

# A Novel Symbolic Regressor Enhancer Using Genetic Programming

Tu-Chin Chiang

Taiwan Evolutionary Intelligence LAB  
Department of Electrical Engineering  
National Taiwan University  
r11921038@ntu.edu.tw

Chi-Hsien Chang

Taiwan Evolutionary Intelligence LAB  
Department of Electrical Engineering  
National Taiwan University  
d07921004@ntu.edu.tw

Tian-Li Yu

Taiwan Evolutionary Intelligence LAB  
Department of Electrical Engineering  
National Taiwan University  
tianliyu@ntu.edu.tw

**Abstract**—This paper proposes a framework combining genetic programming (GP) with other symbolic regression (SR) methods, called the symbolic regressor enhancer (SRE). The basic idea is to use the syntax tree of the expression obtained from other SR methods to improve both the efficiency and the quality of the evolutionary procedure. Specifically, this paper investigates on the different ways of hybridization, elitist selection, and ranging crossover to assemble the proposed SRE. The effectiveness of SRE is demonstrated with the Taylor polynomial, the fast function extraction, and the GP-based SR methods, including Operon, the GP variant of gene-pool optimal mixing evolutionary algorithm, the epsilon-lexicase selection, and gplearn. Out of 28 benchmarks from the SR benchmark and the Feynman SR database, the statistical test indicates that SRE applied to each selected SR method significantly outperforms the respective SR method in at least 8 and at most 21 benchmarks.

**Index Terms**—Symbolic regression, genetic programming

## I. INTRODUCTION

Symbolic regression (SR) aims to discover optimal mathematical expressions that capture relationships within the data in syntax format [1], [2]. Genetic programming (GP) [3] is a commonly-used technique for finding SR models [4]–[7]. GP leverages evolutionary mechanisms, such as crossover and mutation, to iteratively generate individuals to find a model that fits the dataset.

In recent years, a promising research area has focused on integrating GP with traditional SR methods [8]–[11]. Among these methods, the Taylor GP (TaylorGP) [9] utilizes the Taylor polynomial (TP), while the GP based evolution of models of models (EMM-GP) [11] incorporates the fast function extraction (FFX) [12] within GP. TaylorGP calculates TPs through finite differences and employs the low-order segments of TP and the features extracted from TP to guide the evolutionary process. EMM-GP utilizes FFX to generate model sets and employs these models as the essential building blocks added into the function set of GP.

Building on insights from the related literature, we explore the potential for improvement by leveraging the entire syntax structure of the returned model through the subsequent evolutionary procedure. This paper proposes a new framework combining GP with other SR methods, called the symbolic regressor enhancer (SRE)<sup>1</sup>. The proposed framework

hybridizes the expression derived from other SR methods with random programs during the population initialization of tree-based GP. This hybridization initiates the evolution with the knowledge learned from other SR methods and aims to improve the efficiency and the quality of the evolutionary procedure. Furthermore, to strengthen the capability of searching for the superior offspring during the evolutionary procedure, we incorporate the elitist selection and the ranging crossover [6] into the SRE framework. The elitist selection preserves the best individuals from the previous generation, while the ranging crossover replaces a tree node with a subtree matching the desired output range.

The effectiveness of SRE is assessed through its application to both traditional and GP-based SR methods. The traditional methods used in our experiments contain TP and FFX. The GP-based methods include Operon [7], the GP variant of gene-pool optimal mixing evolutionary algorithm (GP-GOMEA) [4], the epsilon-lexicase selection (EPLEX) [5], and gplearn [13], which are widely used in contemporary SR research [2], [14], [15].

The rest of this paper is organized as follows. Section II introduces the proposed SRE framework. Section III explores the variants of the components in SRE. Section IV details the setup of the experiments, the results, and the discussions. Finally, Section V concludes the paper.

## II. METHODOLOGY

The SRE framework, as shown in Fig. 1, consists of two phases: the population generation and the evolutionary procedure. In the population generation phase, an SR method is selected to be combined with GP to form an initial population. Each program in the population is generated by combining the expression  $E$  from the SR method and a random program  $G_i$  from GP through the hybridization function  $H(E, G_i)$ . This hybridization produces a tree-structured program with  $E$  and  $G_i$  as the left and right subtrees of  $H$ , respectively. An overview of the hybridization and an example are illustrated in Fig. 2, where  $H$  is multiplication ( $H_\times$ ),  $E$  is  $2x + 3$ ,  $G_i$  is  $\sin(x)$ , and  $x$  represents the variable. The resulting hybrid program is  $H_\times(E, G_i) = (2x + 3) \times \sin(x)$ . The introduction of  $E$  aims to capture the underlying syntax format of the data,

<sup>1</sup>The source code is available at <https://github.com/tuchin32/teil-sre.git>.

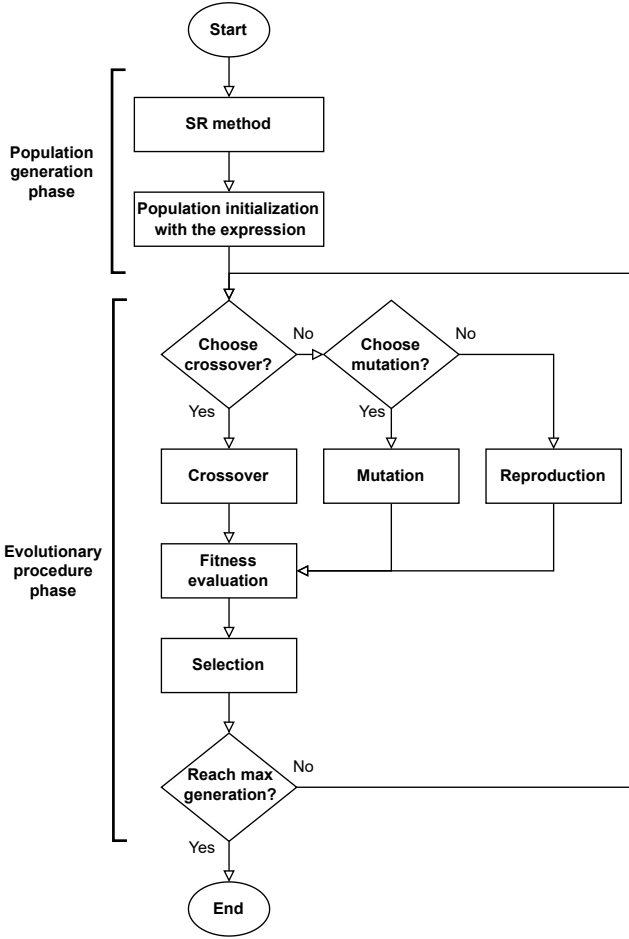


Fig. 1. The SRE framework flowchart includes two phases: the population generation and the evolutionary procedure.

and SRE utilizes the learned syntax format in anticipation of the enhanced performance.

