Are Mutants a Valid Substitute for Real Faults in Software Testing?

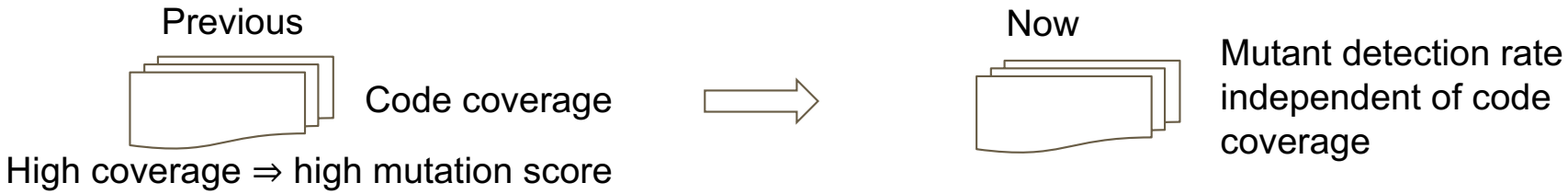
Anna Deng & Ping Lin

What is a Mutant

- Created by systematically injecting small artificial faults into the program under test
- Mutation operators
  - Replace constants, replace operators, modify branch conditions, delete statements etc.
- Mutation score
  - Percentage of mutants that a test suite can distinguish from the original program.
  - Killed Mutants / Total number of Mutants

Test Metrics

- A good test suite detects real faults
- Test quality metrics
  - Test suite evaluation, selection, minimization, generation...
- Problem: ability to detect real faults is unknown
- Solution: use a proxy metric for test quality
  - Code coverage ratio
  - **Mutant detection rate $\stackrel{?}{=}$ Real fault detection rate**



