# UniTSyn: A Large-Scale Dataset Capable of Enhancing the Prowess of Large Language Models for Program Testing

Yifeng He, Jiabo Huang, Yuyang Rong, Yiwen Guo, Ethan Wang, Hao Chen
UC Davis

Noyce Symposium 2024

## Project Statement

The remarkable capability of large language models (LLMs) in generating high-quality code has drawn increasing attention in the software testing community. However, existing code LLMs often demonstrate unsatisfactory capabilities in generating accurate, complete tests since they were trained on code snippets collected without differentiating between code for testing and for other purposes.

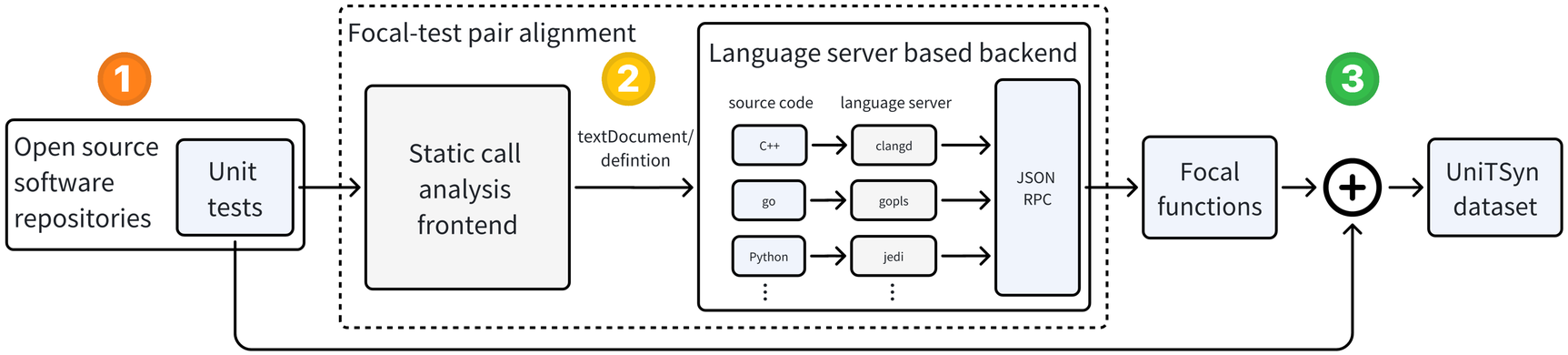
## Project Goals

In this paper, we present a large-scale dataset, UniTSyn, which can enhance LLMs for Unit Test Synthesis. Associating tests with the tested functions is crucial for LLMs to infer the expected behavior and the logic paths to be verified. By leveraging Language Server Protocol, UniTSyn achieves the challenging goal of collecting focal-test pairs without per-project execution setups or per-language heuristics, which tend to be fragile and difficult to scale. Containing 2.7 million focal-test pairs across five mainstream programming languages, it can enhance the test generation ability of LLMs. Our research questions are:

1. How accurate are the test cases generated by LLMs?
2. How many of the generated tests are complete?
3. Is it necessary to train LLMs with pairwise focal and test functions?
4. What are the effects of training with multilingual testing code?

## Methods

1. We download open-source software repositories and extract their unit tests. 2. We designed a flexible, unified static analyzer to find calls to focal functions from the unit tests, which decreases the complexity of performing call analysis for each language. We integrated the Language Server Protocol (LSP) into the dataset-building process to harness its language extensibility and call-definition matching ability. 3. We store the aligned function-level focal-test pairs as training data.

