From Six to Two: SLIM-DUO, a Simplified Extension of SLIM-GSGP

Liah Rosenfeld*
lrosenfeld@novaims.unl.pt
NOVA Information Management School (NOVA IMS),
Universidade Nova de Lisboa, Campus de Campolide,
1070-312 Lisboa, Portugal

Leonardo Vanneschi
lvanneschi@novaims.unl.pt
NOVA Information Management School (NOVA IMS),
Universidade Nova de Lisboa, Campus de Campolide,
1070-312 Lisboa, Portugal

Abstract

Geometric Semantic Genetic Programming (GSGP) is an extension of Genetic Programming (GP) that captured the interest of researchers because of its ability to induce a unimodal error surface for any supervised learning problem. Although still a recent development, the Semantic Learning with Inflate and Deflate Mutations (SLIM-GSGP) extension of GSGP has already attracted significant attention due to its novel ability to generate offspring that are smaller than their parents, effectively addressing the problem of steady model growth in GSGP. This paper presents SLIM-DUO, an extension of SLIM-GSGP that integrates the six existing SLIM-GSGP variants into solely two unified formulations: DUO-MUL and DUO-SUM. This integration streamlines benchmarking and hyperparameter exploration while aiming to retain comparable predictive performance at approximately one third of the computational cost. Across five test problems, SLIM-DUO achieves predictive performance comparable to that of SLIM-GSGP, with model size differences that remain within previously observed dataset-dependent variability among SLIM variants. Overall, SLIM-DUO substantially reduces the computational effort required during both the configuration and benchmarking phases, highlighting a favorable balance between model size and computational complexity and preserving solution quality.

Keywords

Genetic Programming, Geometric Semantic Genetic Programming, SLIM-GSGP, Model Interpretability, Parameter Simplification

ACM Reference Format:
Liah Rosenfeld and Leonardo Vanneschi. 2026. From Six to Two: SLIM-DUO, a Simplified Extension of SLIM-GSGP. In *Genetic and Evolutionary Computation Conference (GECCO '26), July 13–17, 2026, San Jose, Costa Rica.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3795095.3805151

1 Introduction

Despite decades of progress, the efficient discovery of accurate, generalizable predictors in large hypothesis spaces remains a persistent challenge in supervised learning. In this setting, Geometric Semantic Genetic Programming (GSGP) [17] has emerged as an extension of Genetic Programming (GP) [20]. GSGP's distinctive appeal lies in the fact that it induces a unimodal error surface for any supervised learning task [17]. This feature has attracted substantial attention in the literature, where it is repeatedly linked to strong generalization, thereby offering a compelling explanation for GSGP's consistent ability to deliver highly accurate models and for its continued use across a wide range of application domains over the past decades [5, 6, 15, 16, 31]. Motivated by these strengths, the community has invested substantial effort in pushing GSGP further, not only by probing its capabilities but also by studying its limitations. As a result, the literature now offers a broad range of refinements that seek to improve both performance and practical usability, extending GSGP beyond its original formulation [3, 4, 8, 11, 25, 28]. One of the most recent steps in this line of work is the Semantic Learning algorithm based on Inflate and Deflate Mutations (SLIM-GSGP) [27], proposed in response to a limitation that has long shadowed GSGP: the tendency of its individuals to grow so large that they become impossible to read and interpret. This growth is especially difficult to mitigate because it is intrinsic to GSGP rather than incidental, arising directly from the original geometric semantic operators, which by construction always generate offspring that are larger than their parents [17]. SLIM-GSGP addresses this issue by introducing a novel genetic operator specifically designed to counteract this systematic expansion. Specifically, SLIM-GSGP proposes complementing standard mutation, termed *inflate mutation* because it systematically increases offspring size, with a novel *deflate mutation* designed to generate offspring smaller than their parents. Despite its recent introduction, SLIM-GSGP has already attracted substantial attention [7, 9, 10, 19, 22, 23, 29] and currently comprises six variants [22, 29], which differ in how semantic mutations are applied during inflation. Reported results show performance comparable to, and sometimes exceeding, standard GSGP, while producing models that are orders of magnitude smaller and more amenable to human interpretation. However, despite its demonstrated success, the presence of six distinct SLIM-GSGP variants creates a practical challenge: when tackling a new problem, practitioners and researchers must benchmark all six in order to properly assess SLIM-GSGP's behavior and capabilities and to identify the most suitable configuration. This need is reinforced by the *No Free Lunch Theorem* [32], which rules out the possibility that any single variant will consistently outperform the others across all tasks. This increases both computational cost (and thus execution time) and the overall complexity of hyperparameter tuning. SLIM-GSGP is particularly appealing because of its potential to produce human-interpretable models. Building on this strength, reducing the number of variant-level choices—and the associated hyperparameter burden—would make SLIM-GSGP even more attractive for both researchers and practitioners.

*Both authors contributed equally to this research.

GECCO '26, San Jose, Costa Rica

ACM ISBN 979-8-4007-2487-9/2026/07
https://doi.org/10.1145/3795095.3805151

In this work, we propose SLIM-DUO to mitigate the computational overhead introduced by the need to explore multiple variants. SLIM-DUO collapses the existing six variants into only two, called DUO-SUM and DUO-MUL, substantially reducing benchmarking effort while retaining competitive performance. This reduction is achieved by integrating the behaviors of the original variants within a single evolutionary run.

This document is organized as follows: Section 2 introduces GSGP and SLIM-GSGP, along with its six existing variants. Section 3 then presents SLIM-DUO. Section 4 describes the five real-life symbolic regression problems used as test cases and details the employed experimental settings. Section 5 reports the obtained results, comparing the two proposed SLIM-DUO variants against the six variants of the standard SLIM-GSGP configuration. Finally, Section 6 concludes the manuscript and suggests directions for future work.

2 Theoretical Background

To introduce the proposed SLIM-DUO formulation, this section first reviews the two underlying algorithms on which it builds: GSGP and the more recent SLIM-GSGP extension.

Geometric Semantic Genetic Programming. GSGP [17] is a variant of GP [20] that replaces GP's traditional syntax-based genetic operators with Geometric Semantic Operators (GSOs). Although acting directly on the syntax of the individuals, unlike conventional operators GSOs have a known and controllable effect on program semantics—i.e., the output vector produced by an individual over a given set of inputs. Also, GSOs induce a unimodal error surface, as explained in detail in [30]. Originally, GSGP relied on Geometric Semantic Mutation (GSM) and Geometric Semantic Crossover (GSC). However, previous work showed that mutation is the most effective operator, with mutation-only GSGP often matching or even outperforming mixed mutation-crossover methods [31]. As a result, much of the recent literature (including SLIM-GSGP [27] and this work) focuses solely on GSM. The GSM operator mutates a parent function $T : \mathbb{R}^n \to \mathbb{R}$ to produce an offspring T_M defined as: $T_M = T + \Delta T$, where, in the most common definition, ΔT is equal to $ms \cdot (T_{R1} - T_{R2})$, with T_{R1} and T_{R2} randomly generated real-valued functions whose outputs lie in $[0, 1]$, and ms the mutation-step hyperparameter. In the semantic space, GSM can be interpreted as a ball mutation, as it generates offspring within a hypersphere of radius ms centered on the parent's semantics [30].