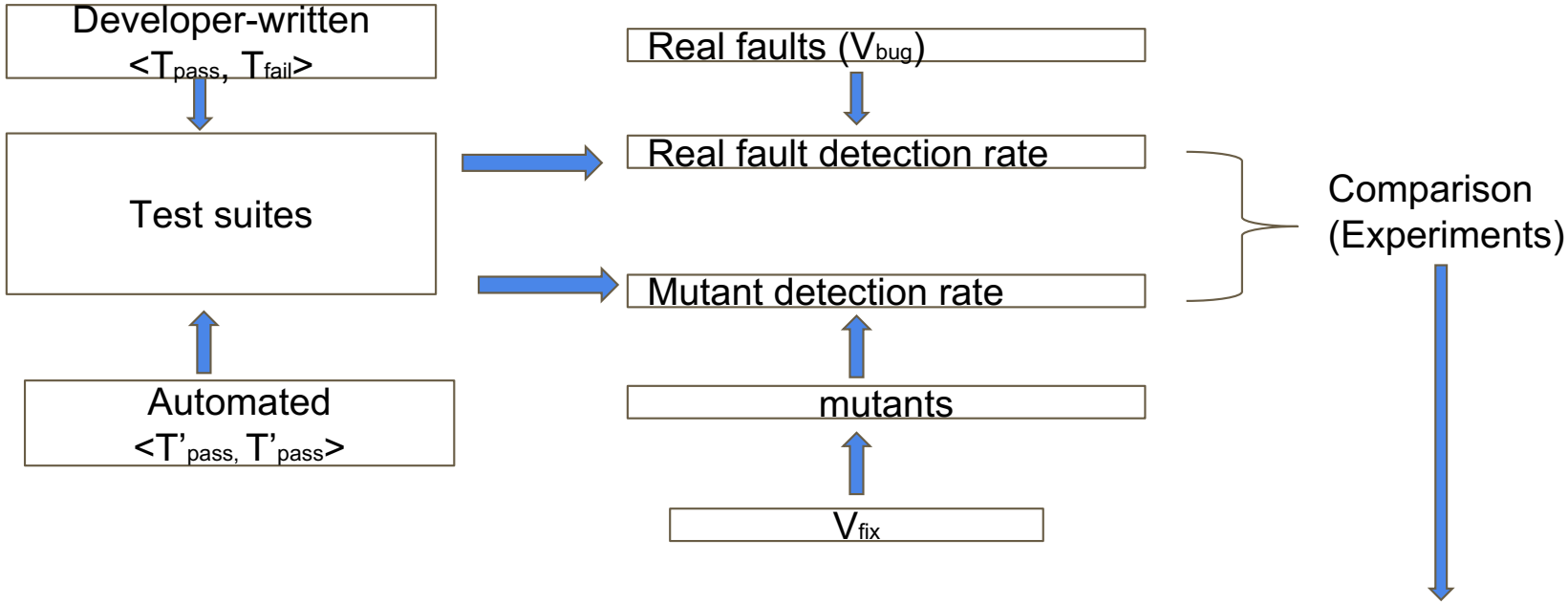
Research Questions

1. Are real faults coupled to mutants generated by commonly used mutation operators?

2. What types of real faults are not coupled to mutants?

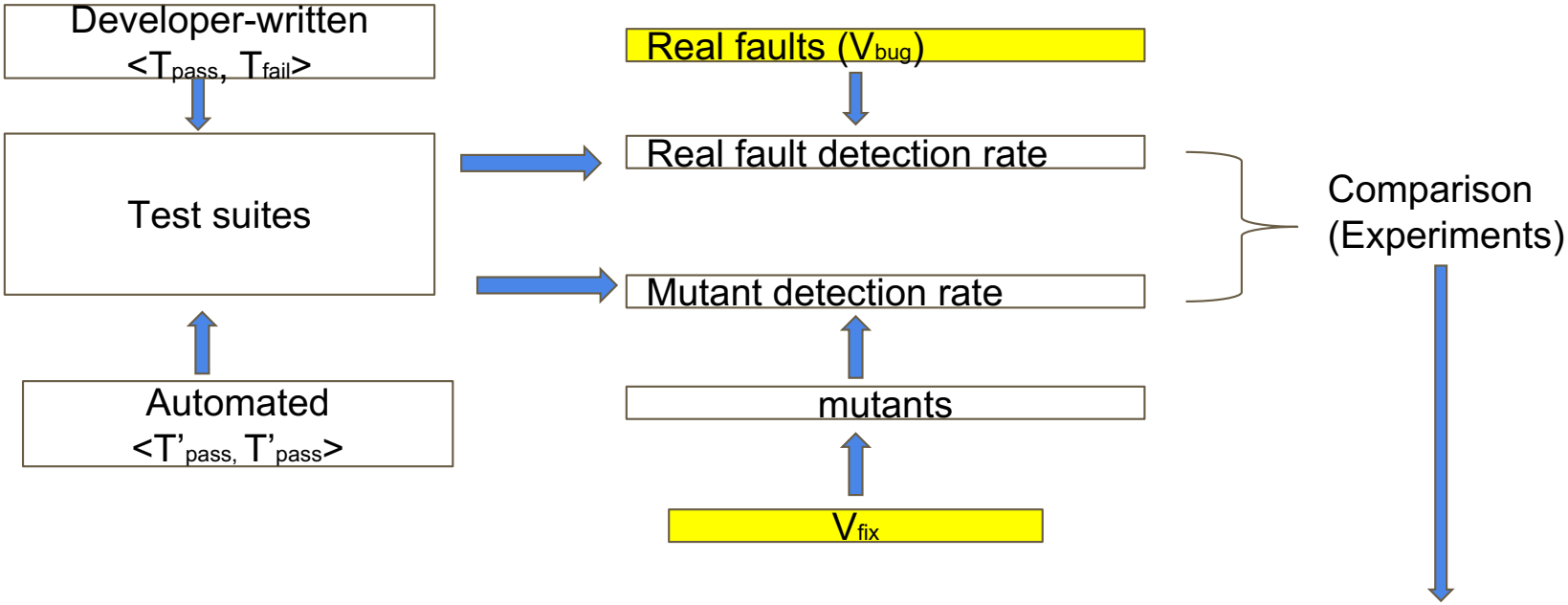
3. Is mutant detection correlated with real fault detection?

Methodology



Are mutants a valid substitute for real faults?

