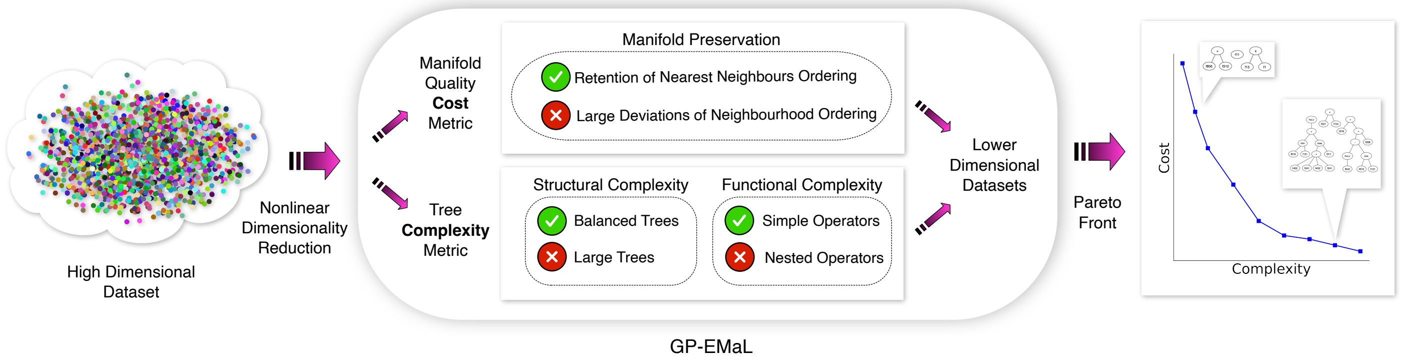


Fig. 1: GP-EMaL Architecture

TABLE II: Comparison of GP-EMaL and GP-MaL-MO

Aspect	GP-EMaL
Complexity Objective	Explicitly optimizes for tree complexity using a parameterized metric with symmetry and size penalties.
Manifold Quality Objective	Retains the measure from GP-MaL-MO, ensuring high-quality embeddings.
Operator Focus	Encourages simpler operators near the root, improving interpretability.
Multi-Objective Design	Focuses on complexity and embedding quality trade-offs.
Innovation	Introduces penalties for asymmetry, nested operations, and configurable operator costs.
Adaptability	Uses user-defined parameters for tuning interpretability in specific applications.

manifold quality. This metric significantly expands on the simpler dimensionality-based approach used in GP-MaL-MO. To better illustrate the differences, a comparison of GP-EMaL and GP-MaL-MO is presented in Table II.

The complexity metric in GP-EMaL is a unified measure that evaluates tree structures by integrating four interdependent factors: symmetry, size, operator usage, and semantic interpretability. These components are designed to work in tandem to ensure a balance between embedding quality and interpretability. By incorporating penalties for structural asymmetry and large size, along with operator-specific costs, GP-EMaL evolves trees that are inherently easier to interpret compared to the baseline method.

For example, the parameterized function set enables domain-specific adjustments, where simpler operators like addition and subtraction may have lower costs in general-use applications, while complex operators like trigonometric or exponential functions might be penalized unless required for specific fields like physics. These operator preferences help maintain interpretability in diverse domains, aligning tree structures with the expectations and expertise of end-users.

To further illustrate the methodology, we detail the key components of the complexity metric in the following sub-subsections. These include penalties for structural asymmetry, scaling penalties for tree size, operator-specific costs, and their integration into the overall metric.

The tree complexity metric, $F(T)$, combines three main components: symmetry balancing, tree size scaling, and operator costs. These components are described below, with each term linked to its corresponding symbol in Table III.

TABLE III: Symbols and Definitions in the Complexity Metric

Symbol	Definition
A_i	Asymmetry penalty at node i
$\text{Size}_{\text{Left}}$	Size of the left subtree at node i
$\text{Size}_{\text{Right}}$	Size of the right subtree at node i
Δ_i	Difference between $\text{Size}_{\text{Left}}$ and $\text{Size}_{\text{Right}}$
α	Ratio of tree size to maximum tree size
S_T	Scaling term for tree size
L_i, R_i	Complexities of the left and right subtrees at node i
$\mathcal{O}(t)$	Asymptotic complexity based on subtree size t

1) Symmetry Balancing Term: The symmetry balancing term A_i is applied at each node to promote balanced tree structures. It penalises asymmetry between the left and right subtrees, defined by their sizes $\text{Size}_{\text{Left}}$ and $\text{Size}_{\text{Right}}$. The penalty is calculated as:

$$A_i = 2^{|\Delta_i|} - 1, \quad \Delta_i = \text{Size}_{\text{Left}} - \text{Size}_{\text{Right}}. \tag{1}$$

This term ensures that balance is considered at all levels of the tree, leading to visually and structurally simpler models.

2) Scaling Term: Recognising the challenges posed by larger trees to interpretability, GP-EMaL incorporates a scaling term S_T to penalise excessive tree sizes. The scaling term is governed by the parameter α, representing the size of tree T relative to a predefined maximum size Size_{Max}. The penalty is defined as:

$$S_T = \begin{cases} 1, & \text{if } \alpha < \mu, \\ 2\alpha, & \text{if } \alpha > \mu, \end{cases} \quad \alpha = \frac{t}{\text{Size}_{\text{Max}}}. \tag{2}$$

This term increases the penalty as the tree size exceeds the threshold μ.

Parameters such as maximum tree size play a crucial role in balancing the trade-off between embedding quality and interpretability in GP-EMaL. This parameter directly affects the complexity of operation trees, which in turn influences the conflicting objectives of embedding accuracy and model simplicity. While systematic sensitivity analysis is beyond the scope of this study, the parameter settings were informed by prior work in GP-based NLDR and other GP literature. Moreover, the choice of maximum tree size is often application-dependent, as users may prioritize either higher interpretability or greater embedding fidelity depending on their specific requirements.

TABLE IV: Summary of Function Cost Sets

Cost	Operator	Complexity	Cost Scaling
sum	$n_i \in (+, -)$	$L_i + R_i$	$\mathcal{O}(t)$
prod	$n_i \in (\times, \div)$	$\max(L_i \times R_i, L_i, R_i)$	$\mathcal{O}(t^2)$
exp	$n_i \in (f_1, .., f_n)$	$2^{(L_i+R_i)}$	$\mathcal{O}(2^t)$

3) Tree Complexity Term: The GP-EMaL approach introduces enhanced flexibility in measuring tree complexity by introducing a parameterisable function set. This set allows users to define both the tree operators and their respective contributions to overall tree complexity. The cost assigned to each operator is proportional to the height of the respective subtree, with the scaling of these costs categorised as linear, quadratic, or exponential.

For example, consider a function set comprising three basic arithmetic operators and a nonlinear operator: $[+, -, \times, sigmoid]$, paired with a function cost set $[sum, sum, prod, exp]$. This configuration results in a higher prevalence of the simpler arithmetic operators $[+, -]$ within the evolved trees due to their lower cost. The multiplication operator $\times$ would appear less frequently, typically higher in the tree, while the nonlinear operator $sigmoid$ would be least common, generally positioned near the top of the tree, directly affecting the features.

This methodological approach in GP-EMaL allows for nuanced control over the complexity of the generated trees. It enables the specification of not only which operators are present but also their preferred locations within the tree structure, thereby influencing both the functional complexity and the interpretability of the trees.

Table IV summarises the settings for these function cost sets. $f_1, .., f_n$ represent the set of non-arithmetic or "special" operations (e.g. trigonometric functions). $\mathcal{O}$ indicates the asymptotic scaling complexity based on the number of nodes in the left and right child subtrees.

The overall tree complexity function, denoted as $F(T)$, is defined as:

$$\begin{aligned} F(T) = S_T \times \Bigg[&\sum_{n_i \in (+,-)} (L_i + R_i + A_i) \\ &+ \sum_{n_j \in (\times,\div)} (\max(L_j \times R_j, L_j, R_j) + A_j) \\ &+ \sum_{n_k \in (f_1,\ldots,f_n)} (2^{(L_k+R_k)} + A_k) \Bigg]. \end{aligned} \tag{3}$$

where A_i refers to the asymmetry penalty at each node as defined in Eq. (1), and S_t is the scaling term for tree size as per Eq. (2).