Semantic Learning with Inflate and Deflate Mutations. As its name suggests, SLIM-GSGP incorporates two distinct types of mutations. The first type, the Inflate Geometric Semantic Mutation (IGSM), corresponds to the GSM defined previously, and results in offspring that are bigger than their parents. The second type, referred to as the Deflate Geometric Semantic Mutation (DGSM), simply removes from the genotype of an individual a term that had previously been added, creating offspring that are smaller than their parents. The reader is referred to [27, 29] for a discussion of why DGSM is still a geometric semantic mutation, creating a random perturbation of the parent's semantics in $[-ms, ms]$ and why DGSM is not a backtracking operator, but rather an operator that enables exploration of the search space by creating new individuals. SLIM-GSGP integrates three distinct methods for defining a function that produces values within the range $[-ms, ms]$ and two different approaches to inducing a ball mutation in the semantic space. By integrating these techniques, we define the six existing variants of SLIM-GSGP.

Different ways of defining ΔT with values in $[-ms, ms]$. Considering the definition of GSM given previously, and assuming that the outputs of the random expressions T_{R1} and T_{R2} are constrained into the interval $[0, 1]$ using a sigmoid function, the previous definition of ΔT can be rewritten as: $\Delta T = \mathrm{SIG2}(T) = ms \cdot (S(T_{R1}) - S(T_{R2}))$, where S is the sigmoid function, and ΔT was renamed SIG2(T) to emphasize the fact that, to define a function with values in $[-ms, ms]$, we have used a *sigmoid* (SIG) to wrap each of the *two* (2) random expressions. Nonetheless, several other ways of defining a ΔT with values in $[-ms, ms]$ can be imagined [27, 29]. For instance, employing a single random expression T_R rather than two can efficiently produce outputs within $[-ms, ms]$, allowing us to save a significant amount of computational resources. This has lead to the two following possible definitions for ΔT: (i) $\Delta T = \mathrm{ABS}(T) = ms \cdot (1 - \frac{2}{1+|T_R|})$ and (ii) $\Delta T = \mathrm{SIG1}(T) = ms \cdot (2 \cdot S(T_R) - 1)$. Function ABS($T$) takes its name from its use of the *absolute value* to normalize the outputs generated by the random expression, while SIG1(T) has this name because it uses the *sigmoid* (SIG) to constrain the outputs of a *single* (1) random expression.

Different ways of obtaining ball mutation on the semantic space. A small perturbation of a numerical value can easily be achieved by either adding a small positive or negative value, or by multiplying the number by a factor slightly above or below 1. Consequently, the inflate mutation may either add any of the ΔT variants defined above, and so its definition is $\mathrm{Inflate}(T) = T + \Delta T$, or multiply by $(1 + \Delta T)$, and so its definition is $\mathrm{Inflate}(T) = T \times (1 + \Delta T)$. Depending on the approach, if the inflate operation involves addition of ΔT , then the deflate operation will involve subtraction of a previously added term. Similarly, if the inflate operation involves multiplication of $(1 + \Delta T)$, then the deflate operation will correspondingly involve division by a term that had previously been multiplied. By combining the three methods to define ΔT with the two techniques for achieving ball mutation, we now define the six distinct variants of SLIM-GSGP, named SLIM+SIG2, SLIM+ABS, SLIM+SIG1, SLIM*SIG2, SLIM*ABS and SLIM*SIG1.

3 From Six to Two: The SLIM-DUO Extension

As discussed in [27], it is particularly convenient to implement each variant of SLIM-GSGP using a linked-list construct to represent the genotypes. An IGSM appends a new "block" (or item) to the list, whereas a DGSM removes a previously added block [22, 27]. Depending on the SLIM-GSGP variant, each block represents a different ΔT. SLIM-DUO builds on the understanding that, once the way of inducing ball mutation is established (i.e., by adding a value close to zero, or multiplying by a value either slightly smaller or slightly larger than 1), there is no principled reason to force all blocks in an individual to come from the same mutation function (SIG2, SIG1, or ABS). An individual can instead contain a heterogeneous mix of blocks, each produced by a different IGSM mutation choice. This allows the six SLIM-GSGP variants—formed by combining three mutation functions with two perturbation methods—to be reduced to two variants defined only by the perturbation: *DUO-SUM* (addition-based) and *DUO-MUL* (multiplication-based). In each variant, the IGSM mutation function can be sampled dynamically among SIG2, SIG1, and ABS, while DGSM removes blocks consistent with the same perturbation mechanism. SLIM-DUO follows the same dynamics as SLIM-GSGP, with only one modification. In standard SLIM-GSGP, whenever mutation is applied, the algorithm selects either IGSM or DGSM (each with its own probability hyperparameter). DGSM removes one block from the individual, whereas IGSM appends a new block whose type is fixed *a priori* by the chosen SLIM variant, meaning that the mutation function is always the same (SIG2, SIG1, or ABS). SLIM-DUO preserves this structure, but modifies the inflate step: when IGSM is selected, the type of block to be added is sampled at that moment by choosing among the three mutation functions (SIG2, SIG1, and ABS), each one with its own probability, that becomes a new hyperparameter. The corresponding block is then appended to the individual, allowing for a SLIM-DUO individual to have a mix of SIG2, SIG1 and ABS functions in its genotype. Exactly as in SLIM-GSGP, when DGSM is chosen, a previously added block is removed, independently from it being a SIG2, SIG1 or ABS block. Finally, as in SLIM-GSGP, individual semantics are computed by either summing up (*sum*) or multiplying (*mul*) the semantics of the different blocks. To highlight the difference between SLIM-GSGP and SLIM-DUO, consider the following example of a possible SLIM-DUO genotype: $T_M = T + ms\cdot(T_{R1} - T_{R2}) + ms\cdot(T_{R3} - T_{R4}+) + ms\cdot(1 - 2/(1 + |T_{R5}|))$. This individual has three blocks, added to the initialized individual: two SIG2 blocks, followed by one ABS block.

Note that the declared objective of SLIM-DUO is not to change the algorithm's behavior or improve its performance. Instead, it serves a pragmatic goal: to allow us to have at our disposal all the functionalities offered by the six-variants, inside a framework that has only two variants, thus significantly facilitating the task of finding the most appropriate configuration to tackle a problem.