Methodology



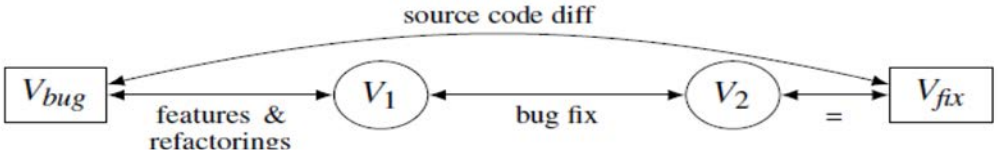
Are mutants a valid substitute for real faults?

Locating and Isolating Real Faults

- Obtain candidate revisions

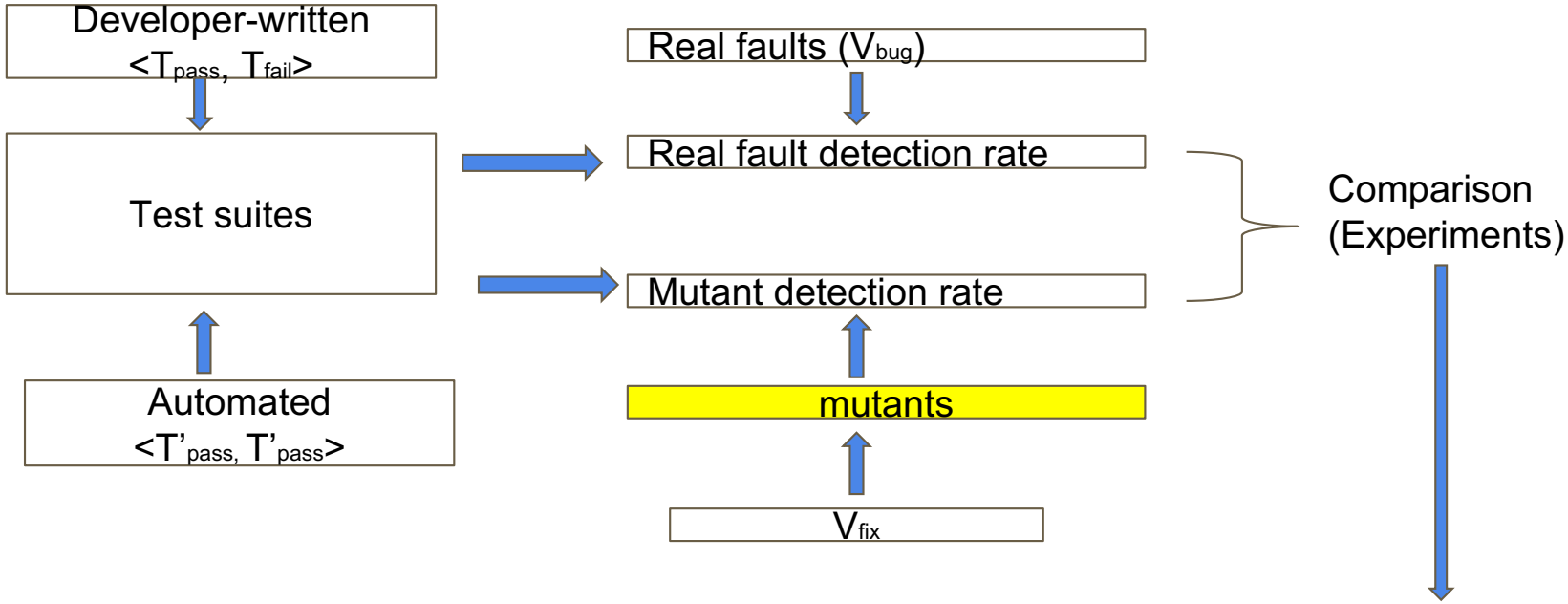
mine version control system: explicit mentions of bug fix → Candidate revision pair <Rev_{bug} , Rev_{fix} >

- Discard non-reproducible faults
  - Reproducible: detect faults with existing tests
  - non-reproducible: code diff empty, test not fail rev_{bug}
- Discard non-isolated faults: manually review → <V1, V2>
  - Unrelated changes affect outcome of generated tests, coverage, mutation score



	Candidate revisions	Compilable revisions	Reproducible faults	Isolated faults
Total	1179	836	437	357

