A General Feature-Informed Crossover for Two-Stage Feature Selection in Symbolic Regression

Hengzhe Zhang, Qi Chen, Bing Xue, Wolfgang Banzhaf, Mengjie Zhang

May 30, 2025

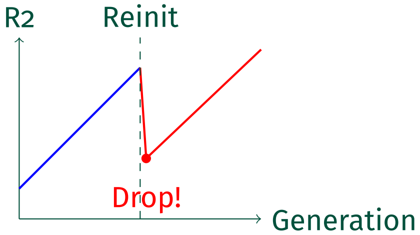
Outline

Introduction

- **Symbolic regression** generates mathematical expressions $f(X)$ that map input data X to target output Y
- **Genetic Programming (GP)** naturally suited for this task:
  - Gradient-free, population-based optimization
  - Creates and optimizes variable-length symbolic expressions
  - Inherent ability to perform feature selection
- **Challenge**: As dimensionality increases, GP effectiveness decreases due to rapid growth of search space

Current Approach:

1. **Stage 1:** Run GP on all features
2. **Feature Selection:** Analyze top individuals
    - Frequency analysis
    - Permutation importance [1]
    - Shapley values [2, 3]
3. **Stage 2:** Reinitialize population with selected features



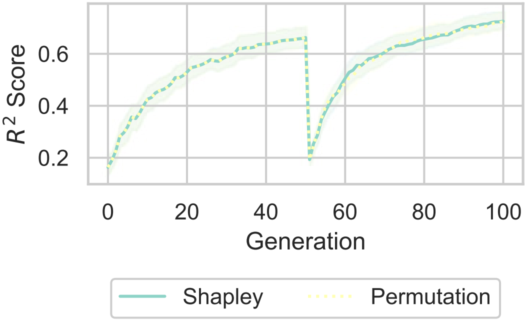
Problem: Reinitialization disrupts effective building blocks accumulated during evolution!

[1] Chen, Qi, Mengjie Zhang, and Bing Xue, 2017“, ‘Feature selection to improve generalization of genetic programming for high-dimensional symbolic regression’”, *IEEE Transactions on Evolutionary Computation*

[2] Wang, Chunyu et al., 2023“, ‘Shapley Value Based Feature Selection to Improve Generalization of Genetic Programming for High-Dimensional Symbolic Regression’”, *Australasian Conference on Data Science and Machine Learning*

[3] Rimas, Mohamad, Qi Chen, and Mengjie Zhang, 2024“, ‘Feature Selection for GPSR Based on Maximal Information Coefficient and Shapley Values’”, *2024 IEEE Congress on Evolutionary Computation (CEC)*

Performance Drop Due to Reinitialization



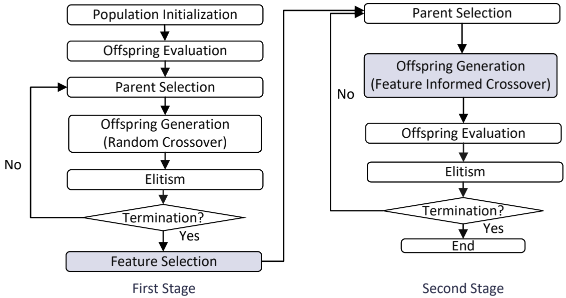
- Significant performance drop occurs at reinitialization point
- Accumulated building blocks are completely discarded
- Several generations needed to recover performance

Our Contributions

1. **Feature-Informed Crossover (FIC) Operator**:
   - Eliminates need for population reinitialization
   - Preserves evolved building blocks after feature selection
   - Gradually eliminates irrelevant features during evolution
2. **Subtree Importance Measurement**:
   - Quantifies importance based on selected features
   - Guides crossover to prioritize useful subtrees
3. **General Applicability**:
   - Works with three different feature selection mechanisms
   - Validated across 98 datasets
   - Demonstrates broad effectiveness

Proposed Method

Algorithm Framework Overview



Key Innovation: No population reinitialization between stages!

