ICLR
International Conference On Learning Representations

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

Tencent AI Lab

RepoGraph:

Enhancing AI Software Engineering with Repository-level Code Graph

Paper: *https://arxiv.org/abs/2410.14684*

Code: *https://github.com/ozyyshr/RepoGraph*

Siru Ouyang 

Wenhao Yu 

Kaixin Ma 

Zilin Xiao 

Zhihan Zhang 

Mengzhao Jia 

Jiawei Han 

Hongming Zhang 

Dong Yu 

AI Software Engineering

Real-world issue in a repository opened by a user.

Bishop88 functions sometimes return incorrect results or errors when using `d2mutau` and `NsVbi` parameters #2013

New issue

Closed **kandersolar** opened this issue on Apr 16 · 7 comments · Fixed by #2032

kandersolar commented on Apr 16 — Member

Describe the bug
The bishop88 functions can sometimes fail, or return incorrect results, when using the `d2mutau` and `NsVbi` parameters.

To Reproduce

Set-up:

```
import pvlib
import numpy as np
import matplotlib.pyplot as plt

sdm_params = {
    'alpha_sc': 0.001, 'gamma_ref': 1.409, 'mu_gamma': 0.001, 'I_o_ref': 3.96e-11,
    'R_sh_ref': 4900, 'R_sh_0': 8800, 'R_sh_exp': 5.5,
    'R_s': 6.76, 'cells_in_series': 264, 'EgRef': 1.121, 'I_L_ref': 2.278883384981941
}
d2mutau = 1.5
NsVbi = 237.6

G = np.linspace(0, 1200)
T = 25
sde_params = pvlib.pvsystem.calcparams_pvsyst(G, T, **sdm_params)
```

`method='brentq'` errors with `ValueError: f(a) and f(b) must have different signs`.

▼ Click to show traceback

```
Traceback (most recent call last):

  Cell In[16], line 17
    mpp = pvlib.singlediode.bishop88_mpp(*sde_params, d2mutau=d2mutau, NsVbi=NsVbi, method='brentq')

  File ~\software\miniconda3\envs\dev\lib\site-packages\pvlib\singlediode.py:577 in bishop88_mpp
    vd = vec_fun(voc_est, *args)

  File ~\software\miniconda3\envs\dev\lib\site-packages\numpy\lib\function_base.py:2372 in __call__
    return self._call_as_normal(*args, **kwargs)

  File ~\software\miniconda3\envs\dev\lib\site-packages\numpy\lib\function_base.py:2365 in _call_as_normal
    return self._vectorize_call(func=func, args=vargs)

  File ~\software\miniconda3\envs\dev\lib\site-packages\numpy\lib\function_base.py:2450 in _vectorize_call
    ufunc, otypes = self._get_ufunc_and_otypes(func=func, args=args)

  File ~\software\miniconda3\envs\dev\lib\site-packages\numpy\lib\function_base.py:2410 in _get_ufunc_and_otype
    outputs = func(*inputs)

  File ~\software\miniconda3\envs\dev\lib\site-packages\pvlib\singlediode.py:572 in <lambda>
    vbr_exp: brentq(fmpp, 0.0, voc,

  File ~\software\miniconda3\envs\dev\lib\site-packages\scipy\optimize\_zeros_py.py:809 in brentq
    r = _zeros._brentq(f, a, b, xtol, rtol, maxiter, args, full_output, disp)

ValueError: f(a) and f(b) must have different signs
```

Assignees
No one assigned

Labels
bug

Projects
None yet

Milestone
v0.10.5

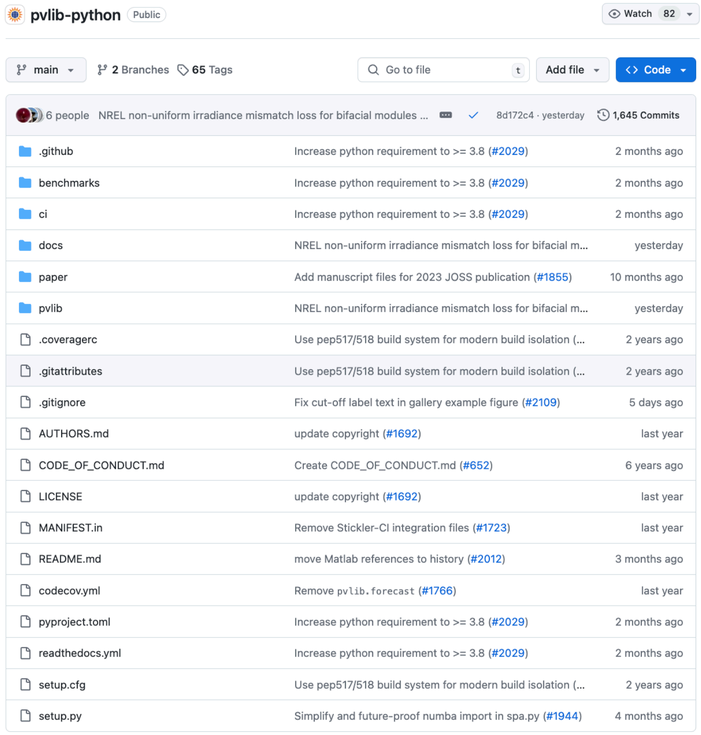
Development
Successfully merging a pull request may close this issue.
Modify initial Voc for bishop88 functions
cwhanse/pvlib-python

Notifications Customize
Subscribe
You're not receiving notifications from this thread.