Methodology

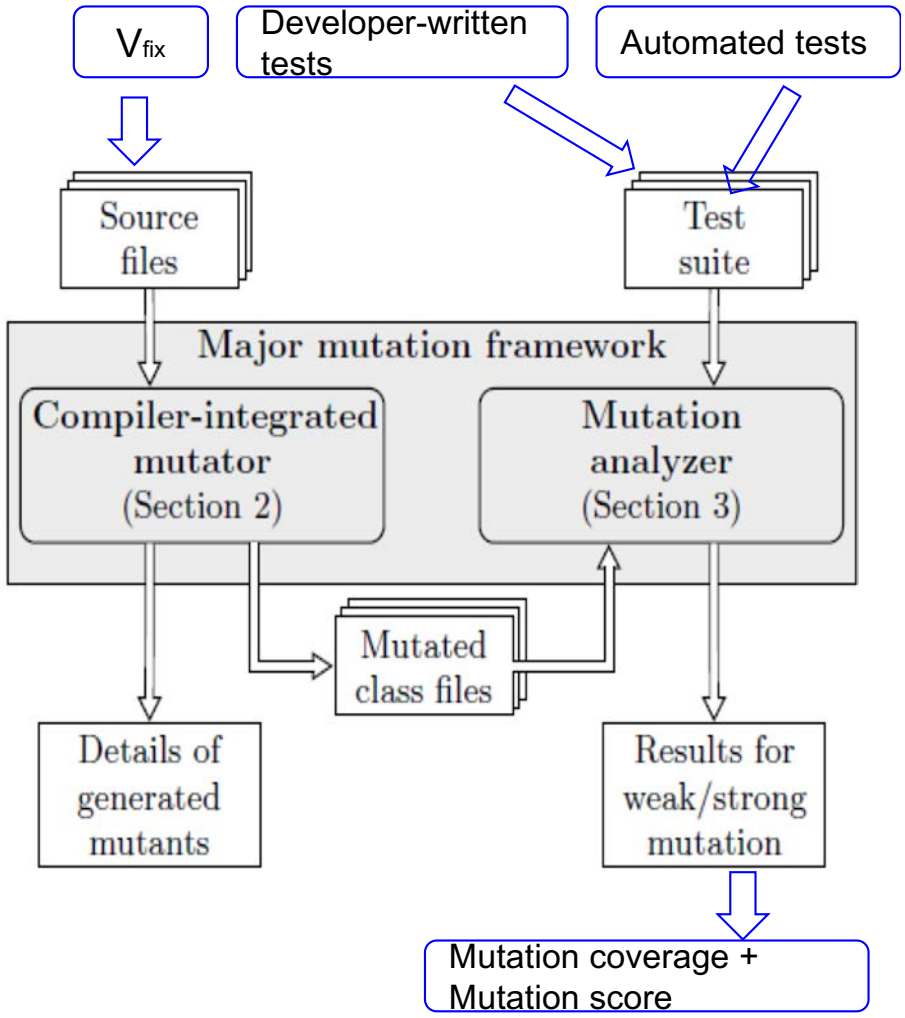


Are mutants a valid substitute for real faults?

