Relieving Coefficient Learning in Genetic Programming for Symbolic Regression via Correlation and Linear Scaling

Qi Chen
Victoria University of Wellington
Wellington, New Zealand
Qi.Chen@ecs.vuw.ac.nz

Bing Xue
Victoria University of Wellington
Wellington, New Zealand
Bing.Xue@ecs.vuw.ac.nz

Wolfgang Banzhaf
Michigan State University
Michigan, USA
Banzhafw@msu.edu

Mengjie Zhang
Victoria University of Wellington
Wellington, New Zealand
Mengjie.Zhang@ecs.vuw.ac.nz

ABSTRACT

The difficulty of learning optimal coefficients in regression models using only genetic operators has long been a challenge in genetic programming for symbolic regression. As a simple but effective remedy it has been proposed to perform linear scaling of model outputs prior to a fitness evaluation. Recently, the use of a correlation coefficient-based fitness function with a post-processing linear scaling step for model alignment has been shown to outperform error-based fitness functions in generating symbolic regression models. In this study, we compare the impact of four evaluation strategies on relieving genetic programming (GP) from learning coefficients in symbolic regression and focusing on learning the more crucial model structure. The results from 12 datasets, including ten real-world tasks and two synthetic datasets, confirm that all these strategies assist GP to varying degrees in learning coefficients. Among the them, correlation fitness with one-time linear scaling as post-processing, due to be the most efficient while bringing notable benefits to the performance, is the recommended strategy to relieve GP from learning coefficients.

CCS CONCEPTS

• **Computing methodologies** → **Genetic programming**.

KEYWORDS

Genetic Programming, Symbolic Regression, Fitness Function, Correlation, Linear Scaling

ACM Reference Format:
Qi Chen, Bing Xue, Wolfgang Banzhaf, and Mengjie Zhang. 2023. Relieving Coefficient Learning in Genetic Programming for Symbolic Regression via Correlation and Linear Scaling. In *Proceedings of The Genetic and Evolutionary Computation Conference 2023 (GECCO '23).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3583131.3595918

GECCO '23, July 15–19, 2023, Lisbon, Portugal

ACM ISBN 979-8-4007-0119-1/23/07…$15.00
https://doi.org/10.1145/3583131.3595918

1 INTRODUCTION

Genetic programming for symbolic regression (GPSR) is a powerful regression technique that aims to automatically learn both the model structure(s) and model coefficients [15]. During an evolutionary process, the model structure can be automatically learnt via genetic operators. The model coefficients are created and optimised using a special type of terminal, i.e. *ephemeral random constant* (ERC) [15]. ERCs in GP individuals are random constants generated from a predefined range during the initialisation phase or in a mutation step. Learning of coefficients happens by moving these ERCs around from tree/model to tree/model using the crossover operator. Due to its random nature, learning optimal coefficients is difficult for GP, and is still one of the (most) significant open issues in GP considered to be addressed by our community[19].

Coefficient learning in GPSR also creates some other issues. The effort of GPSR spent on learning coefficients may prohibit it from finding regression models with the desired structure/shape. Note that the shape of a model refers to the overall pattern of the mathematical model's graph or equation. It describes how the model behaves and changes as its inputs or coefficients are varied. For example, a linear model has a shape of a straight line, while an exponential model has a shape that curves upward rapidly as the input increases. In GPSR, various error measures, e.g. root/mean squared errors (RMSE), relative squared errors (RSE), and mean absolute errors (MAE), are normally used to determine the quality of GPSR models during the evolutionary process. Due to the direct (local) comparison between predicted and target values in these error measures, GPSR is forced to first get the range right. Generally, the selection pressure on getting the right range is so high that it causes GP to spend most effort in finding good coefficient values. Once these are found, the diversity of a population drops, making it much more difficult to find the desired regression model structure. Keijzer [13] has found a huge difference between 98% and 16% in the successes rate when using a standard GPSR on two simple problems of X^2 and $X^2 + 100$, and a large difference in search efficiency as well. Keijzer [13] also proposed to use linear scaling on the GPSR model prior to calculating the error. This is a simple but effective way to obtain constants that otherwise need to be found during the evolutionary process of GP, and it enables GP to concentrate on the more important problem of inducing a regression model with the desired shape.

Recently, Haut et al. [11] explored the use of correlation as the fitness function in GPSR. During the evolutionary process, GPSR tries to find models that maximise the Pearson Correlation between the predictions and the target variable. With linear scaling as an alignment step at the end of the evolutionary process in GPSR, the use of correlation obtains notable gains over using RMSE as fitness function, not only in prediction accuracy but also efficiency in terms of the number of data points needed to train regression models. They ascribe the advantage of the correlation based evaluation to emphasis on the global features of a model, instead of the point-to-point comparisons in the commonly used error measures. However, the effect of a correlation based fitness function on relieving GPSR from learning coefficients has not been thoroughly investigated. Furthermore, there is another question when using the correlation coefficient as a fitness value for GPSR models. Specifically, in statistical analysis, the correlation coefficient measures the degree of association between two variables, thus the question here is whether the *degree of association* between the predictions and the target outputs can be used as a measure of the *degree of coincidence* since the best coincidence of two functions/regression models means their best fitting. In [11], Haut et al. considered a large number of synthetic benchmark symbolic regression tasks with a varying number of data points and varying levels of noise. However, the effect of the correlation based fitness function on general real-world symbolic regression tasks is not clear yet.

This work aims to answer these questions and further explore the benefits of correlation based fitness function and linear scaling on relieving coefficients learning in GPSR. In addition, this work will further investigate the importance of searching for models with a good shape while utilising linear scaling for alignment on improving the performance of a GPSR system. More specifically, there are three objectives in this contribution:

(1) Investigate the effect of the Pearson correlation coefficient on making GPSR more focused on searching for the desired shape, thus relieving coefficients learning in GPSR;
(2) Investigate the effect of linear scaling either as a post processing step or prior to each fitness evaluation of GP individuals on enhancing the coefficient learning in GPSR;
(3) Investigate whether a correlation coefficient could be a general evaluation strategy for GPSR that can improve its learning efficiency and generalisation ability over commonly used error measures, without much additional or even less running overhead.