3 participants

https://github.com/pvlib/pvlib-python/issues/2013

AI Software Engineering



pvlib-python Public

Watch 82

main · 2 Branches · 65 Tags · Go to file t · Add file · Code

6 people · NREL non-uniform irradiance mismatch loss for bifacial modules ... · 8d172c4 · yesterday · 1,645 Commits

.github	Increase python requirement to >= 3.8 (#2029)	2 months ago
benchmarks	Increase python requirement to >= 3.8 (#2029)	2 months ago
ci	Increase python requirement to >= 3.8 (#2029)	2 months ago
docs	NREL non-uniform irradiance mismatch loss for bifacial m...	yesterday
paper	Add manuscript files for 2023 JOSS publication (#1855)	10 months ago
pvlib	NREL non-uniform irradiance mismatch loss for bifacial m...	yesterday
.coveragerc	Use pep517/518 build system for modern build isolation (...	2 years ago
.gitattributes	Use pep517/518 build system for modern build isolation (...	2 years ago
.gitignore	Fix cut-off label text in gallery example figure (#2109)	5 days ago
AUTHORS.md	update copyright (#1692)	last year
CODE_OF_CONDUCT.md	Create CODE_OF_CONDUCT.md (#652)	6 years ago
LICENSE	update copyright (#1692)	last year
MANIFEST.in	Remove Stickler-CI integration files (#1723)	last year
README.md	move Matlab references to history (#2012)	3 months ago
codecov.yml	Remove `pvlib.forecast` (#1766)	last year
pyproject.toml	Increase python requirement to >= 3.8 (#2029)	2 months ago
readthedocs.yml	Increase python requirement to >= 3.8 (#2029)	2 months ago
setup.cfg	Use pep517/518 build system for modern build isolation (...	2 years ago
setup.py	Simplify and future-proof numba import in spa.py (#1944)	4 months ago

Bishop88 functions sometimes return incorrect results or errors when using `d2mutau` and `NsVbi` parameters #2013

New issue

Closed · **kandersolar** opened this issue on Apr 16 · 7 comments · Fixed by #2032

kandersolar commented on Apr 16 · Member

Describe the bug

The bishop88 functions can sometimes fail, or return incorrect results, when using the `d2mutau` and `NsVbi` parameters.

To Reproduce

Set-up:

```
import pvlib
import numpy as np
import matplotlib.pyplot as plt

sdm_params = {
    'alpha_sc': 0.001, 'gamma_ref': 1.409, 'mu_gamma': 0.001, 'I_o_ref': 3.96e-11,
    'R_sh_ref': 4900, 'R_sh_0': 8800, 'R_sh_exp': 5.5,
    'R_s': 6.76, 'cells_in_series': 264, 'EgRef': 1.121, 'I_L_ref': 2.278883384981941
}
d2mutau = 1.5
NsVbi = 237.6

G = np.linspace(0, 1200)
T = 25
sde_params = pvlib.pvsystem.calcparams_pvsyst(G, T, **sdm_params)
```

`method='brentq'` errors with `ValueError: f(a) and f(b) must have different signs`.

▼ Click to show traceback

```
Traceback (most recent call last):

  Cell In[16], line 17
    mpp = pvlib.singlediode.bishop88_mpp(*sde_params, d2mutau=d2mutau, NsVbi=NsVbi, method='brentq')

  File ~\software\miniconda3\envs\dev\lib\site-packages\pvlib\singlediode.py:577 in bishop88_mpp
    vd = vec_fun(voc_est, *args)

  File ~\software\miniconda3\envs\dev\lib\site-packages\numpy\lib\function_base.py:2372 in __call__
    return self._call_as_normal(*args, **kwargs)

  File ~\software\miniconda3\envs\dev\lib\site-packages\numpy\lib\function_base.py:2365 in _call_as_normal
    return self._vectorize_call(func=func, args=vargs)

  File ~\software\miniconda3\envs\dev\lib\site-packages\numpy\lib\function_base.py:2450 in _vectorize_call
    ufunc, otypes = self._get_ufunc_and_otypes(func=func, args=args)

  File ~\software\miniconda3\envs\dev\lib\site-packages\numpy\lib\function_base.py:2410 in _get_ufunc_and_otype
    outputs = func(*inputs)

  File ~\software\miniconda3\envs\dev\lib\site-packages\pvlib\singlediode.py:572 in <lambda>
    vbr_exp: brentq(fmpp, 0.0, voc,

  File ~\software\miniconda3\envs\dev\lib\site-packages\scipy\optimize\_zeros_py.py:809 in brentq
    r = _zeros._brentq(f, a, b, xtol, rtol, maxiter, args, full_output, disp)

ValueError: f(a) and f(b) must have different signs
```

Assignees
No one assigned

Labels
bug

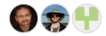
Projects
None yet

Milestone
v0.10.5

Development
Successfully merging a pull request may close this issue.
Modify initial Voc for bishop88 functions
cwhanse/pvlib-python

Notifications · Customize
Subscribe
You're not receiving notifications from this thread.

