



ReCodeAgent: A Multi-agent Workflow for Language-Agnostic Translation and Validation of Large-Scale Repositories

Ali Reza Ibrahimzada¹, Brandon Paulsen², Daniel Kroening²
Reyhaneh Jabbarvand¹

¹University of Illinois Urbana-Champaign ²Amazon

*41st IEEE/ACM International Conference on Automated Software Engineering (ASE)
October 12-16, 2026, Munich, Germany*

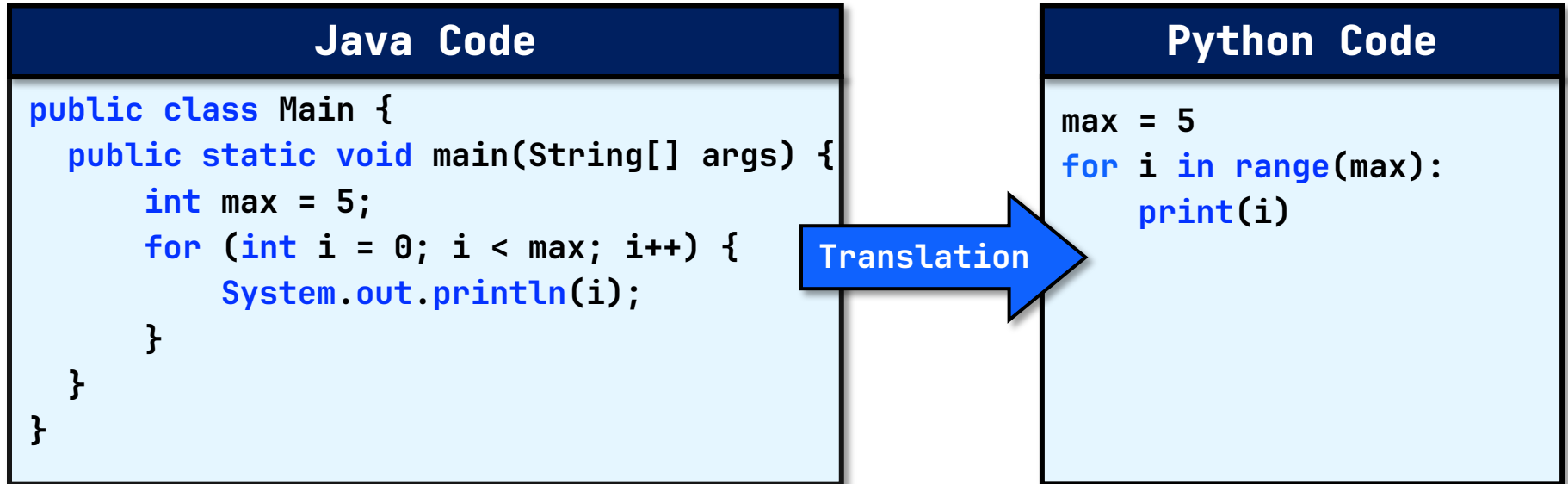


UNIVERSITY OF
ILLINOIS
URBANA-CHAMPAIGN



What is Code Translation?

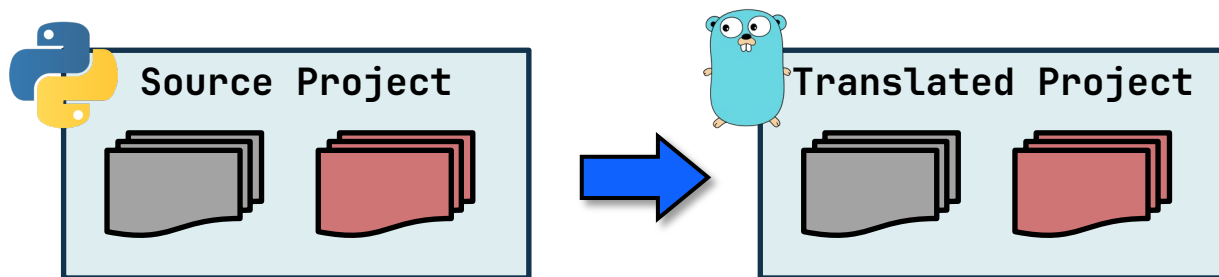
- Code translation converts source code from one programming language (PL) to another



Why Code Translation?