- Feature selection occurs at midpoint of evolution
- Population continues with feature-informed operators
- Building blocks preserved throughout process

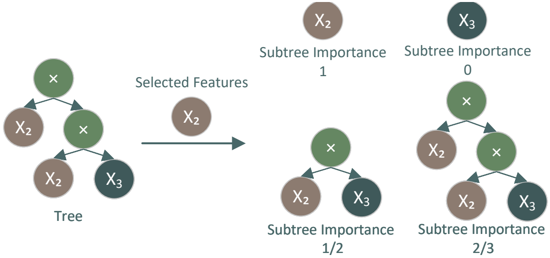
Importance calculation:

$$w_\psi = \frac{\omega_w}{\sigma_w}$$

where:

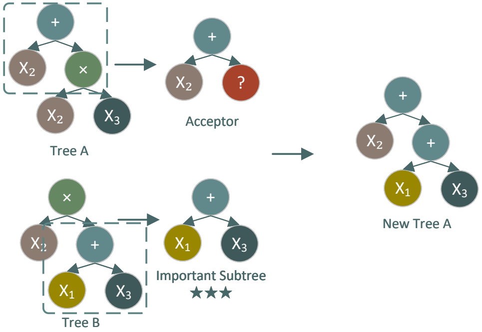
- σ_w: total terminal nodes in subtree ψ
- ω_w: weight of selected terminals in ψ
- Node weight = 1 if selected feature or constant, 0 otherwise

Normalization prevents size bias in importance



Crossover Process:

1. **Probability Conversion:**
   - Convert subtree weights to probabilities
   - Add 1 to each weight for smoothing
   - Normalize to sum to 1
2. **Subtree Selection:**
   - **Donor:** Select based on importance probability
   - **Acceptor:** Select uniformly at random
3. **Replacement:**
   - Replace acceptor with donor subtree
   - Preserve high-importance building blocks



Experimental Settings

Feature Selection Methods:

- **Frequency-based**: Count feature usage
- **Permutation-based**: Measure removal impact

$$I(X_j) = \mathbb{E}[R^2_{\text{orig}} - R^2_{\text{perm}(X_j)}]$$

- **Shapley-based**: Marginal contributions

Datasets:

- 98 datasets from Penn ML Benchmark
- Fewer than 2000 instances each
- 80:20 train/test split
- Features standardized

GP Parameters:

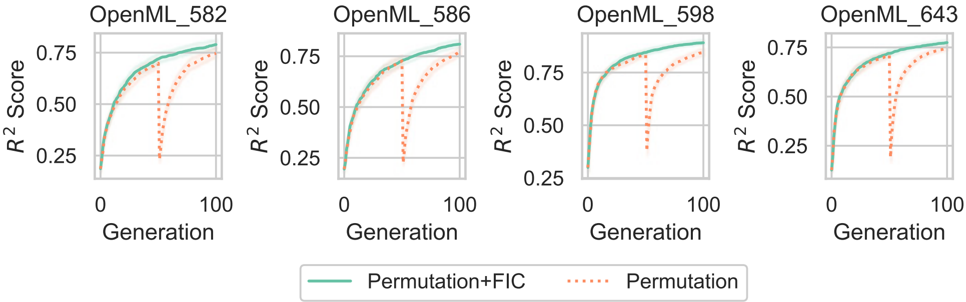
- Population: 100
- Generations: 50 + 50 (two stages)
- Max depth: 10
- Crossover/Mutation: 0.9/0.1
- Selection: Lexicase
- Functions: +, -, *, AQ, etc.

Evaluation:

- R^2 score metric
- 30 runs (6 groups × 5-fold CV)
- Wilcoxon signed-rank test
- Significance level: 0.05

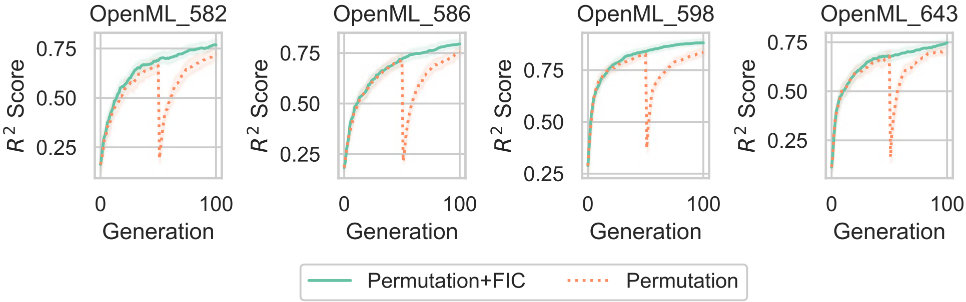
Experimental Results

Training Performance Comparison



- **FIC significantly outperforms baseline** on 73/98 datasets (permutation)
- **No performance drop** at stage transition
- **Smooth evolution** preserves building blocks
- Similar improvements with Shapley (72/98) and Frequency (75/98) methods