3 participants



https://github.com/pvlib/pvlib-python/issues/2013

AI Software Engineering

Copilot -> Autopilot

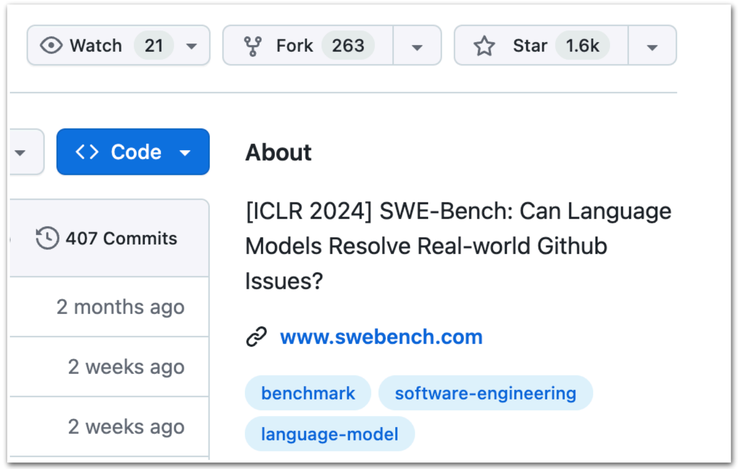
Systematic problem-solving

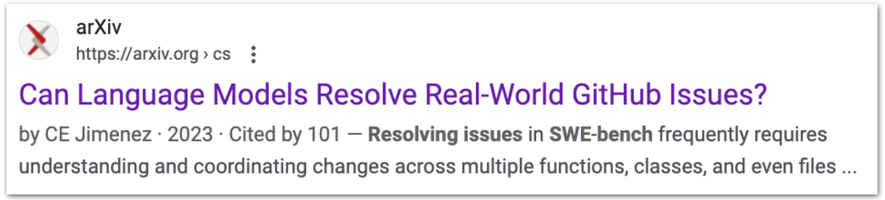
Autonomous software engineering capability

Repository-level understanding

https://github.com/pvlib/pvlib-python/issues/2013

Introducing SWE-bench





August 13, 2024

Introducing SWE-bench Verified

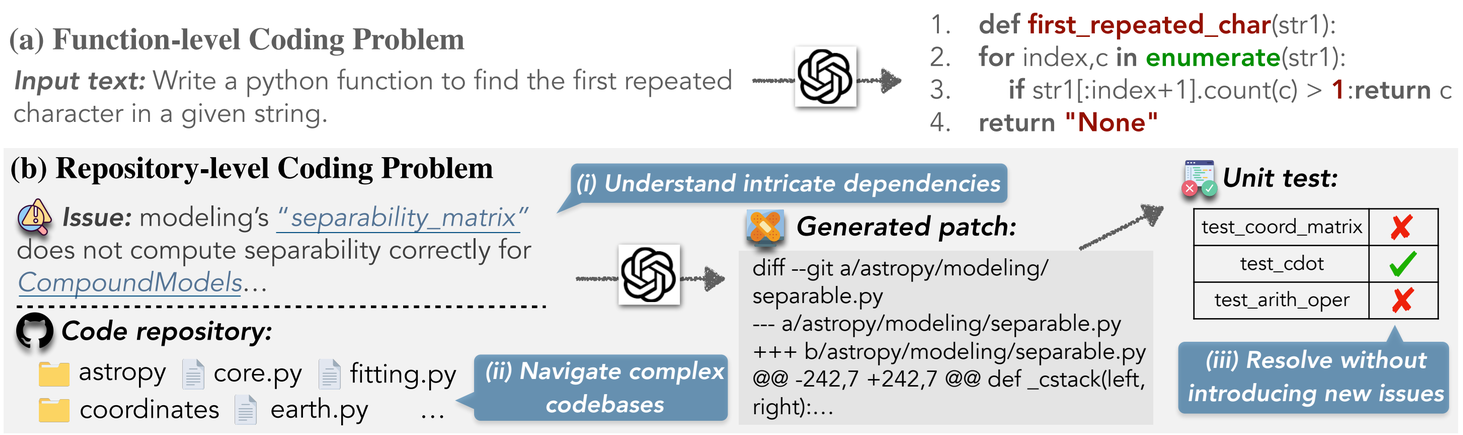
We're releasing a human-validated subset of SWE-bench that more reliably evaluates AI models' ability to solve real-world software issues.