4 Test Problems and Experimental Settings

Table 1 summarises the five real-world symbolic regression benchmarks used in this study, all of which are well established in the GP community and match the datasets adopted in [27]. Across these five test problems, the two SLIM-DUO variants were evaluated against all existing SLIM-GSGP variants. For each method, 30 independent runs were performed and, for each one of these runs, different training and test set were generated using a 70%/30% random split (Monte Carlo cross-validation [30]). In line with [27], individuals were constructed using the four basic arithmetic operators +, −, ∗, and division /, protected as in [13]. Evolution was run for 2000 generations with a population size of 100 individuals. Although this population size may be considered too small in the context of standard GP, this is the typical magnitude in the context of GSGP, where smaller population sizes left to evolve for many generations are usual [31], and also some forms of geometric semantic Hill Climbing (i.e., with only one individual being optimized) have been used [12]. Fitness was measured using Root Mean Squared Error (RMSE). The terminal set comprised the input variables, with dimensionality matching the number of features in each dataset. Population initialization was done using the ramped half-and-half algorithm with a maximum tree depth of 6, while parent selection used tournament selection with tournament size 2. Elitism was also applied by copying the best individual unchanged into the next generation. Exactly as in [27, 29], mutation settings were dataset-dependent, with Concrete as the sole exception. For all other datasets, inflate/deflate rates were 0.3/0.7 and the step size was sampled uniformly from $[0, 1]$ at each inflate event. For Concrete, both rates were set to 0.5 and the step size was sampled uniformly from $[0, 0.3]$. In the SLIM-DUO versions, the inflate mutation selects SIG2, SIG1 and ABS blocks with uniform probability (1/3 rate each).

Table 1: Used datasets overview presenting number of instances, number of features, and original references.

Dataset	# Instances	# Features	Reference
Concrete	1030	8	[6]
Energy	330	8	[26]
Istanbul	536	7	[1]
PPB	234	626	[2]
RBSP	372	107	[21]

5 Experimental Results

Figure 1 shows the median training and test RMSE of the elite individual across generations, calculated over 30 runs for each dataset. Statistical significance of final-generation test RMSE differences was assessed using a Wilcoxon rank-sum (Mann–Whitney U) test ($\alpha = 0.05$), with p-values reported in Table 2. The table compares DUO-SUM and DUO-MUL against their operator-matched SLIM variants (+ and ∗, respectively) and against each other; significant results are bold, with ✓ indicating the left method is better and ⋆ indicating the opposite.

An analysis of the results reported in Figure 1 and Table 2 reveals that, across most datasets, the performance of the different SLIM variants—encompassing both the original six SLIM-GSGP variants and the SLIM-DUO variants—is largely comparable. The DUO-MUL variant consistently matches or surpasses the performance of its DUO-SUM counterpart, a trend observed in four out of the five case studies. However, within this subset, DUO-MUL achieves statistically significant improvements over DUO-SUM in only one dataset (RSBP). The Istanbul dataset constitutes a clear exception to this behavior, exhibiting an inverted trend in which DUO-SUM significantly outperforms DUO-MUL. Beyond the comparison between the two SLIM-DUO variants, further insights arise when contrasting them with their SLIM-GSGP counterparts. Across all case studies, both DUO-SUM and DUO-MUL exhibit performance broadly comparable to that of the corresponding operator-based SLIM variants, with statistically significant improvements and degradations observed in equal number, and no significant differences in the remaining cases. In particular, DUO-SUM significantly outperforms SLIM+SIG2 on the Concrete dataset, is significantly outperformed by the same variant on the Energy dataset, and achieves statistically comparable performance across all remaining variants and case studies. Similarly, although DUO-MUL performs comparably

