



ARTICLE

A Multi-Tree Genetic Programming Framework for Multi-View Feature Construction and Fusion

Dongmin Zhang^{1,#}, Chao Sun^{1,#}, Yikun Zhang^{2,*}, Runyao Yin² and Chen Chen¹

¹Southwest China Institute of Electronic Technology, Chengdu, China

²School of Future Science and Engineering, Soochow University, Suzhou, China

*Corresponding Author: Yikun Zhang. Email: 20265258040@stu.suda.edu.cn

#These authors contributed equally to this work and share first authorship

Received: 02 June 2026; Accepted: 28 July 2026; Published: 22 October 2026

ABSTRACT: Multi-view tabular data poses challenges due to heterogeneous feature spaces, divergent distributions, and implicit cross-view relationships. Traditional methods further struggle because their hand-crafted fusion strategies cannot adequately capture complex nonlinear interactions among views. To address this issue, this paper proposes a genetic programming (GP)-based multi-view feature fusion method. It employs a multi-tree GP framework: each view is assigned a dedicated tree for intra-view feature selection and construction, and a fusion tree combines their outputs for cross-view feature-level fusion. An enhanced feature construction strategy further enriches the final representation by exploiting subtree information. Together with decision-level fusion, GP-based multi-view feature fusion framework (GPMVF) forms a two-level fusion framework that jointly handles intra-view construction, cross-view fusion, and classification. Experiments on eight public multi-view datasets show that the proposed method achieves better performance than the compared baseline method.

KEYWORDS: Genetic programming; multi-view learning; feature construction; feature fusion

1 Introduction

Multi-view data are widely encountered in many real-world tasks [1]. For example, in antenna design, radiation performance can be described by multiple complementary views: near-field distribution, far-field pattern, and impedance characteristics. Jointly analyzing these views enables more accurate and robust electromagnetic performance assessment. In such data, the same sample can be described by multiple groups of features collected from different sources, modalities, or feature construction processes [2]. Multi-view learning has shown important practical value in a variety of application scenarios, such as intelligent diagnosis based on fused tongue-image features [3]. It has also been successfully applied to facial micro-expression recognition from different perspectives [4]. In addition, multi-view learning has played an important role in visual understanding tasks such as multi-view 3D object recognition [5]. Compared with single-view learning, multi-view learning benefits not only from richer descriptive information, but also from the ability to jointly exploit the consistency and complementarity across views [1]. Its advantage stems not merely from additional data, but from how different views collaborate to enhance discrimination [6].

However, effectively utilizing multi-view information remains a challenging problem [7]. This issue is particularly evident in multi-view tabular data, where different views may exhibit heterogeneity in feature semantics, dimensionality, and statistical distributions, and where the number of samples is often limited [8].

In such settings, some views may contain high-dimensional but weakly informative features, while others may provide relatively compact yet more discriminative representations. The imbalance in view quality and scale makes it difficult to directly apply conventional multi-view strategies in a stable and effective way [9]. At the same time, the relationships across views in tabular data are often implicit rather than structurally aligned, which makes cross-view interactions harder to model. Under such conditions, a learning model must preserve useful view-specific characteristics while also capturing latent interactions across views [10]. The performance of the multi-view method is highly sensitive to how cross-view relationships are modeled [11]. Common strategies for handling multi-view information span representation learning, feature-level fusion, and decision-level fusion [7], with hybrid approaches that integrate feature-level and decision-level fusion also receiving attention [12]. Nevertheless, many existing methods still rely heavily on manually specified fusion structures or carefully designed combination rules, which may limit their adaptability across datasets with different view settings.

To reduce the reliance on manually designed fusion rules and better handle the heterogeneity of multi-view tabular data, a more adaptive feature construction mechanism is needed. Genetic programming (GP) provides such a possibility, as it can evolve tree-structured programs to construct nonlinear feature transformation expressions directly from raw features [13]. This property makes GP particularly suitable for tabular classification tasks, where informative patterns may arise from complex feature interactions that are difficult to specify in advance. Prior studies have shown that GP can effectively construct discriminative features for classification and can even evolve multiple constructed features within a single individual to enrich the representation space [14].

Nevertheless, most existing GP-based studies are still centered on single-view learning. Although multi-tree GP has shown advantages in handling multiple outputs, modular subtasks, and heterogeneous information sources [15], related work in the multi-view context is still scarce [16]. As a result, the potential of GP for multi-view tabular feature fusion has not been fully explored. In addition, many traditional GP-based feature construction strategies mainly use the output of the root node as the final constructed feature. Although this design is simple and intuitive, it may ignore the hierarchical information contained in internal subtrees. For multi-view data, this limitation may become more prominent, because informative transformations may emerge not only within each view but also at different structural levels of the evolved trees.

To address the above issues, this paper proposes a GP-based multi-view feature fusion framework (GPMVF). In the proposed framework, each view is assigned a dedicated view-specific tree to perform intra-view feature construction, while an additional fusion tree is introduced to model nonlinear interactions among different views. Furthermore, an enhanced feature construction strategy is designed to integrate the root output, informative subtree-constructed features, and the original terminal features actually involved in the construction process, so as to obtain a richer and more discriminative representation. On this basis, the framework further combines feature-level fusion with decision-level fusion based on support vector machine (SVM) classifiers to improve the effectiveness and robustness of multi-view classification.

The main contributions of this paper can be summarized as follows:

1. A multi-tree GP-based framework is adapted and integrated for multi-view tabular classification, where view-specific trees are used for intra-view feature construction and a fusion tree is employed to model latent inter-view relationships.
2. Within this framework, an enhanced feature representation strategy is adopted to exploit not only the root outputs of GP trees, but also informative subtree-level constructed features and the original terminal features involved in the construction process, thereby enriching the final feature representation.

3. A dual-layer fusion architecture is established for multi-view classification, in which feature-level fusion is achieved by the fusion tree and decision-level fusion is further performed on the outputs of multiple classifiers.

The remainder of this paper is organized as follows. [Section 2](#) reviews the related work on multi-view learning, GP, and multi-tree GP. [Section 3](#) presents the proposed GPMVF framework. [Section 4](#) describes the experimental settings. [Section 5](#) reports and discusses the experimental results from multiple perspectives. Finally, [Section 6](#) concludes this paper and outlines possible directions for future work.

2 Related Work

2.1 Multi-View Learning

Multi-view learning jointly models multiple feature representations of the same instances, exploiting cross-view consistency and complementarity to achieve better performance than any single view alone. Early studies mainly focused on how different views could provide complementary information for one another during the learning process. A representative example is the co-training framework, which was proposed to improve learning performance by leveraging complementary supervision from different views [17]. In unsupervised settings, correlation-based modeling ideas were further introduced into multi-view learning. For example, canonical correlation analysis was incorporated into multi-view clustering to capture the consistency across different views through their correlated structures [18]. Later, graph-based methods were also explored, and a cross-view graph diffusion strategy was proposed to strengthen information interaction among different views through graph propagation [19].

With the continued development of multi-view learning, research focus gradually shifted from shallow cross-view alignment to richer representation learning and latent-space modeling. In this line of work, the key objective is usually to improve the consistency and separability of learned representations across views. For example, a multi-view latent-space fusion representation method based on deep feature enhancement was proposed, showing that joint hidden-space modeling can improve the quality of multi-view representations [20]. More recent reviews have pointed out that multi-view learning is evolving from traditional shallow fusion toward more complex joint representation learning frameworks [21]. The role of self-supervision and contrastive alignment in deep multi-view clustering was further investigated, reflecting the trend of integrating multi-view learning with deeper representation learning frameworks [22]. In addition, Transformer-based mechanisms were incorporated into deep contrastive multi-view clustering, which suggests that attention modeling and deep feature fusion have become important directions in recent research [23].

Besides representation learning, fusion-based learning is another important branch of multi-view research. From the perspective of information integration, existing methods can generally be divided into feature-level fusion, decision-level fusion, and hybrid fusion [7]. Feature-level fusion usually combines information from multiple views before classification, which is helpful for mining cross-view correlations in the feature space. Decision-level fusion trains classifiers for different views separately and then combines their outputs, which provides a relatively flexible way to utilize heterogeneous views. Hybrid fusion attempts to combine the advantages of both paradigms. For example, a hierarchical classifier was constructed by combining feature-level and decision-level fusion, indicating that multi-level fusion is a meaningful direction for improving learning performance [12].

Although the above studies have substantially advanced multi-view learning, some limitations still remain when they are applied to heterogeneous tabular data. As pointed out in the existing literature, constructing a stable and effective fusion mechanism is still a major challenge in multi-view classification [7].

In many cases, existing methods still rely on manually designed fusion structures, predefined combination rules, or carefully specified learning objectives. As a result, they are often not sufficiently flexible for datasets with large differences in view number, feature dimensionality, and distribution characteristics. More importantly, these methods do not directly focus on automatic nonlinear feature construction from raw multi-view tabular features. This limitation motivates the development of a more automated framework that can simultaneously support intra-view feature construction and inter-view fusion, which is one of the main goals of this paper.