Download SWE-bench Verified ↗

Leaderboard

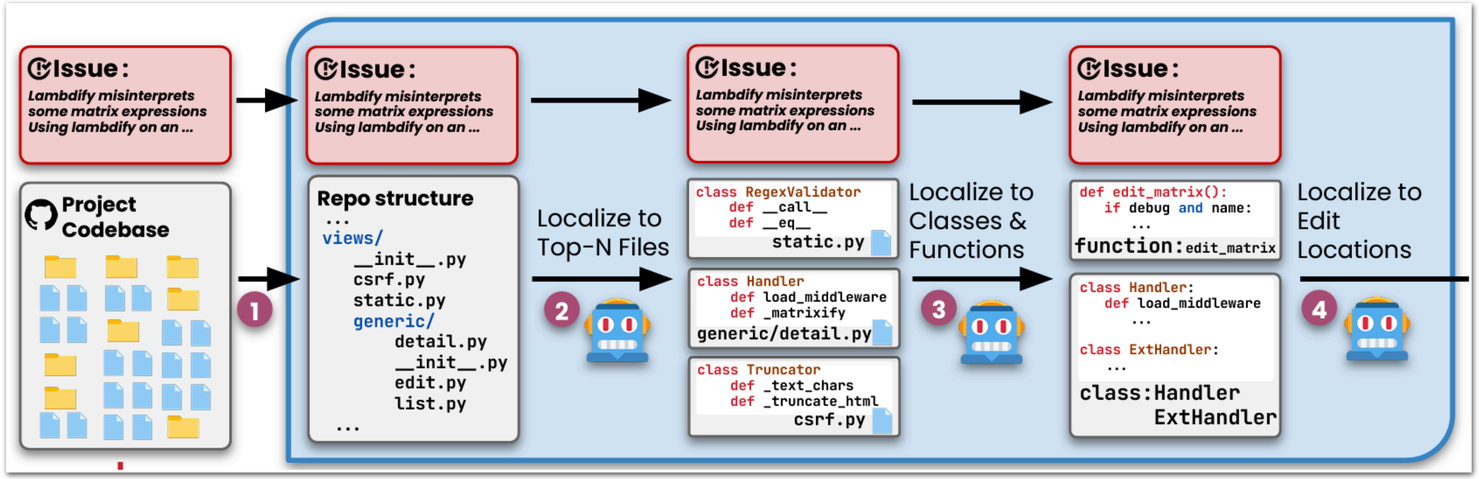
Lite | Verified | Full

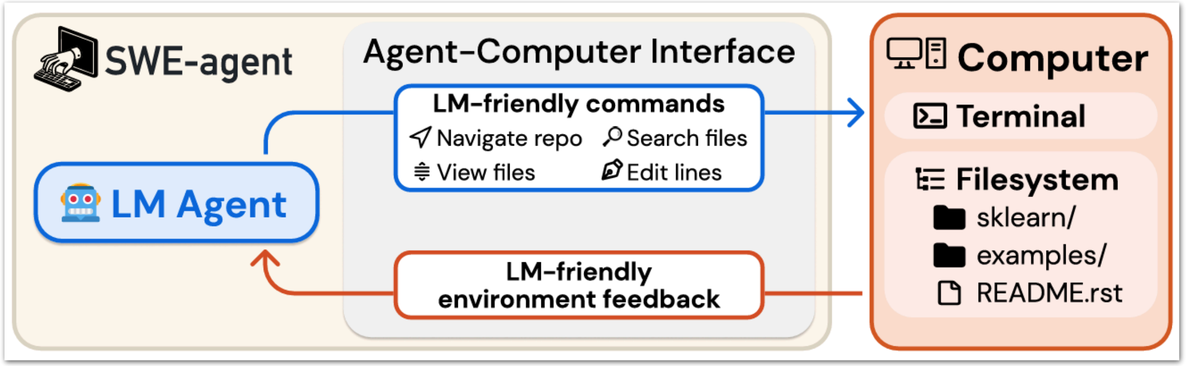
Model	% Resolved	Date	Logs	Trajs	Site
🥇 CodeStory Aide + Mixed Models	43.00	2024-07-02	🔗	-	🔗
🥈 AbanteAI MentatBot + GPT 4o (2024-05-13)	38.00	2024-06-27	🔗	-	🔗
🥉 Gru(2024-08-11)	35.67	2024-08-11	🔗	🔗	🔗
Bytedance MarsCode Agent + GPT 4o (2024-05-13)	34.00	2024-07-23	🔗	-	🔗
Alibaba Lingma Agent	33.00	2024-06-22	🔗	-	🔗
Factory Code Droid	31.33	2024-06-17	🔗	-	🔗
AutoCodeRover (v20240620) + GPT 4o (2024-05-13)	30.67	2024-06-21	🔗	-	🔗
Amazon Q Developer Agent (v20240719-dev)	29.67	2024-07-21	🔗	-	🔗
🤠 Agentless + RepoGraph + GPT-4o	29.67	2024-08-08	🔗	🔗	🔗
CodeR + GPT 4 (1106)	28.33	2024-06-04	🔗	-	🔗
MASAI + GPT 4o (2024-05-13)	28.00	2024-06-12	🔗	-	🔗
SIMA + GPT 4o (2024-05-13)	27.67	2024-07-06	🔗	🔗	🔗
🤠 Agentless + GPT 4o (2024-05-13)	27.33	2024-06-30	🔗	-	🔗
🤠 ✅ Moatless Tools + Claude 3.5 Sonnet	26.67	2024-06-23	🔗	🔗	🔗
🤠 ✅ OpenDevin + CodeAct v1.8	26.67	2024-07-25	🔗	🔗	🔗
IBM Research Agent-101	26.67	2024-06-12	🔗	-	🔗
🤠 Aider + GPT 4o & Claude 3 Opus	26.33	2024-05-23	🔗	-	🔗
🤠 ✅ Moatless Tools + GPT 4o (2024-05-13)	24.67	2024-06-17	🔗	🔗	🔗
OpenCSG StarShip CodeGenAgent + GPT 4 (0613)	23.67	2024-05-24	🔗	-	🔗


