Using Machine Learning to Predict the Sequences of Optimization Passes

Presenters: Anurag Bangera, Chirag Bangera, Jonhan Chen, Richard Wang

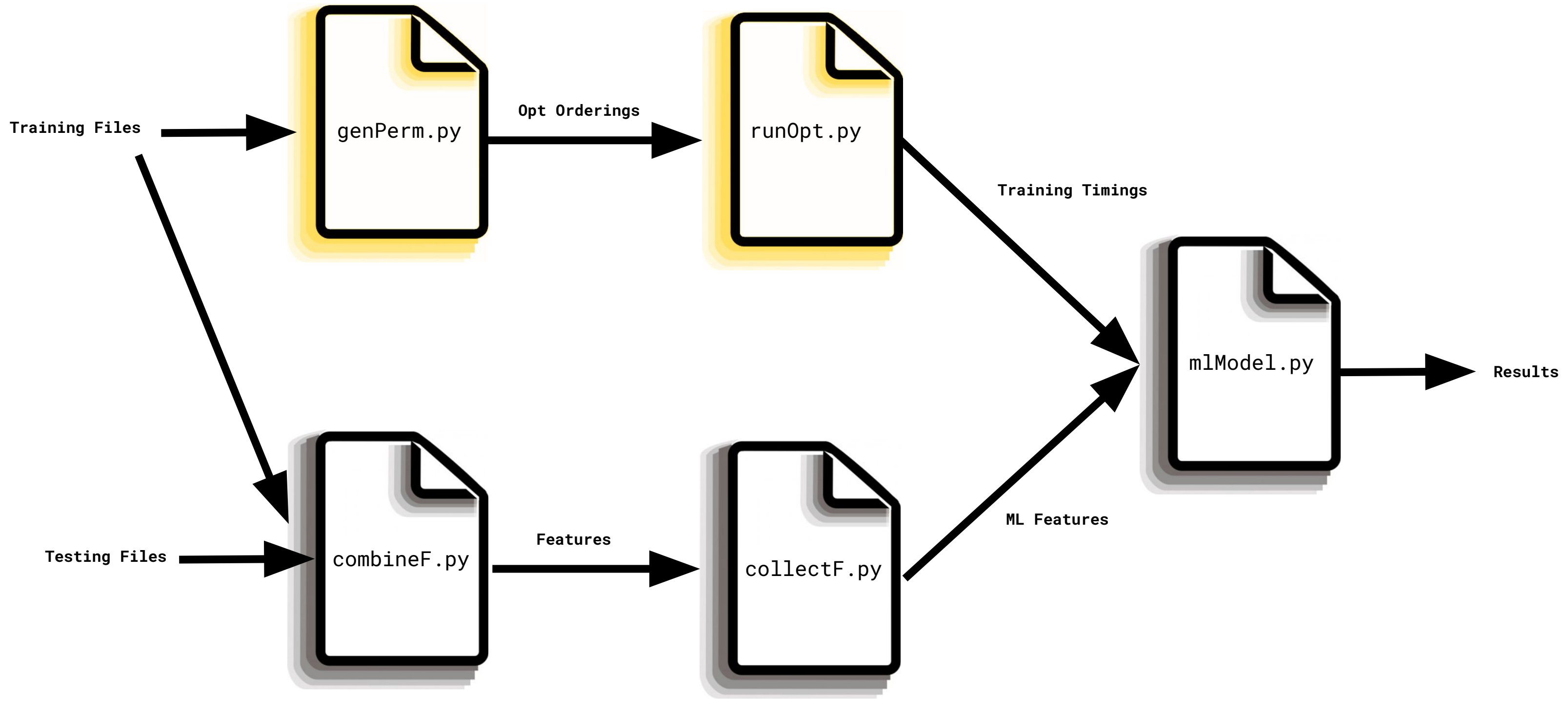
December 13th, 2023

Table of Contents

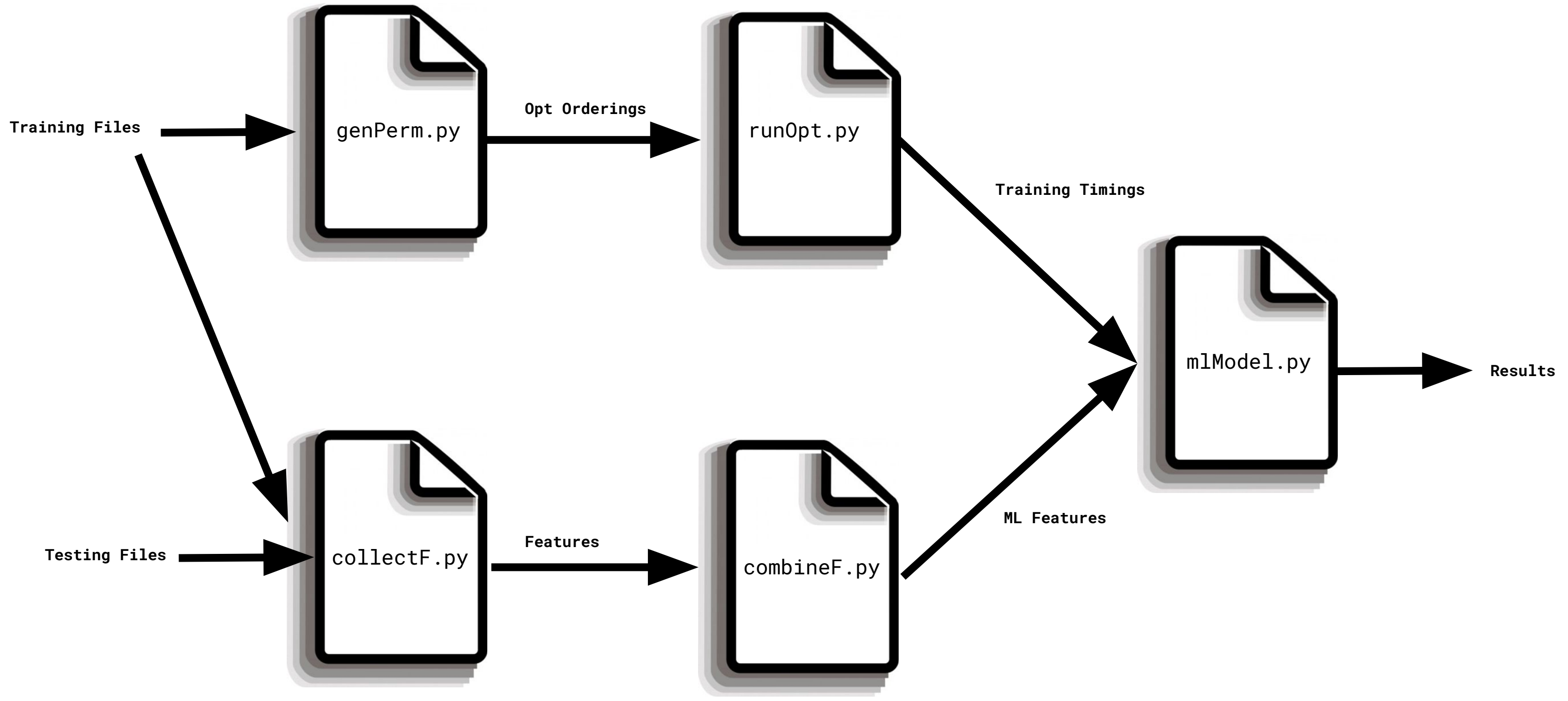
Project Summary

- Using Machine Learning to Optimize Phase Ordering
- Gathering Training Data from Training Files
  - Find “Best” Phase Orderings
  - Collect Program Features
- Train ML Models on Gathered Data
  - KNN, SVR, Random Forest, Ada Boost, Gradient Boosting
- Used Trained Models on Test Files