In the evolutionary procedure phase, to leverage the learned structure of  $E$  from other SR methods, we simply forbid the node or nodes representing  $H$  from changing during each recombination. Furthermore, several evolutionary mechanisms are utilized to enhance superior offspring production during selection and recombination. By employing the appropriate mechanisms, we strengthen the capability of the SRE framework. The choices of alternative hybridization functions in the first phase and evolutionary mechanisms in the second phase are investigated in the following section.

### III. SRE ALGORITHM

Based on the aforementioned framework, we determine appropriate hybridization functions and evolutionary mechanisms to define the SRE algorithm. We initially conduct pre-experiments to test TP as the SR method with a relatively small number of function evaluations (NFE). The application of SRE to other SR methods is extended in the subsequent section to demonstrate the effectiveness of our choices. In our pre-experiments, we first describe the benchmark problems

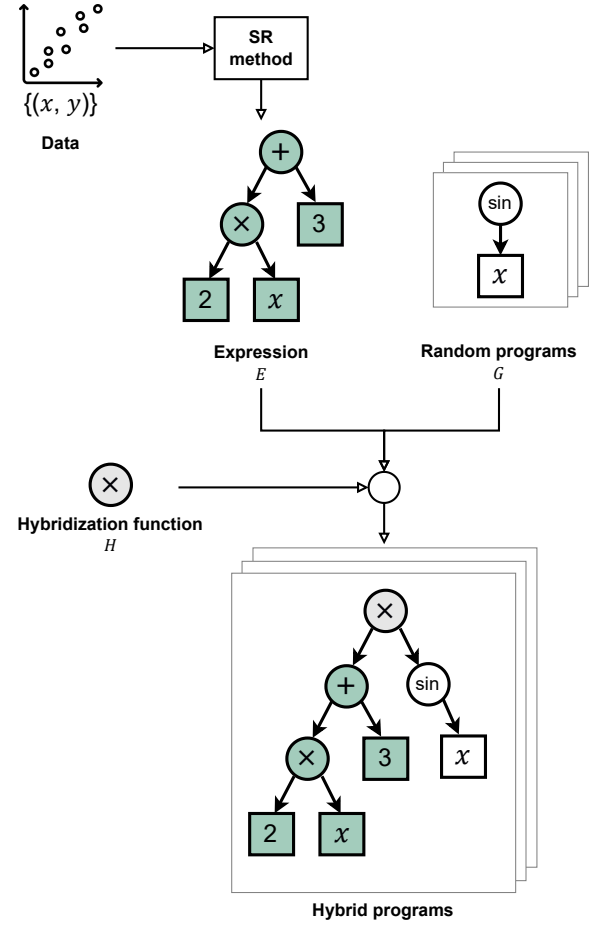


Fig. 2. Overview of the hybridization framework. The expression  $E$ , sourced from other SR methods, is integrated with random programs  $G$  through the hybridization function  $H$  to form a population composed of hybrid programs.

and the setup of this study. Then, we provide the results and discussions on the chosen candidates.

In this study, 28 commonly-used benchmark problems with the known ground truth models [2], [9], [16], [17] from the SR benchmark [18] and the Feynman SR database [19] are considered. The benchmark problems are detailed in Table I. Additionally, the parameter settings for the variants of SRE are provided in Table II. Different sets of population size and maximum generation are employed in the following studies and specified in the respective description. As for the function set,  $\div^*$  is the protected division defined by

$$\div^*(x, y) = \begin{cases} \frac{x}{y} & \text{if } |y| \geq 0.001 \\ 1 & \text{otherwise,} \end{cases}$$

and  $\exp^*$  is the protected exponential function defined by

$$\exp^*(x) = \begin{cases} e^x & \text{if } x \leq 10 \\ e^{10} & \text{otherwise.} \end{cases}$$

The fitness of each program is evaluated using the mean squared error (MSE). For each benchmark, the dataset is randomly split into 70% for training and 30% for testing.

TABLE I

THE PROBLEMS FROM THE SR BENCHMARK AND THE FEYNMANN SR DATABASE.

WE CATEGORIZE THE BENCHMARKS INTO TWO GROUPS: POLYNOMIAL-EXPRESSIBLE AND POLYNOMIAL-INEXPRESSIBLE.

THE NOTATION  $t: [a, b]$  REPRESENTS THE BOUNDARY OF THE UNIFORM DISTRIBUTION OF THE VARIABLE  $t$ , WHERE  $a$  AND  $b$  DEFINE THE RANGE OF  $t$ .

Category	Problem	Name	Expression	Dataset	Samples
Polynomial expressible	$f_1$	Nguyen-4	$x^6 + x^5 + x^4 + x^3 + x^2 + x$	$x: [-1, 1]$	20
	$f_2$	Nguyen-3	$x^5 + x^4 + x^3 + x^2 + x$	$x: [-1, 1]$	20
	$f_3$	Nguyen-2	$x^4 + x^3 + x^2 + x$	$x: [-1, 1]$	20
	$f_4$	Nguyen-1	$x^3 + x^2 + x$	$x: [-1, 1]$	20
	$f_5$	Koza-3	$x^6 - 2x^4 + x^2$	$x: [-1, 1]$	20
	$f_6$	Koza-2	$x^5 - 2x^3 + x$	$x: [-1, 1]$	20
	$f_7$	Nguyen-12	$x^4 - x^3 + y^2/2 - y$	$x: [-1, 1]$	400
	$f_8$	Keijzer-15	$x^3/5 + y^3/2 - y - x$	$x: [-1, 1]$	400
Polynomial inexpressible	$f_9$	Korns-11	$6.87 + 11 \cos(7.23x^3)$	$x: [-1.3, 1.3]$	40
	$f_{10}$	Nguyen-5	$\sin(x^2) \cos(x) - 1$	$x: [-1.6, 1.6]$	40
	$f_{11}$	Nguyen-6	$\sin(x) + \sin(x + x^2)$	$x: [-1, 1]$	20
	$f_{12}$	Keijzer-11	$xy + \sin((x-1)(y-1))$	$x, y: [-1, 1]$	400
	$f_{13}$	Keijzer-13	$6 \sin(x) \cos(y)$	$x, y: [-1, 1]$	400
	$f_{14}$	Nguyen-9	$\sin(x) + \sin(y^2)$	$x, y: [-1, 1]$	400
	$f_{15}$	Nguyen-10	$2 \sin(x) \cos(y)$	$x, y: [-1, 1]$	400
	$f_{16}$	Vladislavleva-1	$\frac{\exp(-(x-1)^2)}{1.2+(y-2.5)^2}$	$x, y: [-1, 1]$	400
	$f_{17}$	Korns-2	$0.23 + 14.2(v+y)/(3w)$	$v, y: [-1, 1], w: [1, 3]$	1000
	$f_{18}$	Vladislavleva-5	$30((x-1)(z-1))/(y^2(x-10))$	$x, z: [0, 4], y: [1, 3]$	1000
	$f_{19}$	Keijzer-7	$\ln(x)$	$x: [0, 4]$	20
	$f_{20}$	Nguyen-8	$\sqrt{x}$	$x: [0, 4]$	20
	$f_{21}$	Nguyen-11	$x^y$	$x, y: [1, 5]$	400
	$f_{22}$	Keijzer-5	$30xz/((x-10)y^2)$	$x, z: [-2, 2], y: [1, 3]$	1000
	$f_{23}$	Feynman-I.6.2	$f = \frac{\exp(-(\theta/\sigma)^2/2)}{\sqrt{2\pi}\sigma}$	$\theta, \sigma: [1, 3]$	100000
	$f_{24}$	Feynman-I.26.2	$\theta_1 = \arcsin(n \times \sin(\theta_2))$	$n: [0, 1], \theta_2: [1, 5]$	100000
	$f_{25}$	Feynman-I.34.14	$\omega = \frac{(1+v/c)}{\sqrt{1-v^2/c^2}} \omega_0$	$c: [3, 10], v: [1, 2], \omega_0: [1, 5]$	100000
	$f_{26}$	Feynman-I.37.4	$Int = I_1 + I_2 + 2\sqrt{I_1 I_2} \cos(\delta)$	$I_1, I_2, \delta: [1, 5]$	100000
	$f_{27}$	Feynman-I.47.23	$c = \sqrt{\gamma \times pr/\rho}$	$\gamma, pr, \rho: [1, 5]$	100000
	$f_{28}$	Feynman-II.24.17	$k = \sqrt{\frac{\omega^2}{c^2} - \frac{\pi^2}{d^2}}$	$\omega: [4, 6], c: [1, 2], d: [2, 4]$	100000