C. Function and terminal sets

Our default function set contains a range of basic and more complex operators. It includes the arithmetic functions (+,−,×, protected ÷), conditional functions (max and min), and nonlinear operators (absolute value, ReLU and the Sigmoid function). The terminal set (leaf nodes) contains each of the features of the dataset.

D. Objective functions

As a multi-objective GP-MaL approach, our method has two objectives: one that minimises the difference between the input feature space and the embedding space; and a second that minimises the complexity of the trees used to transform the input space into the embedding space.

As our focus in this work is on reducing the complexity of the learned trees, we utilise the same first objective (Cost) as in GP-MaL-MO, which has shown to be an effective measure:

$$Cost(I, X) = \frac{1}{|X|} \sum_{x \in X} \frac{(1 - \text{corr}(N, N'))}{2} \tag{4}$$

for GP individual I and dataset X with data point $x \in X$. *corr* represents Spearman's rank correlation coefficient and $|X|$ is the number of data points. N and N' is the ordering of I's neighbours in the high-dimensional and embedded spaces, respectively. The 2 in the denominator is a scaling factor that constrains the complexity function to $[0, 1]$. This equation penalises GP individuals where the embedding has a different neighbourhood structure from the input dataset, where a value of 0 represents a perfect preservation of structure, and 1 represents the worst possible result (a complete reversal of neighbour orderings).

Whereas GP-MaL-MO simply uses embedding dimensionality (number of trees) as the complexity measure, our proposed GP-EMaL uses the new complexity metric introduced in Section III-B. Given that a GP-EMaL individual contains multiple trees (embedding dimensions), we sum across the tree complexity metric (Eq. (3)) to get the overall complexity of a given GP individual (I):

$$Complexity(I) = \sum_{T \in I} F(T) \tag{5}$$

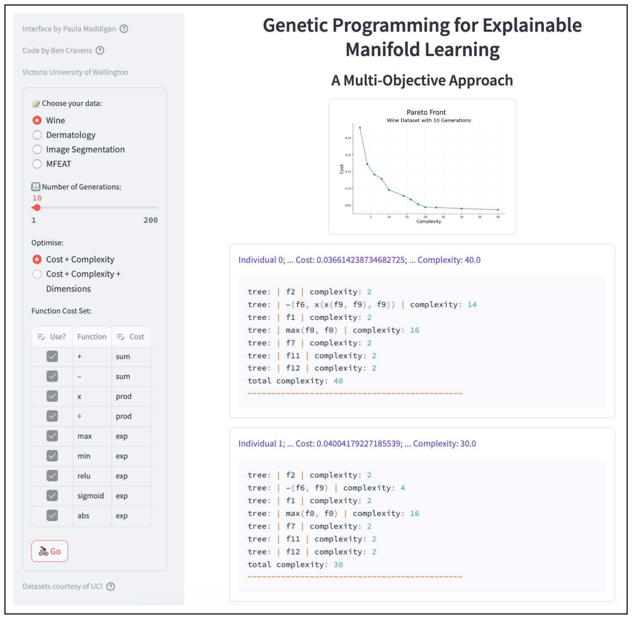
where T is a tree in I.

These two objective functions are generally conflicting, as less complex trees represent functions with fewer degrees of freedom and, therefore, cannot retain the neighbourhood structure of the input space as accurately as more complex trees, leading to a higher cost. By using the MOEA/D algorithm to evolve GP individuals according to these two objectives, we are able to produce a population of GP individuals to obtain an approximate Pareto front, where individuals represent different (non-dominated) tradeoffs between embedding quality (Cost) and tree complexity.

E. Implementation

The GP-EMaL algorithm is built using Python with the code publicly available[1]. A Streamlit web-based application is also accessible for ease of use[2] depicted in Fig. 2.

[1]https://github.com/cravies/GP-EMaL
[2]https://gp-emal.streamlit.app/


Interface by Paula Maddigan
Code by Ben Cravens
Victoria University of Wellington
Genetic Programming for Explainable Manifold Learning
A Multi-Objective Approach
Pareto Front
Wine Dataset with 10 Generations
Cost
Complexity
Datasets courtesy of UCI
Go

Choose your data: [✓ Wine] Dermatology Image Segmentation MFEAT

Number of Generations: 10 (1 – 200)

Optimise: [✓ Cost + Complexity] Cost + Complexity + Dimensions

Function Cost Set:

Use?	Function	Cost
✓	+	sum
✓	–	sum
✓	x	prod
✓	÷	prod
✓	max	exp
✓	min	exp
✓	relu	exp
✓	sigmoid	exp
✓	abs	exp

Individual 0; ... Cost: 0.036614238734682725; ... Complexity: 40.0

```
tree: | f2 | complexity: 2
tree: | -(f6, x(x(f9, f9), f9)) | complexity: 14
tree: | f1 | complexity: 2
tree: | max(f0, f0) | complexity: 16
tree: | f7 | complexity: 2
tree: | f11 | complexity: 2
tree: | f12 | complexity: 2
total complexity: 40
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Individual 1; ... Cost: 0.04004179227185539; ... Complexity: 30.0

```
tree: | f2 | complexity: 2
tree: | -(f6, f9) | complexity: 4
tree: | f1 | complexity: 2
tree: | max(f0, f0) | complexity: 16
tree: | f7 | complexity: 2
tree: | f11 | complexity: 2
tree: | f12 | complexity: 2
total complexity: 30
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Fig. 2: Streamlit Application Interface.

IV. Experiment Design

We evaluated our new GP-EMaL approach using the following methodology, divided into three distinct analyses: *Predictive Performance*, *Complexity Summary Statistics*, and *Visual Comparisons*.

A. Predictive Performance

To assess the effectiveness of GP-EMaL, we adopted a widely used approach for comparing manifold learning (MaL) algorithms, focusing on the classification accuracy achievable in the embedded spaces they produce. Classification accuracy serves as a practical and unbiased proxy for evaluating the preservation of data structures within embeddings.

For this evaluation, we trained KNN and RF classifiers on embeddings generated by both the proposed GP-EMaL method and the baseline GP-MaL-MO method. Ten-fold cross-validation was employed to ensure robustness, using random forests (RFs) with 100 trees and k-nearest neighbors (KNN). RF was chosen for its stability and general reliability, while KNN was included due to its dependence on high-quality embeddings for accurate distance-based predictions.

The evaluation used seven datasets (listed in Table V), primarily from the UCI repository [42], along with the COIL20 dataset [43]. These datasets were selected to represent a diverse range of features, instances, and class distributions. For each dataset, classification accuracy was recorded over 30 runs, using the parameter settings outlined in Table VI. These settings were chosen based on established practices in the GP literature [14], [15], ensuring robust performance. Tree depths, population size, generations, and mutation/crossover probabilities were informed by preliminary tests to balance computational efficiency and embedding quality, while complexity penalties directly align with our interpretability objectives.

The nature of our research and methodology introduces significant challenges in applying traditional statistical techniques to validate the findings and compare GP-EMaL with GP-MaL-MO. Developing a unified accuracy measure that appropriately

TABLE V: Classification Datasets for Evaluation

Dataset	Features	Instances	Classes
Wine	13	178	3
Dermatology	34	358	6
Image Segmentation	19	2310	7
MFEAT	649	2000	10
MNIST	784	2000	2
COIL20	1024	1440	20
Isolet	617	1560	26

TABLE VI: GP-EMaL Parameter Settings

Parameter	Setting
Generations	1000
Population Size	100
"Standard" Mutation	15%
Add/Remove Tree Mutation	15%
Crossover	70%
Max. No. Trees	max(2, $\lceil \#\text{features} \div 2 \rceil$)
Min. Tree Depth	2
Max. Tree Depth ($Size_{Max}$)	14
Decomposition	Tchebycheff
Pop. Initialisation	Ramped Half-and-half
Function Set	$+ - \times \div$ ReLu sigmoid max min abs
Cost Set	sum sum prod prod exp exp exp exp exp

accounts for complexity — such as tree size, use of expensive operators, or feature utilisation — involves subjective trade-offs that vary depending on the application context.