Project Structure

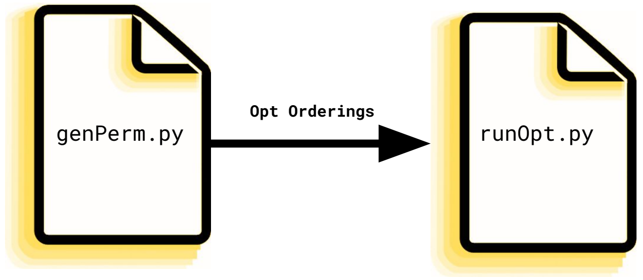


Project Structure



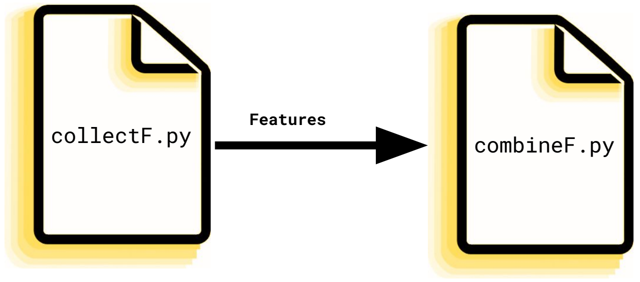
Gathering Training Data: Timings

- generatePermutations.py
  - Works with a static set of optimizations
  - Generates permutations of optimization orderings on training files
- runOptimizations.py
  - Gathers the “best” orderings by profiling each permutation



Gathering Training Data: Feature Collection

- collectFeatures.py
  - Finds and stores all features from training data
  - Finds and stores all features from test data
- combineFeatures.py
  - Transforms features to be ML Model Readable



Gathering Training Data: LLVM Passes

1. Function Pass
    a. High-level and control flow
2. Loop Pass
    a. Depth and loop hierarchy
3. Module Pass
    a. Static instruction counts
    b. Compare our results to the features used by our research paper

65 total features

Function Pass

- Total Basic Block Count
- Average Instruction Count Per BB
- Dynamic Instruction Count Categories
- Static Instruction Count Categories

30 total features

- Dynamic-Static Instruction Count Ratios
- Biased/Unbiased Branch Counts
- Loop Count
- Basic Blocks Per Loop
- Average Static Instruction Count Categories Per Loop
- Recursive Call Count