TABLE II  
THE PARAMETER SETTINGS FOR GP APPROACHES.

Parameter	Value
Runs per benchmark	30
Population size	200, 400, 500
Maximum generation	20, 40, 50
Function set	$+, -, \times, \div, *, \sin, \cos, \exp^*$
Terminal set	variable, constant
Fitness function	MSE

To find an appropriate hybridization function and evolutionary mechanisms, we sequentially investigate each component. First, three hybridization functions are employed with standard evolutionary mechanisms. Then, utilizing the selected hybridization function, we investigate techniques aiming at facilitating progressive improvement in fitness and consider the semantics of subtrees for the recombination procedure.

#### A. The Choice for The Hybridization Function

The hybridization function  $H$  determines how to combine the expression  $E$  and the random program  $G_i$ . Here we investigate the effects of using addition ( $H_+$ ), multiplication ( $H_\times$ ), and the transformation ( $H_T$ ) as the hybridization functions.

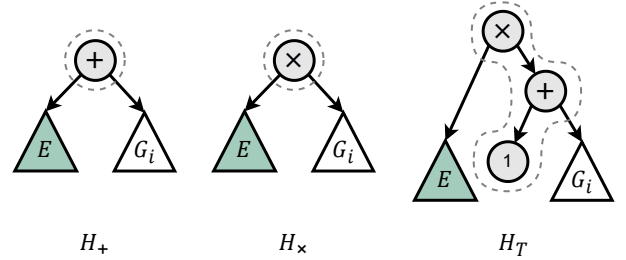


Fig. 3. Tree structures depict the expression  $E$ , a random program  $G_i$ , and the three hybridization functions:  $H_+$ ,  $H_\times$ , and  $H_T$ . Dotted line enclosed areas correspond to the structure of the hybridization functions.

The tree structures of the three hybridization functions are illustrated in Fig. 3, and  $H_T$  is defined by

$$H_T(E, G_i) = E \times (1 + G_i).$$

Note that the potential candidates for  $H$  extend beyond the three hybridization functions we choose.

The investigation results of two metrics, the coefficient of determination ( $R^2$ ) and the interquartile range (IQR), are presented in Fig. 4 and Table III. The results show that  $H_\times$  has an advantage over  $H_+$  and  $H_T$ . The population size is 200 and the maximum generation is 20 in this study.

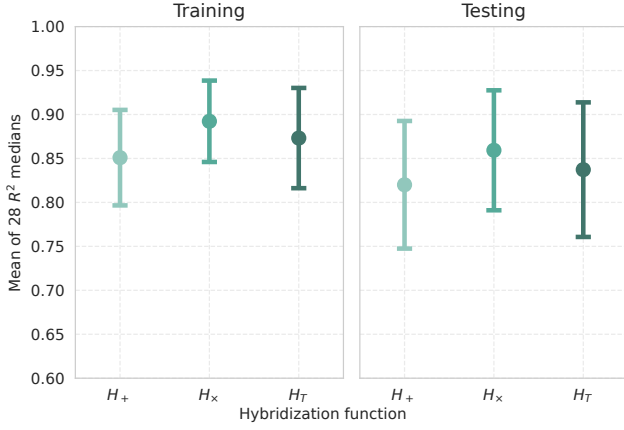


Fig. 4. Results on the three hybridization functions. Points indicate the mean of the 28  $R^2$  medians across all benchmarks, and bars show the corresponding standard deviations of each point.

TABLE III

MEANS OF THE 28 RANKINGS IN TERMS OF IQR ACROSS ALL BENCHMARKS OF THE THREE HYBRIDIZATION FUNCTIONS ON THE TESTING DATASET. A HIGHER RANKING MATCHES A SMALLER IQR.

	$H_+$	$H_\times$	$H_T$
Means of the rankings	2.00	<b>1.86</b>	2.14

First, the means of 28  $R^2$  medians across all benchmarks are depicted in Fig. 4 for both training and testing stages, with each  $R^2$  median computed from 30 independent runs. The mean of  $R^2$  medians of  $H_\times$  is 1.05 times greater than that of  $H_+$  and 1.02 times greater than that of  $H_T$ . These results indicate that  $H_\times$  demonstrates relative generalizability across the benchmarks. Furthermore, given that  $H_\times$  exhibits superior distribution in terms of  $R^2$  medians, we assess the consistency by computing rankings based on the IQR of 30  $R^2$  scores from each independent run on the testing dataset per benchmark. A smaller IQR corresponds to a higher ranking and generally indicates more consistent performance. The results, as shown in Table III, indicate that  $H_\times$  achieves the highest mean rankings among the three hybridization functions. Since  $H_\times$  demonstrates relative generalizability and more consistent performance, we select multiplication ( $H_\times$ ) as the hybridization function for our proposed method.