Moreover, GP-MaL-MO was not explicitly optimised with respect to complexity metrics, whereas GP-EMaL incorporates these as primary optimisation objectives. This fundamental difference means that direct statistical comparisons would inherently favour GP-EMaL, potentially misrepresenting the strengths of GP-MaL-MO. Additionally, the two methods generate differing numbers of individuals, leading to unbalanced datasets that violate key assumptions of many statistical tests.

Instead, we adopt a qualitative approach supported by detailed visualizations (Figs. 10 to 16) and summary statistics (Fig. 17). This approach highlights the nuanced trade-offs between embedding quality and complexity, which are integral to multi-objective optimization.

We also acknowledge the broader need in future research for standardized methodologies and statistical tools to analyze multi-objective methods like these, particularly when balancing conflicting objectives.

B. Complexity Summary Statistics

To complement the predictive performance analysis, we gathered summary statistics across all datasets. For both GP-EMaL and GP-MaL-MO, we measured the average number of nodes, the number of unique features used, and the complexity of the nodes chosen. These statistics offer a high-level comparison of the two methods in terms of their complexity and explainability.

C. Visual Comparisons

Finally, to directly contrast the explainability of the two approaches, we compared typical example individuals generated by GP-MaL-MO and GP-EMaL on the COIL20 dataset. As

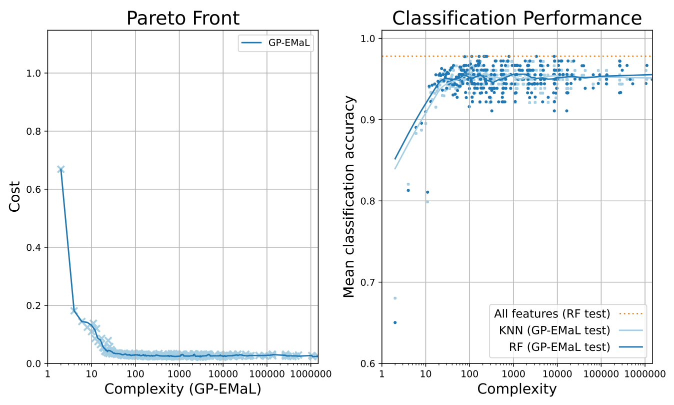


Fig. 3: Front and Classif. Performance on Wine

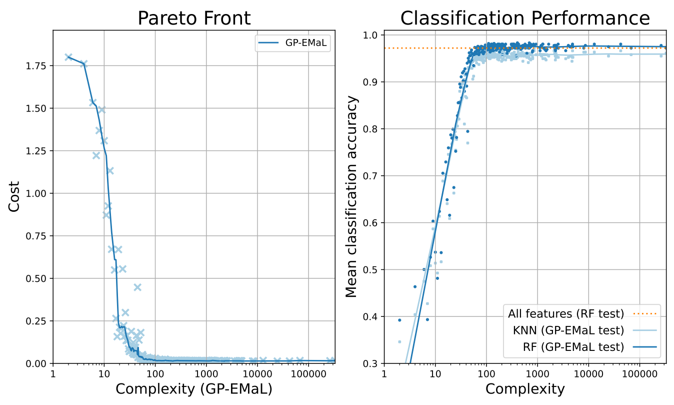


Fig. 4: Front and Classif. Performance on Dermatology

the dataset with the highest number of features in our tests, COIL20 represents a particularly challenging case for producing explainable non-linear dimensionality reduction (NLDR) models.

V. Results

A. Predictive Performance

We begin by exploring the trade-off between complexity and classification accuracy in the individuals produced by GP-EMaL in Section V-A1.

1) GP-EMaL Complexity vs. Classification Accuracy: Figures 3 to 9 show the results for GP-EMaL using the seven datasets, in order of roughly increasing dataset complexity. For each dataset, we show two visualisations. The first depicts the approximated Pareto front. The cost (reduction in manifold quality) is measured on the y-axis, with the tree complexity metric (in log scale) shown on the x-axis. As both cost and complexity should be minimised, points closer to the bottom-left are of higher quality. The second plot shows the Classif. Performances of both KNN (lighter blue) and RF (darker blue) predictors using the new embeddings. A smoothing spline is run on the results to average and smooth the curves over the 30 runs. As classification accuracy should be maximised, points closer to the top-left of the second set of plots are better-performing. We also include a dotted orange horizontal line, which is the performance of the RF classifier when using all features. This line represents a benchmark of what a good classifier could achieve on the source high-dimensional dataset.

The Wine dataset (Fig. 3) is the least complex of our tested datasets, with only 13 features and 178 instances. The approximated Pareto front shows that GP individuals with very low complexity ($\approx$30 as measured by Eq. (4)) are able to achieve a very low cost, strongly retaining the structure of the original high-dimensional space by using only simple operators. A similar result is seen in the classification plots, where both the KNN and Random Forest classifiers perform well with above 90% accuracy using the embeddings produced by a simple GP individual.

The Dermatology dataset (Fig. 4) exhibits similar behaviour to that of Wine but requires a slightly more complex solution ($\approx$75) to generate an embedding with a cost near zero. Both

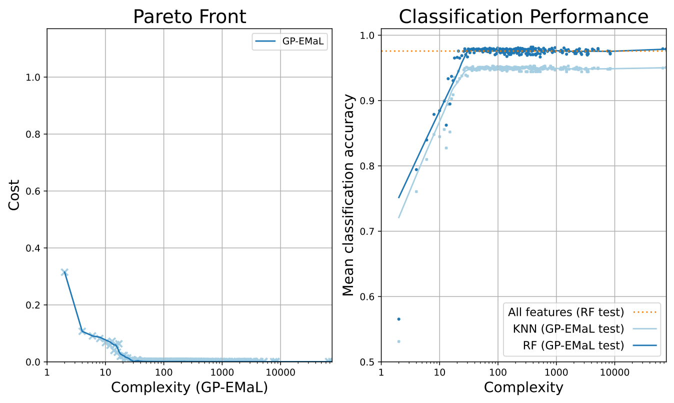


Fig. 5: Front and Classif. Performance on Image Segmentation

RF and KNN achieve in excess of 90% classification accuracy, matching the baseline performance of all features.

The Image Segmentation results (Fig. 5) resemble those of Wine, given the similar number of features in both datasets. This is the first set of results where there is a substantial performance difference between RF and KNN. The RF classifer achieves over 95% accuracy with a GP complexity of only $\approx$30, whereas the KNN classifier is slightly worse across all complexities.

The MFEAT and MNIST datasets show comparable patterns in their results (Figs. 6 and 7). They show convergence to low-cost solutions at a slightly higher complexity ($\approx$100) than the simpler datasets, achieving over 90% classification accuracy on RF and KNN. These are the first datasets where there is a (very small) gap in performance between RF using all features and RF using the embedding produced by even the most complex GP trees. This is an inherent trade-off: on sufficiently complex problems, a more complex (less explainable) model is needed to achieve the best performance. However, as we show in the rest of this paper, a much simpler model often can still achieve very high performance — a trade-off that is easily preferable in many applications.