2 BACKGROUND AND RELATED WORKS

2.1 Correlation Coefficients

One of the most prevalent correlation coefficients is Pearson's product-moment correlation coefficient r [24], also known as Pearson correlation coefficient, which measures the linear relationship between two random variables. It refers to the ratio of the covariance of the two variables to the product of their standard deviations, with a value ranging between -1 and 1. The further away r is from 0, the stronger the linear relationship between the two variables, while the sign of r corresponds to the direction of the relationship.

Another commonly used correlation coefficient is Spearman correlation, which is the sample correlation coefficient of the ranks based on continuous data [2]. Spearman correlation is used to measure the monotonic relationship between two variables, i.e. whether one variable tends to take either a large/small value by increasing the value of the other variable.

The R-square (R^2) is a concept related to the correlation coefficients and also a commonly used measure for the goodness of regression fits in statistics. Throughout the literature, there are many different formulas to define R^2 [16]. One definition of R^2 is to take the square of the Pearson correlation coefficient r. R^2 measures the fraction of the variability in one variable that can be explained by the variability in the other variable through their linear relationship, or vice versa. Note that R^2 is calculated only on the basis of the Pearson correlation coefficient. Thus, it is not appropriate to compute R^2 on the basis of rank correlation coefficients such as the Spearman. In this work, we will investigate the effect of R^2 as the fitness function taking its most commonly used form of

$$R^2 = \left(\frac{\sum_{i=1}^{n} (\hat{y}_i - \overline{\hat{y}})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n} (\hat{y}_i - \overline{\hat{y}}) \sum_{i=1}^{n} (y_i - \overline{y})}} \right)^2 \tag{1}$$

where $\hat{y}_i$ and y_i are the prediction and the target values, and $\overline{y}$ is the average of target values. Typically, R^2 is within the range of $[0, 1]$.

R^2 and its extended forms have been used in a number of regression tasks [7, 10, 29]. Fumo et al. [10] utilised R^2 for measuring the quality of simple and multivariate linear regression models for resident energy consumption prediction. Chicco et al. [7] compared the property and effort of R^2 and Symmetric mean absolute percentage error (SMAPE) theoretically and empirically in their work. They found that R^2 is more truthful and informative than SMAPE. More specifically, R^2 generally generates a high score only if the regression model correctly predicts most of the ground truth elements for each ground truth group, considering their distribution. The error measure, i.e., SMAPE, focuses on the relative distance between each predicted value and its corresponding target value instead without considering their distribution. Zhang et al. [29] extended R^2 for generalized linear models by giving it a new definition. The advantage of the new definition is that it only needs to know the mean and variance functions instead of the complete specification of the likelihood function. Many works in traditional regression have realised the importance of R^2 on measuring model quality, but there is only a small number of works in GPSR [11] that has used R^2 as a performance metric.

2.2 Coefficient Learning in GPSR

Using linear scaling for optimising coefficients in the GPSR model is not new. Keijzer et al. [13] proposed to incorporate linear regression/scaling into GPSR to remove the search of coefficients from GP runs. Later, Chen et al. [5] and Virgolin et al. [27] introduced linear scaling into semantic GPSR to learn better coefficients thus helping to achieve the desired semantics.

Ryan and Keijzer [23] investigated how coefficients/constants could be effectively evolved during the evolutionary process in GPSR. They found that without mutation on these coefficients, only a small number of them can survive to the final stage of the evolutionary process, which may force GPSR to synthesise the

desired coefficients at a cost of performance. But uniform mutation generally increases this number, and thus is more likely to lead to better performance in GPSR. Chen et al. [3] extended the idea from [26, 30] and applied gradient search for optimising the coefficients in GPSR. Recently, Dick [8] examined the use of stochastic gradient descent techniques for learning coefficients in GPSR with Z-score standardisation, which is considered to be an important element to apply stochastic gradient descent effectively. Kommenda et al. [14] used nonlinear least squares to optimise the coefficient in GPSR with Levenberg-Marquardt where automatic differentiation is applied to calculate gradients. They achieved a notable improvement in the prediction performance of GP. Based on [14], Rockett [22] explored the influence of coefficient optimisation on the performance of GPSR with and without feature standardisation under the multi-objective GP framework. Instead of Levenberg-Marquardt, they employed Sequential Linear Quadratic Programming which can minimise arbitrary functions but requires the existence of at least the second derivatives of the fitness function. In Sobania et al. [25], instead of ERC, constants that could be optimised were used during the evolutionary process in GPSR with the Sequential Least Squares Programming method. In this way, GPSR evolves regression models with better fitness but a smaller size. Haut et al. [11] utilised Pearson correlation as the fitness function in GPSR. During the evolutionary process, their GPSR method aims to find models that maximise Pearson correlation between the predictions and the target variable. The authors ascribe the advantage of the correlation based evaluation to its emphasis on the global features of a model, with the coefficients in a correlation measure being less important than in the commonly used error measure.

3 LIBERATING GPSR FROM COEFFICIENT LEARNING

This work explores the effect of a combination of fitness evaluation strategy and linear scaling on coefficient learning in GPSR. In this section, we will present the fitness function considered in this work and the method to perform linear scaling.

3.1 Error Measures and Correlation Coefficients

The general method of determining the fitness of a GPSR model is to measure how close the predicted outputs $\hat{y}$ are to the target outputs y over the training data, using an error measure like root mean squared Error (RMSE) as shown in Equation (2).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}} \tag{2}$$

where $\hat{y}_i$ and y_i are the predicted and the target values, respectively. n is the number of training instances. These error measures generally represent the average distance that the target values away from the predictions of GPSR models. Thus, GPSR models with a smaller error value are better since the smaller error value indicates that they are closer to the target outputs.

During the evolutionary process, GPSR models are typically evolved to obtain a smaller error value. The selection pressure pushes the models to get the outputs to the right range close to the target outputs first, rather than focusing on the more important task of finding the right model structure. Moreover, models with the desired structure might be overlooked due to the distance accumulated over all prediction points. Keijzer et al. [13] proposed to evaluate the error between the linear scaled outputs $a\hat{y} + b$ and the target values y:

$$RMSE_{ls} = \sqrt{\frac{\sum_{i=1}^{n}(a\hat{y}_i + b - y_i)^2}{n}} \tag{3}$$

In such a way GP is freed (or at least relieved) from searching for the right range of outputs and GP can focus more on searching for expressions with the right shape. However, a more straightforward way to do this would be to apply a fitness measure which focuses on measuring the shape of the function represented by models directly, thus driving the search of GPSR for equations whose shape is most similar to that of the target data.