- Use cases of code translation:
 - Application modernization
 - Migrating legacy software to cloud-native applications
- Transpiler techniques, e.g., C2Rust, CxGO, Java2C#
 - **Scales** to large codebases but produces **non-idiomatic code** and **does not generalize** to many PLs
- LLMs (becoming prevalent in 2022)  **AI**   
 - **Generalizes well** and **generates more natural code** but **does not scale** to large codebases

End-to-End Agentic Code Translation



- Challenge 1: True PL-agnosticism
- Challenge 2: Context degradation
- Challenge 3: Dichotomy of test translation
- Challenge 4: Process-centric evaluation

Challenge 1: True PL-agnosticism

- Existing techniques use naive tools for exploration

```
class BasicParser:  
    def __init__(self):  
        self.var = 1  
  
    def get_var(self):  
        return self.var
```

Read

Scroll

ls

CodeQL

```
class BasicParserTest:  
  
    def test1(self):  
  
    def test2(self):
```

```
===== get_directory_tree =====  
|- python/  
    |- src/  
        |- main/  
            |- __init__.py  
            |- BasicParser.py  
        |- test/  
            |- BasicParserTest.py
```

```
===== get_file_structure =====  
"filepath": "/../..../"  
"language": "java/python/.."  
"skeleton": {  
    "imports": [],  
    "classes": ["BasicParser"],  
    "functions": ["__init__", "get_var"],  
    "globals": []  
}
```

Challenge 2: Context degradation

- Single context window can make it challenging for agents

```
class BasicParser:  
    def __init__(self):  
        self.var = 1
```

```
    def get_var(self):  
        return self.var
```

```
class BasicParserTest:
```

```
    def test1(self):
```

```
    def test2(self):
```

10K LoC

Pre-translation Steps

1. Analyze source project
2. Analyze library usage
3. Extract fragments
4. Create target project skeleton
5. Create implementation plan

Challenge 3: Dichotomy of Test Translation

Source
PL
Test

```
@Test
public void testClassPattern() throws Exception {
    final Options options = PatternOptionBuilder.parsePattern("c+d+");
    final C...
    final C...

    assertEquals...
    assertEquals...
}
```

Translator

- Translate code
- Translate test
- Repair, if necessary

```
", "System.DateTime"}));
("c"));
```

One
Agent

```
...
assert
...
```

Validator

- Check translated tests
- Independently execute tests
- Provide independent quality assurance report to Translator

Two
Agents

```
# In Py
c_value
# The t
from datetime import datetime as dt
self.assertEqual(dt, c_value)
...
```

lent to Calendar in Java)