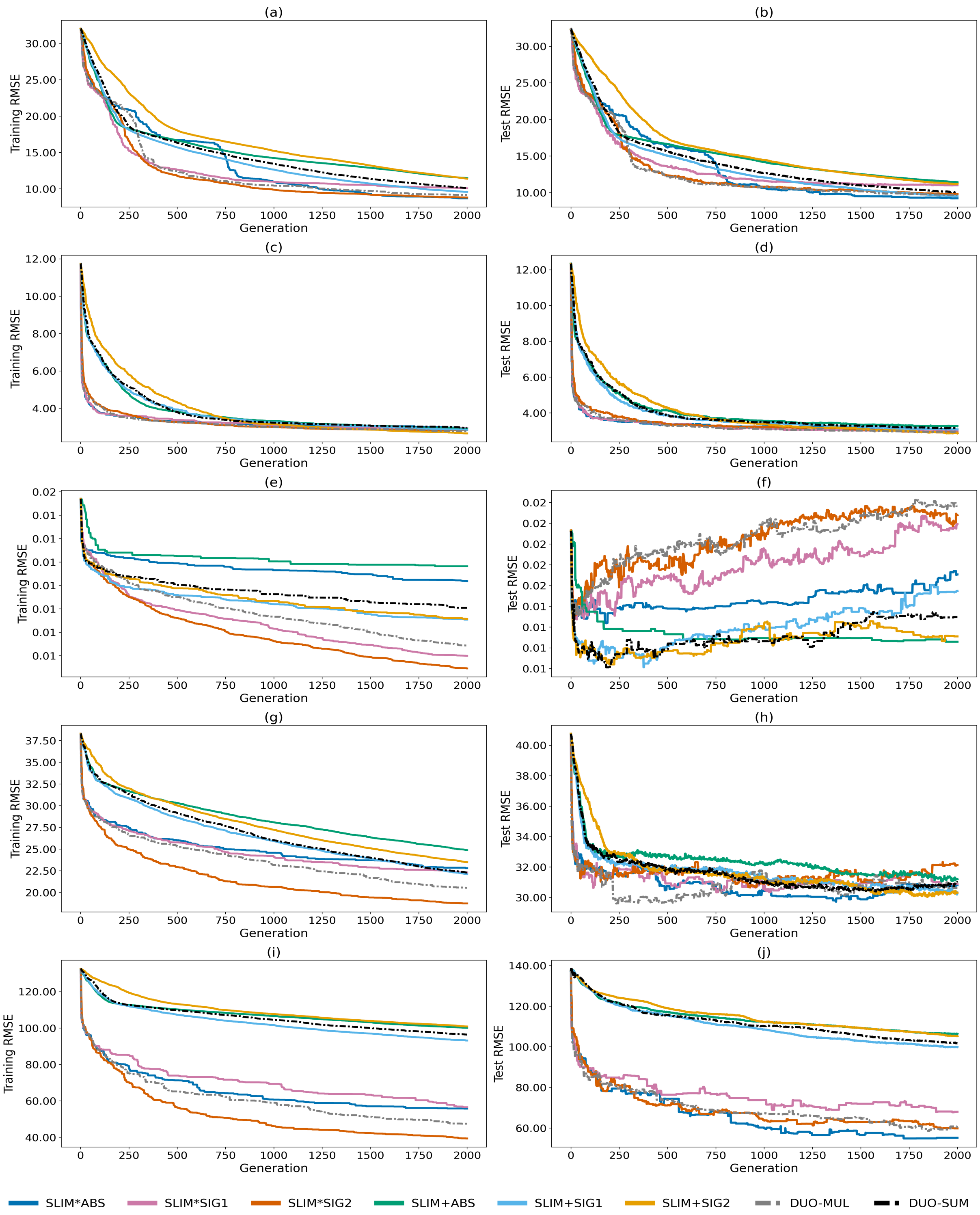

Figure 1: Median RMSE per generation. Training results are shown in the left column and test results in the right column. The plots are: (a) and (b): Concrete, (c) and (d): Energy, (e) and (f): Istanbul, (g) and (h): PPB, (i) and (j): RBSP.

Table 2: Mann–Whitney U p-values for final-generation test RMSE. Statistically significant values ($\alpha = 0.05$) are bold, with ✓ indicating the left method outperforms the right and ★ indicating the reverse.

	Concrete	Energy	Istanbul	PPB	RSBP
DUO-SUM vs SLIM+SIG2	$\mathbf{2.92 \times 10^{-2}}$ ✓	$\mathbf{2.07 \times 10^{-2}}$ ★	9.35×10^{-1}	7.96×10^{-1}	7.28×10^{-1}
DUO-SUM vs SLIM+ABS	5.75×10^{-2}	2.84×10^{-1}	1.62×10^{-1}	4.83×10^{-1}	4.64×10^{-1}
DUO-SUM vs SLIM+SIG1	2.71×10^{-1}	8.77×10^{-1}	2.52×10^{-1}	5.49×10^{-1}	6.41×10^{-1}
DUO-SUM vs DUO-MUL	7.96×10^{-1}	5.94×10^{-2}	$\mathbf{1.34 \times 10^{-5}}$ ✓	9.71×10^{-1}	$\mathbf{2.03 \times 10^{-9}}$ ★
DUO-MUL vs SLIM*SIG2	9.82×10^{-1}	6.20×10^{-1}	3.63×10^{-1}	3.71×10^{-1}	7.84×10^{-1}
DUO-MUL vs SLIM*ABS	2.46×10^{-1}	4.46×10^{-1}	$\mathbf{4.98 \times 10^{-4}}$ ★	9.59×10^{-1}	9.71×10^{-1}
DUO-MUL vs SLIM*SIG1	1.22×10^{-1}	7.84×10^{-1}	4.73×10^{-1}	5.59×10^{-1}	$\mathbf{4.36 \times 10^{-2}}$ ✓

to its SLIM counterparts in most scenarios, it achieves statistically significant improvements over SLIM*SIG1 on the RBSP dataset, while being significantly outperformed by SLIM*ABS on the Istanbul dataset These results are noteworthy, as they indicate that the proposed SLIM-DUO formulation achieves overall performance comparable to the existing SLIM-GSGP variants, despite operating within a substantially reduced hyperparameter space.

Figure 2 shows the median elite size (number of nodes) across generations over 30 runs for each dataset, with Wilcoxon rank-sum (Mann–Whitney U) p-values reported in Table 3. From these results, we can see that SLIM-DUO can, in several settings, produce statistically larger elites than some of the baseline SLIM variants. However, observing Figure 2, we can clearly see that SLIM-DUO never produces neither the smallest nor the largest models among the compared variants, always positioning itself in approximately a median position among the results of its SLIM counterparts. Also, this behavior is not unique to SLIM-DUO: the six original SLIM variants already exhibit substantial and dataset-dependent differences in elite size, with no single variant consistently dominating across problems.

The main practical advantage of SLIM-DUO lies in reducing the number of configurations that must be evaluated to achieve strong performance on a new task. While SLIM-DUO may produce slightly larger individuals than the smallest SLIM models (but always smaller than the largest SLIM models), this does not hinder its effectiveness. In fact, there are instances where this even comes with improved predictive performance. For example, on the RSBP dataset, DUO-MUL achieves statistically better RMSE than SLIM*SIG1, despite producing statistically larger models. There are also cases where this pattern does not hold uniformly. On the Istanbul dataset, DUO-MUL is outperformed by the SLIM*ABS variant while still producing larger individuals. However, as will be discussed in Section 5.2, this dataset exhibits behavior that differs from the general trend. Overall, these results indicate that SLIM-DUO retains the predictive competitiveness of the SLIM framework, while collapsing a six-way variant selection problem into only two candidates.

5.1 Elite Block Composition Analysis

To further characterise SLIM-DUO solutions beyond predictive performance and overall size, we analyse the elite block composition, i.e., the distribution of block types (SIG1, SIG2, or ABS) in the best individual at each generation. As previously mentioned, in our experiments new blocks are generated via inflate mutation, where the block type is sampled uniformly at random, introducing no *a priori* bias toward any type. Thus, any departure from a uniform mix within elites reflects selection dynamics and the relative usefulness of each block type for the task. Figure 3 shows, for each dataset and both SLIM-DUO variants, how the median number of SIG1, SIG2, and ABS blocks in elite individuals evolves across generations. The Wilcoxon rank-sum (Mann–Whitney U) test was also applied to compare DUO-SUM and DUO-MUL by block counts across datasets. While the p-values are omitted due to space constraints, all pairwise differences are significant ($p < 0.01$). Consistent with earlier elite-size trends, DUO-MUL yields elites with significantly fewer blocks when compared to DUO-SUM on all datasets except Istanbul, where it produces significantly more. Figure 3 also highlights systematic differences in block composition: DUO-SUM is generally dominated by SIG1 block types (with Energy being an exception and remaining relatively balanced), whereas DUO-MUL tends to retain more SIG2 block types. Thus, despite uniform block sampling, results show that selection drives distinct, reproducible internal structures under additive versus multiplicative updates. As will be discussed in Section 6, this could motivate future work to exploit block-type bias during evolution, rather than only to analyze them *a posteriori*.