Correlation as a measure of association between two variables can be used in the fitness function to relieve GPSR from searching for the right coefficients and focus more on the structure of the model. For a regression task which typically has a continuous target variable and predictions, we can consider the Pearson correlation coefficient r. r measures a linear relationship between two variables with a definition as given in Equation (4).

$$r = \frac{\sum_{i=1}^{n}(\hat{y}_i - \overline{\hat{y}})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \overline{\hat{y}})\sum_{i=1}^{n}(y_i - \overline{y})}} \tag{4}$$

where $\hat{y}_i$ and y_i are the predicted and the target values/outputs, respectively, $\overline{\hat{y}}$ is the average of predicted values while $\overline{y}$ is the averaged target values.

Comparing Equations (2) and (4), a key difference between them is that the latter fitness function has an additional component of the averaged predicted output $\overline{\hat{y}}$. It is considered in relation to the corresponding target outputs. The correlation function looks at the relative position of predicted data points, and compares that with the relative position in the target model, thus incorporating information about the shape of equations represented by GPSR models into the fitness function.

Based on [11], in this work, we also examine the maximisation of the R^2, which is the squared Pearson correlation coefficient between the predictions of GPSR models and the target outputs driving the evolutionary process. This is actually equal to minimising $1 - R^2$ which ranges at $[0, 1]$. Thus, the correlation based fitness function in this work is as follows.

$$1 - R^2 = 1 - \left(\frac{\sum_{i=1}^{n}(\hat{y}_i - \overline{\hat{y}})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \overline{\hat{y}})\sum_{i=1}^{n}(y_i - \overline{y})}}\right)^2 \tag{5}$$

where $\hat{y}_i$ and y_i are the prediction and the target values, and $\overline{y}$ is the average of target values.

3.2 Linear Scaling with Correlation and Error Measures

When using RMSE as fitness function for GPSR, the effect of using linear scaled model outputs during the evolutionary process is to change the view of the selection operator on the goodness of fit of individuals. Even taking the simplest form of linear scaling, the

outputs of the model with an optimal slope and intercept, GP will focus its search on expressions that are close in shape to the target.

The case is different when using correlation. Here, due to the scale and translation invariance of correlation, these are left to a post-processing step of linear regression/scaling [11]. This latter operation will not change the correlation between predictions and the target variables, and is therefore not necessary during the evolutionary process.

To perform a linear scaling, in [13], a deterministic calculation has been used where $b = \frac{\sum[(\hat{y}_i - \overline{\hat{y}})(y_i - \overline{y})]}{\sum[(y - \overline{y})^2]}$, $a = \overline{\hat{y}_i} - b\overline{y}$. In this work, to prevent the model from overfitting the training data, a Ridge regression with an L2 penalty term for large coefficients is performed to find an optimal slope a and intercept b where a and b are found by minimising Equation (6):

$$\sum_{i=0}^{n}(y_i - (a\hat{y}_i + b))^2 + \lambda(a^2 + b^2) \tag{6}$$

where the penalty term $\lambda = 1$ is chose empirically in this work.

4 EXPERIMENTAL SETTINGS

To compare the effect of RMSE and correlation coefficient combined with linear scaling on relieving GPSR from learning coefficients and evolving SR models with a desire shape, a set of experiments has been conducted. We will introduce the key components and settings of these experiments in this section.

4.1 Benchmark Datasets

Twelve widely used benchmark symbolic regression tasks are used in this work. They are taken from the UCI machine learning repository [17], and recent research on symbolic regression [1, 8]. The numbers of features and instances in the datasets are summarised in Table 1. While the first ten datasets are real-world tasks, the last two datasets are synthetic datasets.

Among the ten real-world datasets, Tower and DowChem are symbolic regression benchmarks recommended in [28], and the other eight real-world datasets are all available in "scikit-learn" [20]. The training set and the test set are provided in Tower. For the other datasets, following previous research on machine learning for regression [12] and GPSR [4], a random split is performed with 80% of the data for training and the rest 20% for testing.

The two synthetic datasets, Keijzer5 and Korns8 are benchmark tasks recommended in [28] and have also been studied in previous work on using correlation as the fitness function [11]. Thus, we also use them in this work. Functions and sampling strategies for generating the datasets are shown in Table 2.

4.2 Benchmark Methods

The following five GP methods are considered and compared with each other in this work:

(1) *GP*: a standard GP method with RMSE using Equation (2) as the fitness function without any scaling in the models. It is used as a baseline for the comparisons.
(2) *GPLLS*: GP with linear scaling as the final post-processing step. It basically still uses RMSE in Equation (2) as the fitness function, but linear scaling with Ridge regression will be applied to the *best-of-run* models. To make a fair and clear comparison, in this work, GPLLS is run independently instead of taking the best individuals directly from GP before post processing.
(3) *GPCorLLS*: a GP method maximising correlation coefficient (minimising $1 - R^2$ in Equation (5)) with linear scaling as the final post-processing step. Similar to GPLLS, linear scaling with Ridge regression will be applied to the *best-of-run* models.
(4) *GPLS*: GP with a scaled error measure. This method uses scaled RMSE as fitness function. Linear scaling with Ridge regression will be applied to *each GPSR model* before calculating RMSE using Equation (2).
(5) *GPCorLS*: a GP method maximising correlation coefficient with linear scaling. Similar to GPLS, linear scaling with Ridge regression will be applied to each GPSR model before evaluation. As mentioned earlier, linear scaling will not change the correlation between two variables, i.e. the fitness value of GPSR models. However, to confirm this point and also investigate how RMSE will change when using linear scaling in this case, we will perform a linear scaling prior to obtaining $1 - R^2$ using Equation (5) for *each GP individual*.

Note that this work does not attempt to compare with a state-of-the-art ERC learning method, e.g., learning coefficients with gradient descent [3] or using nonlinear least squares [14, 22], since the main aim of this work is to investigate whether and how the change of the fitness function from the error based to the correlation based can liberate coefficient learning in GPSR.

The parameter settings are summarised in Table 3 where most of them are typical settings for GPSR. The analytic quotient operator (*AQ*) in GPSR has shown to be a better choice than 'protected division' in generating models with good generalisation abilities [6, 18], thus *AQ* is employed to replace the commonly used protected division in the function set in this work. All the GP methods are implemented under the DEAP package [9].