2.2 GP for Feature Construction and Classification

GP has long been regarded as an important approach for automatic feature construction. In the foundational work on GP, candidate solutions were represented as tree-structured programs and evolved through population-based search, which provides a natural mechanism for automatically constructing new expressions from primitive operators and original variables [13]. Subsequent studies further extended the capability of GP in automatically discovering reusable structures, thereby strengthening its ability to generate more complex feature expressions [24]. Early applications of GP to image analysis also illustrated its potential for automatic extraction of complex features [25]. In addition, strongly typed GP was introduced to enforce type consistency during evolution, providing an important foundation for later constrained GP designs when operators and data structures need to be matched semantically [26]. Owing to its symbolic tree representation, GP is particularly suitable for tasks in which nonlinear feature transformations need to be learned directly from data rather than specified manually.

Recent advances have extended GP-based feature construction beyond single-tree representations to multi-tree architectures, where multiple program trees are evolved within a single individual. For example, an adaptive multi-tree GP approach that automatically adjusts the number of trees based on fitness feedback and the current evolutionary state [15] was designed, to address a key limitation of existing multi-tree GP methods where the number of trees per individual needs to be predetermined, which can reduce parameter tuning effort while improving adaptability across datasets. Beyond multi-tree architectures, multi-population strategies have emerged as a promising direction for handling heterogeneous or multi-view data. A multi-population ensemble GP framework that integrates cooperative coevolution with multi-view learning [27] was proposed, where the feature space is decomposed into multiple subsets, and each population evolves independently on a distinct view while interacting through an ensemble-based fitness mechanism. This approach enhances diversity, reduces premature convergence, and improves classification accuracy by processing disparate data views in parallel.

Although these studies have greatly advanced GP-based feature construction, most of them are still mainly oriented toward single-view learning scenarios. Existing work has demonstrated the ability of GP to perform nonlinear feature transformation, multiple-feature construction, and representation extension in specific domains, but it does not yet directly address the problem of jointly modeling view-specific information and cross-view interactions in multi-view tabular data. Therefore, there is still a need for a more specialized framework that can combine the automatic feature construction ability of GP with the fusion requirements of heterogeneous multi-view data, which forms an important motivation for the present work.

2.3 Multi-Tree GP for Multi-Source Learning

Multi-tree GP can be regarded as an important extension of standard single-tree GP. By allowing one individual to contain multiple trees, this framework provides a more flexible way to handle tasks involving multiple constructed features, modular subtasks, or heterogeneous information sources. An adaptive multi-tree GP method was proposed for the collaborative construction of multiple new features,

showing that the multi-tree structure has clear advantages in complex feature learning [15]. Related studies on multidimensional GP representations further suggest that multi-tree or multidimensional structures can improve the expressive ability of learned features in classification tasks [28].

Multi-tree GP has also been explored in a wider range of supervised learning problems. A new multi-tree GP approach was proposed for feature construction in high-dimensional classification, verifying the effectiveness of the multi-tree framework in complex high-dimensional data scenarios [29]. A dual-population GP approach with active meta-learning [30] was designed for ontology matching, which co-evolves two-layer similarity features, actively refines training labels, and meta-learns expert aggregation, outperforming state-of-the-art methods on OAEI benchmarks. An adaptive particle swarm optimization framework that integrates GP [31] was proposed, to dynamically refine the fitness function and velocity update mechanism for automating similarity feature combination in knowledge graph alignment, which achieves superior matching accuracy and efficiency across diverse alignment tasks. These studies indicate that, compared with standard single-tree representations, multi-tree designs are better suited to problems that require multiple coordinated transformation processes. More recently, multi-tree GP was further shown to support not only feature learning but also effective feature fusion in multimodal medical image classification [16].

Although these studies have demonstrated the potential of multi-tree GP in multi-source learning, existing work still focuses mainly on images, signals, and other domain-specific data. In contrast, research on multi-view tabular data remains relatively limited. In particular, current studies do not yet provide a unified framework that can simultaneously support intra-view feature construction and inter-view feature fusion for heterogeneous tabular views. Therefore, there is still a need for a more specialized multi-tree GP framework to address the characteristics of multi-view tabular classification, which directly motivates the method proposed in this paper.

3 Proposed Method

3.1 Overall Algorithm

The proposed method follows the standard GP evolutionary process, including population initialization, individual evaluation, selection, crossover, mutation, and population updating. Unlike conventional single-tree GP, each individual in the proposed method comprises multiple view-specific trees and one view-fusion tree. The evolutionary process therefore performs both intra-view feature construction and inter-view feature fusion within a unified framework. During initialization, an expression space is defined for each view based on the number of views in the dataset, along with the corresponding function set, terminal set, and depth constraints, ensuring that each individual has a complete multi-tree structure.

During evaluation, the original features of each view are first transformed by the corresponding view-specific tree, which applies the enhanced feature construction strategy to produce an enriched view-dependent representation. The resulting representations are then fed into the view-fusion tree to generate a fused cross-view representation for classification and fitness evaluation. During evolution, parent individuals are selected, offspring are generated through genetic operations, and elite individuals are retained when the population is updated. To preserve the semantic roles of the trees within the multi-tree structure, crossover is performed only between subtrees at the same position, while both crossover and mutation are subject to structural and depth constraints to maintain individual validity and stability. This process is repeated until the termination condition is met, after which the best individual in the final population is used for feature construction and classification on the test set. Fig. 1 illustrates the overall workflow of the proposed algorithm.

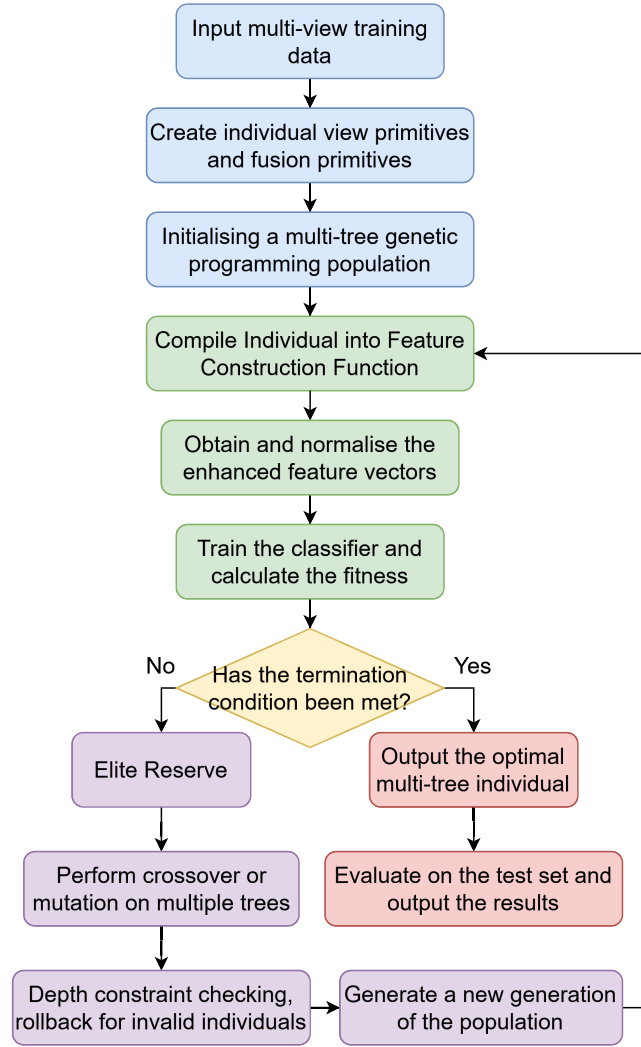


Figure 1: Overall workflow of the proposed multi-tree GP algorithm.

The main innovations of the proposed method are its multi-tree representation and multi-view fusion mechanism, which are described in detail in the following subsections.

3.2 Multi-Tree GP Representation

To jointly model intra-view discriminative information and inter-view complementary information in multi-view tabular data, we design a multi-tree GP individual. Unlike a conventional single-tree GP representation, each individual in the proposed framework comprises multiple view-specific trees and one view-fusion tree. This structure enables intra-view nonlinear feature construction and cross-view feature-level fusion within a unified evolutionary framework [16].

Let a multi-view sample be represented as $X = \{x^{(1)}, x^{(2)}, \dots, x^{(v)}\}$, where v denotes the number of views. For each view, a dedicated view-specific tree is used to evolve nonlinear representations from the original feature space. A view-fusion tree is then introduced to take the outputs of all view-specific trees as input and learn the relationships among the different views. This individual design therefore provides a clear functional division between intra-view feature transformation and inter-view feature fusion.

Fig. 2 illustrates a GP individual for a two-view dataset, in which each individual comprises two view-specific trees and one view-fusion tree.