The most complex datasets, COIL20 and Isolet (Figs. 8 to 9) also show an asymptotic pattern, where cost approaches zero at a complexity of $\approx$75. While the RF classification performance on COIL20 is near the baseline, on Isolet, there is a gap of around 10%, even at the highest complexities. If very high

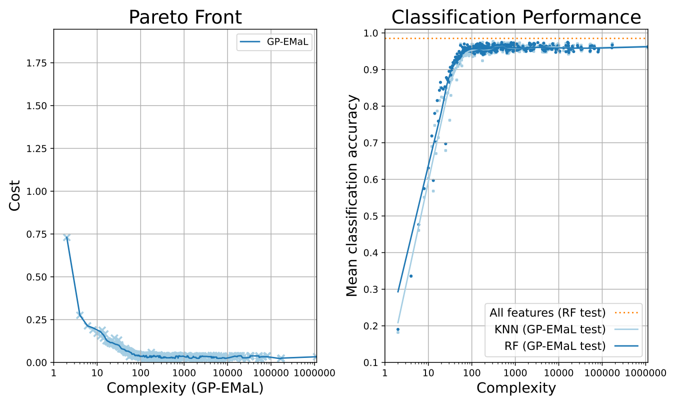


Fig. 6: Front and Classif. Performance on the MFEAT dataset

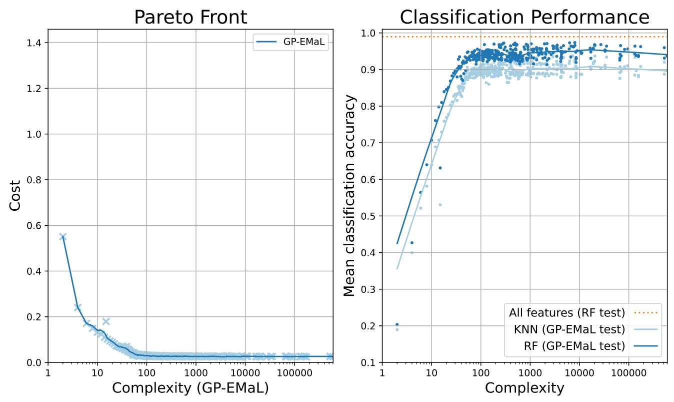


Fig. 8: Front and Classif. Performance on COIL20

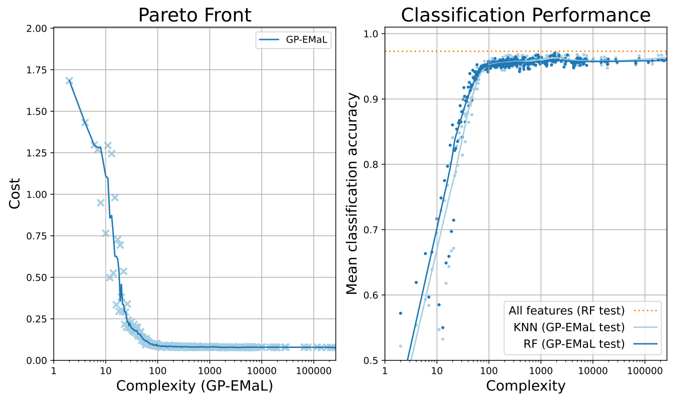


Fig. 7: Front and Classif. Performance on MNIST

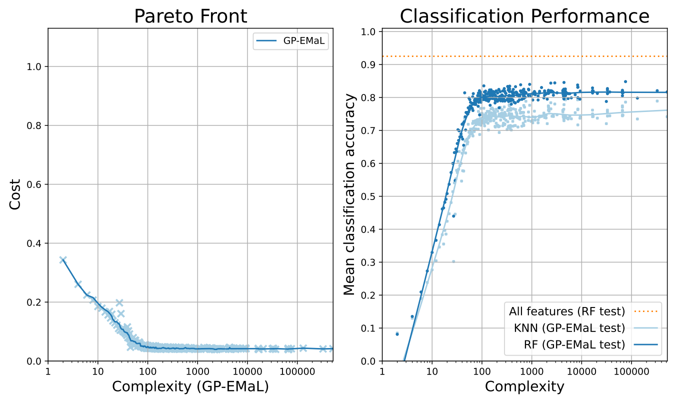


Fig. 9: Front and Classif. Performance on Isolet

performance is crucial on Isolet, the original GP-MaL-MO approach may be required (as discussed later) instead.

2) Comparing GP-EMaL to GP-MaL-MO: To compare GP-EMaL to GP-MaL-MO as fairly as possible, we evaluate the performance of each method along three different signals of complexity/interpretability. These are:

1) The total number of nodes used in the GP individual, which gives a "raw" measure of tree complexity;
2) The total number of *exp* cost operators used in the GP individual, which measures how many of the most difficult-to-understand functions are used; and
3) The number of unique features used, i.e. the number of dimensions in the high-dimensional input data that were used to construct the embedding. Using fewer unique features will make a GP individual easier to understand, as it requires less interpretation of different aspects of the data domain.

We plot each of these measures against the classification accuracy for all individuals in the fronts evolved by both GP-EMaL and GP-MaL-MO across all our runs. We use only RF for computing classification accuracy, as it always outperforms KNN and hence provides a more accurate measure of embedding quality. The plots for each of the datasets are shown in Figs. 10 to 16, where each orange × symbol shows one GP individual evolved by GP-EMaL and each ○ symbol shows one GP individual evolved by GP-MaL-MO. We also include an overlay within each plot to show a closer perspective of the boundary between the GP-EMaL and GP-MaL-MO regions. For all of these plots, points closer to the top-left of the plot are of higher quality (having higher classification accuracy at a lower complexity). As with our earlier results, we include a dotted horizontal line to represent the "baseline" performance, which uses the same RF classifier with all the original features of the dataset.

Across nearly all of the datasets, the proposed GP-EMaL method is able to achieve similar performance to the existing GP-MaL-MO method despite having smaller trees with fewer complex operators and using fewer unique features. This is a clear testament to the complexity function proposed in this work — by using a more sophisticated measure of complexity (compared to just the number of trees in GP-MaL-MO) as our second objective, we can produce explainable trees that effectively preserve data structure in the embedded space.

This improvement in GP-EMaL is especially clear in the first three datasets (Figs. 10 to 12). On these, GP-EMaL is able to retain sufficient manifold structure to reach the baseline accuracy while clearly using much less complex trees than GP-MaL-MO; GP-EMaL effectively dominates GP-MaL-MO across the three different complexity measures. GP-EMaL is consistently less complex at high levels of accuracy; on the Image Segmentation dataset, for example, GP-MaL-MO has accuracies as low as 70%, despite using all features in the dataset and using trees with many more nodes. Finally, GP-EMaL provides a much better approximation of the low-