Table 1: Benchmark Datasets.

Datasets	# Feature	#Instances	#Training	#Test
Tower	25	4999	3999	1000
Bodyfat	14	251	200	51
CalHouse	8	20640	16512	4128
BstHouse	13	506	404	102
Concrete	8	1030	824	206
Dowchem	57	1065	852	213
Parkinsons	18	5874	4699	1175
Yacht	6	307	245	62
Energy	8	767	613	154
WineRed	11	1598	1278	320
Keijzer5	3	11000	1000	10000
Korns8	5	20000	10000	100000

Table 2: The Synthetic Functions.

	Function	Traning/Test Sets
Keijzer5	$30xz/(x-10)v$	Training: $x, z = U[-1,1]$, $v = U[1,2]$ Test: $x, z = U[-1,1]$,$v = U[1,2]$
Korns8	$6.87(11 * \sqrt{7.23 * x_0 x_3 x_4})$	Training/Test: $x_1 - x_5 = U[-50,50]$

Table 3: Parameter Settings for GP Runs

Parameter	Value
Population Size	1024
Maximal #Generations	100
Initial Crossover & Mutation Rates	0.7 & 0.3
Elitism	10
Initial and Maximum Tree Depth	2-6&10
Initialisation	Ramped half-and-half
Tournament size	7
Basic Function Set	$+, -, *, AQ = x/\sqrt{(1+y)^2}$
number of runs	50

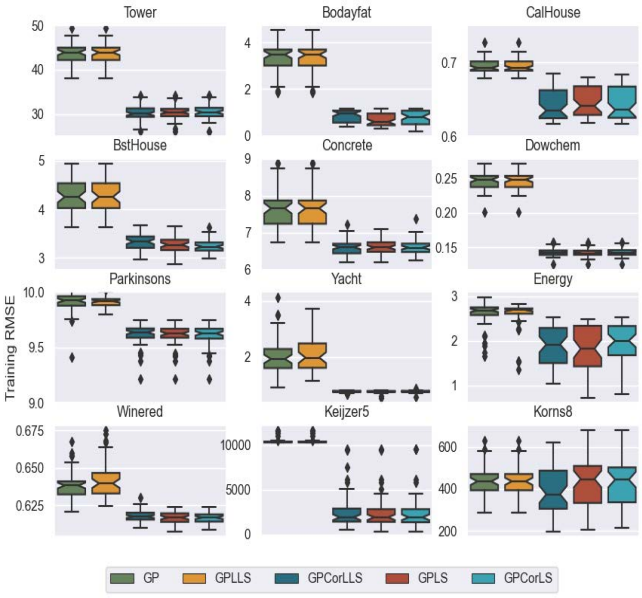
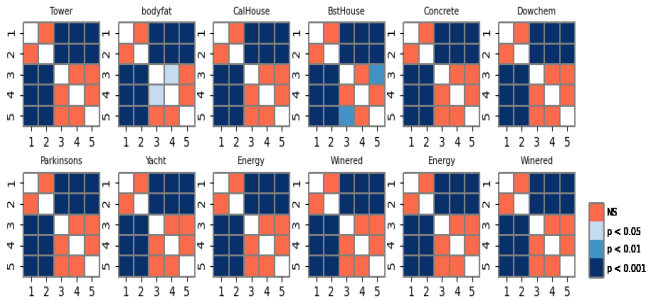


Figure 1: Boxplots on the *Training* *RMSE*s



1-5 stands for GP, GPLLS, GPCorLLS, GPLS, and GPCorLS, respectively.

Figure 2: Statistical Significance Test Results on the *Training* sets with P values.

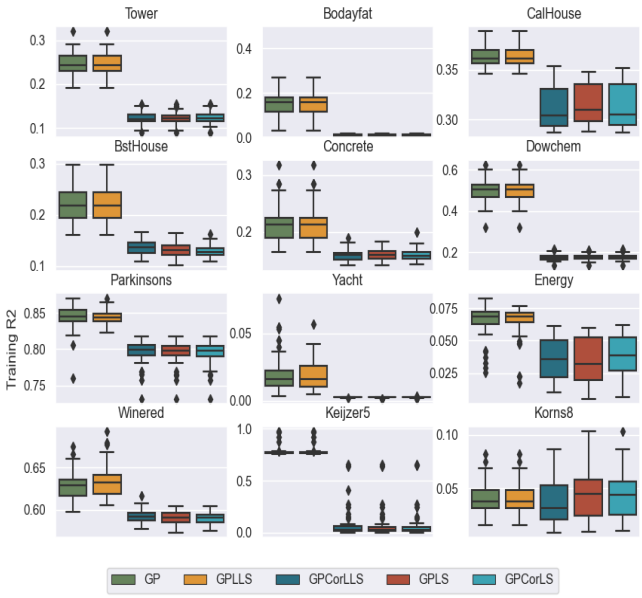


Figure 3: Boxplots on the Training R^2

5 RESULTS AND ANALYSIS

This section presents a comparison of the five GP methods with different evaluation strategies. RMSE and R^2 of the best-of-run GPSR models on training and test sets are reported. A non-parametric statistical significance test, i.e. the Friedman test with post-hoc Nemenyi test [21] at a significance level of 0.05 is conducted to compare training and test RMSEs among the five GP methods. The evolutionary plots of R^2 are also presented for a detailed examination of the learning process and corresponding generalisation performance of GP with the five different evaluation strategies. Further analysis on the size of the learnt models and the computational time is also presented.

5.1 Comparison on the Learning Performance

The distribution of training RMSEs of the best-of-run models of the five GP methods is shown in Figure 1. Statistical significance results are shown in heat maps with p values of the post-hoc Nemenyi test on each pair of methods in Figure 2, where 1-5 stands for GP, GPLLS, GPCorLLS, GPLS, and GPCorLS, respectively.

5.1.1 Comparing standard GP with the four GP methods with coefficient learning. As shown in Figure 1, the four GP methods with coefficient learning generally outperform GP on most of the 12 training sets. GPLLS slightly outperforms GP but not significantly on any of the training sets. The other three GP methods, GPCorLLS, GPLS, and GPCorLS all have much lower learning RMSEs than standard GP on 11 of the 12 datasets except for Korns8. On Korns8, the final three GP methods have a smaller best RMSE, but a larger variance than GP, and there is not any significant difference between the two groups of methods. The comparison results indicate that utilising linear scaling with RMSE or correlation can relieve GPSR from searching for coefficients thus improving its learning performance to some different degrees.