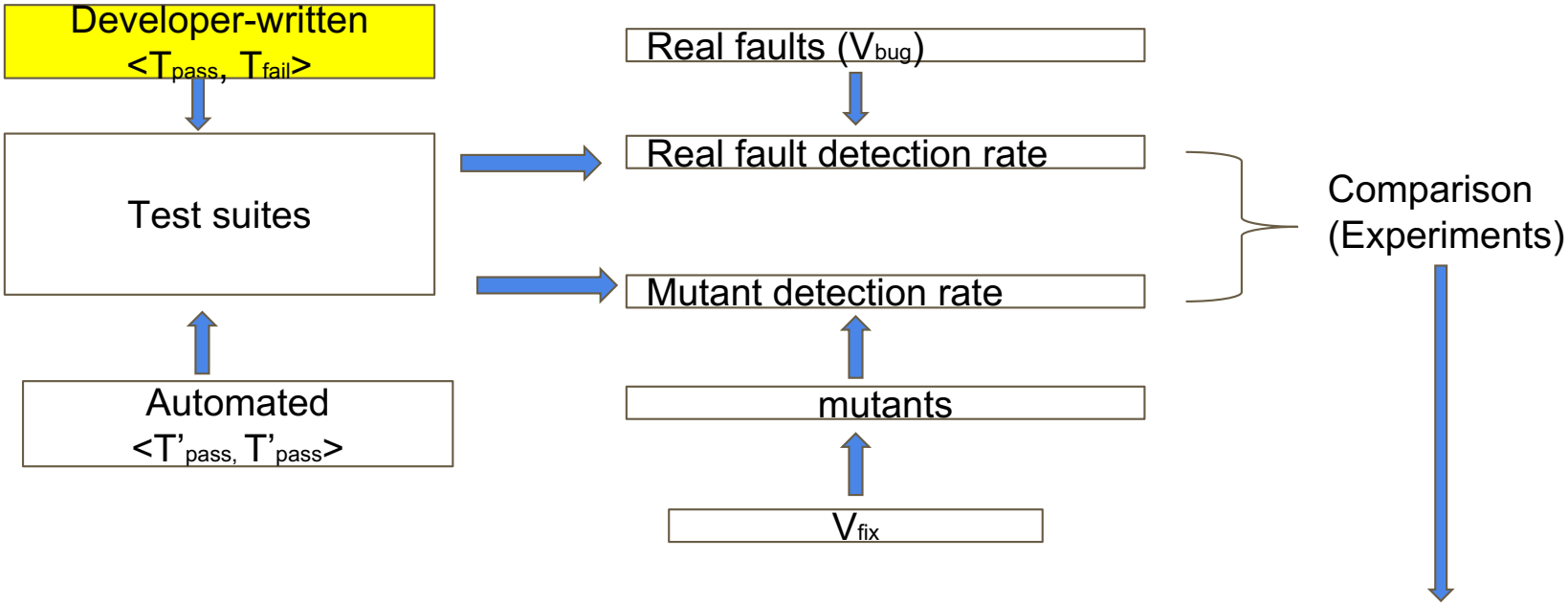
Mutant Generation

- Major mutation framework
  - Generate mutants
  - Mutation analyses: coverage, score

- Mutation Operators:
  - Replace constants
  - Replace operators
  - Modify branch conditions
  - Delete statements



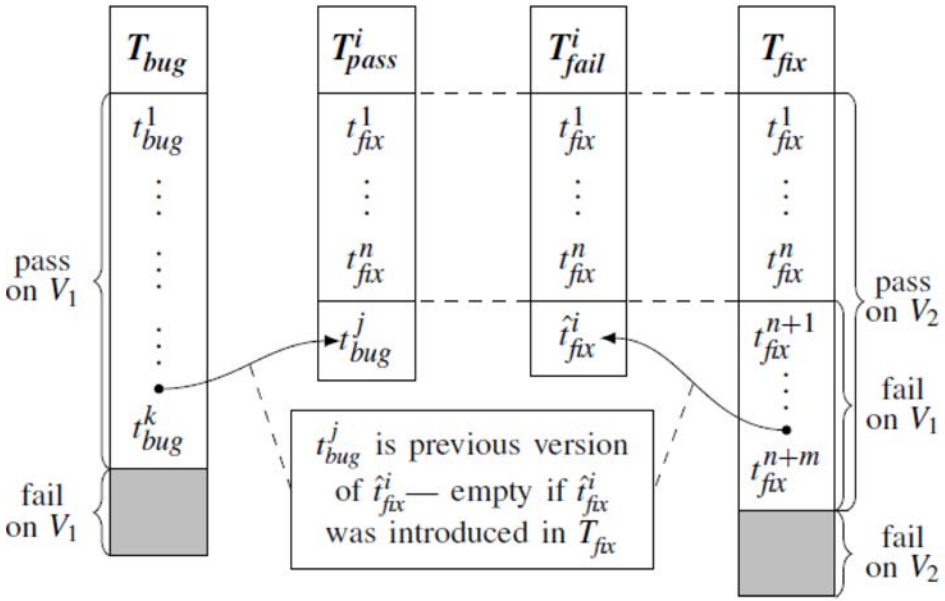
Methodology



Are mutants a valid substitute for real faults?