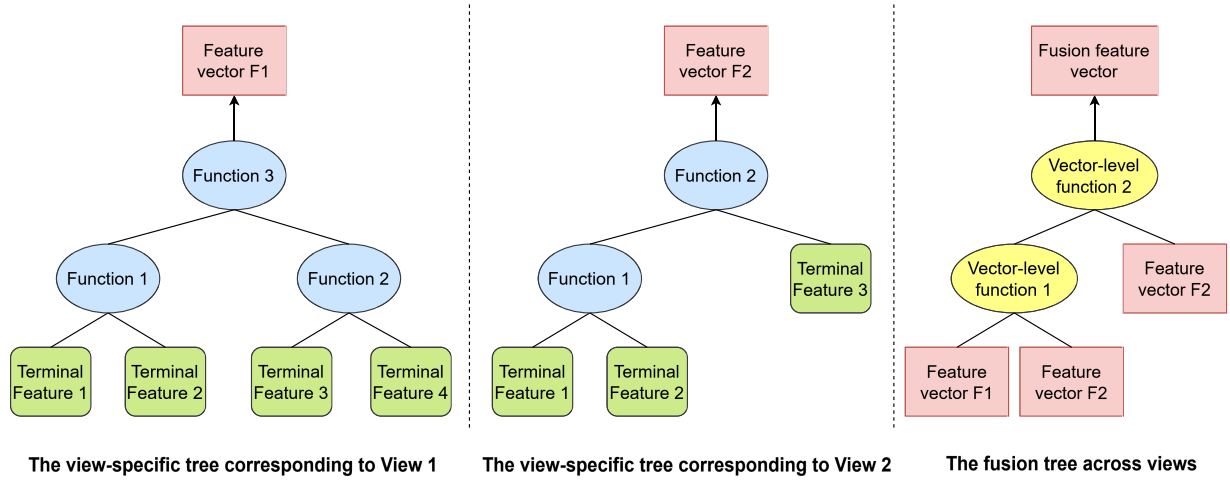


Figure 2: An example of a multi-tree GP individual.

3.2.1 View-Specific Tree

A view-specific tree performs nonlinear transformations on the original features of a single view and serves as the basic component of a multi-tree GP individual. For the i -th view in a multi-view dataset, an independent program tree is assigned to that view. By recursively combining function and terminal nodes, the tree automatically constructs new feature representations for the corresponding view.

Because different views may vary in data distribution, feature dimensionality, and discriminative information, the “one view, one tree” strategy helps preserve the structural characteristics of each view and improves the model’s adaptability to heterogeneous views. Each view-specific tree receives only the features of its corresponding view and cannot directly access information from other views. Its primary role is to perform intra-view feature transformation and evolve more discriminative high-level representations by capturing latent feature relationships through combination, reconstruction, and transformation. This process reduces the model’s dependence on manually designed features.

The view-specific trees operate in parallel within the same individual and collaborate under the guidance of the overall fitness. Rather than optimizing its corresponding view in isolation, each tree contributes jointly with the other view-specific trees to the overall multi-view fusion task.

The view-specific tree primarily performs scalar feature operations. Its function and terminal sets are listed in Table 1.

Table 1: Function set and terminal set of the view-specific tree.

Type	Name	Arity	Description
Binary operator	Add	2	Addition of two input features
Binary operator	Subtract	2	Subtraction of two input features
Binary operator	Multiply	2	Multiplication of two input features
Binary operator	Divide	2	Protected division; returns 1 when the divisor is close to 0

(Continued)

Table 1 (continued)

Type	Name	Arity	Description
Unary operator	Negation	1	Negation of the input feature
Unary operator	Absolute value	1	Absolute value transformation of the input feature
Unary operator	Sine transformation	1	Sine transformation of the input feature
Unary operator	Cosine transformation	1	Cosine transformation of the input feature
Terminal set	Feature variable	–	Original input features, where d is the dimensionality of the input features
Constant term	Random constant	–	Random constants in the interval from –1 to 1

3.2.2 View-Fusion Tree

After the view-specific trees construct features for their respective views, a view-fusion tree is used to perform cross-view information interaction and feature-level fusion. The view-fusion tree occupies the upper level of the multi-tree individual. It takes as input the feature vectors generated by the view-specific trees rather than the original dataset features. After the intra-view nonlinear transformations, the fusion tree learns relationships among the views to form a higher-level joint feature representation.

The fusion tree uses functions designed for vector inputs. Vector-level operators, including element-wise addition, element-wise multiplication, maximum, and mean, automatically combine and transform the multi-view features. Compared with fixed concatenation or simple weighting, this program-tree-based fusion mechanism is more flexible and can better exploit the consistency and complementarity among views.

Because the feature vectors generated by different view-specific trees may have different lengths, a feature-vector length alignment function is introduced. This function uses the maximum length among the current input vectors as the target length and pads shorter vectors with trailing zeros, ensuring that all input vectors have consistent dimensions. The vector-level operators can then be applied normally. In addition, a type-constraint mechanism ensures consistent input and output types for all nodes, thereby preventing mismatched connections between structural units [26].

The function and terminal sets of the view-fusion tree are listed in Table 2.

Table 2: Function set and terminal set of the view-fusion tree.

Type	Name	Input Form	Description
Vector fusion operator	Add	i vectors	Element-wise addition of the input vectors
Vector fusion operator	Multiply	i vectors	Element-wise multiplication of the input vectors
Vector fusion operator	Maximum	i vectors	Element-wise maximum of the input vectors
Vector fusion operator	Mean	i vectors	Element-wise mean of the input vectors
Terminal set	–	–	$V_1, V_2, \dots, V_v$, feature vectors generated by the view-specific trees, where v is the number of views

3.3 Enhanced Feature Selection and Construction Strategy Based on GP Subtrees

To better exploit the hierarchical construction information encoded in GP trees, we adopt an enhanced feature construction strategy based on subtree-constructed features. Rather than using only the final feature constructed at the root node, the proposed method also incorporates informative subtree features and the original terminal features used in the tree, thereby producing a richer feature representation.

Let T denote a program tree. Let $f_T(x)$ denote the final feature constructed at its root node, $S_1(x), S_2(x), \dots, S_m(x)$ denote the features constructed by its internal subtrees, and $x_0, x_1, \dots, x_k$ denote the terminal features used in the tree. The enhanced feature vector V can then be expressed as

$$V = [f_T(x), S_1(x), S_2(x), \dots, S_m(x), x_0, x_1, \dots, x_k]. \quad (1)$$

In implementation, these components are obtained by traversing the string representation of the evolved program tree and evaluating the whole-tree expression, subtree expressions, and used terminal features for the current sample. To maintain semantic consistency across trees and views, the final constructed feature is always placed first in the vector rather than being concatenated solely according to the traversal order. This arrangement helps the subsequent fusion tree use the main constructed feature more consistently.

Fig. 3 illustrates the example tree $\text{Mul}(\text{Add}(x_0, x_1), \text{Neg}(x_2))$. Under the enhanced feature construction strategy, this tree generates the feature vector $[f, s_1, s_2, x_0, x_1, x_2]$. Compared with the conventional approach that uses only the root output, this strategy preserves construction results from multiple levels and provides a richer feature representation.

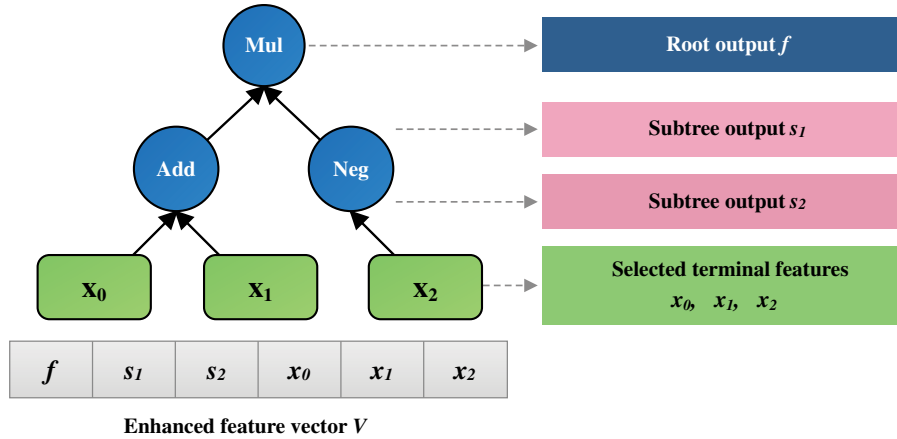


Figure 3: Illustration of enhanced feature construction from a GP tree.

3.4 Dual-Layer Fusion Architecture for Multi-View Learning

3.4.1 Feature-Level Fusion Mechanism

At the feature level, the view-fusion tree jointly models information from multiple views. The view-specific trees first apply nonlinear mappings to the original features of their corresponding views to generate intra-view constructed features. The view-fusion tree then takes the output vectors of these trees as input and uses vector-level operators to learn the relationships among the view representations, thereby forming a joint cross-view feature representation. Because this process occurs during feature construction, it constitutes feature-level fusion.

Unlike the direct concatenation of features from different views, the fusion tree can adaptively search for an effective cross-view fusion form during genetic evolution. It can therefore better exploit the consistency

and complementarity of multi-view data and provide a more discriminative feature basis for subsequent decision-level fusion.