5.1.2 GPLS vs GPLLS and GPCorLS vs GPCorLLS. When comparing the learning performance of GPLS and GPLLS, as shown in Figure 1, GPLS has a much smaller training RMSE than GPLLS on 11 of the 12 training sets, except for Korns8. The comparison confirms that when evaluating models with their RMSEs, utilising linear scaling

prior to *every* evaluation brings more benefits. It encourages GP to focus on searching models with the desire structure, and allows models with a good structure but bad coefficients to survive.

Comparing the learning performance between GPCorLLS and GPCorLS, which are the two GP methods maximising correlations (minimising $1 - R^2$) between the model predictions and the target variable, there is not much difference between the two methods. This is consistent with the assumption that performing linear scaling prior to the evaluation of each GPSR model is not necessary if the correlation-based measures are used as fitness function. The small difference between the two methods is due probably to the randomness brought by linear scaling.

The two sets of comparisons confirm that linear scaling is needed prior to every evaluation when utilising RMSE as the fitness function, and this should apply to other error-based fitness evaluations while performing one-step post-hoc linear scaling on the best-of-run model at the end of the evolutionary process is sufficient when utilising the correlation-based fitness function.

5.1.3 Comparison between GP with RMSE and GP with correlation-based fitness functions. Comparing GPCorLS with GPLS, on all the 12 datasets, GPCorLS has a similar training RMSE to that of GPLS. There is not any significant difference between the learning performance of the two methods. A similar pattern can be found when comparing the learning performance of GPCorLLS and GPLS. However, regarding the other set of comparison between GPCorLLS and GPLLS, on 11 of the 12 training sets except for Korns8, GPCorLLS obtains significantly smaller RMSEs than GPLLS, which are all significant. On Korns8, the two methods have similar performance.

The similar learning performance of GPCorLS/GPCorLLS vs. GPLS, and the notably better learning performance of GPCorLLS over GPLLS indicate that utilising correlation coefficient as the fitness function will be less demanding for linear scaling than the commonly used RMSE, without sacrificing the learning performance.

5.1.4 Comparing the Training R^2 Distributions. We also present the distribution of 50 training R^2s of evolved models in the five GP methods in Figure 3. Note that a larger R^2 indicates a larger correlation between the outputs of a model and the target variable, while the level of R^2 also can tell the difficulty of the problems. Generally, a relatively smaller R^2 of the examined algorithms in one dataset indicates it is more difficult, e.g. among the ten datasets, Parkinson and WineRed are generally the most difficult problems for the five GP methods.

As shown in Figure 3, on the training sets, GP and GPLLS have very much the same R^2 distribution (note that the small difference in the two boxplots on some training sets, e.g. Yacht and WineRed, is due to the small difference in the boxplot outlier identification). The R^2s of GP and GPLLS are usually much smaller than those of the other three GP methods on 11 of the 12 training sets except for Korns8. This pattern is consistent with that on RMSEs. More specifically, for a GP method that obtains the smallest RMSEs, e.g. GPLS on Bodyfat and GPCorLLS on Korns8, it also obtains the largest R^2/correlation coefficient on the corresponding training set. Correlation coefficients and RMSEs are consistent indicators of the training performance. It also indicates that correlation which is the

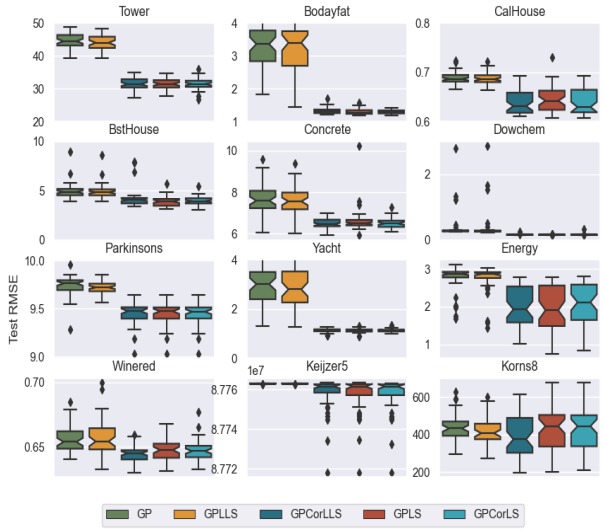
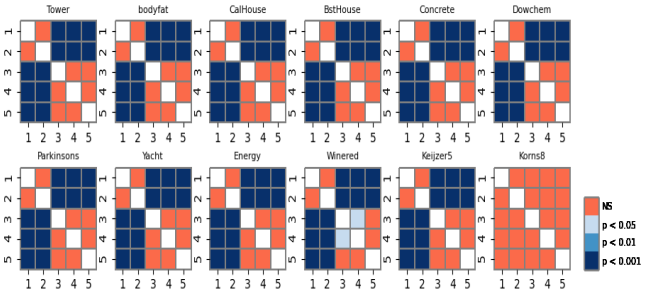


Figure 4: Boxplots on the *Test* *RMSE*



1-5 standards for GP, GPLLS, GPCorLLS, GPLS, and GPCorLS, respectively.

Figure 5: Statistical Significance Test Results on *Test* sets with P values.

degree of association between the predictions and the target outputs can be used as a measure of the degree of coincidence.

5.2 Comparisons on the Test Performance

The distribution of the test RMSEs of the best-of-run models in the five GP methods is shown in Figure 4. The statistical significance test results are shown in Figure 5. From these two figures, we can easily see that the overall pattern on the test sets is similar to that on the training sets. The four GP with coefficient learning methods generally outperform GP on all the 12 test sets. GPLLS outperforms GP slightly but not significantly on any of the test sets. The other three GP methods, GPCorLLS, GPLS, and GPCorLS have much lower test RMSEs than standard GP on 11 of the 12 datasets except for Korns8. On Korns8, there is not any significant difference among all the five GP methods.