Obtaining Developer-written Test Suites

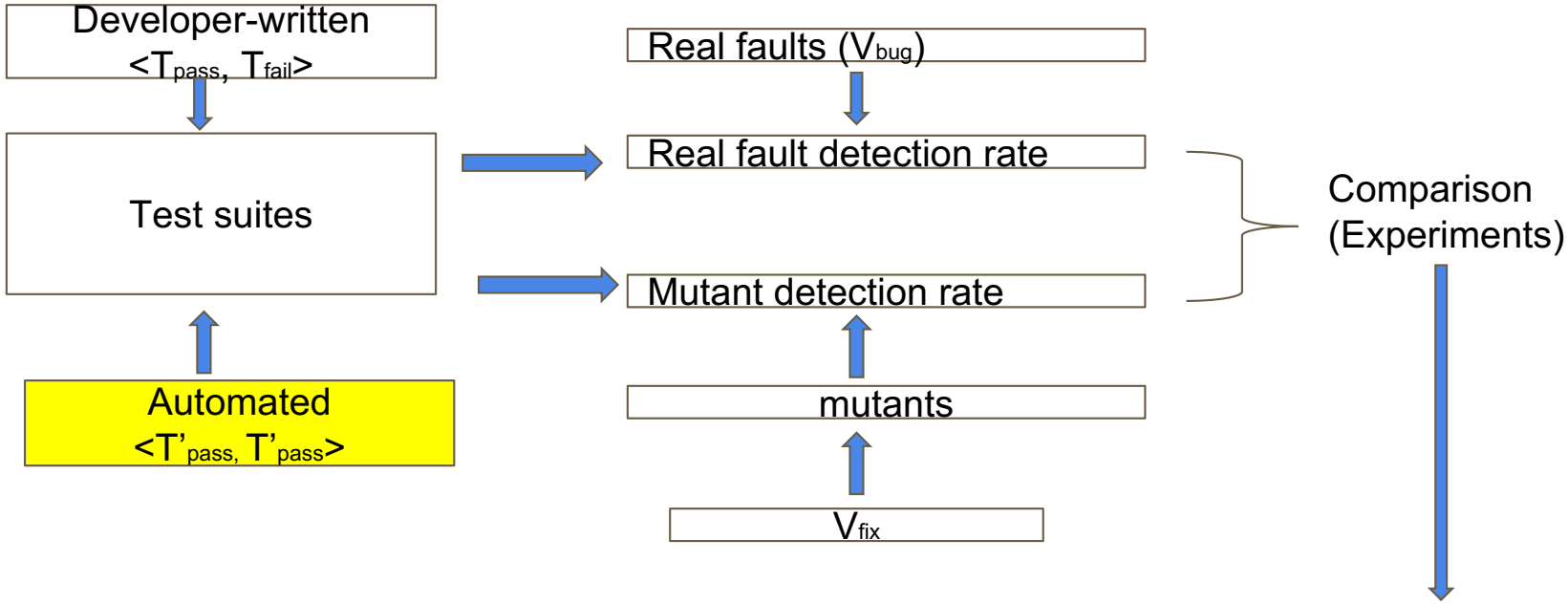
- 480 test suite pairs <T_{pass}, T_{fail}>