Test Performance Results

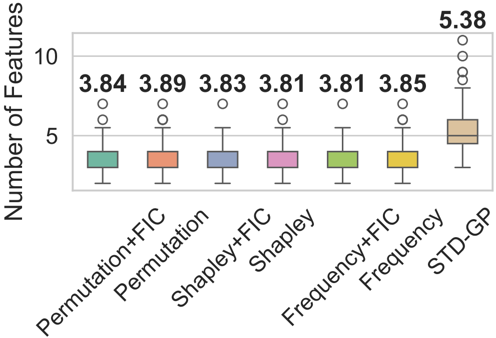


- **Generalization improvement:** 42/98 datasets (permutation)
- **Consistent across methods:** Shapley (39/98), Frequency (41/98)
- **Avoids performance cliff** at stage transition

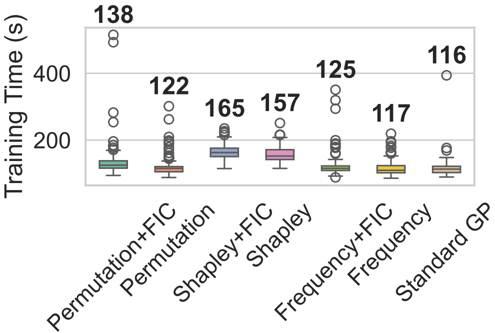
Table: Statistical comparison of test R^2 scores

	Permutation	Shapley+FIC	Shapley	Frequency+FIC	Frequency
Permutation+FIC	42(+)/56(∼)/0(-)	0(+)/98(∼)/0(-)	38(+)/60(∼)/0(-)	0(+)/98(∼)/0(-)	43(+)/54(∼)/1(-)
Shapley+FIC	—	—	39(+)/59(∼)/0(-)	0(+)/98(∼)/0(-)	41(+)/57(∼)/0(-)
Frequency+FIC	—	—	—	—	41(+)/56(∼)/1(-)

- **Consistent improvements** across all three feature selection methods
- **Robust performance**: Very few cases where FIC performs worse
- **General applicability** demonstrated



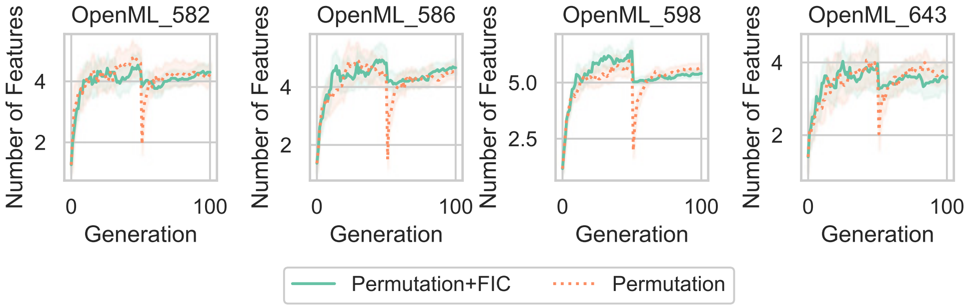
Number of Features Used



Training Time

- **Feature reduction**: No significant difference vs. reinitialization
- **Interpretability maintained**: Comparable feature counts
- **Computational efficiency**: Minimal training time increase
- **Cost reduction**: Fewer features needed for deployment

Feature Evolution Trajectories



- **Building block preservation**: Avoids drastic feature reduction
- **Gradual refinement**: Natural evolution toward optimal feature set

Conclusions

- **Problem Addressed**: Reinitialization in two-stage feature selection disrupts building blocks
- **Solution Proposed**: Feature-Informed Crossover (FIC) operator
  - Preserves evolved building blocks
  - Gradually eliminates irrelevant features
  - Avoids performance drops during stage transitions
- **Experimental Validation**:
  - Significant improvements across 98 datasets
  - Consistent results with three feature selection methods
  - Maintained interpretability with comparable feature counts
  - Minimal computational overhead
- **Broad Applicability**: Works with frequency, permutation, and Shapley-based selection

Future Directions

- **Multi-stage Extension**:
  - Iterative feature selection with FIC
- **Advanced Importance Measures**:
  - Real-time feature importance calculation

Thank You!

Questions & Discussion

Hengzhe Zhang, Qi Chen, Bing Xue,
Wolfgang Banzhaf, Mengjie Zhang

hengzhe.zhang@ecs.vuw.ac.nz