The comparisons between the generalisation performance of the two GP with RMSE methods, and the two GP with correlation coefficient methods also show a similar pattern to that on the training sets. As shown in Figure 4, GPLS generally achieves notably smaller

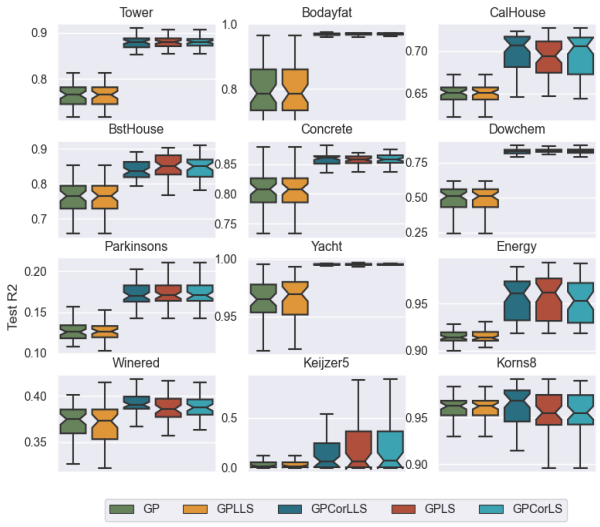


Figure 6: Boxplots on the Test R^2

RMSEs than GPLLS on all the test sets where on 11 of the 12 test sets are significant but not on Korns8. GPCorLS and GPCorLLS also achieve similar generalisation performance as what they have shown on the training sets. The advantage of GPLS over GPLLS on the test sets confirms that when using RMSE as the fitness function utilising linear scaling prior to every evaluation also produces more benefits in enhancing the generalisation of GP than just using linear scaling once at the end of the evolutionary process. But this is not the case when using correlation coefficient as the fitness function. In the latter case, performing a final step of linear scaling is sufficient to bring notable improvement in the generalisation performance of GP.

Regarding the comparison between the generalisation performance of GP with RMSE and GP with correlation-based fitness functions, Figure 4 clearly shows that the three methods GPLS, GPCorLS and GPCorLLS obtain similar generalisation performance. This confirms that to obtain the same level of generalisation benefits, utilising correlation-based fitness function with one-step (post-processing) linear scaling is more efficient than RMSE with linear scaling for each individual.

5.2.1 The Test R^2 Distributions. We also present the distribution of test R^2s of evolved models in the five GP methods in Figure 6. Similar to the comparisons on RMSEs, GPLLS and GP have the same level of R^2, while GPCorLLS, GCorLS, and GPLS obtain similar R^2s, which are generally much higher than those of GPLLS and GP, i.e. a much larger correlation value. Moreover, the GP method which obtains a lower test RMSE also has a larger R^2s on the test sets, e.g. GPLS on BstHouse and GPCorLS on WineRed and Korns8, and this pattern can be found on all the 12 test sets and also consistent with what we have found on the training sets.

Comparing the R^2s on the 12 test sets, the five GP methods obtain relatively smaller R^2s on DowChem, Parkinsons, WineRed, and Keijzer5 than on the other eight test sets. This is particularly the case on Keijzer5 where the five GP methods have a median R^2 close to 0, which means a small correlation. Among these four test sets, linear scaling helps to improve the correlation values most on Dowchem from a median value of 0.5 in GP/GPLLS to around 0.9 in the other three GP methods. Linear scaling achieves the smallest effort on Korns8. This is probably due to GP already obtaining a relatively small R^2, there is not much space for linear scaling to improve it. This also explains why the four GP methods with linear scaling cannot achieve much generalisation gain on Korns8.

5.3 Further Analysis of Evolutionary Training and Test R^2

To examine the learning and test performance in more detail, Figures 7 and 8 show the evolving plots of the best-of-generation models on the training and test R^2 respectively. Note that here in the two figures, only the performance of models in four GP methods have been presented as standard GP and GPLLS share the same evolutionary plots on R^2.

As shown in Figure 7, the five GP methods generally obtain stable learning performance with a small variance among 50 different runs during the evolutionary process on all the training sets. Compared with GP/GPLLS, the other three GP methods including GPLS with help of linear scaling and the two GP with correlation-based fitness functions, have a much higher R^2 at the very beginning of the evolutionary process. Moreover, they also have a much higher convergence rate on the learning performance than GP/GPLLS on most test sets. This is particularly the case on Bodyfat and Yacht.

The evolutionary plots on the test sets have a similar pattern to that on the training sets. The five GP methods generally obtain stable generalisation performance among different runs on most of the 12 test sets except for Bodyfat, BstHouse and Keijzer5. While on Bodyfat GPCorLS and GPCorLLS have a higher R^2 at the later generations than the very first generations, on BstHouse, the test R_2 plots of the two methods are relatively fluctuating. On Keijzer5, the three GP methods, GPCorLS, GPCorLLS and GPLS, have a large variance on the test R^2 among different runs, which indicates the less stable generalisation errors.

As shown in Figures 7 and 8, for the learning and test evolutionary plots of the two synthetic datasets, i.e. Keijzer5 and Korns8, linear scaling plays a different role. On Keijzer5, the four GP methods have a similar training/test R^2, and the difference between GP/GPLS and the other three GP methods becomes larger along the evolutionary process, which means linear scaling works at the later stage of the evolutionary process. While on Korns8, the case is completely different. The effort of linear scaling can be found in the first several generations of the evolutionary process. The three GP with linear scaling methods have a much larger R^2 than GP/GPLLS, but the difference becomes much smaller after a few generations and not much difference can be found at the later stage of the evolutionary process.

5.4 Comparisons on the Model Size and the Computational Cost

The model size of the learnt/best-of-run GPSR models and computational cost of the learning process in the five GP methods are summarised in Table 4. Note that, since GPLLS performs linear scaling as a single post-processing step, it does not change the learnt