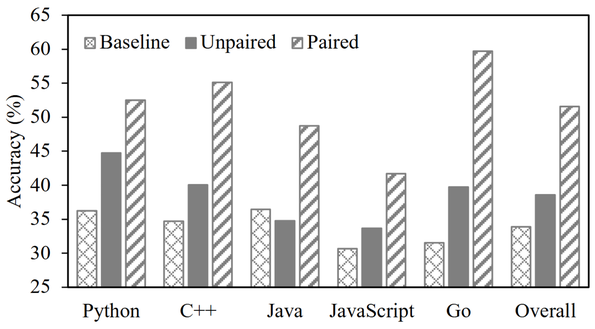



## Results

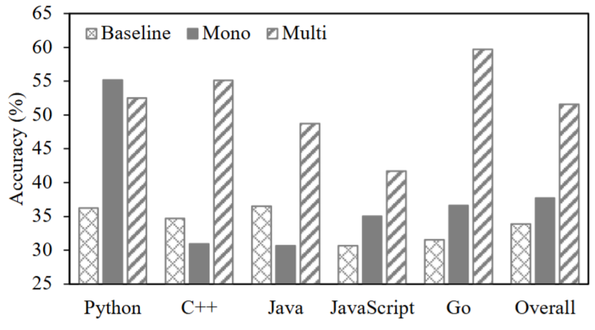
| Model | #Params | Py | C++ | Java | JS | Go | Avg |
|---|---|---|---|---|---|---|---|
| CodeT5p | 770M | 30.6 | 33.7 | 26.9 | 37.1 | 32.9 | 32.2 |
| CodeGen2 | 1.0B | 34.0 | 40.7 | 24.1 | 30.5 | 36.1 | 33.1 |
| WizardCoder | 1.0B | 36.8 | 43.9 | 28.7 | 31.3 | 47.7 | 37.7 |
| InCoder | 1.3B | 34.2 | 33.5 | 22.6 | 24.4 | 31.5 | 29.2 |
| SantaCoder | 1.1B | 36.2 | 34.7 | 36.5 | 30.6 | 31.5 | 33.9 |
| CAT-LM† | 2.7B | 37.5 | 31.6 | 34.4 | 29.2 | 36.9 | 33.9 |
| **UniTester† (Ours)** | 1.1B | **52.5** | **55.1** | **48.8** | **41.7** | **59.7** | **51.5** |

| Model | Python | | C++ | | Java | | Javascript | | Go | |
|---|---|---|---|---|---|---|---|---|---|---|
| | #Pass | Line | #Pass | Line | #Pass | Line | #Pass | Line | #Pass | Stmt |
| CodeT5p | 10.0 | 5.72 | 0.7 | 0.43 | 40.3 | 4.22 | 4.9 | 2.07 | 1.7 | 0.73 |
| CodeGen2 | 4.1 | 2.41 | 11.6 | 7.07 | 52.3 | 5.12 | 48.5 | 27.65 | 19.2 | 10.99 |
| WizardCoder | 16.1 | 9.39 | 3.7 | 2.24 | 47.7 | 5.62 | 9.2 | 5.50 | 0.7 | 0.42 |
| InCoder | 3.0 | 1.76 | 0.0 | 0.00 | 15.0 | 1.54 | 0.5 | 0.29 | 1.3 | 0.78 |
| SantaCoder | 4.5 | 2.62 | 4.9 | 2.99 | 50.1 | 4.74 | 5.9 | 3.53 | 0.7 | 0.43 |
| CAT-LM† | 35.9 | 19.51 | 0.0 | 0.00 | 0.9 | 0.07 | 9.2 | 4.53 | 0.0 | 0.00 |
| **UniTester† (Ours)** | 41.2 | 20.71 | 28.1 | 13.39 | 103.1 | 10.78 | 53.3 | 27.59 | 36.0 | 12.39 |

RQ 1 & 2: Our model trained on our dataset achieves the best assertion accuracy and branch/statement coverage.




RQ 3: Training with function-level aligned focal-test pairs increases assertion accuracy in all five languages.




RQ 4: For Python, the monolingual model demonstrated superior capability in assertion accuracy. For other languages with stricter syntax, the multilingual model achieves better results.