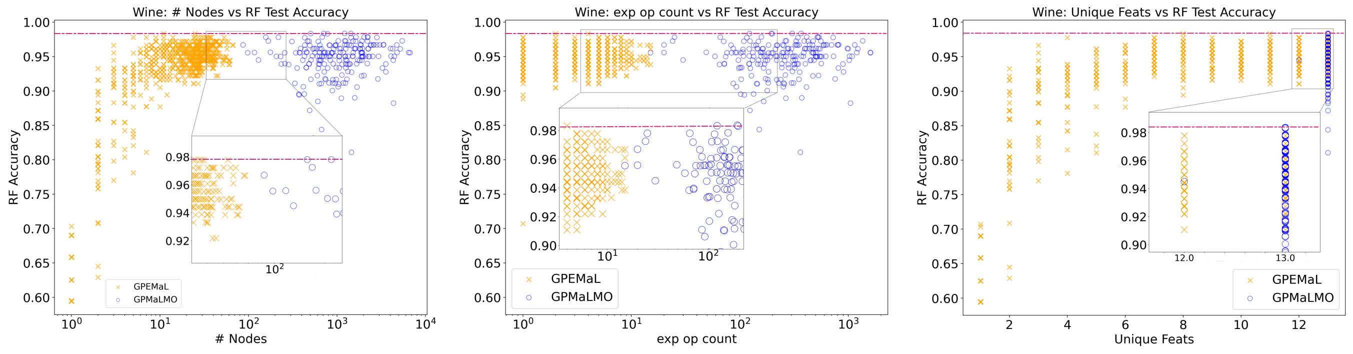


Fig. 10: Performance Comparisons using the Wine dataset

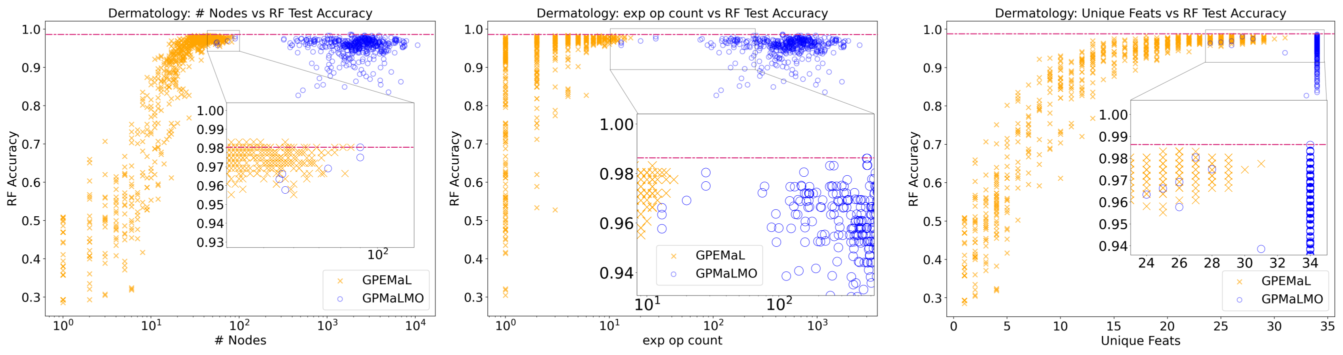


Fig. 11: Performance Comparisons using the Dermatology dataset

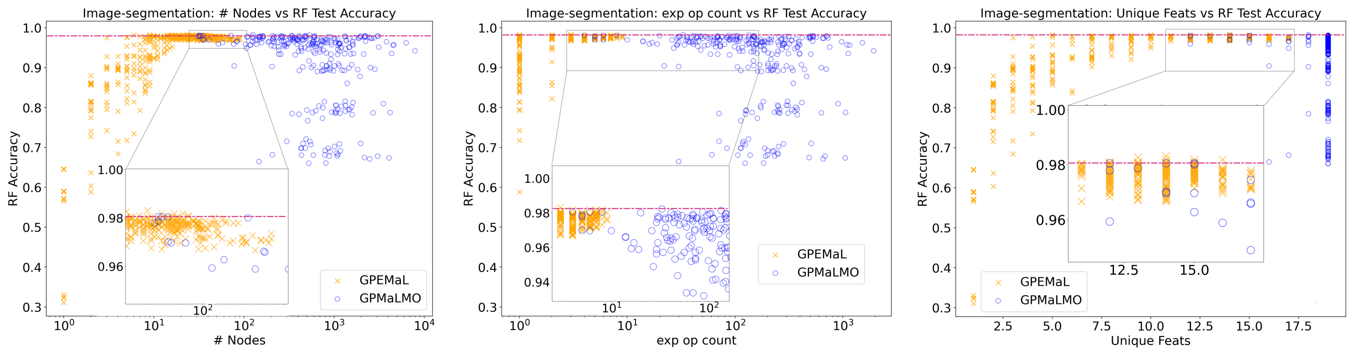


Fig. 12: Performance Comparisons using the Image Segmentation dataset

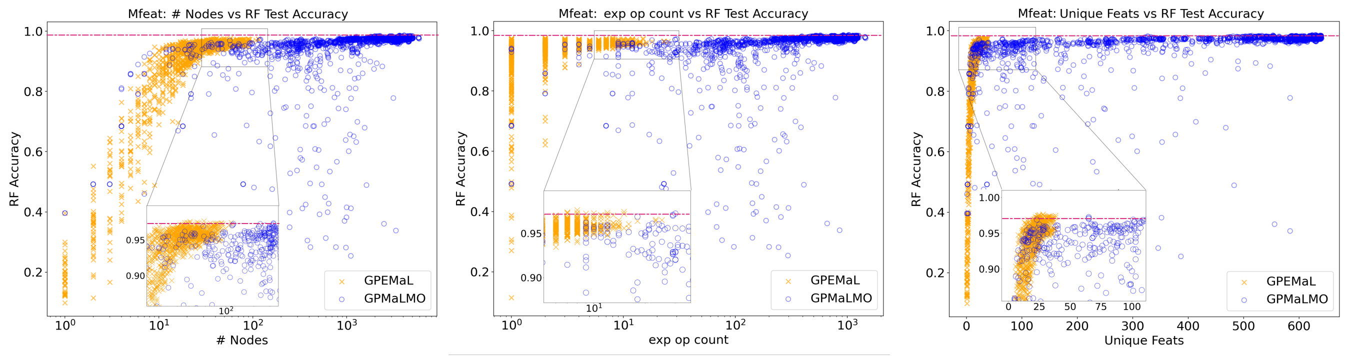


Fig. 13: Performance Comparisons using the MFEAT dataset

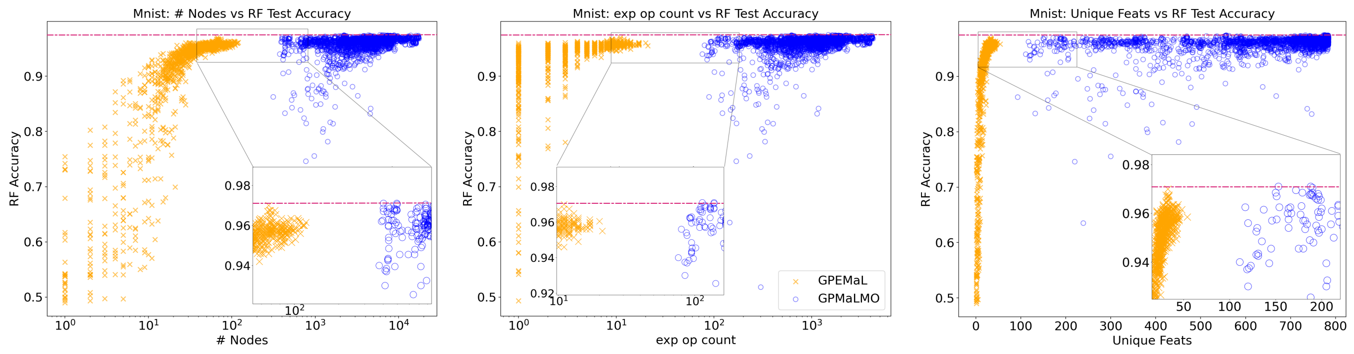


Fig. 14: Performance Comparisons using the MNIST dataset

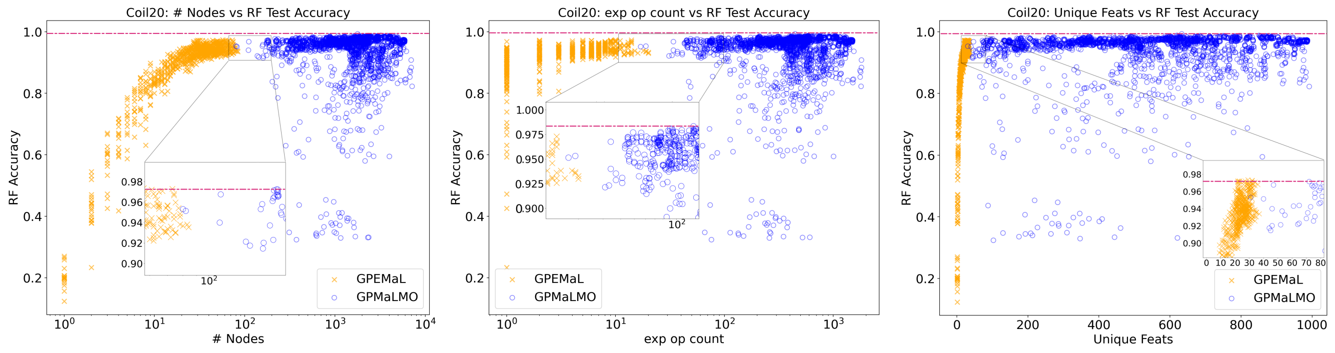


Fig. 15: Performance Comparisons using the *COIL20* dataset

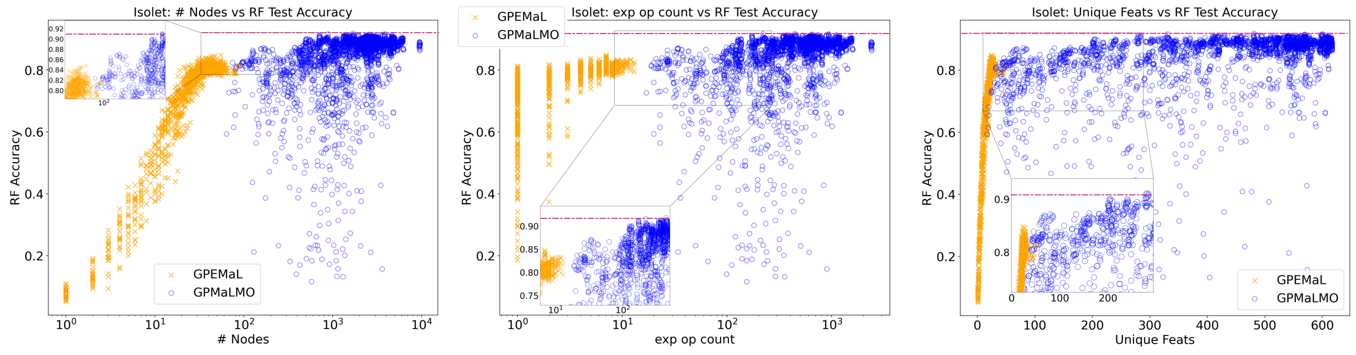


Fig. 16: Performance Comparisons using the Isolet dataset

complexity part of the Pareto front: it produces extremely simple individuals (e.g. using only a few nodes and/or one or two features) that often can achieve over 80% accuracy — a trade-off that may be desirable on some problem domains.

On the MFEAT dataset (Fig. 13), we observe that GP-MaL-MO has a small but consistent decrease in RF accuracy for less-complex GP individuals. At similar levels of complexity, GP-EMaL outperforms GP-MaL-MO. For example, GP-MaL-MO achieves around 93% accuracy for a GP individual with 100 nodes, whereas GP-EMaL achieves close to the baseline accuracy (around 97%) with as few as 40 nodes. Similar patterns are seen for the number of exponential operators and unique features. Indeed, GP-EMaL can consistently reach the baseline accuracy when using only 25 unique features, whereas GP-MaL-MO has significant variance in accuracy.

As the complexity of our datasets further increases, a small gap in classification accuracy between the two methods begins to emerge. On MNIST (Fig. 14), GP-EMaL is very slightly ($\leq$ 1%) below the baseline accuracy, whereas GP-MaL-MO is able to reach baseline performance. There is, however, a marked complexity gap between the two methods: GP-EMaL is nearly an order of magnitude less complex than GP-MaL-MO (e.g. 100 vs 500 nodes) despite the only 1% difference in performance. A very similar result is observed on the COIL20 dataset (Fig. 15), albeit with a smaller gap in complexity and performance between the most-complex GP-EMaL individual and the least-complex GP-MaL-MO individual.

On the most complex dataset, Isolet (Fig. 16), GP-MaL-

TABLE VII: Summary Statistics

Statistic	Description