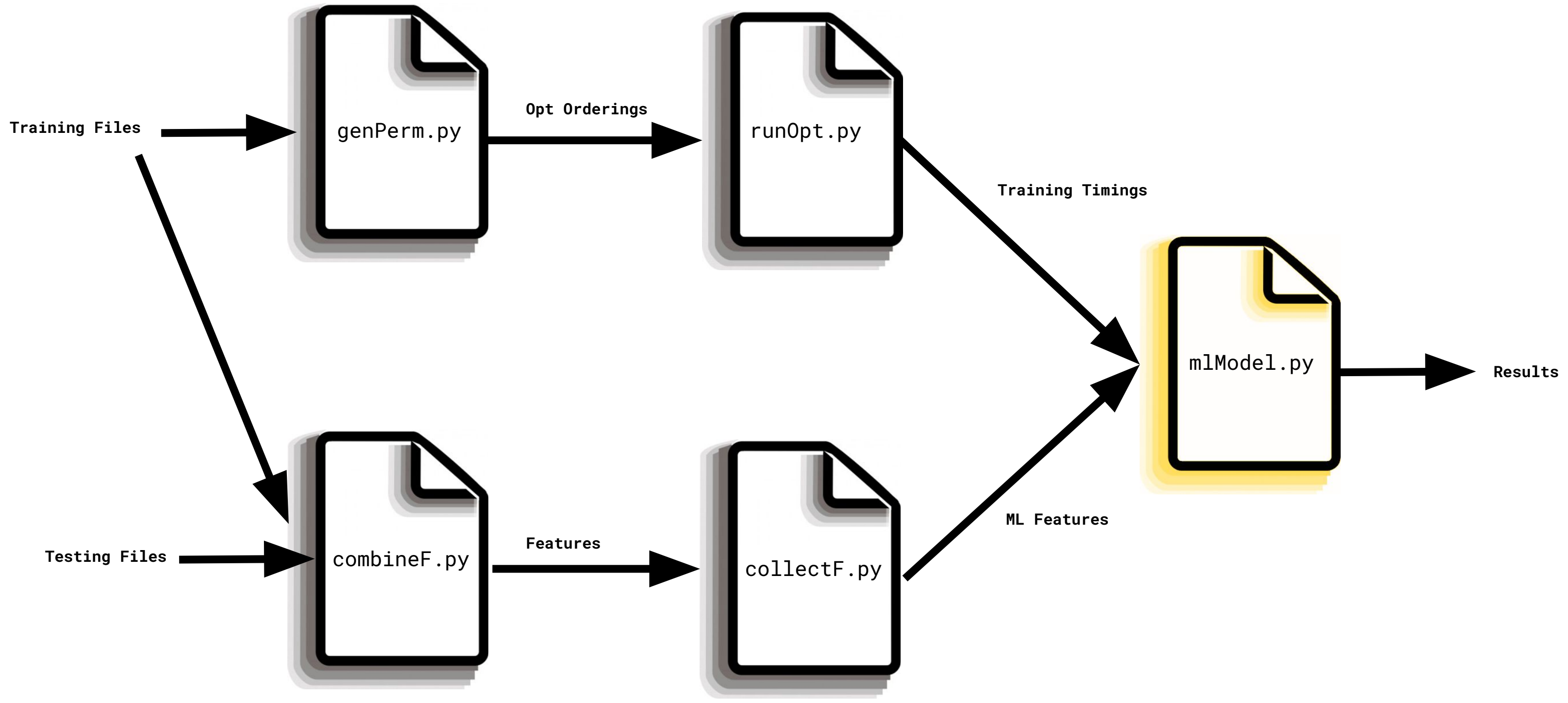
Loop Pass

- Number Of Loops With Nesting
- Number Of Outermost Loops
- Average Loop Nesting Depth
- Average Outermost Loop Depth
- Deepest Loop Nesting Depth

5 total features

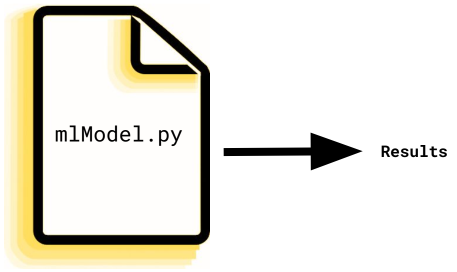
35 total features combined with function pass

Project Structure



Machine Learning Models

- mlModel.py
  - Contains:
    - K Nearest Neighbors
    - Support Vector Regression (SVR)
      - Kernels: Linear, Poly, RBF
    - Random Forest
  - Ensemble Methods (combine multiple models)
    - Gradient Boost
    - AdaBoost