5.2 The Istanbul Test Case: Unravelling an Inverted Performance Pattern

An examination of the results presented in Section 5 reveals that the Istanbul dataset exhibits a distinctive behavior compared to the remaining case studies. Specifically, it is the only dataset in which the DUO-SUM variant achieves lower test-set RMSE than its DUO-MUL counterpart, thereby breaking the broader pattern observed across the remaining datasets, where DUO-MUL performs better than DUO-SUM, albeit not always with statistical significance. In addition, Istanbul is also the only dataset in which DUO-SUM produces smaller elite individuals, reversing the clearly consistent pattern observed elsewhere, where DUO-MUL yields smaller models. This inversion, affecting both predictive performance and model size, marks the Istanbul dataset as a noticeable outlier among the considered benchmarks. The emergence of this unexpected behavior naturally motivates further investigation. In particular, these findings prompt the question of whether dataset-specific characteristics could be leveraged to determine, *a priori*, which SLIM-DUO variant is most appropriate for a given problem. If such dataset-specific *a priori* indicators could be identified, the primary objective of SLIM-DUO would be further strengthened: not only reducing the original six variants to two, but potentially enabling the automatic

Table 3: Mann–Whitney U test p-values for final-generation elite size. Statistically significant values ($\alpha = 0.05$) are bold, with ✓ indicating the left method outperforms the right and ⋆ indicating the reverse.

	Concrete	Energy	Istanbul	PPB	RSBP
DUO-SUM vs SLIM+SIG2	$\mathbf{3.02 \times 10^{-11}}$ ✓	$\mathbf{4.97 \times 10^{-11}}$ ✓	$\mathbf{3.09 \times 10^{-4}}$ ✓	$\mathbf{6.12 \times 10^{-10}}$ ✓	1.71×10^{-1}
DUO-SUM vs SLIM+ABS	$\mathbf{1.33 \times 10^{-6}}$ ⋆	6.04×10^{-2}	$\mathbf{3.00 \times 10^{-11}}$ ⋆	$\mathbf{6.82 \times 10^{-3}}$ ⋆	9.53×10^{-1}
DUO-SUM vs SLIM+SIG1	$\mathbf{4.07 \times 10^{-11}}$ ⋆	$\mathbf{2.49 \times 10^{-8}}$ ⋆	$\mathbf{3.59 \times 10^{-5}}$ ✓	$\mathbf{3.65 \times 10^{-8}}$ ⋆	$\mathbf{2.92 \times 10^{-2}}$ ⋆
DUO-SUM vs DUO-MUL	$\mathbf{3.02 \times 10^{-11}}$ ⋆	$\mathbf{3.01 \times 10^{-11}}$ ⋆	$\mathbf{3.02 \times 10^{-11}}$ ✓	$\mathbf{3.02 \times 10^{-11}}$ ⋆	$\mathbf{2.98 \times 10^{-11}}$ ⋆
DUO-MUL vs SLIM*SIG2	$\mathbf{4.18 \times 10^{-9}}$ ✓	$\mathbf{1.41 \times 10^{-9}}$ ✓	$\mathbf{3.34 \times 10^{-11}}$ ✓	$\mathbf{1.95 \times 10^{-10}}$ ✓	$\mathbf{1.59 \times 10^{-10}}$ ✓
DUO-MUL vs SLIM*ABS	$\mathbf{3.32 \times 10^{-6}}$ ⋆	$\mathbf{3.33 \times 10^{-9}}$ ⋆	$\mathbf{3.01 \times 10^{-11}}$ ⋆	$\mathbf{2.57 \times 10^{-10}}$ ⋆	$\mathbf{7.88 \times 10^{-8}}$ ⋆
DUO-MUL vs SLIM*SIG1	$\mathbf{1.64 \times 10^{-5}}$ ⋆	$\mathbf{1.60 \times 10^{-6}}$ ⋆	$\mathbf{4.22 \times 10^{-4}}$ ✓	$\mathbf{6.22 \times 10^{-7}}$ ⋆	$\mathbf{1.85 \times 10^{-7}}$ ⋆

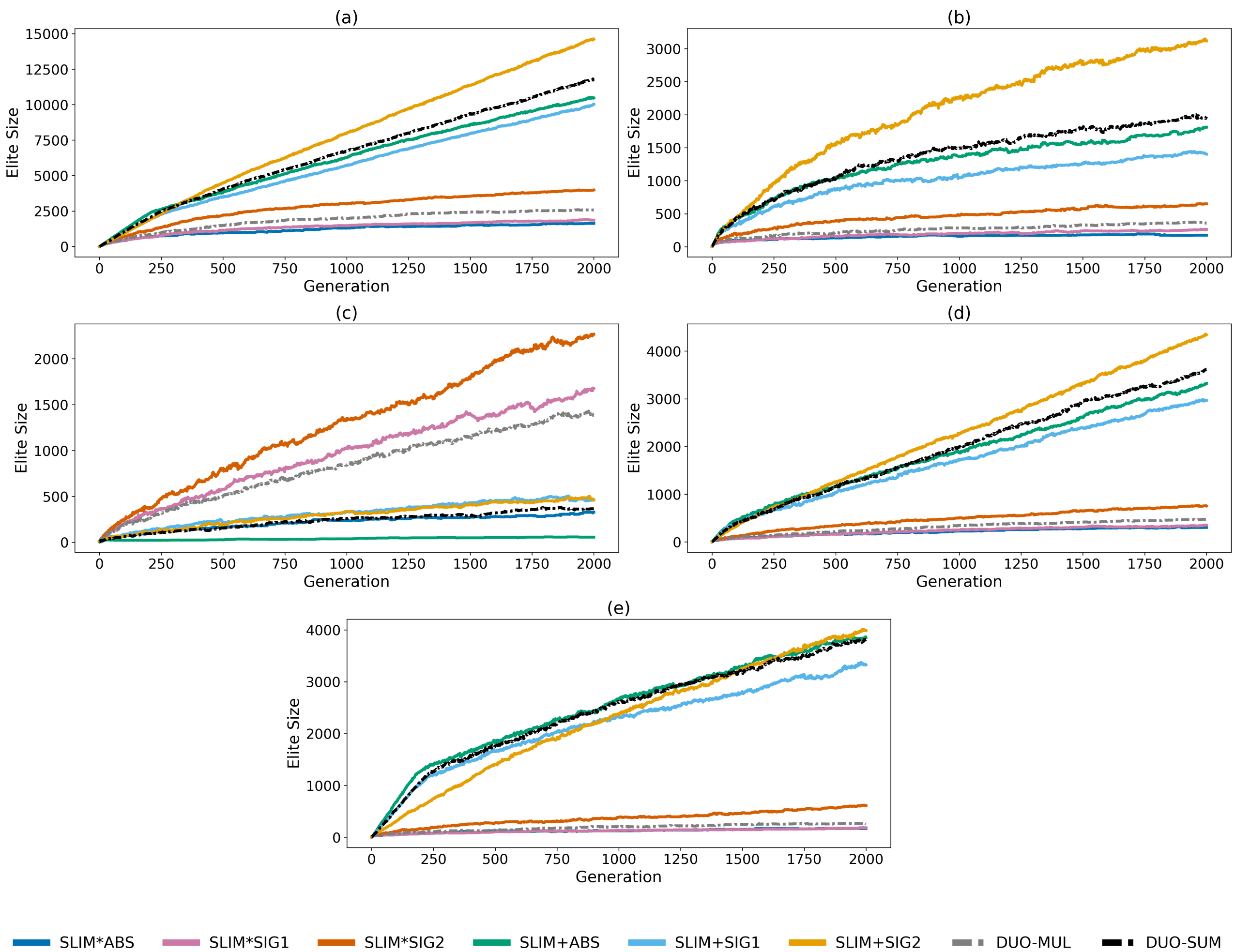

Figure 2: Median elite size per generation across datasets. The plots are (a): Concrete, (b): Energy, (c): Istanbul, (d): PPB, (e): RBSP.

choice of a single, well-suited variant based on the properties of the dataset.