(a) Function-level Coding Problem
Input text: Write a python function to find the first repeated character in a given string.
1. def first_repeated_char(str1):
2. for index,c in enumerate(str1):
3. if str1[:index+1].count(c) > 1:return c
4. return "None"
(b) Repository-level Coding Problem
Issue: modeling's "separability_matrix" does not compute separability correctly for CompoundModels…
Code repository:
astropy
core.py
fitting.py
coordinates
earth.py
…
(i) Understand intricate dependencies
(ii) Navigate complex codebases
Generated patch:
diff --git a/astropy/modeling/separable.py
--- a/astropy/modeling/separable.py
+++ b/astropy/modeling/separable.py
@@ -242,7 +242,7 @@ def _cstack(left, right):…
Unit test:
test_coord_matrix
test_cdot
test_arith_oper
(iii) Resolve without introducing new issues

Related works

- Localize-then-edit
  - RAG
  - AutoCodeRover
  - Agentless
- Agent
  - SWE-agent
  - OpenDevin
  - Aider





Related works

- Localize-then-edit 📍
  - RAG
  - AutoCodeRover
  - Agentless
- Agent 🤖
  - SWE-agent
  - OpenDeven
  - Aider

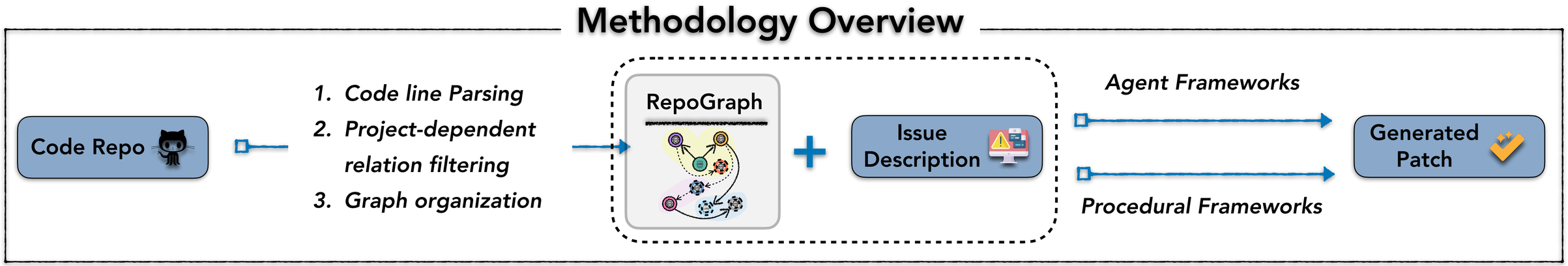
	Features
SWE-agent	open, goto, scroll_down, scroll_up, search_dir, search_file, find_file, edit, create, submit
OpenDeven	Flexible commands
Aider	<<<<<<< SEARCH from flask import Flask ======= import math from flask import Flask >>>>>> REPLACE {fence[1]} {fence[0]} mathweb/flask/app.py
RAG	File-level indexing
AutoCodeRover	search_class, search_method_in_file, search_method_in_class, search_method, search_code, search_code_in_file
Agentless	Repo-level structure localization