# $\mathcal{O}(2^n)$ nodes	total number of $\mathcal{O}(2^n)$ cost operators
# $\mathcal{O}(n^2)$ nodes	total number of $\mathcal{O}(n^2)$ cost operators
# $\mathcal{O}(n)$ nodes	total number of $\mathcal{O}(n)$ cost operators
# $\mathcal{O}(1)$ nodes	total number of $\mathcal{O}(1)$ cost operators (feature nodes)
# nodes	total number of nodes
# unique feat	number of unique feature nodes

MO is clearly able to attain higher classification accuracy than GP-EMaL — but not without using more complex GP trees. GP-MaL-MO requires very complex trees (≈1000 nodes, ≈100 complex operators, ≈300 unique features) to achieve a baseline level of accuracy. It is interesting to note that together, GP-EMaL and GP-MaL-MO almost form a combined front, with neither method having individuals that Pareto-dominate the other. On large, complex datasets such as Isolet, it may be desirable to run both GP-EMaL and GP-MaL-MO and produce a combined front to provide the user with a comprehensive set of trade-offs from very simple to very complex models.

In conclusion, the results demonstrate a clear trade-off between complexity and predictive accuracy across various datasets. This trade-off is more evident in datasets with higher complexity. For most datasets, GP-EMaL offers an improvement over GP-MaL-MO by producing substantially more explainable GP individuals without compromising on predictive accuracy. For the more complex datasets, GP-MaL-MO may have an edge in accuracy, but GP-EMaL stands out as a robust alternative, especially when the focus is on interpretability, with only a minimal performance decrease.

B. Summary Statistics

To understand the overall complexity of the two methods across all the datasets, we calculated several summary statistics for both GP-MaL-MO and GP-EMaL. These statistics are described in Table VII.

Fig. 17 shows the summary statistics in the form of histograms for each of the two GP methods. By looking at the x-axis values, we can see that GP-EMaL produces much less complex trees than GP-MaL-MO. Indeed, the maximum value for GP-EMaL on each of the graphs is actually below the *median* value on the corresponding graph for GP-MaL-MO. The median and maximum number of nodes used by GP-EMaL are around 100× smaller than that of GP-MaL-MO.

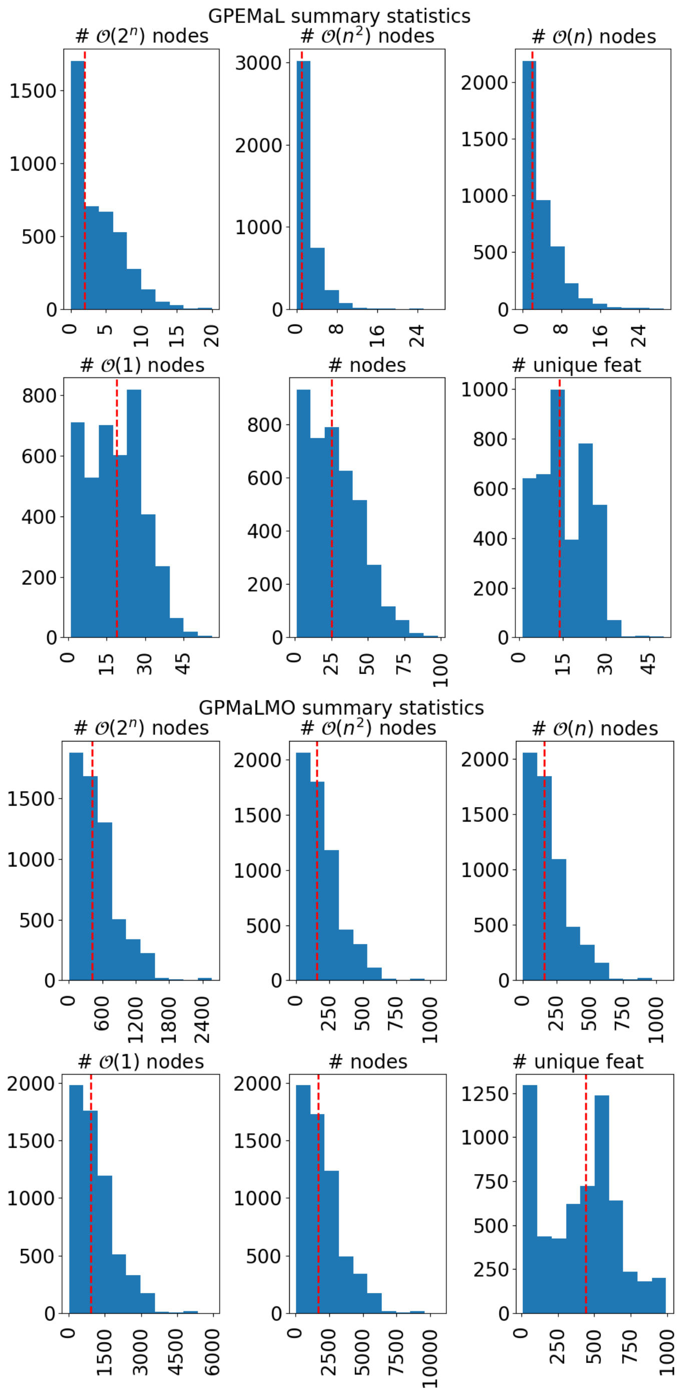


Fig. 17: Summary statistics for the two GP methods across all datasets. The red line represents the median result.