In the present work, we limit our investigation to a preliminary speculative analysis, intended to shed light on the atypical behavior found by the results on the Istanbul dataset. To this end, we initially considered the statistical properties of the target variable across all datasets as an initial, general characterization and overview. The resulting summary statistics are reported in Table 4, where we can easily observe that Istanbul differs from the remaining case studies in a few aspects. Most notably, its target variable operates on a substantially smaller numerical scale, as reflected by its markedly lower *std*, range, and *IQR*. In fact, the target values in Istanbul lie

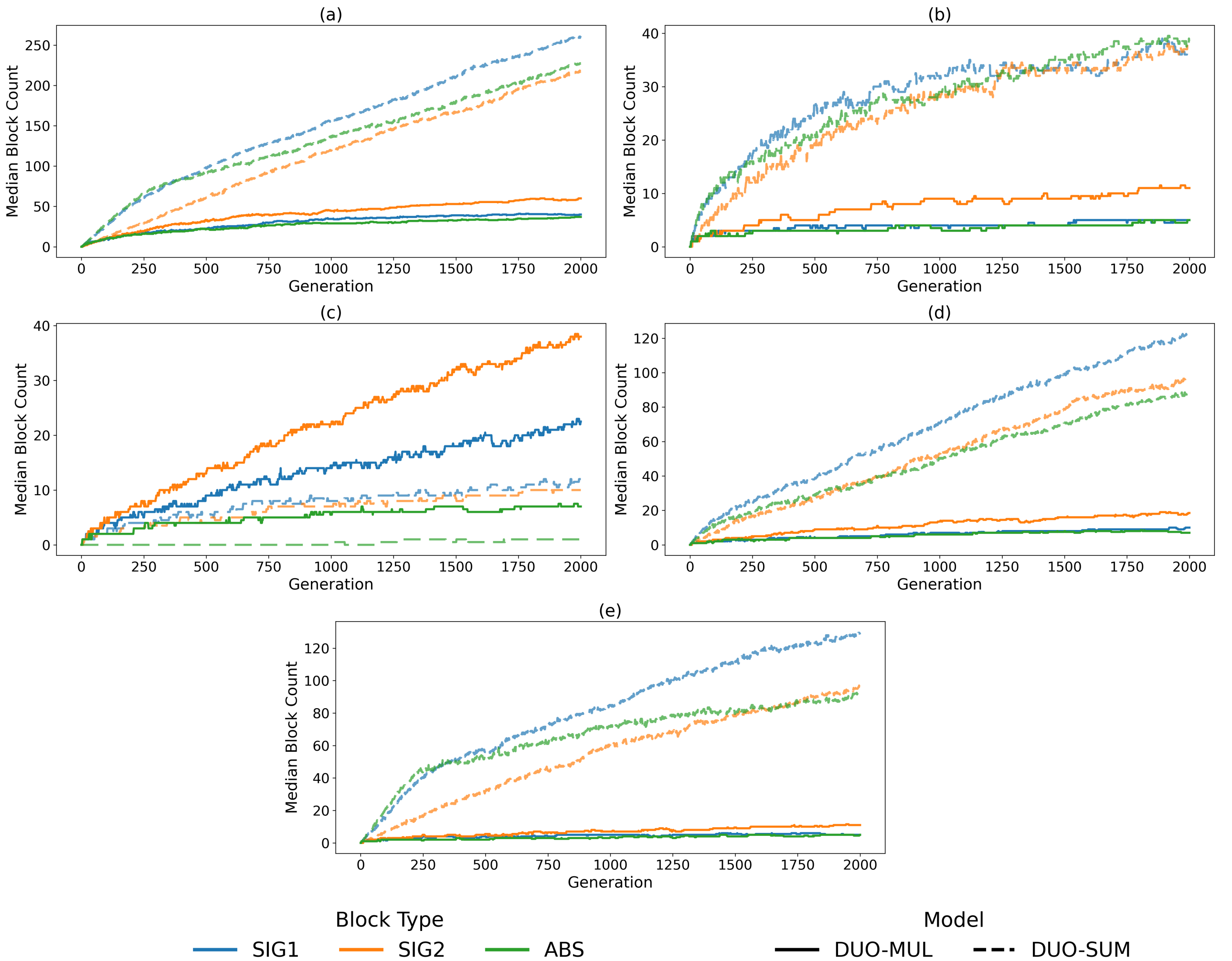

Figure 3: Evolution of the median of the block type count in elite individuals over generations. Solid lines indicate DUO-MUL and dashed lines indicate DUO-SUM. The plots are (a): Concrete, (b): Energy, (c): Istanbul, (d): PPB, (e): RBSP

Table 4: Summary statistics of the target variable for each dataset.

Dataset	std	IQR	range	skew	kurt	$\text{outlier}_{\text{MAD2.5}}$
RSBP	162.63	220.00	980.00	1.261	2.195	0.089
Concrete	16.71	22.43	80.27	0.417	−0.314	0.092
Istanbul	0.0211	0.0236	0.1853	−0.080	1.824	0.146
Energy	9.85	16.09	36.91	0.528	−0.949	0.127
PPB	57.43	44.00	626.50	6.654	65.238	0.258

very close to zero. While this observation may appear superficial at first, it becomes potentially informative once it is considered in light of the two SLIM-DUO perturbation strategies. Recall that an inflate mutation in DUO-SUM applies an additive perturbation, i.e., it adds values close to 0 to the current semantics, whereas DUO-MUL applies a multiplicative perturbation, i.e., it scales the semantics by a factor close to 1. Although both mechanisms represent "small" semantic changes in general terms, they are not equivalent in their effective impact when the semantics (and, consequently, the target values) are confined to a very small range. In particular, the additive update introduces an *absolute* change, whose magnitude does not depend on the current semantic value, meaning that a perturbation deemed *close to zero* may still constitute a non-negligible relative change under the Istanbul scale. By contrast, the multiplicative update induces a *relative* change that is proportional to the current semantics, and therefore tends to remain small when the semantics themselves are small. As a result, DUO-SUM may effectively operate more aggressively than DUO-MUL on Istanbul, providing a plausible interpretation for the inverted trend compared to the results observed on the other datasets.