3.4.2 Decision-Level Fusion Mechanism

After feature-level fusion, decision-level fusion is introduced during classification to make fuller use of the multi-view information. Specifically, classifiers are trained on both the enhanced feature vectors constructed by the view-specific trees and the joint feature representation generated by the fusion tree, after which their predictions are integrated. Consequently, both the local discriminative information from individual views and the global information in the fused features contribute to the final classification decision, forming a dual-layer mechanism that combines feature-level and decision-level fusion.

Three decision-level fusion strategies are considered: voting, weight, and single.

The voting strategy combines the predictions of multiple classifiers through majority voting. All classifiers have equal voting status, and each contributes one vote to the final class. The class receiving the most votes is selected as the final prediction. If multiple classes receive the same number of votes, a predefined rule is applied to resolve the tie. This strategy requires no additional parameter settings and is straightforward to implement. In multi-view tasks, it integrates complementary information from different view classifiers and reduces the influence of unstable predictions from an individual view by aggregating multiple decisions. Because the strategy combines discrete class labels produced by the classifiers, it constitutes a hard-decision fusion mechanism.

The weight strategy combines the predictions of different classifiers through weighted fusion. All classifiers associated with the view-specific trees and the fusion-tree classifier are assigned equal weights, and the final prediction is obtained by averaging their posterior probabilities. For a dataset with V views, $V + 1$ classifiers participate in the fusion, each with a weight of $1/(V + 1)$. This strategy provides a simple and stable soft-decision fusion mechanism without introducing randomness into the weight assignment. Unlike the voting strategy, which combines discrete class labels, the weight strategy uses the probabilistic outputs of all classifiers and therefore preserves more decision information during fusion.

The single strategy does not further integrate the predictions of multiple classifiers; only the prediction of the classifier trained on the fusion-tree features is used as the final output. This strategy serves as a reference setting for evaluating classification performance when feature-level fusion is applied without subsequent decision-level integration across views. Comparing single with voting and weight therefore reveals the contribution of decision-level fusion to the overall model performance.

Together, the three strategies form the decision-level component of the dual-layer fusion mechanism. Voting implements hard-decision fusion through majority voting, weight implements soft-decision fusion through equal-weight probabilistic integration, and single provides a reference for assessing the gain achieved by decision-level fusion.

4 Experimental Settings

4.1 Datasets

To evaluate GPMVF on multi-view tabular classification tasks, experiments are conducted on eight representative public datasets spanning scene recognition, object recognition, handwritten digit recognition, bioinformatics, face recognition, and plant classification. The datasets vary in the number of views, samples, and classes, as well as in the dimensionality of each view, providing diverse settings for evaluating multi-view feature construction and fusion.

Table 3 summarizes the key characteristics of the eight datasets, including their application domains, numbers of views, classes, and samples, and the dimensionality of each view.

Table 3: Summary of the multi-view datasets used in the experiments.

Dataset	Type	Views	Classes	Samples	Feature Dimensions
Scene15	Scene image classification	3	15	4485	20, 59, 40
ALOI	Object image classification	2	100	10,800	77, 64
HandWritten	Handwritten digit recognition	3	10	2000	240, 64, 216
BDGP	Bioinformatics classification	2	5	2500	1750, 79
Prokaryotic	Bioinformatics classification	3	4	551	393, 3, 438
yale_mtv	Face recognition	3	15	165	4096, 3304, 6750
100leaves	Plant leaf classification	3	100	1600	64, 64, 64
mfeat	Handwritten digit recognition	6	10	2000	216, 76, 64, 6, 240, 47

Note: ALOI, Amsterdam Library of Object Images; BDGP, Berkeley Drosophila Genome Project; mfeat, Multiple Features; yale_mtv, multi-view Yale Face Database.

4.2 Parameter Settings

The main evolutionary parameters are set as follows: the population size is 100, the number of generations is 50, the crossover probability is 0.8, the mutation probability is 0.19, the elitism ratio is 0.01, and the tournament size is 5. These settings follow [16]. To control structural complexity, the maximum depths of the view-specific trees and the fusion tree are set to 10 and 8, respectively. Classification performance is evaluated using stratified five-fold cross-validation.

4.3 Evaluation Metrics

Classification accuracy is used as the primary evaluation metric because it directly measures the proportion of correctly classified samples. It is defined as

$$\text{Accuracy} = \frac{N_c}{N}, \quad (2)$$

where N_c denotes the number of correctly classified samples and N denotes the total number of samples.

5 Results and Discussion

5.1 Comparison with Baseline Method

To evaluate GPMVF on multi-view classification tasks, we compare it with a representative baseline that trains an independent SVM classifier for each view and combines the predictions through voting. Because GPMVF involves stochastic evolutionary search, it is run repeatedly on each dataset, and its results are reported as means with standard deviations. The baseline is deterministic under the current setting and is therefore evaluated once per dataset.

Table 4 reports the accuracy, F1 score, and Kappa coefficient of the two methods. For each dataset, a two-sided exact Wilcoxon signed-rank test compares the repeated GPMVF results with the corresponding fixed baseline value to determine whether the observed differences are statistically significant.

Table 4: Comparison between GPMVF and the baseline method.

Dataset	GPMVF (Mean $\pm$ Std)			Per-View SVM Voting			<i>p</i> -Value (Wilcoxon)		
	Acc.	F1	Kappa	Acc.	F1	Kappa	Acc.	F1	Kappa
mfeat	0.9818 $\pm$ 0.0046	0.9817 $\pm$ 0.0046	0.9797 $\pm$ 0.0051	0.9850	0.9849	0.9833	0.078125 ($\approx$)	0.105469 ($\approx$)	0.085938 ($\approx$)
100leaves	0.9540 $\pm$ 0.0123	0.9518 $\pm$ 0.0124	0.9535 $\pm$ 0.0124	0.8475	0.8432	0.8460	0.001953 ($>$)	0.001953 ($>$)	0.001953 ($>$)
Scene15	0.7711 $\pm$ 0.0140	0.7664 $\pm$ 0.0145	0.7539 $\pm$ 0.0150	0.6696	0.6497	0.6448	0.001953 ($>$)	0.001953 ($>$)	0.001953 ($>$)
yale_mtv	0.9267 $\pm$ 0.0393	0.9254 $\pm$ 0.0401	0.9214 $\pm$ 0.0421	0.9778	0.9771	0.9762	0.001953 ($<$)	0.003906 ($<$)	0.003906 ($<$)
HandWritten	0.9550 $\pm$ 0.0071	0.9552 $\pm$ 0.0069	0.9500 $\pm$ 0.0079	0.9675	0.9678	0.9639	0.001953 ($<$)	0.001953 ($<$)	0.001953 ($<$)
Prokaryotic	0.8777 $\pm$ 0.0193	0.8717 $\pm$ 0.0211	0.7900 $\pm$ 0.0321	0.7946	0.7636	0.6132	0.001953 ($>$)	0.001953 ($>$)	0.001953 ($>$)
BDGP	0.9548 $\pm$ 0.0099	0.9552 $\pm$ 0.0097	0.9435 $\pm$ 0.0124	0.9280	0.9279	0.9100	0.001953 ($>$)	0.001953 ($>$)	0.001953 ($>$)
ALOI	0.9488 $\pm$ 0.0076	0.9487 $\pm$ 0.0076	0.9483 $\pm$ 0.0077	0.7182	0.7038	0.7153	0.001953 ($>$)	0.001953 ($>$)	0.001953 ($>$)

Note: Bold values indicate the best results. The symbols ($>$, $<$, and $\approx$) denote significant improvement, degradation, or no significant difference between GPMVF and the baseline based on the Wilcoxon signed-rank test. Abb: ALOI, Amsterdam Library of Object Images; BDGP, Berkeley Drosophila Genome Project; mfeat, Multiple Features; yale_mtv, multi-view Yale Face Database.

In terms of mean accuracy, GPMVF outperforms the baseline on five datasets: 100leaves, Scene15, Prokaryotic, the Berkeley Drosophila Genome Project (BDGP), and the Amsterdam Library of Object Images (ALOI). The improvements are statistically significant on all five datasets ($p < 0.05$), and the F1 score and Kappa coefficient show similar trends. The baseline achieves higher mean scores on Multiple Features (mfeat), the multi-view Yale Face Database (yale_mtv), and HandWritten. For mfeat, however, none of the differences across the three metrics is statistically significant ($p > 0.05$). The baseline performs significantly better on yale_mtv and HandWritten.

Overall, GPMVF is particularly effective when complementary cross-view information can be exploited. However, the simpler voting baseline can remain competitive or superior when the original view representations are highly discriminative, indicating that the advantages of GPMVF are dataset-dependent.

5.2 Ablation Study

To assess the contributions of the main components of GPMVF, we compare it with two ablation variants. The simple feature selection and construction variant retains the root-node feature and the terminal features used in the GP expression but excludes intermediate subtree-constructed features. The direct-concatenation GP variant merges all views into a single input vector and performs GP-based feature construction without preserving the view structure or using a dedicated fusion tree. All three methods are run 10 times on each dataset, with results reported as means with standard deviations. Two-sided exact Wilcoxon signed-rank tests are used to assess statistical significance. Table 5 compares GPMVF with the simple variant.