### B. The Choices for The Evolutionary Mechanisms

First, to aim for progressive improvement in fitness and retain elitist individuals, we investigate the applicability of the two elitism mechanisms, the optimal mixing (OM) [20] and the  $(\mu + \lambda)$ -selection [21]. A key component of OM involves the use of additional function evaluations to assess whether the temporal individuals improve. The  $(\mu + \lambda)$ -selection chooses  $\mu$  individuals from the combined population consisting of  $\mu$  parent individuals and  $\lambda$  offspring individuals. In this study, the baseline is the tournament selection (TS) [22], a commonly-used selection scheme in GP [23], and OM is incorporated

TABLE IV  
MEANS OF THE 28 RANKINGS IN TERMS OF MSE MEDIAN OVER 30 INDEPENDENT RUNS ON ALL BENCHMARKS ON THE TESTING DATASET. A HIGHER RANKING MATCHES A LOWER MSE MEDIAN.

	TS	OM with TS	$(\mu + \lambda)$ -selection
Means of the rankings	2.04	2.64	<b>1.25</b>

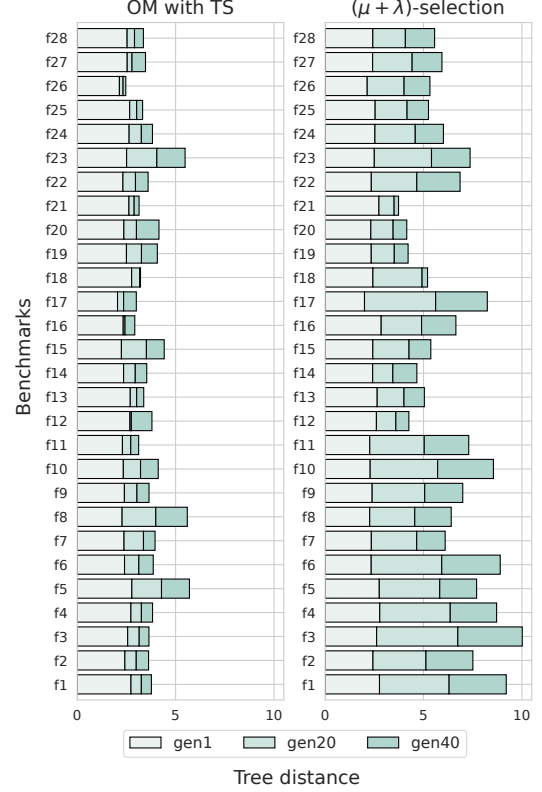


Fig. 5. Means of the tree distances between the best program and the others in the population at generations 1, 20, and 40. The disparity in tree distance between generations 1 and 40 is more pronounced through OM with TS than through the  $(\mu + \lambda)$ -selection across most benchmarks.

with TS. The population size is set to 400 and the maximum generation is set to 40.

The results, as shown in Table IV, indicate that utilizing the  $(\mu + \lambda)$ -selection leads to improvement over the other two schemes. An observation is that the mean ranking of OM with TS is even lower than that of TS alone. To understand this, we evaluate the impact on population diversity in the training stage by comparing the average tree distance with respect to OM with TS and the  $(\mu + \lambda)$ -selection. Specifically, the tree distance is defined in [24], and the average tree distance is calculated from the distances between the best program and the others in the population. The trends illustrated in Fig. 5 reveal that the aggregated tree distance of the  $(\mu + \lambda)$ -selection is greater than that of OM with TS on most benchmarks, indicating that population diversity decreases drastically when using OM with TS. Therefore, we utilize the relatively higher population diversity contributed by the  $(\mu + \lambda)$ -selection to

---

**Algorithm 1: SRE**

---

**item** :  $P$ : population,  $O$ : offspring population,  
 $E$ : expression,  $G$ : random programs,  
 $H_{\times}$ : hybridization function

**input** :  $S$ : SR method,  $D$ : dataset

**output**: best program

```
1  $E \leftarrow$  execute  $S$  given  $D$ 
2  $P \leftarrow \emptyset$ 
3 for  $i = 1$  to the population size do
4    $G_i \leftarrow$  generate a random program using GP
5    $P_i \leftarrow H_{\times}(E, G_i)$ 
6    $P \leftarrow P \cup P_i$ 
7 while yet to reach the maximum generation do
8    $O \leftarrow \emptyset$ 
9   for  $P_i \in P$  do
10     $O_i \leftarrow$  execute ranging crossover given  $P_i$  and
         $P$  with probability  $\alpha$ , mutation given  $P_i$  with
        probability  $\beta$ , or reproduction given  $P_i$  with
        probability  $1 - \alpha - \beta$ 
11    Evaluate the fitness of  $O_i$  with  $D$ 
12     $O \leftarrow O \cup O_i$ 
13    $P \leftarrow$  execute  $(\mu + \lambda)$ -selection given  $P$  and  $O$ 
14 return the best program in  $P$  in terms of fitness
```

---

mitigate the risk of converging to local optima and facilitate the improvement in fitness.

Furthermore, to identify segments for offspring using semantics of subtrees during recombination, we compare the ranging crossover [6] to the standard crossover utilizing  $H_{\times}$  and the  $(\mu + \lambda)$ -selection. The ranging crossover utilizes the output ranges of subtrees from a pool to identify which subtree is beneficial for the replaced tree node. The population size and the maximum generation are 400 and 40, respectively. The experiment results show that the median of the MSEs over 30 independent runs with the ranging crossover is less than that with the standard crossover in 57.14% of the benchmarks on the testing dataset. Additionally, The results of the two-sided Mann-Whitney U test [25] indicate that significant improvements occur in 21.42% of the benchmarks, with the significance level of 0.05. Therefore, we adopt the ranging crossover to improve the quality of offspring production.

According to the empirical results, we propose the determined SRE algorithm. The pseudo-code is presented in Algorithm 1. Note that in the evolutionary procedure (lines 7 to 13), we introduce two probability parameters  $\alpha$  and  $\beta$  to determine the recombination event for each program.

#### IV. EXPERIMENT RESULTS

This section first describes the benchmark problems and the setup of the experiments. Then, the experiment results and the associated discussions are presented.

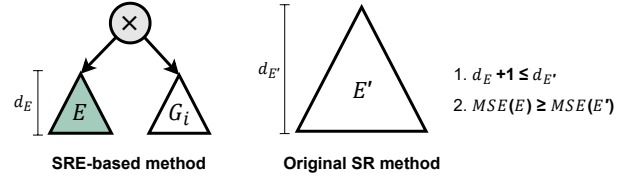


Fig. 6. Constraints on the relationship between expressions  $E$  from the SRE-based method and  $E'$  representing the respective SR method. Symbols  $d_E$  and  $d_{E'}$  denote the maximum tree depth of  $E$  and  $E'$ , respectively.

##### A. Test Problems and Experiment Setup

In the following experiments, we use the same 28 benchmarks [18], [19] as introduced in Section III. Additionally, we categorize these benchmark problems into two groups based on function characteristics: polynomial-expressible ( $f_1$  to  $f_8$ ), purely consisting of finite-order functions, and polynomial-inexpressible ( $f_9$  to  $f_{28}$ ). As for the parameter settings, we increase the population size and the maximum generation while keeping the other settings consistent with those outlined in Table II. Specifically, we set the population size to 500 and the maximum generation to 50.