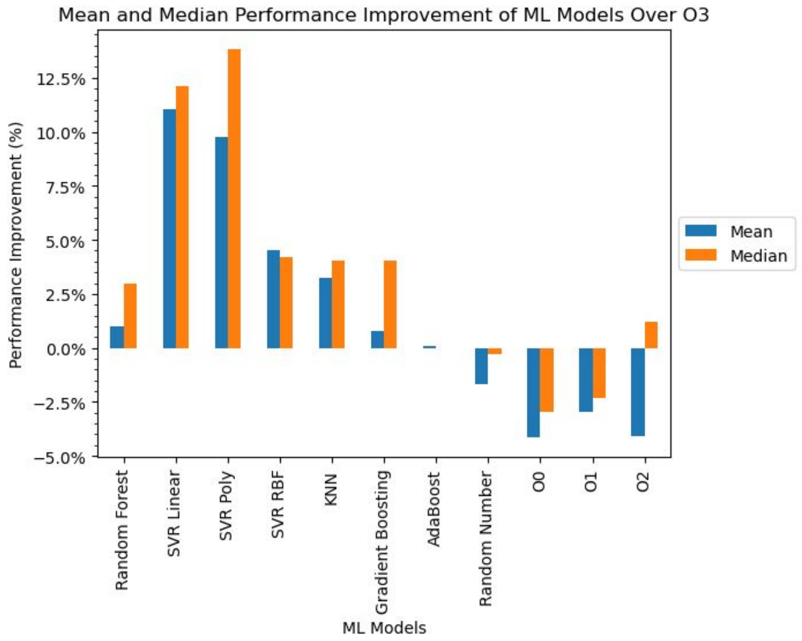


Demo

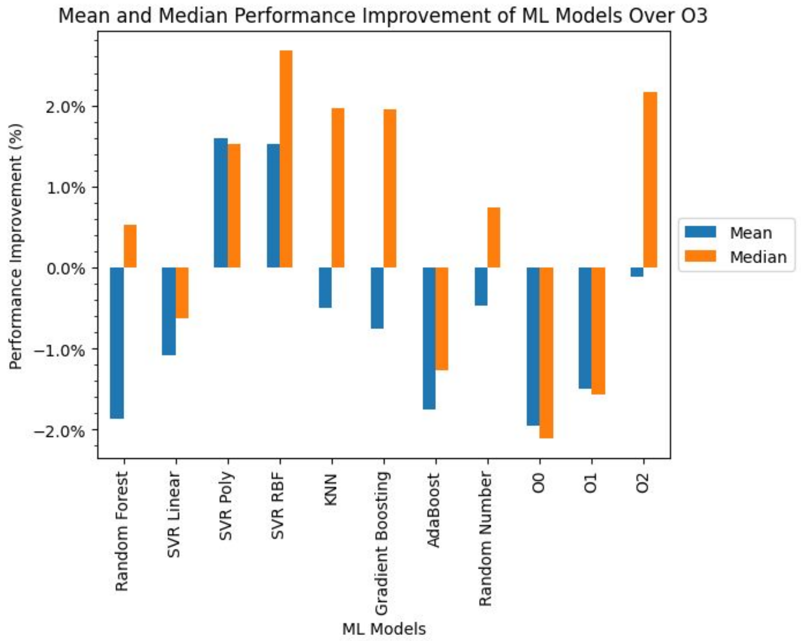
https://github.com/EECS-583-Group-24/ML-LOOP/tree/main/demo

Results

- Our ML Results with our Features

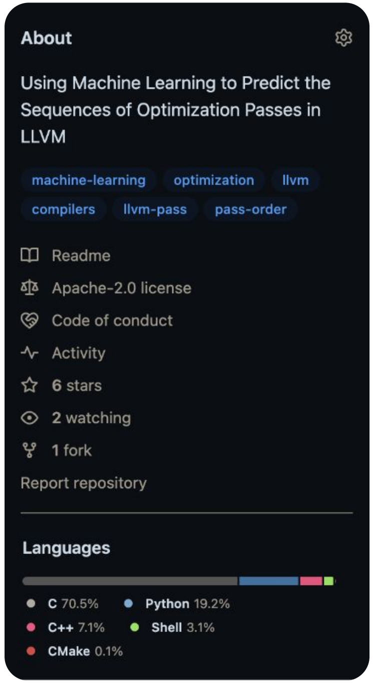


- Original Paper Features



Future Works

- Github: Link
  - More Testing / Training data
    - More Diverse Data Set
    - More Complex Data Set
  - Better ML Models
    - Better Features
    - Better Reinforcement Learning
    - Better Training Heuristic
  - Get Open Source Contributors
    - More Forks / Stars