GPMVF significantly outperforms the simple variant on mfeat, 100leaves, Scene15, Prokaryotic, BDGP, and ALOI across all three metrics. This result indicates that incorporating subtree-constructed features enriches the final representation and improves classification performance on most datasets. The differences on yale_mtv and HandWritten are not statistically significant, showing that the benefit of the enhanced feature construction strategy is dataset-dependent.

On yale_mtv, GPMVF and the simple variant achieve nearly identical mean performance, with no significant difference. This may be because the high dimensionality and small sample size make effective feature construction difficult, so the additional subtree features do not yield a consistent gain. On HandWritten, the simple variant is slightly better on average, but the difference is again not significant. The original view features may already be highly discriminative, limiting the additional benefit of subtree-level enhancement.

Table 5: Comparison between GPMVF and the simple feature selection and construction method in the ablation study.

Dataset	GPMVF (Mean $\pm$ Std)			Simple Feature (Mean $\pm$ Std)			<i>p</i> -Value (Wilcoxon)		
	Acc.	F1	Kappa	Acc.	F1	Kappa	Acc.	F1	Kappa
mfeat	0.9818 $\pm$	0.9817 $\pm$	0.9797 $\pm$	0.9708 $\pm$	0.9707 $\pm$	0.9675 $\pm$	0.001953 (>)	0.001953 (>)	0.001953 (>)
	0.0046	0.0046	0.0051	0.0033	0.0033	0.0037			
100leaves	0.9540 $\pm$	0.9518 $\pm$	0.9535 $\pm$	0.9258 $\pm$	0.9238 $\pm$	0.9250 $\pm$	0.001953 (>)	0.001953 (>)	0.001953 (>)
	0.0123	0.0124	0.0124	0.0121	0.0122	0.0122			
Scene15	0.7711 $\pm$	0.7664 $\pm$	0.7539 $\pm$	0.7267 $\pm$	0.7178 $\pm$	0.7062 $\pm$	0.001953 (>)	0.001953 (>)	0.001953 (>)
	0.0140	0.0145	0.0150	0.0131	0.0159	0.0141			
yale_mtv	0.9267 $\pm$	0.9254 $\pm$	0.9214 $\pm$	0.9267 $\pm$	0.9258 $\pm$	0.9215 $\pm$	0.921875 ($\approx$)	1.000000 ($\approx$)	0.906250 ($\approx$)
	0.0393	0.0401	0.0421	0.0150	0.0150	0.0161			
HandWritten	0.9550 $\pm$	0.9552 $\pm$	0.9500 $\pm$	0.9570 $\pm$	0.9573 $\pm$	0.9522 $\pm$	0.796875 ($\approx$)	0.625000 ($\approx$)	0.757812 ($\approx$)
	0.0071	0.0069	0.0079	0.0071	0.0071	0.0080			
Prokaryotic	0.8777 $\pm$	0.8717 $\pm$	0.7900 $\pm$	0.8018 $\pm$	0.7783 $\pm$	0.6288 $\pm$	0.001953 (>)	0.001953 (>)	0.001953 (>)
	0.0193	0.0211	0.0321	0.0201	0.0264	0.0437			
BDGP	0.9548 $\pm$	0.9552 $\pm$	0.9435 $\pm$	0.9430 $\pm$	0.9435 $\pm$	0.9288 $\pm$	0.009766 (>)	0.005859 (>)	0.009766 (>)
	0.0099	0.0097	0.0124	0.0141	0.0138	0.0176			
ALOI	0.9488 $\pm$	0.9487 $\pm$	0.9483 $\pm$	0.9248 $\pm$	0.9251 $\pm$	0.9240 $\pm$	0.001953 (>)	0.001953 (>)	0.001953 (>)
	0.0076	0.0076	0.0077	0.0065	0.0064	0.0066			

Note: Bold values indicate the best results. The symbols (> and $\approx$) denote whether GPMVF is significantly better to the simple variant according to the Wilcoxon signed-rank test. Abb: ALOI, Amsterdam Library of Object Images; BDGP, Berkeley Drosophila Genome Project; mfeat, Multiple Features; yale_mtv, multi-view Yale Face Database.

Table 6 compares GPMVF with the direct-concatenation GP method. GPMVF achieves higher mean scores on all eight datasets, and most differences are statistically significant across the three metrics. This result supports preserving the multi-view structure during feature construction and using a dedicated fusion tree for inter-view modeling rather than concatenating all features into a single input space.

Table 6: Comparison between GPMVF and the direct-concatenation GP method in the ablation study.

Dataset	GPMVF (Mean $\pm$ Std)			Direct-Concatenation GP (Mean $\pm$ Std)			<i>p</i> -Value (Wilcoxon)		
	Acc.	F1	Kappa	Acc.	F1	Kappa	Acc.	F1	Kappa
mfeat	0.9818 $\pm$	0.9817 $\pm$	0.9797 $\pm$	0.9545 $\pm$	0.9546 $\pm$	0.9495 $\pm$	0.001953 (>)	0.001953 (>)	0.001953 (>)
	0.0046	0.0046	0.0051	0.0118	0.0119	0.0131			
100leaves	0.9540 $\pm$	0.9518 $\pm$	0.9535 $\pm$	0.9283 $\pm$	0.9256 $\pm$	0.9275 $\pm$	0.001953 (>)	0.001953 (>)	0.001953 (>)
	0.0123	0.0124	0.0124	0.0102	0.0110	0.0103			
Scene15	0.7711 $\pm$	0.7664 $\pm$	0.7539 $\pm$	0.7457 $\pm$	0.7408 $\pm$	0.7267 $\pm$	0.005859 (>)	0.005859 (>)	0.005859 (>)
	0.0140	0.0145	0.0150	0.0151	0.0165	0.0162			
yale_mtv	0.9267 $\pm$	0.9254 $\pm$	0.9214 $\pm$	0.8222 $\pm$	0.8136 $\pm$	0.8095 $\pm$	0.001953 (>)	0.001953 (>)	0.001953 (>)
	0.0393	0.0401	0.0421	0.0419	0.0458	0.0449			
HandWritten	0.9550 $\pm$	0.9552 $\pm$	0.9500 $\pm$	0.9393 $\pm$	0.9394 $\pm$	0.9325 $\pm$	0.003906 (>)	0.001953 (>)	0.003906 (>)
	0.0071	0.0069	0.0079	0.0094	0.0096	0.0105			
Prokaryotic	0.8777 $\pm$	0.8717 $\pm$	0.7900 $\pm$	0.8054 $\pm$	0.7902 $\pm$	0.6625 $\pm$	0.001953 (>)	0.005859 (>)	0.003906 (>)
	0.0193	0.0211	0.0321	0.0535	0.0709	0.1127			
BDGP	0.9548 $\pm$	0.9552 $\pm$	0.9435 $\pm$	0.7746 $\pm$	0.7759 $\pm$	0.7183 $\pm$	0.001953 (>)	0.001953 (>)	0.001953 (>)
	0.0099	0.0097	0.0124	0.0204	0.0206	0.0255			
ALOI	0.9488 $\pm$	0.9487 $\pm$	0.9483 $\pm$	0.9270 $\pm$	0.9272 $\pm$	0.9263 $\pm$	0.001953 (>)	0.001953 (>)	0.001953 (>)
	0.0076	0.0076	0.0077	0.0059	0.0059	0.0060			

Note: Bold values indicate the best results. The symbols (>) denote significant improvement based on the Wilcoxon test. Abb: ALOI, Amsterdam Library of Object Images; BDGP, Berkeley Drosophila Genome Project; mfeat, Multiple Features; yale_mtv, multi-view Yale Face Database

To quantify the contribution of each component, the comparison between GPMVF and the simple variant isolates the effect of enhanced feature construction, while the comparison between GPMVF and direct-concatenation GP reflects the contribution of the multi-tree multi-view fusion mechanism. In terms of average performance, the enhanced feature construction module brings gains of 0.0242 in accuracy, 0.0267 in F1 score, and 0.0358 in Kappa coefficient, and its improvements are more consistent across datasets. By comparison, the multi-tree multi-view fusion mechanism brings gains of 0.0591 in accuracy, 0.0611 in F1 score, and 0.0734 in Kappa coefficient, its improvements are larger on some datasets but more dataset-dependent. Combining both components produces the best overall performance.

These results show that the two components are complementary: subtree-enhanced feature construction improves intra-view feature expressiveness, while multi-tree fusion strengthens the modeling of complementary information across views.

5.3 Comparative Analysis of Fusion Strategies

We compare three fusion strategies under the proposed multi-tree GP framework: voting, weight, and single. The single strategy uses only the classifier trained on the fusion-tree output and therefore relies exclusively on feature-level fusion. Voting and weight additionally combine the outputs of the view-specific classifiers, thereby introducing decision-level fusion. Table 7 reports the accuracy, F1 score, and Kappa coefficient obtained by the three strategies.