In this paper, we apply SRE to both traditional and GP-based SR methods to assess whether the proposed algorithm enhances the performance of the respective SR methods in terms of MSE. The traditional methods contain TP and FFX, while the GP-based methods include Operon [7], GP-GOMEA [4], EPLEX [5], and gplearn [13]. TP is selected for its independence from function evaluation when both the polynomial degree and the point are provided, while FFX is chosen for its low cost in function evaluation [12]. The selected GP-based SR methods are commonly-used in SR research, and Operon represents a state-of-the-art GP approach for SR [2], [14], [15]. To denote the developed hybrid approaches, we prefix each selected SR method with “SRE-”.

For a fair comparison, two constraints are imposed on the hybrid program generated by each SRE-based method during population generation and the expression  $E'$  representing the respective SR method, as visualized in Fig. 6. The first constraint requires that the maximum tree depth of  $E$  ( $d_E$ ) is less than that of  $E'$  ( $d_{E'}$ ). The second constraint requires that the MSE of  $E$  is not less than that of  $E'$ . To implement these constraints,

- *For TP*: Several TPs with polynomial degrees ranging from 1 to 6 are computed by approximating at the data point closest to the midpoint of the input boundaries. We select the best expression ( $E'$ ) from the 6 computed polynomials in terms of MSE for TP. Then, among the polynomials with a degree lower than that of  $E'$ , we select the best expression ( $E$ ) that adheres to the constraints for SRE-TP.
- *For FFX*: The algorithm returns several expressions with a trade-off between the number of basis functions and errors [12]. We select the best expression ( $E'$ ) in terms of MSE for FFX; then, among the remaining expressions

TABLE V

THE MSE MEDIANS OVER 30 INDEPENDENT RUNS AND THE TWO-SIDED MANN-WHITNEY U TEST RESULTS ON THE TESTING DATASET. THE SYMBOLS +,  $\approx$ , AND  $-$  INDICATE WHETHER EACH SRE-BASED APPROACH IS SIGNIFICANTLY BETTER, INDIFFERENT, OR SIGNIFICANTLY WORSE THAN THE RESPECTIVE SR METHOD IN TERMS OF MSE. THE LAST ROW SUMMARIZES THE STATISTICAL TEST RESULTS ACROSS 28 BENCHMARKS.