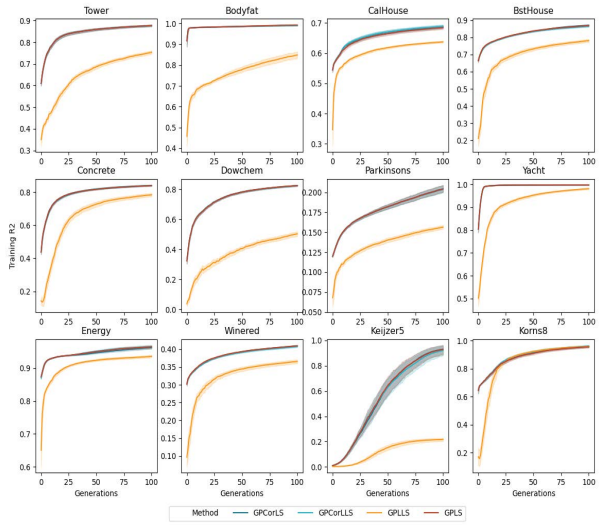


Figure 7: The Training R^2

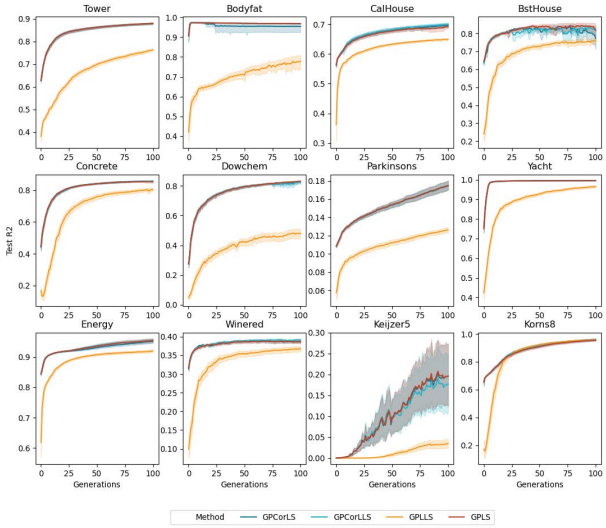


Figure 8: The Test R^2

model thus it shares the same model size with GP. In addition, the difference in their computational cost is generally small and can be almost ignored. So here, we present their results together.

We can see from Table 4, on most datasets except for Bodyfat and Yacht, the other three GP methods have a slightly larger model size than that of GP/GPLLS. As aforementioned, on Bodyfat and Yacht, the three GP with linear scaling methods converge early. This also explains why their learnt models on these two datasets are generally smaller than that in GP/GPLLS.

Comparing the computational time of the five GP methods, as shown in Table 4, GP/GPLLS generally spends a shorter computation time than the other three methods. This is not unexpected. For the other three GP methods, GPCorLLS spends additional effort on obtaining the correlation coefficient and performing linear scaling

Table 4: Model Size (#Node) and Computational Time (in Second)

Method		Model Size	Time		Model Size	Time
GP/GPLLS	Tower	112.59±26.59	44.53±25.61	DowChem	89.08±17.92	5.95±6.5
GPCorLLS		118.59±32.21	39.34±26.03		114.18±30.8	7.37±11.76
GPLS		121.2±32.58	45.78±30.79		117.98±27.92	14.0±19.56
GPCorLS		118.55±32.64	47.1±26.72		113.16±21.63	17.48±23.17
GP/GPLLS	Bodyfat	111.2±34.45	6.17±6.2	Parkinsons	168.55±56.24	44.17±28.29
GPCorLLS		109.29±28.14	9.3±17.41		207.57±44.41	44.17±27.4
GPLS		104.96±22.96	15.52±16.39		212.88±48.31	44.33±30.42
GPCorLS		111.94±30.07	6.31±7.37		213.12±47.0	49.83±29.07
GP/GPLLS	CalHouse	106.67±21.12	129.09±158.77	Yacht	172.47±45.44	15.6±20.04
GPCorLLS		131.0±40.63	131.25±154.7		111.94±23.68	6.76±5.94
GPLS		133.12±32.18	199.01±204.39		110.88±22.38	15.55±18.62
GPCorLS		142.55±45.86	193.53±199.48		110.35±22.95	13.98±17.83
GP/GPLLS	BstHouse	116.63±23.44	7.96±10.81	Energy	129.33±33.3	13.56±20.04
GPCorLLS		129.98±31.91	9.51±12.42		152.31±35.75	18.16±20.66
GPLS		124.47±26.96	22.16±22.0		151.73±39.87	23.59±25.55
GPCorLS		133.57±32.67	10.8±17.37		149.82±39.73	20.55±24.08
GP/GPLLS	Concrete	114.31±25.05	9.53±11.38	WineRed	112.27±27.63	12.47±16.15
GPCorLLS		141.24±33.98	15.04±18.56		133.82±30.6	20.09±23.92
GPLS		141.16±34.87	29.62±27.96		138.39±31.07	31.21±29.74
GPCorLS		133.08±31.24	11.3±13.45		138.02±34.51	25.66±23.84
GP/GPLLS	Keijzer5	337.53±60.8	46.31±25.72	Korns8	140.51±31.59	53.37±30.62
GPCorLLS		330.67±90.44	50.72±22.93		165.69±48.9	54.48±28.35
GPLS		328.31±90.08	53.01±30.3		182.67±72.35	45.52±33.14
GPCorLS		328.92±95.88	50.18±28.2		175.82±62.83	46.08±25.46

for the evolved models while the other two GP methods perform linear scaling for each model during the evolutionary process which is generally more time-consuming. Among the three GP methods which notably enhance the performance of GP, GPCorLLS is the most efficient. It has a shorter computational time than GPCorLS and GPLS on nine of the 12 datasets. On Bodyfat, Concrete and Korns8, GPCorLS has a smaller computational cost than GPCorLLS. This could be due to the (slightly) smaller models in GPCorLS which save the evaluation time. The large difference in the evaluation time for small and large models also explains why the standard deviation of the computation time on some datasets with a large number of instances, e.g. CalHouse, is so large. In general, the slightly higher computational time in GP than GP is worth considering the significant improvement in the learning and generalisation performance of GP.

Comparing GPCorLLS with GPLS which are the two GP methods we recommend when considering linear scaling for learning coefficient for GP, on 11 of the 12 datasets except for Korns8, GPCorLLS is more efficient than GPLS.

6 CONCLUSIONS AND FUTURE WORKS

This work investigated and compared two sets of evaluation strategies combined with linear scaling for GPSR to relieve it from learning coefficients using genetic operators and focused more on searching symbolic regression models with the desired structure. The investigations in this work confirm that both, an error measure with linear scaling prior to each evaluation and the Pearson correlation coefficient based measure to search for models with a similar structure to the desired models, can achieve the goal of relieving GP from learning coefficients with enhanced learning and generalisation ability. A comparison on model size and computational costs confirms that freeing/relieving GPSR from learning coefficient leaves more space for searching different model structures. This way GP learns models with a slightly higher complexity but better quality. Compared to RMSE which needs to perform linear scaling prior to every evaluation, the correlation-based fitness function only needs to perform a one-step linear scaling of evolved models, which saves linear scaling effort and is more efficient.