C. Visual Comparisons

Finally, we present a typical individual evolved by GP-EMaL (Fig. 18) and GP-MaL-MO (Fig. 19) on the complex COIL20 dataset[3]. We chose the individual with median classification accuracy across all runs to represent the "elbow" of the approximated Pareto Front. While GP-EMaL performs slightly worse than GP-MaL-MO on classification accuracy (mirroring our earlier results), the individual itself is markedly less complex: the biggest tree produced by GP-EMaL has seven nodes, compared to 31 in GP-MaL-MO. The deepest tree in GP-EMaL is only a depth of four; GP-MaL-MO produces a tree with a depth of ten.

While GP-MaL-MO may be technically *interpretable* to a domain expert, it is clearly very difficult to *explain* compared to the individuals produced by GP-EMaL. The trees evolved by GP-EMaL consistently exhibit simpler and more balanced structures due to the explicit optimization of tree complexity. Complex operators such as sigmoid and ReLU predominantly appear at the leaf nodes, while simpler, more intuitive operators like subtraction and multiplication occur nearer the

[3]Further sample tree outputs are available at github.com/cravies/GP-EMaL

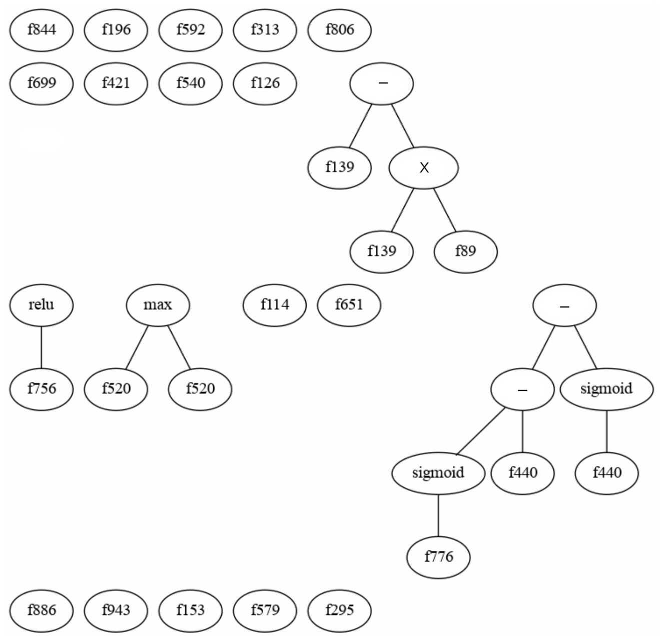


Fig. 18: A typical GP-EMaL individual containing 20 trees on COIL20, with 94.2% test accuracy and a complexity of 78.

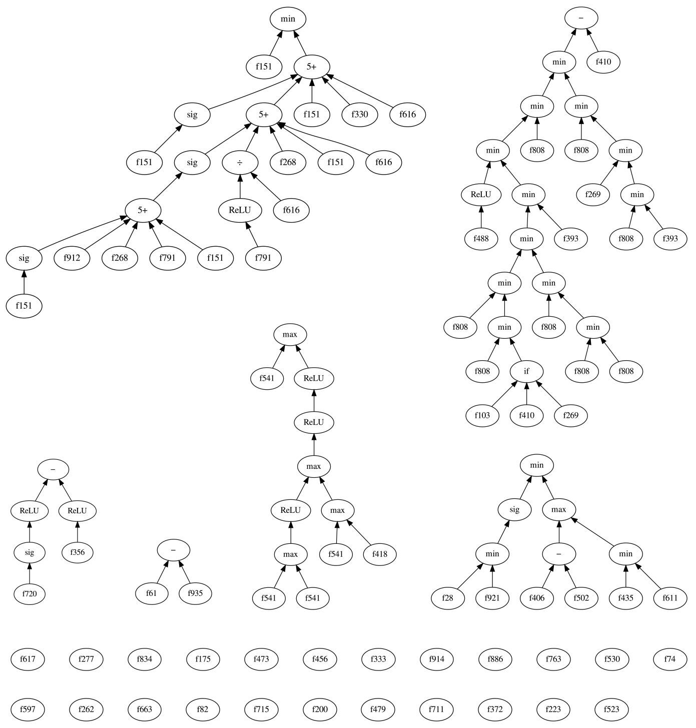


Fig. 19: A typical GP-MaL-MO individual containing 29 trees on COIL20, with 99.1% test accuracy.

root. This hierarchical organization naturally aligns with how domain experts tend to interpret functional mappings.

These observed structural advantages directly reflect GP-EMaL's algorithmic design choices, particularly its novel complexity metric incorporating penalties for tree size, asymmetry, and operator complexity. By explicitly optimizing tree complexity alongside manifold preservation, GP-EMaL consistently generates models that are inherently easier for domain experts to interpret without significantly sacrificing embedding quality. However, interpretability ultimately remains domain- and user-dependent; future work should involve domain experts to empirically validate and refine these interpretability gains in practical scenarios.

VI. Conclusions

This study's exploration into GP-EMaL represents a significant stride in the realm of interpretable manifold learning using genetic programming. GP-EMaL distinguishes itself by its emphasis on interpretability, a critical aspect often overlooked in traditional approaches. Notably, while GP-EMaL demonstrates a slight decrease in performance on higher-complexity datasets, this reduction is generally modest, typically under 5%. This is a small price to pay considering the complexity of previous methods, which were so intricate that they were impractical for tasks where interpretability is paramount.

Furthermore, our findings underline the delicate balance between interpretability and performance in machine learning models. The open-source availability of GP-EMaL is intended to foster more research and practical applications in this field. Future work could validate interpretability with user testing and further explore methods to enhance embedding quality. Furthermore, expanding GP-EMaL to optimise three objectives — neighbourhood structure preservation, embedding complexity, and dimensionality — simultaneously would offer a more nuanced approach, allowing for solutions tailored to specific requirements and contexts.

References

[1] J. Birjandtalab, M. B. Pouyan, and M. Nourani, "Nonlinear dimension reduction for eeg-based epileptic seizure detection," in *2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, 2016, pp. 595–598.

[2] D. Singh, H. Climente-Gonzalez, M. Petrovich, E. Kawakami, and M. Yamada, "Fsnet: Feature selection network on high-dimensional biological data," in *2023 International Joint Conference on Neural Networks (IJCNN)*, 2023, pp. 1–9.

[3] S. Galelli and A. Castelletti, "Tree-based iterative input variable selection for hydrological modeling," *Water Resources Research*, vol. 49, no. 7, pp. 4295–4310, 2013.

[4] A. Gracia, S. González, V. Robles, and E. Menasalvas, "A methodology to compare dimensionality reduction algorithms in terms of loss of quality," *Information Sciences*, vol. 270, pp. 1–27, 2014.

[5] European Parliament and Council of the European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council. [Online]. Available: https://data.europa.eu/eli/reg/2016/679/oj

[6] L. Longo, M. Brcic, F. Cabitza, J. Choi, R. Confalonieri, J. D. Ser, R. Guidotti, Y. Hayashi, F. Herrera, A. Holzinger, R. Jiang, H. Khosravi, F. Lecue, G. Malgieri, A. Páez, W. Samek, J. Schneider, T. Speith, and S. Stumpf, "Explainable artificial intelligence (XAI) 2.0: A manifesto of open challenges and interdisciplinary research directions," *Information Fusion*, vol. 106, p. 102301, 2024.

[7] S. Ali, T. Abuhmed, S. El-Sappagh, K. Muhammad, J. M. Alonso-Moral, R. Confalonieri, R. Guidotti, J. Del Ser, N. Díaz-Rodríguez, and F. Herrera, "Explainable artificial intelligence (XAI): What we know and what is left to attain trustworthy artificial intelligence," *Information Fusion*, vol. 99, p. 101805, 2023.