Challenge 4: Process-centric Evaluation

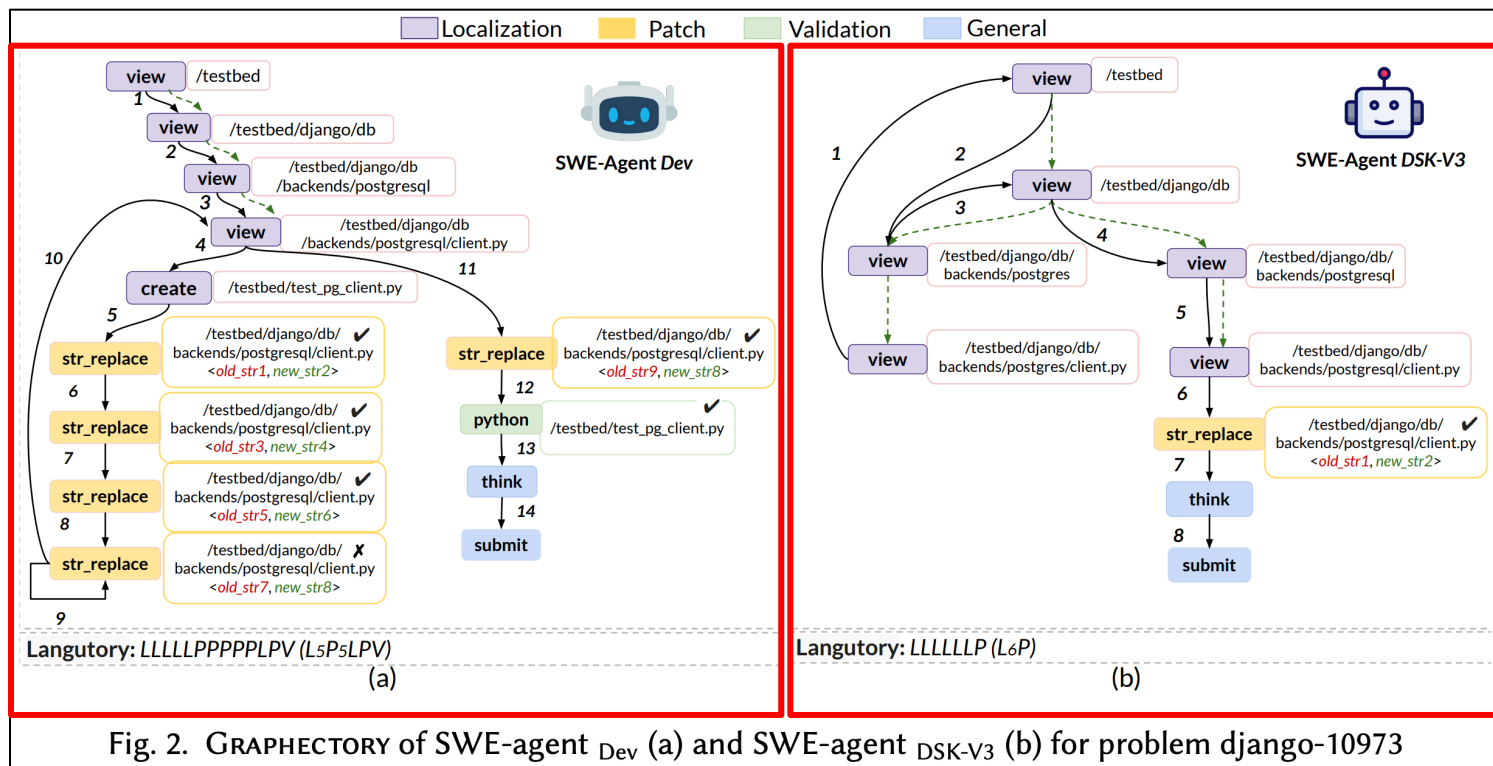
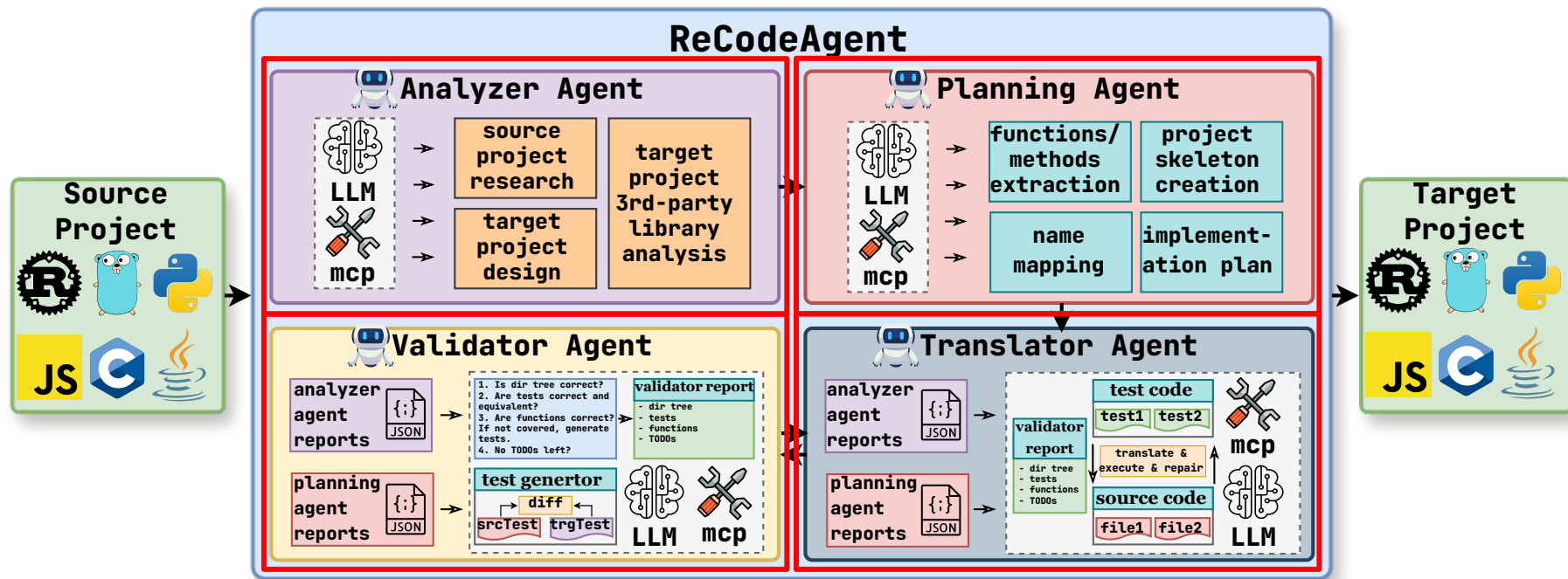