Istanbul also stands out not only for its scale, but for the shape of its distribution. Its target variable has skewness near zero, suggesting an approximately symmetric distribution, yet that symmetry does not come with light tails. On the contrary, the excess kurtosis (about 1.8) points to heavier tails. Consistently, the MAD2.5 rule in Table 4—which labels as outliers observations lying beyond the median ±2.5 times the Median Absolute Deviation (MAD) [14]—flags roughly 15% of the data. This combination of near-symmetry and heavy tails is comparatively unusual across the datasets (Energy is the closest, though it exhibits fewer outliers and higher skewness) and could help account for the inverted pattern observed in the Istanbul test case. Beyond the simple target summaries in Table 4, richer dataset meta-features may help explain the atypical behavior observed on Istanbul. For instance, PCA-based descriptors could reveal effective dimensionality, while KNN-based measures could proxy local smoothness and noise, potentially clarifying when additive (DUO-SUM) versus multiplicative (DUO-MUL) updates are favored. While this type of detailed investigation constitutes a relevant and promising future work direction, it falls beyond the scope of the present work.

6 Conclusions and Future Work

We introduced SLIM-DUO, a new extension of SLIM-GSGP [22, 29], obtained by combining two semantic perturbation schemes, additive (*sum*) and multiplicative (*mul*), with three bounded mutation functions (SIG2, SIG1, and ABS). These 3×2 combinations define the six ball-mutation strategies that make up the standard six-variant SLIM-GSGP design space. While this six-variant repertoire offers flexibility, it also forces practitioners to benchmark multiple configurations for every new task, since no single variant can *a priori* be expected to dominate under the No Free Lunch Theorem [32]. This amplifies runtime and variant-level hyperparameter complexity. SLIM-DUO relieves this bottleneck by collapsing the design space to just two models, DUO-SUM and DUO-MUL, while still capturing the behaviors of all six SLIM variants within a single framework: at each inflate mutation, the newly added block is sampled uniformly from SIG1, SIG2, and ABS. In doing so, SLIM-DUO preserves the expressivity of the original method, but eliminates the need for six-way variant selection, leaving only the perturbation mode—additive or multiplicative—as the remaining choice and thereby enabling faster, simpler, and more scalable experimentation. From the presented results, it can be observed that SLIM-DUO's formulation does not compromise predictive performance despite its substantially smaller hyperparameter space. The results indicate that the simplified SLIM-DUO approach consistently preserves the performance of the original framework, generally yielding statistically comparable results. Cases in which SLIM-DUO either outperforms or is outperformed by the best performing SLIM-GSGP counterparts are both rare and occur with similar frequency, reinforcing the overall parity in performance. These findings support the claim that SLIM-DUO offers a more efficient search space without sacrificing solution quality. However, it is important to note that this simplification is not universally accompanied by improved compactness. In fact, SLIM-DUO can in some cases produce statistically larger elite individuals than some baseline variants. Nevertheless, the observed size increases are generally moderate in magnitude, and—crucially—remain within the range of variability already exhibited by the original SLIM variants across the studied datasets. In other words, while SLIM-DUO may sometimes favor slightly larger models, compared to the smallest SLIM-GSGP solutions, this effect does not appear to undermine the main result that predictive performance is preserved under a substantially simpler and more efficient benchmarking setup.

Research on SLIM still remains at an early stage, with ample room for further insights and improvements. We argue that SLIM-DUO supports this agenda by lowering the barrier to experimentation and enabling more efficient and systematic studies of SLIM. In the future, it would be interesting to combine the DUO formulation with recent extensions, such as SLIM crossover operators [19] and SLIM binary-classification schemes [7]. Another promising direction is to move beyond the uniform block-selection scheme used here and explore alternative ways of setting the selection probabilities of the SIG2, SIG1, and ABS blocks, possibly also extending the framework to other types of blocks. For example, Operator Equalisation [24] could be repurposed to regulate block-type frequencies, based on their observed contribution to fitness, model size, interpretability, or other chosen criteria. The inverted pattern observed on the Istanbul test case also raises a broader question, that merits further investigation: can one predict, from dataset characteristics, whether DUO-SUM or DUO-MUL is the more suitable choice for a given problem? Investigating this would require extending the study to a larger pool of datasets and computing a richer set of meta-data descriptors, with the goal of distilling practical indications that relate dataset properties to the preferred SLIM-DUO variant. The rich benchmark suite presented in [18], alongside with a comparison of SLIM-DUO with the state-of-the-art regression algorithms discussed in that same work can be an appropriate future step. If successful, this would lessen the need to benchmark both DUO variants on every new task, streamlining the experimental workflow and freeing computational budget for a deeper exploration of the remaining hyperparameters.

Acknowledgments

This work was supported by national funds through FCT (Fundação para a Ciência e a Tecnologia), under the project - UID/04152/2025 - Centro de Investigação em Gestão de Informação (MagIC)/NOVA IMS - https://doi.org/10.54499/UID/04152/2025 (2025-01-01/2028-12-31) and UID/PRR/04152/2025 https://doi.org/10.54499/UID/PRR/04152/2025 (2025-01-01/2026-06-30). The manuscript was reviewed using AI-based language-editing assistance to improve English readability.

References

[1] Oguz Akbilgic, Hamparsum Bozdogan, and M Erdal Balaban. 2014. A novel hybrid RBF neural networks model as a forecaster. *Statistics and Computing* 24, 3 (2014), 365–375.
[2] 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.
[3] Illya Bakurov, Jose Manuel Munoz Contreras, Mauro Castelli, Nuno Rodrigues, Sara Silva, Leonardo Trujillo, and Leonardo Vanneschi. 2024. Geometric semantic genetic programming with normalized and standardized random programs. *Genetic Programming and Evolvable Machines* 25, 1 (2024), 6.
[4] Lorenzo Bonin, Luigi Rovito, Andrea De Lorenzo, and Luca Manzoni. 2024. Cellular geometric semantic genetic programming. *Genetic Programming and Evolvable Machines* 25, 1 (2024), 8.
[5] Mauro Castelli, Leonardo Trujillo, and Leonardo Vanneschi. 2015. Energy Consumption Forecasting Using Semantic-Based Genetic Programming with Local Search Optimizer. *Intell. Neuroscience* 2015, Article 57 (jan 2015), 1 pages.

doi:10.1155/2015/971908

[6] Mauro Castelli, Leonardo Vanneschi, and Sara Silva. 2013. Prediction of high performance concrete strength using Genetic Programming with geometric semantic genetic operators. *Expert Systems with Applications* 40, 17 (2013), 6856 – 6862.