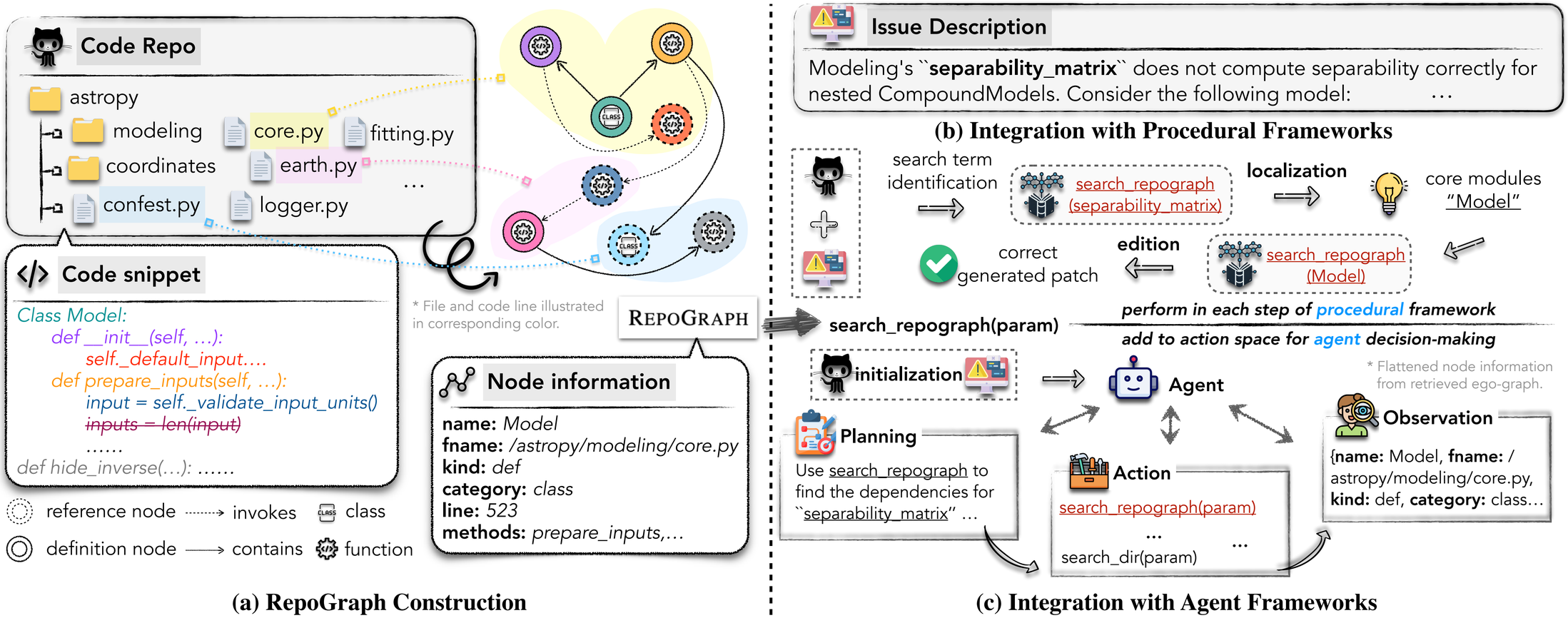
Related works

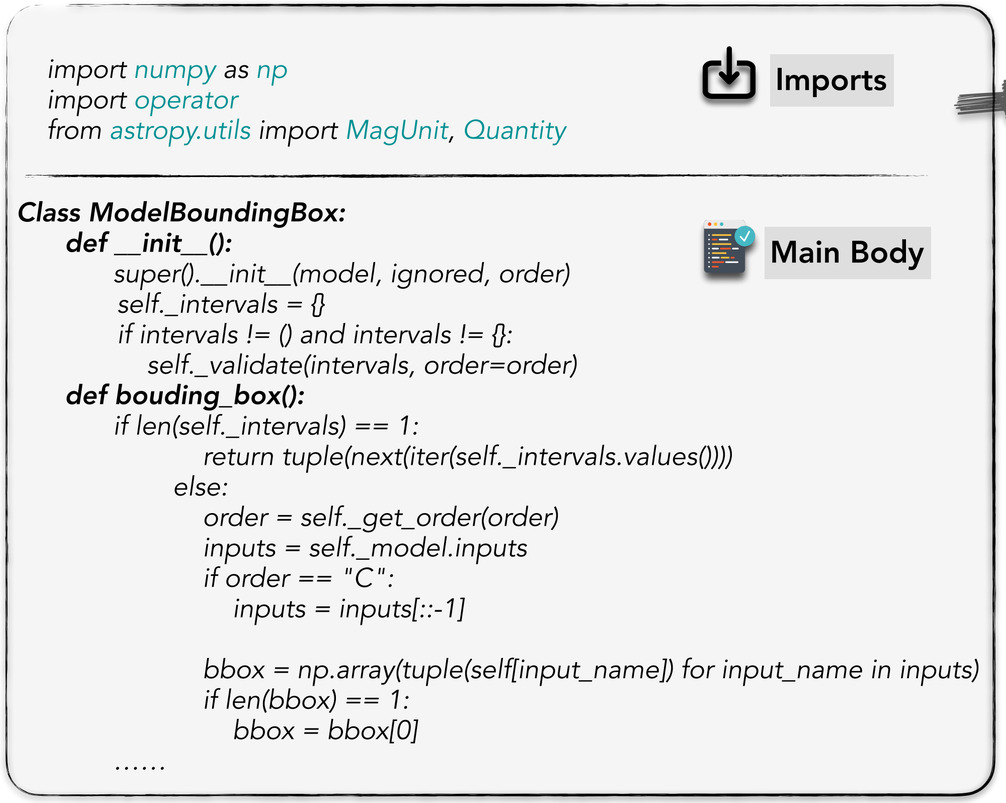
Close-sourced models are generally stronger.

Providing suitable/ comprehensive context for LLMs to solve the issue is important.