Fig. 2. GRAPHECTORY of SWE-agent_{Dev} (a) and SWE-agent_{DSK-V3} (b) for problem django-10973

*Figure from Graphectory: <https://doi.org/10.1145/3798271>

ReCodeAgent



MCP Tools

Server	API
Language Server Protocol	<code>definition(symbol_name: str)</code>
	<code>diagnostics(file_path: str)</code>
	<code>edit_file(edits: List, file_path: str)</code>
	<code>hover(file_path: str, line: int, column: int)</code>
	<code>references(symbol_name: str)</code>
	<code>rename_symbol(file_path: str, line: int, column: int, new_name: str)</code>
Project Analyzer	<code>get_directory_tree(path: str, print_dirs_only: bool)</code>
	<code>get_file_structure(language: str, file_path: str)</code>

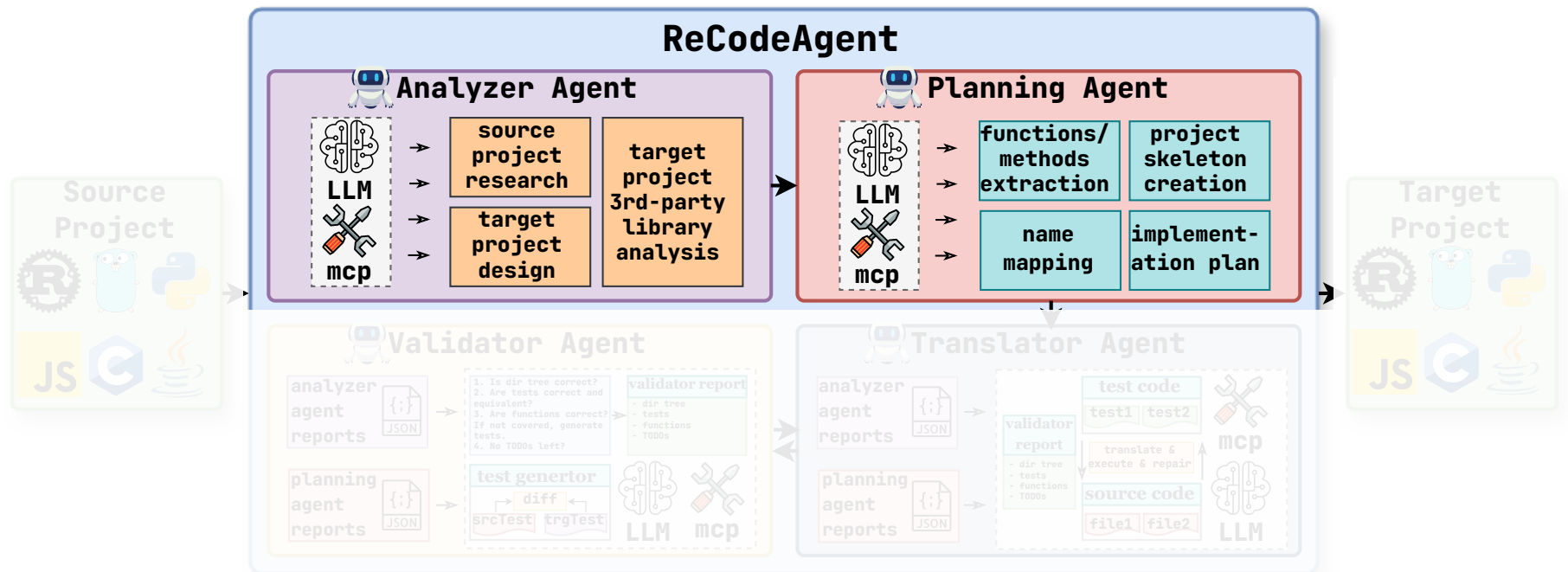
Hover Tool

```
class OptionComparator:
    __serialVersionUID: int = 530546787396684014
    def compare(self, opt1: Option, opt2: Option) -> int:
        return (
            opt1.getKey().casefold() < opt2.getKey().casefold()
        ) - (
            opt1.getKey().casefold() > opt2.getKey().casefold()
        )
```