[7] Leon Debatin. 2025. *Enhancing Symbolic Regression-Based GSGP for Binary Classification.* Master's thesis. Universidade NOVA de Lisboa (Portugal).

[8] Grant Dick. 2024. An ensemble learning interpretation of geometric semantic genetic programming. *Genetic programming and evolvable machines* 25, 1 (2024), 9.

[9] Davide Farinati, Gloria Pietropolli, and Leonardo Vanneschi. 2025. Exploring the Impact of Data Scale on Mutation Step Size in SLIM-GSGP. In *European Conference on Genetic Programming (Part of EvoStar).* Springer, 35–51.

[10] Davide Farinati and Leonardo Vanneschi. 2025. An empirical study on the feature selection ability of SLIM-GSGP. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion.* 607–610.

[11] Akira Hara, Jun-ichi Kushida, Ryo Tanemura, and Tetsuyuki Takahama. 2016. Deterministic crossover based on target semantics in geometric semantic genetic programming. In *2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI).* IEEE, 197–202.

[12] Jan-Benedikt Jagusch, Ivo Gonçalves, and Mauro Castelli. 2018. Neuroevolution under unimodal error landscapes: an exploration of the semantic learning machine algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (Kyoto, Japan) *(GECCO '18).* Association for Computing Machinery, New York, NY, USA, 159–160. doi:10.1145/3205651.3205778

[13] JRGP Koza. 1992. On the programming of computers by means of natural selection. *Genetic programming* (1992).

[14] Christophe Leys, Christophe Ley, Olivier Klein, Philippe Bernard, and Laurent Licata. 2013. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of experimental social psychology* 49, 4 (2013), 764–766.

[15] Francesco Marchetti, Gloria Pietropolli, Federico Julian Camerota Verdù, Mauro Castelli, and Edmondo Minisci. 2024. Automatic design of interpretable control laws through parametrized Genetic Programming with adjoint state method gradient evaluation. *Applied Soft Computing* 159 (2024), 111654.

[16] James McDermott, Alexandros Agapitos, Anthony Brabazon, and Michael O'Neill. 2014. Geometric semantic genetic programming for financial data. In *Applications of Evolutionary Computation: 17th European Conference, EvoApplications 2014, Granada, Spain, April 23-25, 2014, Revised Selected Papers 17.* Springer, 215–226.

[17] Alberto Moraglio, Krzysztof Krawiec, and Colin G Johnson. 2012. Geometric semantic genetic programming. In *Parallel Problem Solving from Nature-PPSN XII: 12th International Conference, Taormina, Italy, September 1-5, 2012, Proceedings, Part I 12.* Springer, 21–31.

[18] Jose Manuel Muñoz Contreras, Leonardo Trujillo, Daniel E. Hernandez, and Luis A. Cardenas Florido. 2024. Benchmarking GSGP: Still competitive 10 years later? *Genetic Programming and Evolvable Machines* 26, 1 (Dec. 2024), 18 pages. doi:10.1007/s10710-024-09504-3

[19] Sofia Carreira Da Conceição Alves Pereira. 2025. *Exploring Crossover Operators in SLIM_GSGP: Towards More Compact and Generalizable Models: a Multi-Phase Evaluation of Novel Crossover Operators for Efficient and Interpretable Evolutionary Learning.* Master's thesis. Universidade NOVA de Lisboa (Portugal).

[20] Riccardo Poli, William B. Langdon, and Nicholas F. McPhee. 2008. *A Field Guide to Genetic Programming* (1st ed.). Lulu Enterprises UK Ltd, UK.

[21] Mohammad Rafiei. 2018. Residential building data set. *UCI Machine Learning Repository* (2018).

[22] Liah Rosenfeld, Davide Farinati, Diogo Rasteiro, Gloria Pietropolli, Karina Rebuli, Sara Silva, and Leonardo Vanneschi. 2025. Slim_gsgp: A Python Library for Non-Bloating GSGP. In *Proceedings of the Genetic and Evolutionary Computation Conference* (NH Malaga Hotel, Malaga, Spain) *(GECCO '25).* Association for Computing Machinery, New York, NY, USA, 1026–1034. doi:10.1145/3712256.3726398

[23] Luigi Rovito, Lorenzo Bonin, Davide Farinati, Leonardo Vanneschi, Luca Manzoni, Andrea De Lorenzo, and Gloria Pietropolli. 2025. Exploring the Integration of Cellular Structures in Genetic Programming-Based Methods. In *European Conference on Genetic Programming (Part of EvoStar).* Springer, 120–138.

[24] Sara Silva, Stephen Dignum, and Leonardo Vanneschi. 2012. Operator equalisation for bloat free genetic programming and a survey of bloat control methods. *Genetic Programming and Evolvable Machines* 13, 2 (2012), 197–238.

[25] Leonardo Trujillo, Emigdio Z-Flores, Perla S Juárez-Smith, Pierrick Legrand, Sara Silva, Mauro Castelli, Leonardo Vanneschi, Oliver Schütze, and Luis Muñoz. 2018. Local search is underused in genetic programming. *Genetic Programming Theory and Practice XIV* (2018), 119–137.

[26] Athanasios Tsanas and Angeliki Xifara. 2012. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and buildings* 49 (2012), 560–567.

[27] Leonardo Vanneschi. 2024. SLIM_GSGP: The non-bloating geometric semantic genetic programming. In *European Conference on Genetic Programming (Part of EvoStar).* Springer, 125–141.

[28] Leonardo Vanneschi, Mauro Castelli, Luca Manzoni, Krzysztof Krawiec, Alberto Moraglio, Sara Silva, and Ivo Gonçalves. 2017. PSXO: population-wide semantic crossover. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion.* 257–258.

[29] Leonardo Vanneschi, Davide Farinati, Diogo Rasteiro, Liah Rosenfeld, Gloria Pietropolli, and Sara Silva. 2025. Exploring non-bloating geometric semantic genetic programming. *Genetic Programming Theory and Practice XXI* (2025), 237–258.

[30] Leonardo Vanneschi and Sara Silva. 2023. *Lectures on Intelligent Systems.* Springer.

[31] Leonardo Vanneschi, Sara Silva, Mauro Castelli, and Luca Manzoni. 2013. Geometric Semantic Genetic Programming for Real Life Applications. In *Genetic Programming Theory and Practice XI*, Rick Riolo, Katya Vladislavleva, and Jason Moore (Eds.). Springer US, Computer Science Collection.

[32] David H Wolpert. 1996. The lack of a priori distinctions between learning algorithms. *Neural computation* 8, 7 (1996), 1341–1390.