Tool	LLM	% Resolved	Avg. $ Cost	Avg. # Tokens	% Correct Location Line	Function	File
Alibaba Lingma Agent [7] 🔒	GPT-4o+ Claude-3.5	99 (33.00%)	-	-	40.0%	58.7%	75.0%
Factory Code Droid [5] 🔒	NA	94 (31.33%)	-	-	36.7%	55.7%	72.7%
AutoCodeRover-v2 [3] 🔒	GPT-4o	92 (30.67%)	-	-	35.0%	52.3%	69.3%
CodeR [17] 🔒	GPT-4	85 (28.33%)	$3.34	323,802	35.7%	52.3%	67.0%
IBM Research Agent-101 [6] 🔒	NA	80 (26.67%)	-	-	39.7%	56.7%	73.3%
OpenCSG StarShip [9] 🔒	GPT-4	71 (23.67%)	-	-	39.0%	61.7%	90.7%
Bytedance MarsCode [8] 🔒	GPT-4o	76 (25.33%)	-	-	37.3%	52.7%	73.7%
Amazon Q Developer [1] 🔒	NA	61 (20.33%)	-	-	34.0%	43.7%	71.7%
RepoUnderstander [41] 🔒	GPT-4	64 (21.33%)	-	-	-	-	-
Aider [21]	GPT-4o+ Claude-3	79 (26.33%)	-	-	35.3%	50.0%	69.7%
AutoCodeRover [65]	GPT-4	57 (19.00%)	$0.45	38,663	29.0%	42.3%	62.3%
SWE-agent [61]	Claude-3	35 (11.67%)	$3.42	221,258	26.3%	36.0%	48.0%
	GPT-4	54 (18.00%)	$2.51	245,008	30.7%	45.3%	61.0%
	GPT-4o	54 (17.00%)	-	-	-	-	-
OpenDevin [10]	GPT-4	50 (16.67%)	-	-	29.0%	39.0%	55.3%
	GPT-4o	52 (17.33%)	-	-	27.3%	39.3%	56.7%
RAG [28]	Claude-3	13 (4.33%)	$0.25	-	22.0%	30.0%	57.0%
	GPT-4	8 (2.67%)	$0.13	-	12.7%	23.3%	47.3%
	Claude-2	9 (3.00%)	-	-	16.7%	24.3%	46.7%
	ChatGPT	1 (0.33%)	-	-	6.3%	11.3%	27.3%
AGENTLESS 😺	GPT-4o	82 (27.33%)	$0.34	42,376	34.3%	51.0%	68.7%


Methodology Overview
Code Repo
1. Code line Parsing
2. Project-dependent relation filtering
3. Graph organization
RepoGraph
Issue Description
Agent Frameworks
Procedural Frameworks
Generated Patch


Code Repo
astropy
modeling
core.py
fitting.py
coordinates
earth.py
confest.py
logger.py
Code snippet
Class Model:
def __init__(self, …):
self._default_input….
def prepare_inputs(self, …):
input = self._validate_input_units()
inputs = len(input)
……
def hide_inverse(…): ……
reference node
invokes
class
definition node
contains
function
* File and code line illustrated in corresponding color.
REPOGRAPH
Node information
name: Model
fname: /astropy/modeling/core.py
kind: def
category: class
line: 523
methods: prepare_inputs,…
(a) RepoGraph Construction
Issue Description
Modeling's ``separability_matrix`` does not compute separability correctly for nested CompoundModels. Consider the following model: …
(b) Integration with Procedural Frameworks
search term identification
search_repograph (separability_matrix)
localization
core modules "Model"
correct generated patch
edition
search_repograph (Model)
search_repograph(param)
perform in each step of procedural framework
add to action space for agent decision-making
initialization
Agent
* Flattened node information from retrieved ego-graph.
Planning
Use search_repograph to find the dependencies for ``separability_matrix'' …
Action
search_repograph(param)
…
search_dir(param)
…
Observation
{name: Model, fname: / astropy/modeling/core.py, kind: def, category: class…
(c) Integration with Agent Frameworks



Project-dependent relation filtering

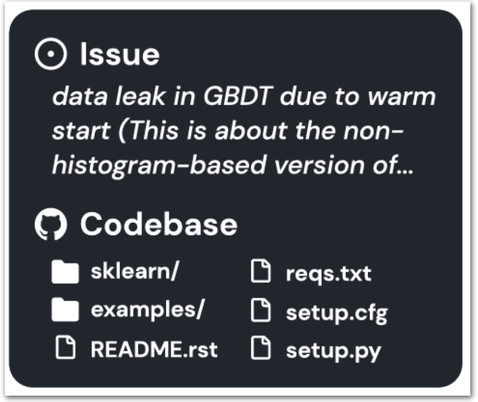
Section 3.1, Step 2

```
Class ModelBoundingBox:
    def __init__():
        super().__init__(model, ignored, order)
        self._intervals = {}
        if intervals != () and intervals != {}:
            self._validate(intervals, order=order)
    def bouding_box():
        if len(self._intervals) == 1:
            return tuple(next(iter(self._intervals.values())))
        else:
            order = self._get_order(order)
            inputs = self._model.inputs
            if order == "C":
                inputs = inputs[::-1]

            bbox = np.array(tuple(self[input_name]) for input_name in ..)
            if len(bbox) == 1:
                bbox = bbox[0]                    .....
```

Evaluation settings

- Datasets:
  - **SWE-bench:** issue resolving from GitHub
    - SWE-bench-Lite: 300 test instances in total



- Augmentation-based evaluation: code graph could be seamlessly integrated with baseline models
  - Agent framework
  - Procedural framework

Main results

Table 2: Results of REPOGRAPH with open-source baselines in two research lines, including procedural and agent frameworks. Numbers of accuracy-related metrics are directly taken from the leaderboard, while the cost-related ones are computed from the corresponding trajectories[6].