<T_{pass}, T_{fail}> only diff by one test for each real fault

Exclude feature tests, unrelated tests

Methodology

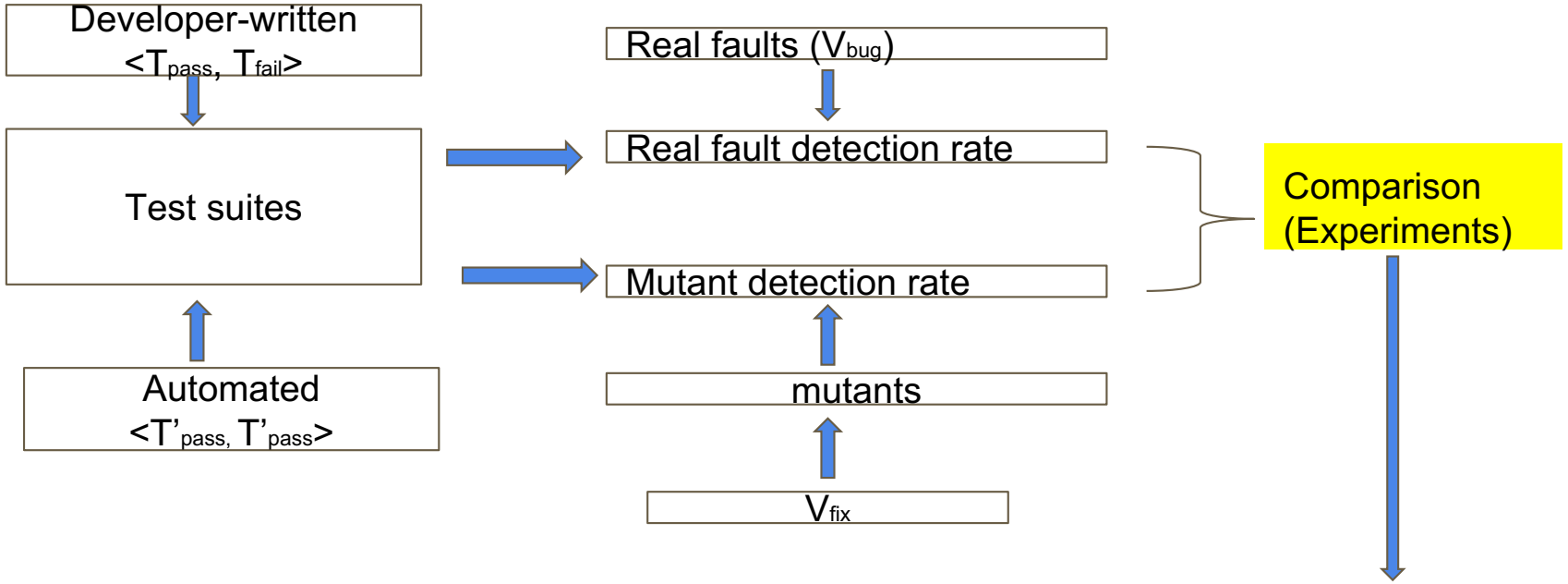


Are mutants a valid substitute for real faults?

Automatically Generating Test Suites

- Tools
  - EvoSuite:
    - Code coverage: Branch coverage, weak mutation testing, strong mutation testing
  - Randoop:
    - Null input, non-null input
  - JCrasher: Unexpected exception that crashes program
- 35,141 automatically-generated test suites
  - Exclude: tests cause compilation errors/fail on V2/non-deterministic tests

Methodology



Are mutants a valid substitute for real faults?

Experiments

RQ1:Are real faults coupled to mutants generated by commonly used mutation operators?

- T_{fail} has higher mutation score than T_{pass}?

RQ2:What types of real faults are not coupled to mutants?

- Not coupled: same mutants detected No. for <T_{pass}, T_{fail}>
- Check diff of mutants detected number between <T_{pass}, T_{fail}>

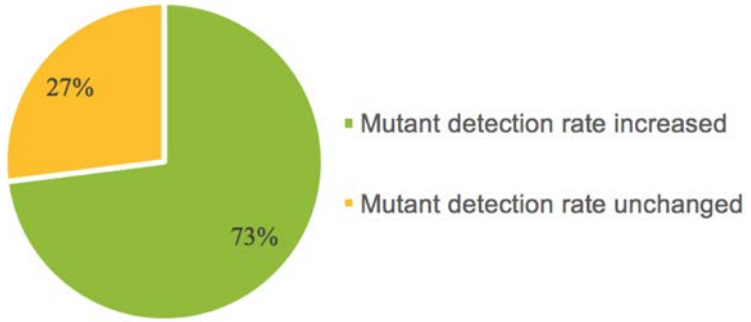
RQ3:Is mutant detection correlated with real fault detection?

- Correlation: mutation score, real fault detection
- <mutation score, real fault detection> stronger <statement coverage, real fault detection>
- Mutation score between <T'_{pass}, T'_{fail}> when coverage is fixed

Evaluation Results:

-- RQ1: Are real faults coupled to mutants generated by commonly used mutation operators?

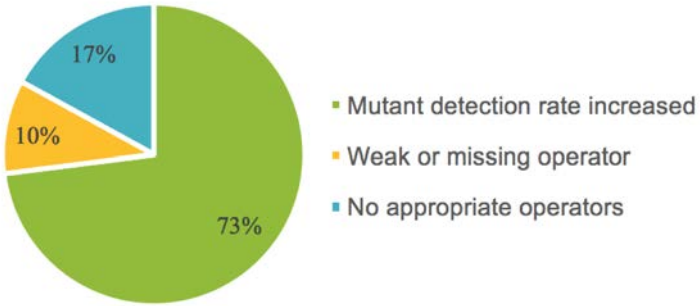
- Mutant detection rate increased for 73% of faults
- On average 2 mutants are coupled to a single real fault
- Conditional & relational operator replacement, statement deletion are more often coupled to real faults



Evaluation Results:

-- RQ2: What types of real faults are not coupled to mutants?