(method) def casefold() -> str
Return a version of the string suitable for caseless comparisons.

```
===== CALL =====
hover(MCP)(file_path: "OptionComp.py", line: 7, column: 27)
===== RESPONSE =====
```python
(method) def casefold() -> str
```
Return a version of the string suitable for caseless comparisons.
```

ReCodeAgent



Analyzer Agent

===== Source Project Research =====

- ## Overview
- ## Directory Structure
- ## Structs and Interfaces
- ## Data Models
- ## Error Handling
- ## Dependencies

==== Third-Party Library Analysis ====

- ## Library A
 - Overview
 - Usages
 - Example
 - Recommendations in Target PL
- ## Library B

===== Target Project Design =====

- ## Overview
- ## Translation Requirements
- ## Source Files to Translate
- ## Module Structure
- ## Error Handling
- ## Third-Party Libraries

Planning Agent

===== Fragment Extraction =====

```
## checkdigit.go
  checkdigit.go:isNumber
  checkdigit.go:NewLuhn
  checkdigit.go:NewDamm
  checkdigit.go:NewUPC
## damm.go
```

===== Name Mapping =====

```
## functions:
  go:isNumber: rs.isNumber
  go.NewLuhn: rs.NewLuhn
  go.NewDamm: rs.NewDamm
  go.NewUPC: rs.NewUPC
## variables:
```

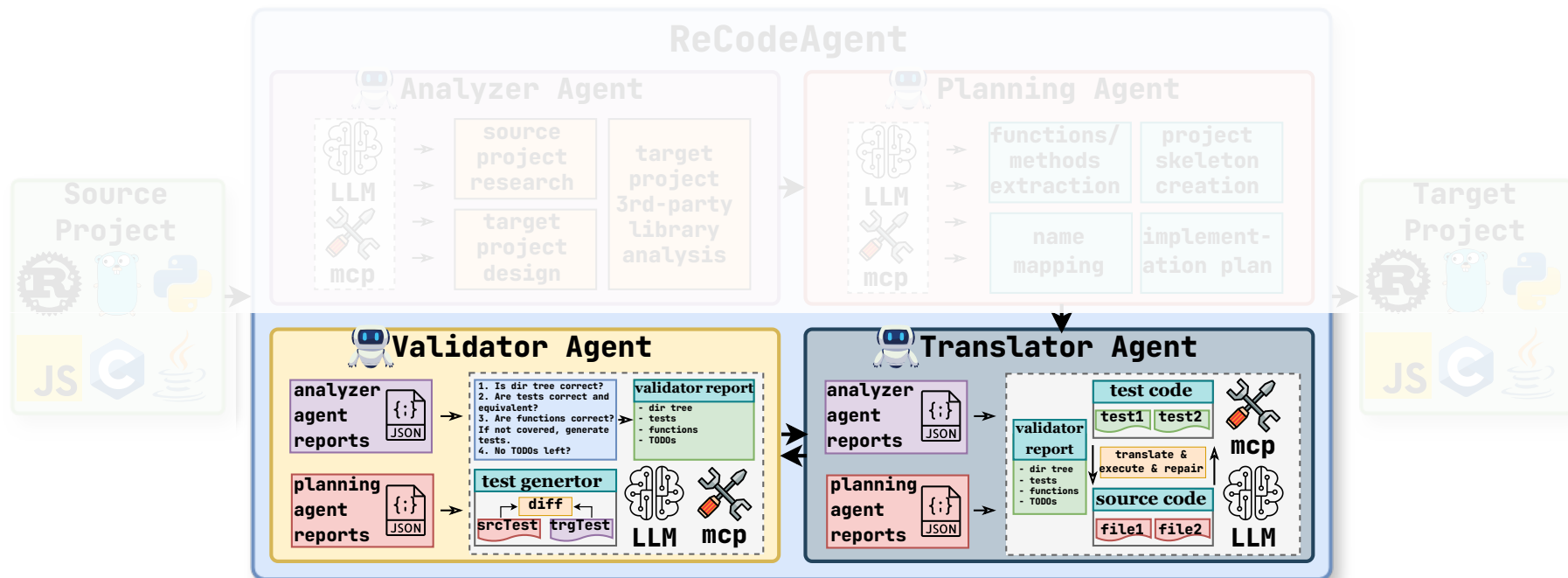
===== Skeleton Generation =====

```
fn isNumber(n: char) -> bool {}
fn NewLuhn() -> impl Pvd {}
fn NewDamm() -> impl Pvd {}
fn NewUPC() -> impl Pvd {}
```