$S$	TP		FFX		Operon		GP-GOMEA		EPLEX		gplearn	
	SRE- $S$	$S$	SRE- $S$	$S$	SRE- $S$	$S$	SRE- $S$	$S$	SRE- $S$	$S$	SRE- $S$	$S$
$f_1$	1.20e-04	<b>0.00e+00*</b>	<b>6.07e-03</b>	6.37e-03	4.58e-03	<b>4.21e-03</b>	1.21e-03	<b>9.64e-04</b>	3.86e-03	<b>4.94e-04*</b>	<b>7.89e-03</b>	1.31e-02
$f_2$	4.39e-06	<b>0.00e+00*</b>	<b>1.70e-03*</b>	4.24e-03	<b>2.03e-03*</b>	3.16e-03	3.30e-04	<b>7.82e-04*</b>	0.00e+00	0.00e+00	<b>2.34e-03</b>	3.56e-03
$f_3$	3.35e-08	<b>0.00e+00*</b>	<b>1.81e-03</b>	3.13e-03	1.94e-03	<b>1.78e-03</b>	<b>7.27e-05</b>	8.59e-05	0.00e+00	0.00e+00	<b>2.39e-03</b>	6.50e-03
$f_4$	0.00e+00	0.00e+00	<b>1.67e-03</b>	1.93e-03	<b>1.36e-03*</b>	2.70e-03	<b>1.03e-32</b>	1.03e-32	0.00e+00	0.00e+00*	<b>2.56e-04*</b>	1.64e-03
$f_5$	9.84e-04	<b>0.00e+00*</b>	<b>1.07e-03*</b>	1.89e-02	1.57e-05	<b>1.21e-05</b>	<b>1.04e-05</b>	1.45e-05	4.49e-04	<b>1.18e-04</b>	<b>3.59e-03</b>	3.73e-03
$f_6$	3.30e-06	<b>0.00e+00*</b>	<b>1.24e-02*</b>	1.88e-02	<b>2.88e-05</b>	3.74e-05	2.59e-06	<b>1.60e-06</b>	<b>1.86e-05</b>	3.16e-05	<b>1.99e-04*</b>	5.09e-03
$f_7$	0.00e+00	0.00e+00	<b>4.65e-03*</b>	2.54e-02	<b>3.37e-02</b>	4.42e-02	<b>6.77e-02*</b>	2.81e-01	3.84e-02	<b>3.68e-02</b>	<b>4.19e-02</b>	5.89e-02
$f_8$	3.02e-03	<b>0.00e+00*</b>	<b>1.76e-04*</b>	6.02e-03	<b>1.71e-03*</b>	4.10e-03	<b>3.37e-04*</b>	2.83e-03	2.93e-03	<b>1.74e-03</b>	<b>4.30e-03*</b>	8.84e-03
$f_9$	<b>1.86e+01*</b>	9.30e+01	<b>1.93e+01*</b>	2.07e+01	<b>1.53e+01*</b>	2.12e+01	<b>1.51e+01*</b>	4.68e+01	<b>1.93e+01</b>	2.27e+01	2.10e+01	<b>1.85e+01</b>
$f_{10}$	<b>2.71e-04*</b>	5.59e-02	<b>1.07e-02*</b>	2.88e-02	2.22e-04	<b>1.84e-04</b>	9.06e-04	<b>6.92e-04*</b>	2.17e-03	<b>1.98e-03*</b>	<b>1.03e-02*</b>	2.55e-02
$f_{11}$	<b>6.24e-05*</b>	1.32e-03	<b>5.77e-04</b>	6.91e-04	2.32e-04	<b>9.35e-05*</b>	0.00e+00	0.00e+00	5.46e-04	<b>3.47e-04*</b>	<b>8.28e-04</b>	1.26e-03
$f_{12}$	<b>5.53e-03*</b>	2.56e-02	<b>2.12e-02*</b>	7.87e-02	<b>3.12e-02</b>	6.32e-02	<b>3.70e-02</b>	1.70e-01	<b>2.05e-02</b>	2.64e-02	<b>2.73e-02*</b>	5.45e-02
$f_{13}$	<b>4.78e-05*</b>	2.16e-02	<b>5.18e-05*</b>	4.78e-03	3.82e-04	<b>3.01e-04</b>	8.04e-04	<b>4.32e-05*</b>	8.64e-03	<b>7.50e-03*</b>	<b>3.99e-03</b>	9.80e-02
$f_{14}$	<b>8.08e-04*</b>	2.74e-03	<b>2.87e-03*</b>	7.64e-02	7.53e-05	<b>5.68e-05</b>	0.00e+00	0.00e+00*	7.17e-04	<b>1.86e-04</b>	<b>5.00e-03*</b>	7.37e-02
$f_{15}$	<b>0.00e+00*</b>	2.74e-03	<b>2.13e-05*</b>	3.67e-04	2.57e-05	<b>1.93e-05</b>	0.00e+00	0.00e+00	<b>3.05e-04</b>	7.62e-04	<b>2.12e-04*</b>	2.16e-02
$f_{16}$	5.77e-02	<b>5.69e-02</b>	<b>8.52e-02*</b>	6.62e-01	1.72e-02	<b>1.22e-02</b>	<b>1.44e-02</b>	4.35e-02	5.31e-02	<b>2.72e-02</b>	<b>4.62e-02</b>	6.11e-02
$f_{17}$	1.66e-04	<b>3.65e-05*</b>	<b>2.48e-03*</b>	1.08e-02	2.88e-07	<b>1.45e-08</b>	<b>2.55e-02*</b>	7.63e-02	<b>5.79e-03</b>	1.21e-02	<b>3.85e-03*</b>	1.72e-02
$f_{18}$	<b>1.10e-03*</b>	4.36e-03	<b>9.24e-03*</b>	1.59e-02	<b>1.59e-02*</b>	2.42e-02	<b>1.27e-02</b>	3.79e-02	<b>7.09e-03</b>	1.69e-02	<b>2.33e-02*</b>	6.17e-02
$f_{19}$	<b>7.01e-02</b>	1.71e-01	<b>3.83e-02</b>	2.69e-02	2.60e-02	<b>2.19e-02</b>	<b>1.74e-01</b>	2.00e-01	<b>1.14e-01</b>	1.99e-01	1.32e-01	<b>1.24e-01*</b>
$f_{20}$	<b>4.67e-06*</b>	4.78e-04	1.69e-04	<b>1.27e-04*</b>	2.61e-05	<b>2.25e-05</b>	2.63e-04	<b>8.30e-05</b>	<b>2.11e-03</b>	2.11e-03	<b>2.07e-03*</b>	9.10e-03
$f_{21}$	<b>1.05e+03*</b>	1.15e+04	<b>2.57e+03*</b>	2.21e+04	<b>1.24e+02</b>	1.35e+02	<b>1.38e+03*</b>	7.71e+03	<b>8.62e+02</b>	1.35e+03	<b>5.33e+02</b>	7.61e+02
$f_{22}$	<b>6.27e-02*</b>	2.98e-01	<b>9.98e-01*</b>	2.42e+00	<b>2.99e-02*</b>	5.55e-02	<b>2.01e-01*</b>	1.29e+00	<b>8.75e-02*</b>	9.13e-02	<b>7.70e-02*</b>	2.75e-01
$f_{23}$	<b>2.51e-05*</b>	4.31e-05	<b>3.51e-04*</b>	4.39e-04	<b>5.87e-05*</b>	8.90e-05	<b>6.91e-05*</b>	5.42e-04	<b>1.40e-04*</b>	2.21e-04	<b>2.65e-04*</b>	7.28e-04
$f_{24}$	<b>1.46e-03*</b>	7.76e-02	<b>1.13e-03*</b>	7.85e-02	4.43e-03	<b>1.76e-03</b>	<b>7.21e-04*</b>	1.45e-03	<b>7.90e-04*</b>	1.29e-03	<b>7.84e-04*</b>	5.00e-03
$f_{25}$	<b>8.54e-03*</b>	1.78e-02	<b>2.82e-02*</b>	5.35e-02	3.81e-02	<b>3.47e-02</b>	<b>9.02e-04</b>	1.81e-03	<b>3.54e-02*</b>	8.96e-02	4.31e-02	<b>2.44e-02</b>
$f_{26}$	<b>4.05e-01*</b>	1.78e+00	<b>3.12e-01*</b>	6.68e-01	<b>5.26e-01</b>	5.90e-01	<b>4.97e-01*</b>	2.36e+00	<b>4.34e-01</b>	7.61e-01	4.54e-01	<b>3.49e-01</b>
$f_{27}$	5.25e-03	<b>3.71e-03</b>	<b>1.48e-02*</b>	1.54e-02	<b>9.34e-03</b>	1.09e-02	<b>4.58e-02*</b>	3.79e-01	<b>4.62e-02*</b>	3.62e-02	3.89e-02	<b>3.75e-02</b>
$f_{28}$	1.25e-03	<b>2.07e-04*</b>	<b>2.02e-03*</b>	2.09e-03	1.15e-03	<b>1.01e-03</b>	6.41e-04	<b>5.57e-04</b>	<b>7.90e-03</b>	1.31e-02	<b>3.76e-03*</b>	4.52e-02
All	15+ / 4 $\approx$ / 9-		21+ / 6 $\approx$ / 1-		8+ / 19 $\approx$ / 1-		14+ / 14 $\approx$ / 0-		8+ / 18 $\approx$ / 2-		15+ / 13 $\approx$ / 0-	

that adhere to the constraints, we select the best expression ( $E$ ) for SRE-FFX.

- *For GP-based methods:* To maintain the same NFE between the SRE-based method and the corresponding GP, we use half of the NFE to generate  $E$  and the other half to execute the evolutionary procedure of the SRE-based method. In contrast,  $E'$  is generated with the full NFE. Note that the NFE in this paper is defined as the product of the population size and the maximum generation.

The aforementioned settings aim for a fair comparison in experiments of this paper. However, several details, such as the maximum degree of the polynomial and the partition of the NFE in the case of the GP-based methods, may require further investigation or customization for specific problems.

To evaluate the effectiveness of SRE, we first compare SRE-based methods with their respective selected SR methods. Regarding tree depth, we impose a restriction that the maximum tree depth for each SRE-based method during the evolutionary process does not exceed that of the corresponding SR method. Furthermore, we establish an overall ranking comparison to assess the improvement of each SRE-based method. In this comparison, each SR method is assigned the same maximum depth for each benchmark.