- Reasons:
  - Weak operator
    - E.g. Argument swapping
  - Missing mutation operator
    - E.g. Type conversions
  - No appropriate operator exists
    - E.g. Algorithm modification, code deletion




```
- Partial newPartial = new Partial(iChronology,
    newTypes, newValues);
+ Partial newPartial = new Partial(newTypes,
    newValues, iChronology);
```

(b) Time-88 fix

```
- FastMath.pow(2 * FastMath.PI, -dim / 2)
+ FastMath.pow(2 * FastMath.PI, -0.5 * dim)
```

(g) Math-929 fix

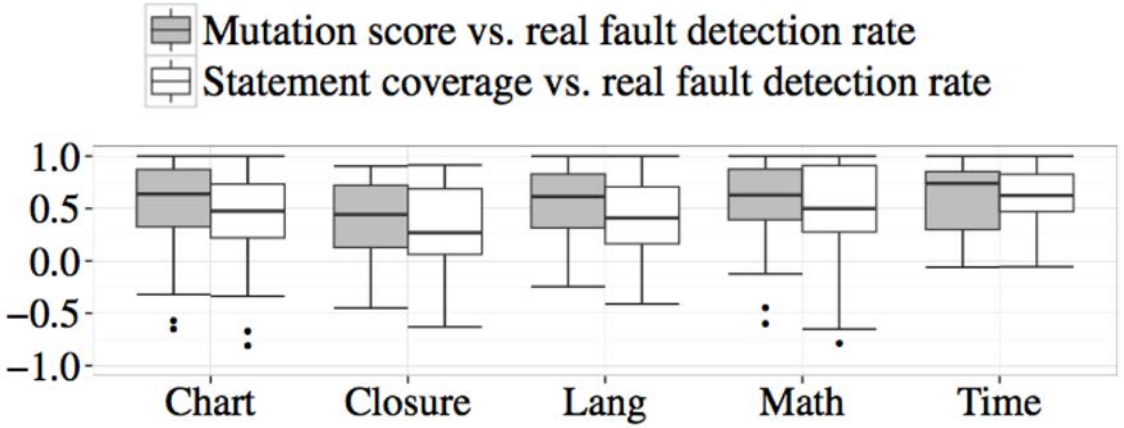
```
  if (childType.isDict()) {
    ...
- } else if (n.getJSType != null &&
-     parent.isAssign()) {
-   return;
  } ...
```

(a) Closure-810 fix

Evaluation Results:

-- RQ3: Is mutant detection correlated with real fault detection?

- Mutation detection is positively correlated with real fault detection
- Mutant detection is more sensitive to faults than statement coverage



(b) Rank-biserial correlation coefficients.

Contributions

1. 357 developer-fixed and manually-verified real faults with test suites
2. Largest study to date of whether mutants are a valid substitute for real faults
3. An investigation showing coupling effect for 73% between mutants and real faults
4. Suggestions for improving mutation analysis and identification of its inherent limitations
5. An analysis showing significantly stronger correlation between mutant detection and real fault detection

Discussion Questions

1. Do you consider 5 subject programs to be enough to conclude the experimental results?

Discussion Questions

2. Can the results be any different using programming languages other than Java?

Discussion Questions

3. Can bias in fault samples impact test results?

Discussion Questions

4. Should we use code coverage or mutation score for test suite minimization?

Discussion Questions

5. Can you think of more possible ways to reduce the unchanged mutant detection rate?

Reference

1. Rene Just, etc. Are Mutants a Valid Substitute for Real Faults in Software Testing?
2. Rene Just. The major mutation framework: efficient and scalable mutation analysis for Java
3. http://www.felienne.com/archives/374
4. https://people.cs.umass.edu/~brun/class/2015Fall/CS521.621/studentSlides/paper9.pdf

Thank You