REFERENCES

[1] Francesco Archetti, Stefano Lanzeni, Enza Messina, and Leonardo Vanneschi. 2007. Genetic programming for computational pharmacokinetics in drug discovery and development. *Genetic Programming and Evolvable Machines* 8, 4 (2007), 413–432.

[2] Bice Cavallo. 2020. Functional relations and Spearman correlation between consistency indices. *Journal of the Operational Research Society* 71, 2 (2020), 301–311.

[3] Qi Chen, Bing Xue, and Mengjie Zhang. 2015. Generalisation and domain adaptation in GP with gradient descent for symbolic regression. In *2015 IEEE congress on evolutionary computation (CEC)*. IEEE, 1137–1144.

[4] Qi Chen, Bing Xue, and Mengjie Zhang. 2019. Improving Generalisation of Genetic Programming for Symbolic Regression with Angle-Driven Geometric Semantic Operators. *IEEE Transactions on Evolutionary Computation* 23, 3 (2019), 488–502.

[5] Qi Chen, Bing Xue, and Mengjie Zhang. 2019. Improving Generalization of Genetic Programming for Symbolic Regression With Angle-Driven Geometric Semantic Operators. *IEEE Transactions on Evolutionary Computation* 23, 3 (2019), 488–502. https://doi.org/10.1109/TEVC.2018.2869621

[6] Qi Chen, Bing Xue, and Mengjie Zhang. 2020. Improving symbolic regression based on correlation between residuals and variables. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*. 922–930.

[7] Davide Chicco, Matthijs J Warrens, and Giuseppe Jurman. 2021. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science* 7 (2021), e623.

[8] Grant Dick. 2022. Genetic Programming, Standardisation, and Stochastic Gradient Descent Revisited: Initial Findings on SRBench. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (Boston, Massachusetts) *(GECCO '22)*. Association for Computing Machinery, New York, NY, USA, 2265–2273. https://doi.org/10.1145/3520304.3534040

[9] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. 2012. DEAP: Evolutionary Algorithms Made Easy. *Journal of Machine Learning Research* 13 (jul 2012), 2171–2175.

[10] Nelson Fumo and MA Rafe Biswas. 2015. Regression analysis for prediction of residential energy consumption. *Renewable and sustainable energy reviews* 47 (2015), 332–343.

[11] Nathan Haut, Wolfgang Banzhaf, and Bill Punch. 2023. Correlation Versus RMSE Loss Functions in Symbolic Regression Tasks. In *Genetic Programming Theory and Practice XIX*. Springer, 31–55.

[12] Quang Nhat Huynh, Shelvin Chand, Hemant Kumar Singh, and Tapabrata Ray. 2018. Genetic Programming With Mixed-Integer Linear Programming-Based Library Search. *IEEE Transactions on Evolutionary Computation* 22, 5 (2018), 733–747.

[13] Maarten Keijzer. 2003. Improving symbolic regression with interval arithmetic and linear scaling. In *Genetic programming*. Springer, 70–82.

[14] Michael Kommenda, Bogdan Burlacu, Gabriel Kronberger, and Michael Affenzeller. 2020. Parameter identification for symbolic regression using nonlinear least squares. *Genetic Programming and Evolvable Machines* 21, 3 (2020), 471–501.

[15] John R Koza. 1992. *Genetic programming: on the programming of computers by means of natural selection*. Vol. 1. MIT press.

[16] Tarald O Kvålseth. 1985. Cautionary note about R 2. *The American Statistician* 39, 4 (1985), 279–285.

[17] M. Lichman. 2013. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[18] Ji Ni and Peter Rockett. 2014. Tikhonov regularization as a complexity measure in multiobjective genetic programming. *IEEE Transactions on Evolutionary Computation* 19, 2 (2014), 157–166.

[19] Michael O'Neill, Leonardo Vanneschi, Steven Gustafson, and Wolfgang Banzhaf. 2010. Open issues in genetic programming. *Genetic Programming and Evolvable Machines* 11, 3 (2010), 339–363.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[21] Dulce G Pereira, Anabela Afonso, and Fátima Melo Medeiros. 2015. Overview of Friedman's test and post-hoc analysis. *Communications in Statistics-Simulation and Computation* 44, 10 (2015), 2636–2653.

[22] Peter Rockett. 2022. Constant optimization and feature standardization in multiobjective genetic programming. *Genetic Programming and Evolvable Machines* 23, 1 (2022), 37–69.

[23] Conor Ryan and Maarten Keijzer. 2003. An analysis of diversity of constants of genetic programming. In *European Conference on Genetic Programming*. Springer, 404–413.

[24] Patrick Schober, Christa Boer, and Lothar A Schwarte. 2018. Correlation coefficients: appropriate use and interpretation. *Anesthesia & Analgesia* 126, 5 (2018), 1763–1768.

[25] Dominik Sobania, Martin Briesch, David Wittenberg, and Franz Rothlauf. 2022. Analyzing optimized constants in genetic programming on a real-world regression problem. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 606–607.

[26] Alexander Topchy, William F Punch, et al. 2001. Faster genetic programming based on local gradient search of numeric leaf values. In *Proceedings of the genetic and evolutionary computation conference (GECCO-2001)*, Vol. 155162. Morgan Kaufmann San Francisco, CA.

[27] Marco Virgolin, Tanja Alderliesten, and Peter AN Bosman. 2019. Linear scaling with and within semantic backpropagation-based genetic programming for symbolic regression. In *Proceedings of the genetic and evolutionary computation conference*. 1084–1092.

[28] David R White, James Mcdermott, Mauro Castelli, Luca Manzoni, Brian W Goldman, Gabriel Kronberger, Wojciech Jaśkowski, Una-May O'Reilly, and Sean Luke. 2013. Better GP benchmarks: community survey results and proposals. *Genetic Programming and Evolvable Machines* 14, 1 (2013), 3–29.

[29] Dabao Zhang. 2017. A coefficient of determination for generalized linear models. *The American Statistician* 71, 4 (2017), 310–316.

[30] Mengjie Zhang and Will Smart. 2004. Genetic programming with gradient descent search for multiclass object classification. In *European Conference on Genetic Programming*. Springer, 399–408.