About

Using Machine Learning to Predict the Sequences of Optimization Passes in LLVM

machine-learning optimization llvm compilers llvm-pass pass-order

Readme
Apache-2.0 license
Code of conduct
Activity
6 stars
2 watching
1 fork
Report repository

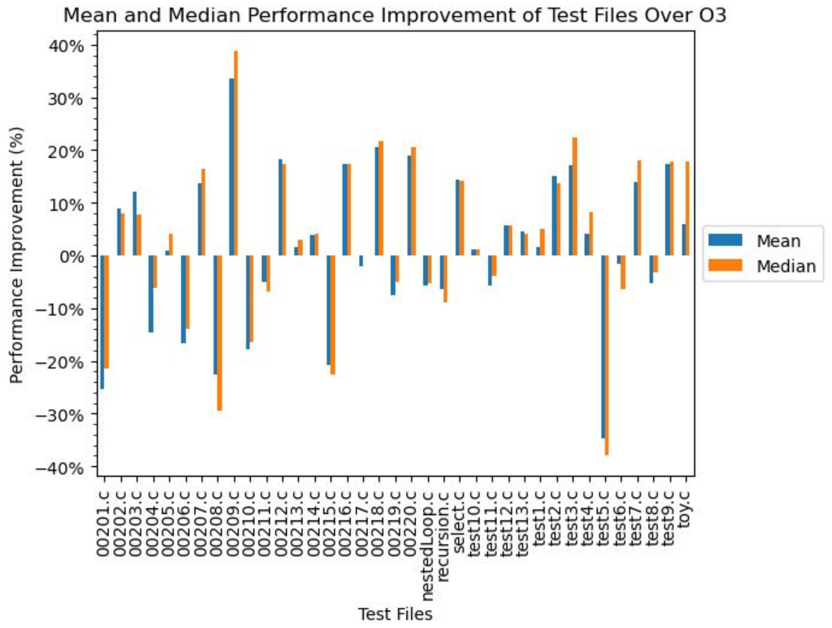
Languages

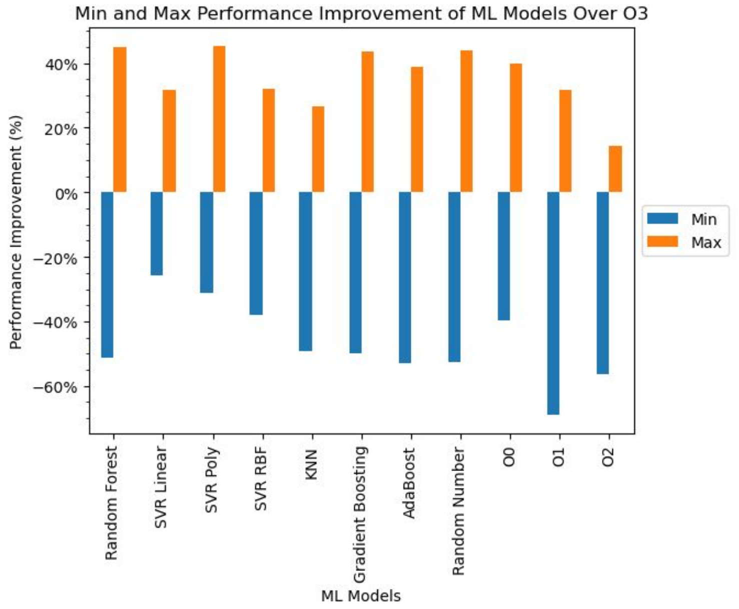
C 70.5% Python 19.2%
C++ 7.1% Shell 3.1%
CMake 0.1%

Thank You!

Any Questions?

Results Appendix





Many more graphs/tables: https://github.com/EECS-583-Group-24/ML-LOOP/tree/main/figures
Graphs for original paper: https://github.com/EECS-583-Group-24/ML-LOOP/tree/paper/figures/originalPaper