Table 7: Comparison of voting, weight, and single fusion strategies.

Dataset	Voting (Acc/F1/Kappa)	Weight (Acc/F1/Kappa)	Single (Acc/F1/Kappa)	Best Strategy
mfeat	0.9695 $\pm$ 0.0108/0.9696 $\pm$ 0.0108/0.9661 $\pm$ 0.0120	0.9818 $\pm$ 0.0046/0.9817 $\pm$ 0.0046/0.9797 $\pm$ 0.0051	0.9770 $\pm$ 0.0062/0.9770 $\pm$ 0.0062/0.9744 $\pm$ 0.0069	Weight
100leaves	0.9280 $\pm$ 0.0087/0.9262 $\pm$ 0.0088	0.9540 $\pm$ 0.0123/0.9518 $\pm$ 0.0124	0.9280 $\pm$ 0.0245/0.9260 $\pm$ 0.0248	Weight
Scene15	0.7409 $\pm$ 0.0091/0.7328 $\pm$ 0.0095/0.7215 $\pm$ 0.0098	0.7711 $\pm$ 0.0140/0.7664 $\pm$ 0.0145/0.7539 $\pm$ 0.0150	0.7570 $\pm$ 0.0179/0.7521 $\pm$ 0.0180/0.7389 $\pm$ 0.0192	Weight
yale_mtv	0.9267 $\pm$ 0.0393/0.9254 $\pm$ 0.0401/0.9214 $\pm$ 0.0421	0.8978 $\pm$ 0.0404/0.8954 $\pm$ 0.0416/0.8905 $\pm$ 0.0433	0.8800 $\pm$ 0.0337/0.8794 $\pm$ 0.0347/0.8715 $\pm$ 0.0361	Voting
HandWritten	0.9540 $\pm$ 0.0104/0.9541 $\pm$ 0.0104/0.9489 $\pm$ 0.0115	0.9550 $\pm$ 0.0071/0.9552 $\pm$ 0.0069/0.9500 $\pm$ 0.0079	0.9510 $\pm$ 0.0052/0.9512 $\pm$ 0.0051/0.9456 $\pm$ 0.0058	Weight
Prokaryotic	0.8000 $\pm$ 0.0307/0.7746 $\pm$ 0.0395/0.6259 $\pm$ 0.0645	0.8777 $\pm$ 0.0193/0.8717 $\pm$ 0.0211/0.7900 $\pm$ 0.0321	0.8464 $\pm$ 0.0590/0.8372 $\pm$ 0.0662/0.7383 $\pm$ 0.1039	Weight
BDGP	0.9398 $\pm$ 0.0221/0.9407 $\pm$ 0.0218/0.9248 $\pm$ 0.0276	0.9548 $\pm$ 0.0099/0.9552 $\pm$ 0.0097/0.9435 $\pm$ 0.0124	0.9312 $\pm$ 0.0175/0.9322 $\pm$ 0.0169/0.9140 $\pm$ 0.0219	Weight
ALOI	0.9225 $\pm$ 0.0149/0.9228 $\pm$ 0.0151/0.9217 $\pm$ 0.0150	0.9488 $\pm$ 0.0076/0.9487 $\pm$ 0.0076/0.9483 $\pm$ 0.0077	0.9345 $\pm$ 0.0128/0.9347 $\pm$ 0.0127/0.9338 $\pm$ 0.0129	Weight

Note: Bold values indicate the best performance among the three fusion strategies. Abb: ALOI, Amsterdam Library of Object Images; BDGP, Berkeley Drosophila Genome Project; mfeat, Multiple Features; yale_mtv, multi-view Yale Face Database.

The single strategy does not achieve the best result on any dataset. The weight strategy performs best on all three metrics for seven datasets: mfeat, 100leaves, Scene15, HandWritten, Prokaryotic, BDGP, and ALOI; voting performs best only on yale_mtv. Although single is close to the other strategies on some metrics, the overall results indicate that adding decision-level fusion generally exploits complementary information more effectively than using the fusion-tree classifier alone. The soft-decision weight strategy also appears more stable and effective than hard-decision voting on most datasets.

5.4 t-SNE-Based Feature Space Visualization

To illustrate the feature engineering capability of GPMVF, we apply t-SNE to visualize the feature space before and after applying the proposed method.

Fig. 4 presents the Prokaryotic results. In the original feature space, samples from different classes overlap and the intra-class distributions are relatively scattered. After GPMVF processing, the samples become more compact and the boundaries between several classes, particularly Class 2 and Class 4, become clearer. This suggests that view-specific feature construction and cross-view fusion can improve feature separability.

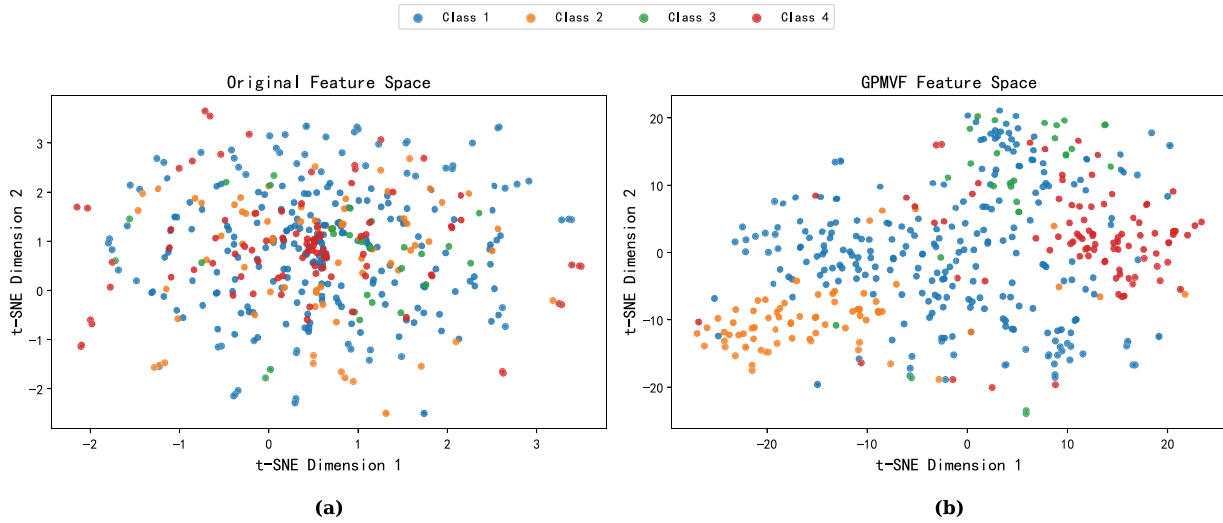


Figure 4: t-SNE visualization of the Prokaryotic dataset: (a) original features; (b) GPMVF-constructed features.

Fig. 5 presents the BDGP results. The original features exhibit inter-class mixing after low-dimensional mapping, whereas the GPMVF-processed features form more distinct clusters with greater intra-class compactness. Together with the classification results, this observation suggests that feature construction and fusion improve the separability of the BDGP representation.

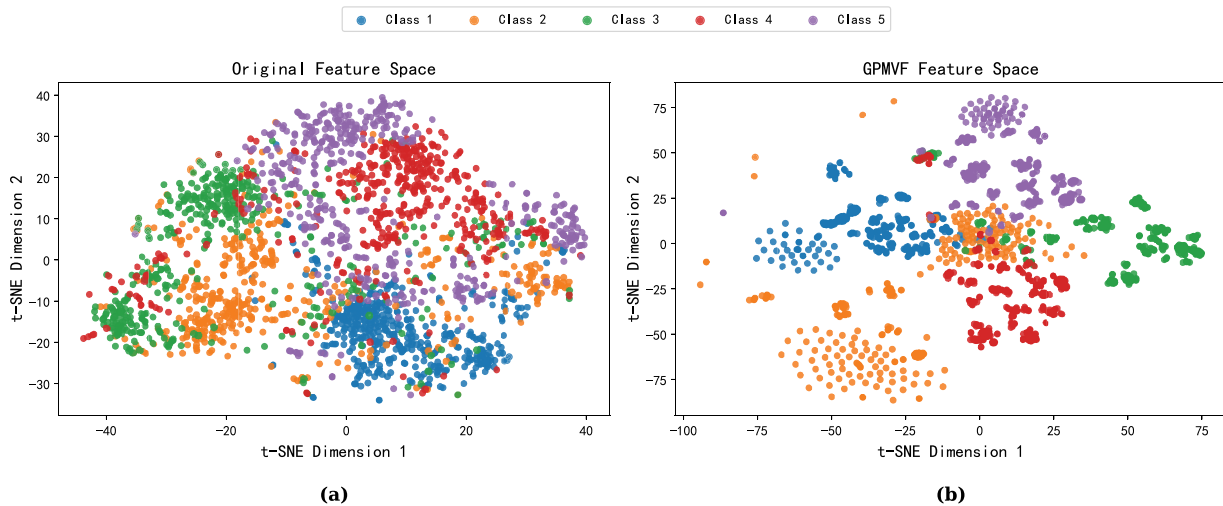


Figure 5: t-SNE visualizations of the BDGP dataset: (a) original features; (b) GPMVF-constructed features.