===== Implementation Plan =====

```
## Overview
## Part A:
  A1: Translate checkdigit.go
## Part B:
  B1: Translate cd_test.go
```

ReCodeAgent



Effectiveness in Code Translation

- $\langle$ Tool, ReCodeAgent $\rangle$
- **86.5%** test pass rate, **60.8%** more than other tools

| Tools | Lines of Code (LoC) | Compilation Success | # Validated Developer Tests | Validated Developer Tests | | |
|------------|---------------------|------------------------------|-----------------------------|------------------------------|-----------------------------|----------------------------|
| | | | | # Tests Executed | # Tests Passed | # Tests Failed |
| Oxidizer | 4167 | $\langle 100, 100 \rangle$ | 229 | $\langle 229, 229 \rangle$ | $\langle 154, 229 \rangle$ | $\langle 75, 0 \rangle$ |
| AlphaTrans | 116085 | $\langle 100, 100 \rangle$ | 1181 | $\langle 611, 1181 \rangle$ | $\langle 188, 1078 \rangle$ | $\langle 423, 103 \rangle$ |
| Skel | 4077 | $\langle 100, 100 \rangle$ | 74 | $\langle 74, 74 \rangle$ | $\langle 69, 74 \rangle$ | $\langle 5, 0 \rangle$ |
| CRUST | 108728 | $\langle 41, 89 \rangle$ | 623 | $\langle 154, 486 \rangle$ | $\langle 131, 441 \rangle$ | $\langle 23, 45 \rangle$ |
| Total | 233057 | $\langle 96.9, 99.4 \rangle$ | 2107 | $\langle 1068, 1970 \rangle$ | $\langle 542, 1822 \rangle$ | $\langle 526, 148 \rangle$ |

Effectiveness in Test Translation

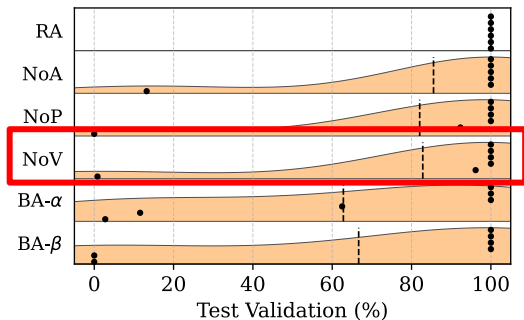
- Comparison between source and translated tests

| Tools | # tests | # Tests Translated / Not Translated | # Tests Matching / Non-Matching # Assertions | # Total / Matching assertEquals Output | Avg. Cosine Similarity | Avg. LoC | Avg. # Method Invocations |
|------------|---------|-------------------------------------|--|--|------------------------|----------------|---------------------------|
| Oxidizer | 229 | 229/0 | 229/0 | 314/314 | 0.92 | <20.21, 20.51> | <10.65, 8.84> |
| AlphaTrans | 1181 | 1181/0 | 1170/11 | 1070/1070 | 0.89 | <12.17, 11.29> | <12.54, 11.94> |
| Skel | 74 | 74/0 | 74/0 | 507/507 | 0.92 | <27.43, 27.54> | <22.64, 21.69> |
| Total | 1484 | 1484/0 | 1473/11 | 1891/1891 | 0.91 | <21.63, 21.58> | <16.40, 15.24> |

Ablation Study

RA: ReCodeAgent, **No (A,P,V):** No (Analyzer, Planning, Validator)
BA-(α , β): Base Agent-(Condense,Concat)

Oxidizer



| | | | | | |
|--------------|-------|-------|------|--------|------|
| RA | 88.3 | 107.6 | 8.8 | 316.6 | 11.9 |
| NoA | 170.3 | 210.8 | 23.4 | 696.3 | 21.0 |
| NoP | 143.6 | 179.9 | 22.2 | 181.6 | 18.3 |
| NoV | 255.3 | 334.3 | 28.0 | 1418.2 | 33.3 |
| BA- α | 95.2 | 95.0 | 0.0 | 0.8 | 2.4 |
| BA- β | 173.7 | 185.3 | 17.5 | 129.0 | 38.5 |
| | NC | TEC | SEC | LC | ALL |