## B. Results and Analyses

1) *Comparisons to the respective SR methods:* The medians of the MSEs and the results from the two-sided Mann-Whitney U test [25] at the significance level of 0.05 on the testing

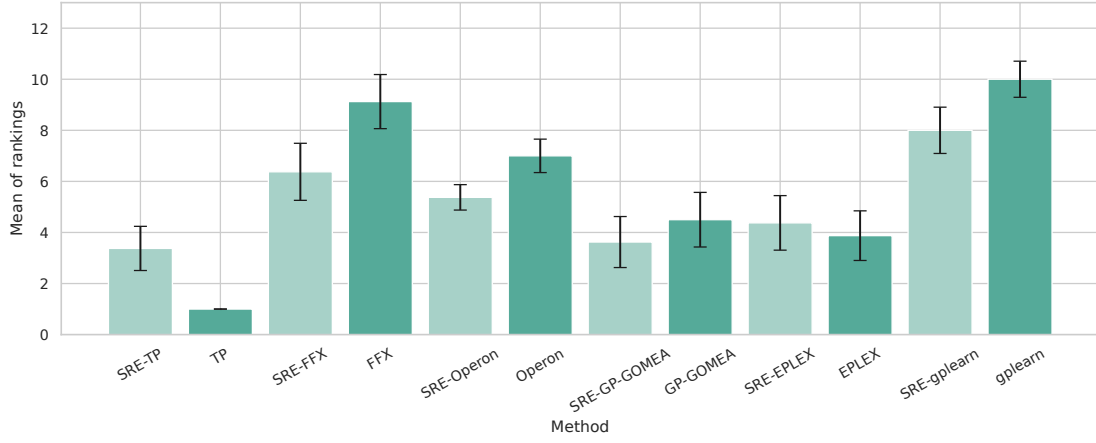
dataset are shown in Table V. Medians less than  $10^{-18}$  are represented as zeros. The smaller medians in each comparison are displayed in bold and additionally marked with a “\*” if the p-value of the corresponding statistical result is below the significance level of 0.05.

Overall, the number of benchmarks where each SRE-based method significantly outperforms the respective SR method exceeds the number where each SRE-based method significantly performs worse. 5 out of 6 SRE-based methods perform significantly worse in only 0 to 2 benchmarks, with the exception that SRE-TP performs significantly worse in 9 benchmarks. This difference is because that TP utilizes correct polynomial degrees for the ground truths in the polynomial-expressible category ( $f_1$  to  $f_8$ ). However, SRE-TP surpasses TP in 15 benchmarks in the polynomial-inexpressible category.

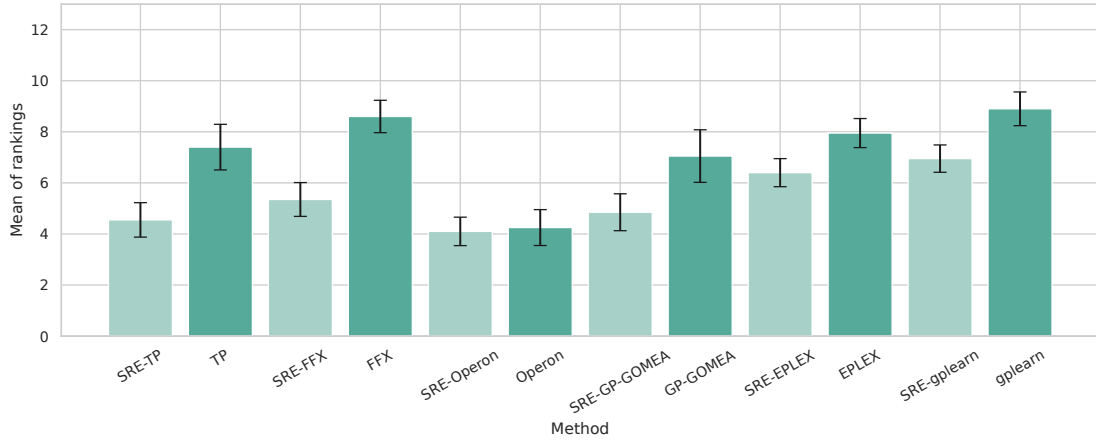
Furthermore, in the case of GP-based SR methods, the numbers where each SRE-based method performs indifferently to the respective SR method either closely match or exceed the numbers of outperformance. This observation is likely due to the exploration ability of GP, and thus SRE provides limited improvement on the selected GP-based SR methods.

2) *Overall ranking comparisons:* For each benchmark, the ranking performance of each method is determined by the median of MSEs on the testing dataset over 30 independent runs. The means of the rankings (MR) of each method are depicted in Fig. 7.

In the ranking results of the polynomial-expressible category (Fig. 7a), SRE-TP takes advantage of TP even with polynomial



(a) Polynomial expressible benchmarks.



(b) Polynomial inexpressible benchmarks.

Fig. 7. MR across 28 benchmarks on the testing dataset. Ranking values are determined by the medians of MSEs from 30 independent runs over 12 methods. The bars represent means, and the whiskers depict standard errors, both in terms of the rank values.

degrees lower than those of the ground truths and secures the second position in terms of MR. Additionally, GP-GOMEA, SRE-GP-GOMEA, EPLEX, and SRE-EPLEX occasionally find solutions that align with the ground-truths and therefore achieve the top positions in terms of MR. This is possibly due to the relatively lower problem difficulty of the benchmarks in the polynomial-expressible category [18], which might favor GP-GOMEA’s focus on handling linkage [4] and the EPLEX’s individual identification [5].

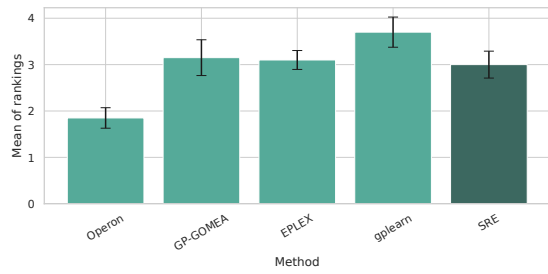
In the ranking results of the polynomial-inexpressible category (Fig. 7b), all SRE-based methods outperform their respective SR methods in terms of MR. Among the top four methods based on MR, SRE-Operon ranks the highest, followed by Operon, SRE-GP-GOMEA, and SRE-TP. Despite overlapping standard errors among the top four methods, three of the four methods are still SRE-based methods.

### C. Discussions

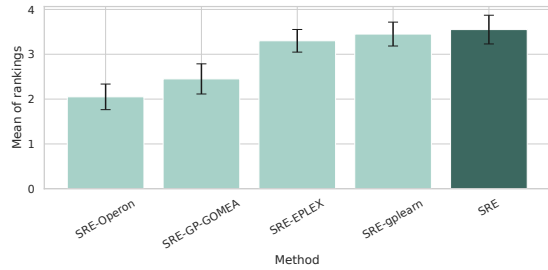
SRE utilizes the expression obtained from other SR methods and leverages the hybridization function and the evolutionary

mechanisms we investigate to achieve the improvement on performance. The effectiveness of SRE is shown by the comparisons in the previous subsection. However, one argument is regarding the necessity of applying SRE to GP-based methods, given the shared essence of GP between SRE and the selected GP-based SR methods. To assess this necessity, we use randomly generated programs as expressions  $E$  by modifying line 1 in Algorithm 1, and we compare the rankings of SRE with the selected GP-based SR methods and the SRE-based methods, respectively. The results of MR on the testing dataset in the polynomial-inexpressible category are illustrated in Fig. 8. In Fig. 8a, SRE ranks third in terms of MR, while in Fig. 8b, SRE falls behind all the SRE-based methods. This suggests that under the constraints of limited NFE and program size, SRE utilizes expressions from other GP-based SR methods and further enhances performance. We hypothesize that SRE conducts a fine-tuning process on the expressions from other GP-based SR methods, although this hypothesis requires further investigation in future works.