To complement the visual analysis, we compare the original and GPMVF-constructed feature spaces using the Silhouette Score, Davies–Bouldin Index, and Calinski–Harabasz Score. All three metrics improve on Prokaryotic and BDGP, consistent with the clearer separation observed in the t-SNE plots.

As shown in Table 8, the metric changes on the four datasets are not fully consistent. Prokaryotic and BDGP show improvement on all three metrics, while yale_mtv improves on two metrics, and HandWritten does not improve on any of them.

Table 8: Changes in clustering quality metrics after GPMVF processing on four datasets.

Dataset	Δ Silhouette Score	Δ Davies–Bouldin Index	Δ Calinski–Harabasz Score
Prokaryotic	+0.061981	+7.490434	+12.764722
BDGP	+0.019607	+3.298324	+82.971161
yale_mtv	−0.030022	+0.062833	+0.400697
HandWritten	−0.028109	−0.095910	−0.133583

Note: BDGP, Berkeley Drosophila Genome Project; yale_mtv, multi-view Yale Face Database.

Fig. 6 presents the yale_mtv results. Because yale_mtv contains relatively few samples but many classes, the difference between the two feature spaces is difficult to assess visually. However, the Davies–Bouldin Index and Calinski–Harabasz Score improve after GPMVF processing, indicating some improvement in clustering structure, although it is insufficient to improve classification performance.

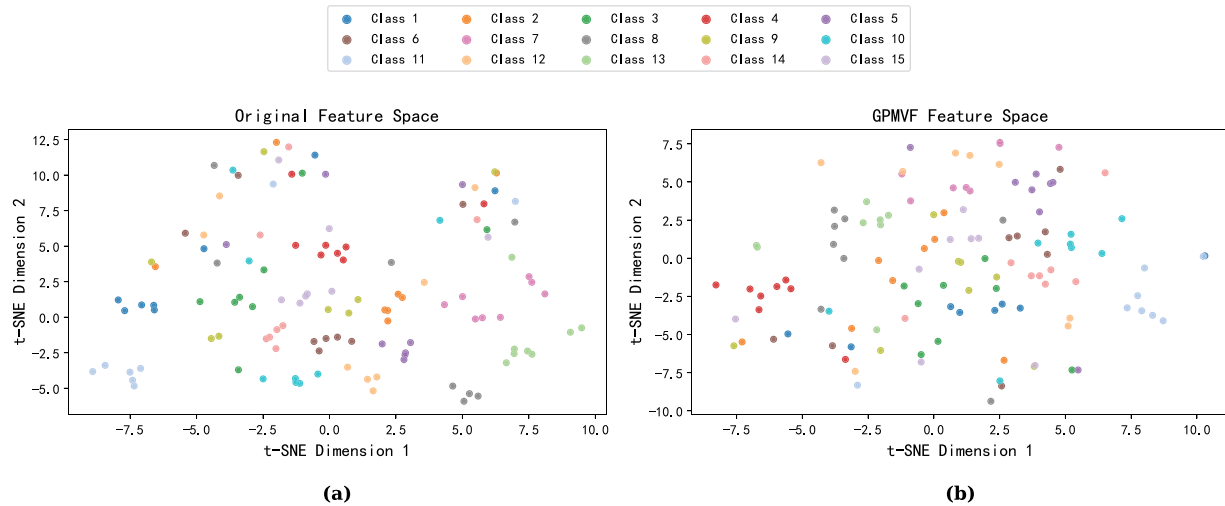


Figure 6: t-SNE visualizations of the yale_mtv dataset: (a) original features; (b) GPMVF-constructed features.

Fig. 7 presents the HandWritten results. The GPMVF-processed feature space does not show clearer class separation than the original feature space, and all three clustering metrics deteriorate. Thus, GPMVF does not identify a more favorable feature space for this dataset, consistent with the absence of improved classification performance.

Together, the t-SNE visualizations and clustering metrics provide a balanced assessment of GPMVF. Both forms of evidence indicate improved feature separability on Prokaryotic and BDGP. yale_mtv shows improvement only in two clustering metrics, while HandWritten shows no improvement in either the visualization or the metrics. The effect of GPMVF on feature separability is therefore dataset-dependent.

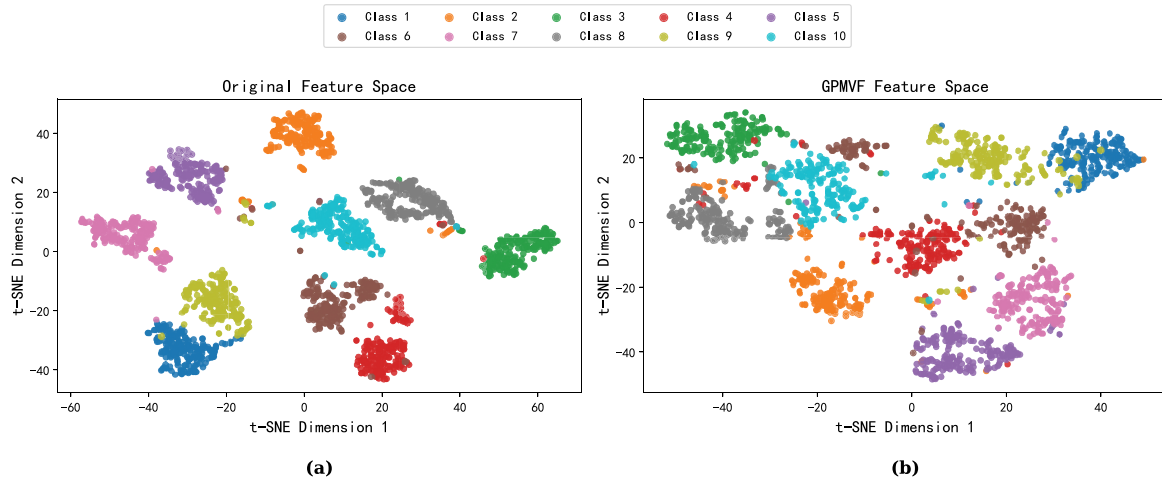


Figure 7: t-SNE visualizations of the HandWritten dataset: (a) original features; (b) GPMVF-constructed features.

5.5 Computational Cost and Scalability Analysis

Computational cost is evaluated using the average wall-clock time of a complete run, including five-fold cross-validation, under the same computing environment. GPMVF requires approximately 2 h on mfeat, 1 h and 40 min on 100leaves, 3 h and 40 min on Scene15, 1 h on yale_mtv, 1.5 h on HandWritten, 30 min on Prokaryotic, 1 h and 15 min on BDGP, and 9 h on ALOI. By contrast, the per-view SVM voting baseline is completed within several minutes on all datasets. The additional cost of GPMVF arises mainly from multi-tree evolutionary search, subtree traversal, view-specific feature construction, and the training of multiple classifiers. Runtime also depends on the number of samples, views, features, and classes, as summarized in Table 3. ALOI has the longest runtime, consistent with its relatively large sample size and number of classes. Thus, the performance gains of GPMVF come with substantial computational cost, and greater efficiency is needed for larger and more complex multi-view datasets.

5.6 Stability Analysis of Fused Feature Dimensionality and Classification Accuracy

We assess the stability of GPMVF using the dimensionality of the final fused features and the test accuracy across repeated runs. Table 9 reports the mean and standard deviation of both quantities on the eight datasets.

Table 9: Stability analysis of fused feature dimensionality and classification accuracy over repeated runs.

Dataset	Fused Feature Dimensionality (Mean $\pm$ Std)	Accuracy (% , Mean $\pm$ Std)
mfeat	87.33 $\pm$ 12.23	98.18 $\pm$ 0.46
100leaves	105.70 $\pm$ 24.72	95.40 $\pm$ 1.23
Scene15	114.20 $\pm$ 19.63	77.11 $\pm$ 1.40
yale_mtv	108.60 $\pm$ 27.38	92.67 $\pm$ 3.93
HandWritten	113.10 $\pm$ 20.16	95.50 $\pm$ 0.71
Prokaryotic	58.10 $\pm$ 21.60	87.77 $\pm$ 1.93
BDGP	139.10 $\pm$ 25.19	95.48 $\pm$ 0.99
ALOI	131.20 $\pm$ 36.49	94.88 $\pm$ 0.76

Note: ALOI, Amsterdam Library of Object Images; BDGP, Berkeley Drosophila Genome Project; mfeat, Multiple Features; yale_mtv, multi-view Yale Face Database.

The fused feature dimensionality varies across runs, indicating that evolution produces representations of different sizes. Nevertheless, classification accuracy remains relatively stable on most datasets: its standard deviation is below 2% on seven of the eight datasets, with only yale_mtv showing greater variation. Thus, despite variability in fused feature dimensionality, GPMVF is generally robust to run-to-run variation in classification performance.