NC : Node Count

(number of distinct actions)

TEC: Temporal Edge Count

(total number of temporal transitions)

SEC: Structural Edge Count

(total number of structural connections)

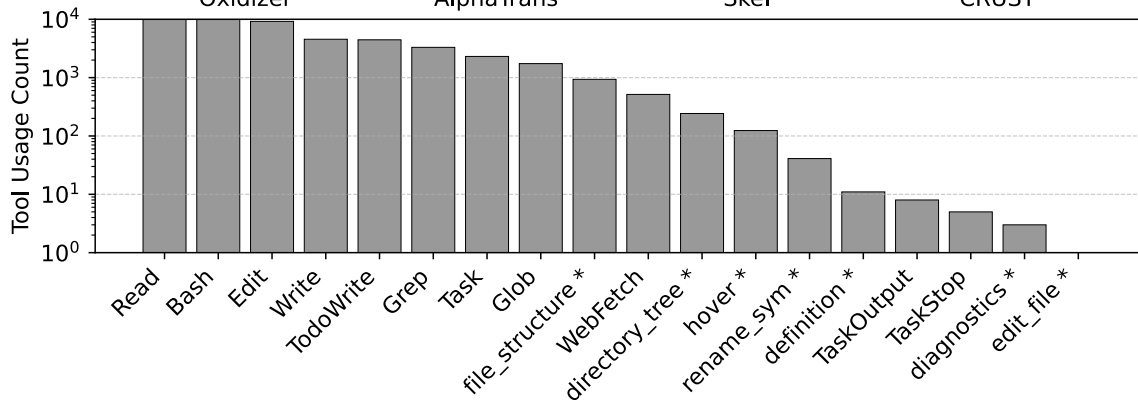
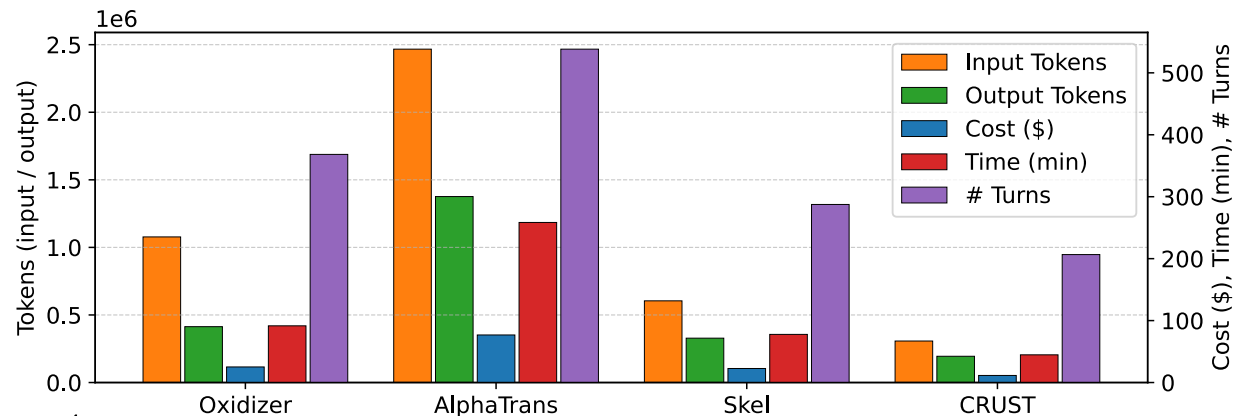
LC : Loop Count

(Number of times the agent decides to repeat a previously executed action)

ALL: Average Loop Length

(The average number of actions involved before an agent decides to repeat a previously executed action)

Cost and Tool Usage Analysis



\$15.3
57 mins

Concluding Remarks

- ReCodeAgent: A multi-agent workflow for end-to-end translation and validation of large-scale repositories
- Evaluated on **118** projects, **6** PLs, and **4** language pairs
- Achieves **86.5%** test pass rate – **60.8%** improvement over other tools
- Achieves **99.3%** correctness in test translation
- Translating projects cost **\$15.3** and **57 minutes** on average



github.com/Intelligent-CAT-Lab/ReCodeAgent