[8] M. A. Ahmad, C. Eckert, and A. Teredesai, "Interpretable machine learning in healthcare," in *Proceedings of the 2018 ACM international conference on bioinformatics, computational biology, and health informatics*, 2018, pp. 559–560.

[9] P. Maddigan and T. Susnjak, "Forecasting patient demand at urgent care clinics using machine learning," *arXiv preprint arXiv:2205.13067*, 2022.

[10] M. Lin, L. Shang, and X. Gao, "Enhancing interpretability in ai-generated image detection with genetic programming," in *IEEE International Conference on Data Mining (ICDM)*. IEEE, 2023, pp. 371–378.

[11] G. Nadizar, E. Medvet, and D. G. Wilson, "Naturally interpretable control policies via graph-based genetic programming," in *Proceedings of the European Conference on Genetic Programming (EuroGP), Lecture Notes in Computer Science*, ser. Lecture Notes in Computer Science, vol. 14631. Springer, 2024, pp. 73–89.
[12] Y. Mei, Q. Chen, A. Lensen, B. Xue, and M. Zhang, "Explainable artificial intelligence by genetic programming: A survey," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 3, pp. 621–641, 2023.
[13] T. Hu, "Genetic programming for interpretable and explainable machine learning," in *Genetic Programming Theory and Practice XIX (GPTP)*. Springer, 2023, pp. 81–90.
[14] A. Lensen, B. Xue, and M. Zhang, "Can genetic programming do manifold learning too?" in *Proceedings of the European Conference on Genetic Programming (EuroGP), Lecture Notes in Computer Science*. Springer International Publishing, 2019, vol. 11451, pp. 114–130.
[15] A. Lensen, M. Zhang, and B. Xue, "Multi-objective genetic programming for manifold learning: balancing quality and dimensionality," *Genetic Programming and Evolvable Machines*, vol. 21, pp. 399–431, 2020.
[16] P. Maddigan, A. Lensen, and B. Xue, "Explaining genetic programming trees using large language models," *arXiv preprint arXiv:2403.03397*, 2024.
[17] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, "Machine learning interpretability: A survey on methods and metrics," *Electronics*, vol. 8, no. 8, 2019.
[18] C. Singh, J. P. Inala, M. Galley, R. Caruana, and J. Gao, "Rethinking interpretability in the era of large language models," *arXiv preprint arXiv:2402.01761*, 2024.
[19] C. Molnar, G. Casalicchio, and B. Bischl, "Interpretable machine learning – a brief history, state-of-the-art and challenges," in *ECML PKDD 2020 Workshops*. Cham: Springer International Publishing, 2020, pp. 417–431.
[20] P. Domingos, "A few useful things to know about machine learning," *Communications of the ACM*, vol. 55, no. 10, pp. 78–87, 2012.
[21] J. Huang, W. Qian, C.-M. Vong, W. Ding, W. Shu, and Q. Huang, "Multi-label feature selection via label enhancement and analytic hierarchy process," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 7, no. 5, pp. 1377–1393, 2023.
[22] R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari, and J. Saeed, "A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction," *Journal of Applied Science and Technology Trends*, vol. 1, no. 2, pp. 56 – 70, May 2020.
[23] L. Van Der Maaten, E. O. Postma, H. J. van den Herik *et al.*, "Dimensionality reduction: A comparative review," *Journal of Machine Learning Research*, vol. 10, no. 66-71, p. 13, 2009.
[24] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," 2020.
[25] G. E. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, pp. 504 – 507, 2006.
[26] L. van der Maaten and G. E. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.
[27] M. Wattenberg, F. Viégas, and I. Johnson, "How to use t-sne effectively," *Distill*, 2016.
[28] J. Dong, J. Zhong, W.-N. Chen, and J. Zhang, "An efficient federated genetic programming framework for symbolic regression," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 7, no. 3, pp. 858–871, 2023.
[29] R. Poli, W. B. Langdon, and N. F. McPhee, *A Field Guide to Genetic Programming*. lulu.com, 2008.
[30] B. Al-Helali, Q. Chen, B. Xue, and M. Zhang, "Genetic programming for feature selection based on feature removal impact in high-dimensional symbolic regression," *IEEE Transactions on Emerging Topics in Computational Intelligence*, pp. 1–14, 2024, early Access.
[31] C. A. C. Coello, "Evolutionary multi-objective optimization: a historical view of the field," *IEEE Comput. Intell. Mag.*, vol. 1, no. 1, pp. 28–36, 2006.
[32] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, 2007.
[33] M. Kommenda, G. Kronberger, M. Affenzeller, S. M. Winkler, and B. Burlacu, *Evolving Simple Symbolic Regression Models by Multi-Objective Genetic Programming*. Springer International Publishing, 2016, ch. Genetic Programming Theory and Practice XIII, pp. 1–19.
[34] A. S. Sambo, R. M. A. Azad, Y. Kovalchuk, V. P. Indramohan, and H. Shah, "Time control or size control? reducing complexity and improving the accuracy of genetic programming models," 2020.
[35] N. Le, H. N. Xuan, A. Brabazon, and T. P. Thi, "Complexity measures in genetic programming learning: A brief review," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, 2016, pp. 2409–2416.
[36] J. Chadalawada, V. Havlicek, and V. Babovic, "A genetic programming approach to system identification of rainfall-runoff models," *Water Resour Manage*, vol. 31, p. 3975–3992, 2017.
[37] G. Campobello, D. Dell'Aquila, M. Russo, and A. Segreto, "Neuro-genetic programming for multigenre classification of music content," *Applied Soft Computing*, vol. 94, p. 106488, 2020.
[38] Y. Tang, H. Jia, and N. Verma, "Reducing energy of approximate feature extraction in heterogeneous architectures for sensor inference via energy-aware genetic programming," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 5, pp. 1576–1587, 2020.
[39] T. Soule and J. A. Foster, "Effects of code growth and parsimony pressure on populations in genetic programming," *Evolutionary Computation*, vol. 6, pp. 293–309, 1998.
[40] G. F. Smits and M. Kotanchek, *Pareto-Front Exploitation in Symbolic Regression*. Springer US, 2005, pp. 283–299.
[41] J.Ni and P.Rockett, "Tikhonov regularization as a complexity measure in multiobjective genetic programming," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 157–166, 2015.
[42] D. Dheeru and E. Karra Taniskidou, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml
[43] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (coil-20)," Columbia University, Tech. Rep., 1996.

Ben Cravens received the B.Sc.(Hons) degree in physics from Otago University, New Zealand, in 2019. He is currently employed in the computer vision and machine learning industry. His research interests include computer vision, explainable AI, and ethical AI in New Zealand.

Andrew Lensen (Member, IEEE) received the B.Sc.(Hons), and Ph.D. degrees in computer science from Victoria University of Wellington, New Zealand, in 2016, and 2019, respectively. He is currently a Senior Lecturer in Artificial Intelligence at the School of Engineering and Computer Science at Victoria University of Wellington.
His current research interests include explainable AI, genetic programming, and interdisciplinary and ethical AI in New Zealand.

Paula Maddigan received the B.Sc. degree (Hons.) in statistics and operations research from Victoria University of Wellington, New Zealand, and the Master of Analytics degree in business from Massey University, New Zealand, in 2022. She is currently pursuing the Ph.D. degree in artificial intelligence. She also has extensive industry experience as a Software Engineer. Her research interests include explainable AI, computer vision, and applied machine learning.

Bing Xue (Fellow, IEEE) received the Ph.D. degree in computer science from the Victoria University of Wellington, New Zealand, in 2014. She is currently a Professor in the School of Engineering and Computer Science, Victoria University of Wellington. She has more than 400 papers published in fully refereed international journals and conferences. Her research focuses mainly on evolutionary computation, feature learning, and machine learning. Prof. Xue has also served as the Associate Editor or a Member of the Editorial Board of ten international journals, including IEEE Transactions on Evolutionary Computation, IEEE Trasnactions on Artificial Intelligence, and IEEE Computational Intelligence Magazine. She is a Fellow with Engineering New Zealand.