6 Conclusion

This paper addresses feature representation and information fusion in multi-view classification through the proposed GPMVF framework. Each view is assigned a view-specific tree for intra-view feature construction, while a fusion tree models the relationships among the resulting view representations. Together with the enhanced feature construction strategy and dual-layer fusion mechanism, GPMVF provides a unified framework for feature construction, cross-view fusion, and classification.

Experiments on eight datasets demonstrate the effectiveness of GPMVF. Compared with the representative baseline, which trains an SVM separately on each view and combines predictions by voting, GPMVF shows a relatively stable advantage, highlighting the value of feature-level joint modeling before classification. The ablation study further validates the proposed design: incorporating subtree-constructed features produces richer and more discriminative representations than retaining only root and terminal features, while the multi-tree structure with a dedicated fusion tree is more effective than treating all views as a single concatenated feature space. The t-SNE analysis also shows that the constructed features can achieve greater intra-class compactness and inter-class separability, indicating improved feature representation.

Future work can further improve the search and evaluation mechanisms of GP to enhance classification performance and reduce redundant feature construction. It is also worthwhile to introduce an adaptive weighting strategy so that different views can be assigned different importance according to their actual contribution during fusion. In addition, the multi-view fusion framework holds potential for integration with soft antenna systems, where each deformable antenna functions as a distinct sensing view, and the dual-layer fusion mechanism can jointly model resonant frequency shifts from multiple embedded antennas to achieve robust shape perception and wireless communication under large structural deformations.

Acknowledgement: None.

Funding Statement: This work was supported in part by the Suzhou Planning Project of Science and Technology (ZXP2025060), and in part by the China Postdoctoral Science Foundation (2025M781499).

Author Contributions: The authors confirm contribution to the paper as follows: conceptualization and methodology, Yikun Zhang; validation and formal analysis, Yikun Zhang and Runyao Yin; writing—original draft preparation, Yikun Zhang; writing—review and editing, Yikun Zhang, Dongmin Zhang, Chao Sun, and Chen Chen; funding acquisition, Dongmin Zhang, Chao Sun, and Chen Chen. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The datasets used in this study are publicly available datasets. Further details about the data and experimental settings are available from the corresponding author upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Li Y, Yang M, Zhang Z. A survey of multi-view representation learning. *IEEE Trans Knowl Data Eng.* 2019;31(10):1863–83. doi:10.1109/tkde.2018.2872063.
2. Zhang R, Nie F, Li X, Wei X. Feature selection with multi-view data: a survey. *Inf Fusion.* 2019;50(1):158–67. doi:10.1016/j.inffus.2018.11.019.
3. Chen K, Zhong L, Zhang Y, Liu Y. Traditional Chinese medicine syndrome identification based on multi-view tongue image feature fusion. *Appl Res Comput.* 2025;42(7):2116–22. (In Chinese).
4. Lu Y, Kasabov N, Lu G. Multi-view geometry consistency network for facial micro-expression recognition from various perspectives. In: *Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN)*; 2021 Jul 18–22; Shenzhen, China. p. 1–8.
5. Qi S, Ning X, Yang G, Zhang L, Long P, Cai W, et al. Review of multi-view 3D object recognition methods based on deep learning. *Displays.* 2021;69:102053. doi:10.1016/j.displa.2021.102053.
6. Sun S. A survey of multi-view machine learning. *Neural Comput Appl.* 2013;23(7):2031–8. doi:10.1007/s00521-013-1362-6.
7. Berahmand K, Daneshfar F, Rahmaninia M, Haghighat M, Jalili M. A comprehensive survey on multi-view classification: methods, applications, and challenges. *ACM Trans Intell Syst Technol.* 2025;16(6):1–34. doi:10.1145/3767728.
8. Zhang Y, Jiao R, Xue B, Zhang M. Evolutionary streaming feature selection via incremental feature clustering. *IEEE Trans Evol Computat.* 2026. doi:10.1109/TEVC.2026.3667279.
9. Zhai S, Jiao R, Xue B, Nojima Y, Zhang M. MILE: multi-expert ensemble with instance selection for multi-class Imbalanced learning. *IEEE Trans Evol Comput.* 2026. doi:10.1109/TEVC.2026.3682346.
10. Xu C, Tao D, Xu C. A survey on multi-view learning. *arXiv:1304.5634.* 2013.
11. Zhao J, Xie X, Xu X, Sun S. Multi-view learning overview: recent progress and new challenges. *Inf Fusion.* 2017;38:43–54. doi:10.1016/j.inffus.2017.02.007.
12. Sun B, Li L, Wu X, Zuo T, Chen Y, Zhou G, et al. Combining feature-level and decision-level fusion in a hierarchical classifier for emotion recognition in the wild. *J Multimodal User Interfaces.* 2016;10(2):125–37. doi:10.1007/s12193-015-0203-6.
13. Koza JR. Genetic programming as a means for programming computers by natural selection. *Stat Comput.* 1994;4(2):87–112. doi:10.1007/bf00175355.
14. Chen Z, Fan Q, Jiao R, Xue B, Huang H, Dai Y, et al. Multi-expert genetic programming based ensemble for long-tailed image classification. *IEEE Trans Evol Computat.* 2026. doi:10.1109/TEVC.2026.3669207.
15. Niu J, Gao X, Bai M, Ma J. Adaptive multi-tree genetic programming for multiple feature construction. In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*; 2025 Jul 14–18; Malaga, Spain. p. 651–4.
16. Wu Z, Xue B, Zhang M. Multitree genetic programming for multimodal learning in multimodal medical image classification. *IEEE Trans Evol Comput.* 2025. doi:10.1109/tevc.2025.3619532.
17. Blum A, Mitchell T. Combining labeled and unlabeled data with co-training. In: *Proceedings of the Annual Conference on Computational Learning Theory*; 1998 Jul 24–26; Madison, WI, USA. p. 92–100.
18. Chaudhuri K, Kakade SM, Livescu K, Sridharan K. Multi-view clustering via canonical correlation analysis. In: *Proceedings of the 26th Annual International Conference on Machine Learning*; 2009 Jun 18–21; Montreal, QC, Canada. p. 129–36.
19. Tang C, Liu X, Zhu X, Zhu E, Luo Z, Wang L, et al. CGD: multi-view clustering via cross-view graph diffusion. *Proc AAAI Conf Artif Intell.* 2020;34(4):5924–31. doi:10.1609/aaai.v34i04.6052.
20. Zheng T, Wu J, Zhang B, Wang L. Enhancing feature deep learning to improve multi-view latent space fusion representation. *J Taiyuan Univ Technol.* 2022;53(4):697–706.
21. Yu Z, Dong Z, Yu C, Yang K, Fan Z, Chen CP. A review on multi-view learning. *Front Comput Sci.* 2024;19(7):197334. doi:10.1007/s11704-024-40004-w.

22. Trosten DJ, Løkse S, Jenssen R, Kampffmeyer MC. On the effects of self-supervision and contrastive alignment in deep multi-view clustering. In: Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2023 Jun 17–24; Vancouver, BC, Canada. p. 23976–85.
23. Li S, Yuan Z, Zhao X. Deep contrastive multi-view clustering with transformer fusion. *Pattern Recognit Artif Intell*. 2025;38(12):1057–74. (In Chinese).
24. Koza JR. Genetic programming II: automatic discovery of reusable programs. Cambridge, MA, USA: MIT Press; 1994.
25. Poli R. Genetic programming for image analysis. Birmingham, UK: University of Birmingham CSRP; 1996. Cognitive Science Research Papers.
26. Montana DJ. Strongly typed genetic programming. *Evol Comput*. 1995;3(2):199–230. doi:10.1162/evco.1995.3.2.199.
27. Khorshidi MS, Yazdanjue N, Gharoun H, Nikoo MR, Chen F, Gandomi AH. Multi-population ensemble genetic programming via cooperative coevolution and multi-view learning for classification. *IEEE Trans Emerg Top Comput Intell*. 2026;10(3):2541–54. doi:10.1109/tetci.2026.3668624.
28. La Cava W, Silva S, Danai K, Spector L, Vanneschi L, Moore JH. Multidimensional genetic programming for multiclass classification. *Swarm Evol Comput*. 2019;44(15):260–72. doi:10.1016/j.swevo.2018.03.015.
29. Chen K, Dao M, Bi Y, Liang J, Wu Z, Wang P. A new multi-tree genetic programming approach to feature construction in high-dimensional classification. *Knowl Based Syst*. 2025;319(2):113643. doi:10.1016/j.knosys.2025.113643.
30. Xue X, Chun-Wei Lin J, Jiang Z. Collaborative ontology matching with dual population genetic programming and active meta-learning. *IEEE Trans Evol Computat*. 2026;30(3):1024–38. doi:10.1109/tevc.2025.3569336.
31. Xue X, Waqas SM, Liu W, Lin JC. Automatic similarity feature combination for knowledge graph alignment via adaptive-designed particle swarm optimization with genetic programming. *IEEE Trans Emerg Top Comput Intell*. 2026;10(4):3149–65. doi:10.1109/TETCI.2026.3683654.