(a) SRE compared to the selected SR methods.



(b) SRE compared to the SRE-based methods.

Fig. 8. MR across 20 benchmarks in the polynomial-inexpressible category on the testing dataset. SRE is compared with the selected SR methods and the SRE-based methods, respectively.

## V. CONCLUSION

In this study, we proposed SRE, a framework that combines GP with other SR methods. The syntax format of the expression from the other SR method can be considered as the learned knowledge of the given data, which aimed to improve the efficiency and the quality of the evolutionary procedure. Our pre-experiments revealed that employing the multiplication as the hybridization function, the  $(\mu + \lambda)$ -selection, and the ranging crossover strengthened the capabilities of SRE. The experiment results with the six selected SR methods indicated that each SRE-based method significantly outperformed its respective SR method in the median MSEs across at least 8 and up to 21 out of 28 benchmarks. Furthermore, SRE-Operon ranked first in terms of MR on the polynomial-inexpressible benchmarks when compared to the other SRE-based methods and the selected SR methods.

As for future work, we would like to test other hybridization functions and analyze the scenarios in which specific hybridization approaches are most effective. Additionally, we would like to explore various combinations of evolutionary mechanisms to further enhance the SRE framework.

## ACKNOWLEDGEMENTS

The authors would like to express thanks for the support provided by the National Science and Technology Council in Taiwan under grant No. NSTC 112-2221-E-002-199 and National Taiwan University under grant No. 112L891103. Additionally, the authors appreciate the comments and the advice received from the reviewers.

## REFERENCES

- [1] P. Orzechowski, W. La Cava, and J. H. Moore, "Where are we now? a large benchmark study of recent symbolic regression methods," in *Proc. Genetic Evol. Comput. Conf.*, 2018, pp. 1183–1190.
- [2] W. La Cava, P. Orzechowski, B. Burlacu, F. de França, M. Virgolin, Y. Jin, M. Kommenda, and J. H. Moore, "Contemporary symbolic regression methods and their relative performance," in *Proc. Neural Inf. Process. Syst. Track Datasets Benchmarks*, vol. 1. Curran, 2021.
- [3] J. R. Koza, "On the programming of computers by means of natural selection," *Genetic Program.*, 1992.
- [4] M. Virgolin, T. Alderliesten, C. Witteveen, and P. A. N. Bosman, "Improving model-based genetic programming for symbolic regression of small expressions," *Evol. Comput.*, vol. 29, no. 2, pp. 211–237, 2021.
- [5] W. La Cava, L. Spector, and K. Danai, "Epsilon-lexicase selection for regression," in *Proc. Genetic Evol. Comput. Conf.*, 2016, pp. 741–748.
- [6] W.-Z. Fang, C.-H. Chang, J.-C. Liu, and T.-L. Yu, "GP with ranging-binding technique for symbolic regression," in *Proc. Genetic Evol. Comput. Conf. Companion*, 2023, pp. 563–566.
- [7] B. Burlacu, G. Kronberger, and M. Kommenda, "Operon C++: an efficient genetic programming framework for symbolic regression," in *Proc. Genetic Evol. Comput. Conf. Companion*, 2020, pp. 1562–1570.
- [8] I. Icke and J. C. Bongard, "Improving genetic programming based symbolic regression using deterministic machine learning," in *IEEE Congr. Evol. Comput.*, 2013, pp. 1763–1770.
- [9] B. He, Q. Lu, Q. Yang, J. Luo, and Z. Wang, "Taylor genetic programming for symbolic regression," in *Proc. Genetic Evol. Comput. Conf.*, 2022, pp. 946–954.
- [10] D. Izzo, F. Biscani, and A. Mereta, "Differentiable genetic programming," in *Eur. Conf. Genetic Program.*, 2017, pp. 35–51.
- [11] M. Semenikina, B. Burlacu, and M. Affenzeller, "Genetic programming based evolution of models of models," in *Comput. Aided Syst. Theory—EUROCAST 2019*, 2020, pp. 387–395.
- [12] T. McConaghy, "FFX: Fast, scalable, deterministic symbolic regression technology," *Genetic Program. Theory Practice IX*, pp. 235–260, 2011.
- [13] T. Stephens, "Genetic programming in Python, with a scikit-learn inspired API: gplearn," 2016.
- [14] F. O. de França, "Transformation-interaction-rational representation for symbolic regression: a detailed analysis of SRBench results," *ACM Trans. Evol. Learn. Optim.*, vol. 3, no. 2, pp. 1–19, 2023.
- [15] P.-A. Kamienny, S. d'Ascoli, G. Lample, and F. Charton, "End-to-end symbolic regression with transformers," *Advances Neural Inf. Process. Syst.*, vol. 35, pp. 10 269–10 281, 2022.
- [16] C. Haider, F. O. de França, G. Kronberger, and B. Burlacu, "Comparing optimistic and pessimistic constraint evaluation in shape-constrained symbolic regression," in *Proc. Genetic Evol. Comput. Conf.*, 2022, pp. 938–945.
- [17] K. Krawiec and U.-M. O'Reilly, "Behavioral programming: a broader and more detailed take on semantic GP," in *Proc. Genetic Evol. Comput. Conf.*, 2014, pp. 935–942.
- [18] J. McDermott, D. R. White, S. Luke, L. Manzoni, M. Castelli, L. Vaneschi, W. Jaskowski, K. Krawiec, R. Harper, and K. De Jong, "Genetic programming needs better benchmarks," in *Proc. Genetic Evol. Comput. Conf.*, 2012, pp. 791–798.
- [19] S.-M. Udrescu and M. Tegmark, "AI Feynman: A physics-inspired method for symbolic regression," *Science Advances*, vol. 6, no. 16, p. eaay2631, 2020.
- [20] D. Thierens and P. A. N. Bosman, "Optimal mixing evolutionary algorithms," in *Proc. Genetic Evol. Comput. Conf.*, 2011, pp. 617–624.
- [21] H.-P. Schwefel, *Numerical Optimization of Computer Models*. John Wiley & Sons, Inc., 1981.
- [22] T. Bäck, "Tournament selection," *Evol. Comput.*, vol. 1, pp. 181–186, 2000.
- [23] Y. Fang and J. Li, "A review of tournament selection in genetic programming," in *Int. Symp. Intell. Comput. Appl.*, 2010, pp. 181–192.
- [24] S.-H. Nienhuys-Cheng, "Distance between herbrand interpretations: A measure for approximations to a target concept," in *Inductive Logic Program.: 7th Int. Workshop*, 1997, pp. 213–226.
- [25] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *Ann. Math. Statist.*, vol. 18, pp. 50–60, 1947.