Methods	LLM	**Accuracy** *resolve*	*# samples*	*patch apply*	**Avg. Cost** *$ cost*	*# tokens*
Procedural frameworks						
RAG	GPT-4	2.67	8	29.33	$0.13	11,736
+REPOGRAPH	GPT-4	5.33 ↑2.66	16 ↑8	47.67 ↑18.34	$0.17	15,439
Agentless	GPT-4o	27.33	82	97.33	$0.34	42,376
+REPOGRAPH	GPT-4o	29.67 ↑2.34	89 ↑7	98.00 ↑0.67	$0.39	47,323
Agent frameworks						
AutoCodeRover	GPT-4	19.00	57	83.00	$0.45	38,663
+REPOGRAPH	GPT-4	21.33 ↑2.33	64 ↑7	86.67 ↑3.67	$0.58	45,112
SWE-agent	GPT-4o	18.33	55	87.00	$2.53	498,346
+REPOGRAPH	GPT-4o	20.33 ↑2.00	61 ↑6	90.33 ↑3.33	$2.69	518,792

RepoGraph brings consistent performance gain for all combinations of frameworks and LLM model bases.

Performance gain brought by RepoGraph is slightly larger on procedural frameworks than agent ones.

Performance gain brought by RepoGraph does not rely on more costs.

RepoGraph variants

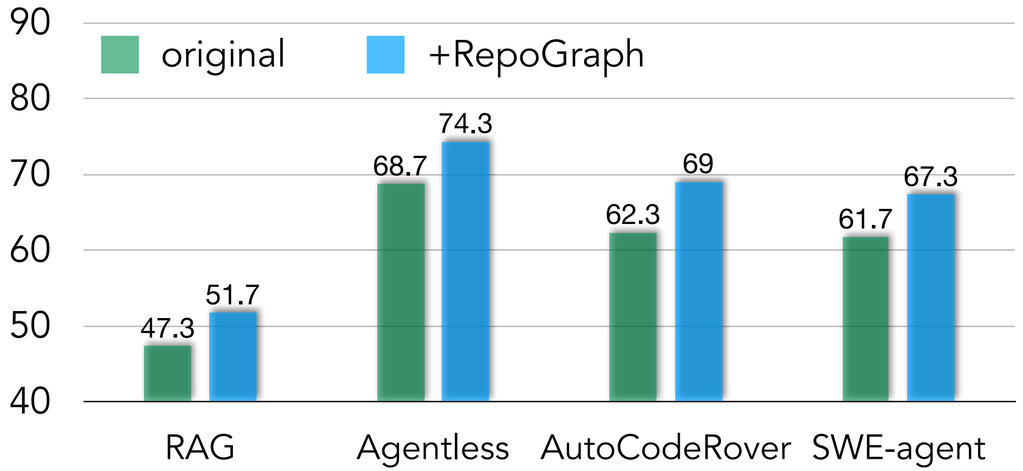
Table 4: The number of nodes, edges, and tokens of REPOGRAPH and its variants. For different retrieval and integration variants, we report the average number on the test set. "summ." refers to the summarized version by LLMs of the retrieved ego-graph.

Metrics	REPOGRAPH	1-hop + flatten	1-hop + summ.	2-hop + flatten	2-hop + summ.
# Nodes	1419.3	11.6	11.6	54.5	54.5
# Edges	26392.1	37.1	37.1	89.9	89.9
# tokens	-	2310.7	717.5	10505.3	1229.2
resolve rate	-	29.67	28.33	26.00	28.67

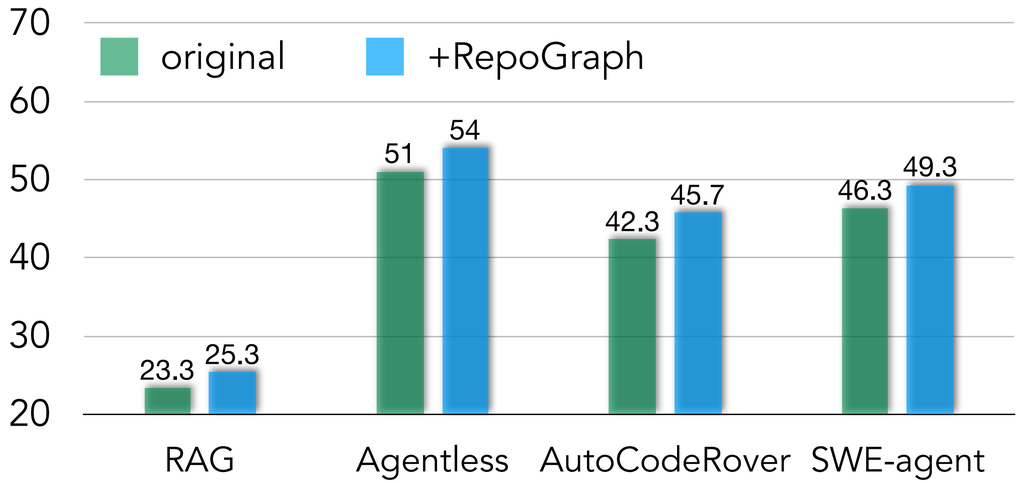
Comprehensive context of RepoGraph

Node and edges grow exponentially when k increases.
Flattening the graph increases the tokens.
Trade-off of token context comprehensiveness and the ability of LLMs to deal with it.

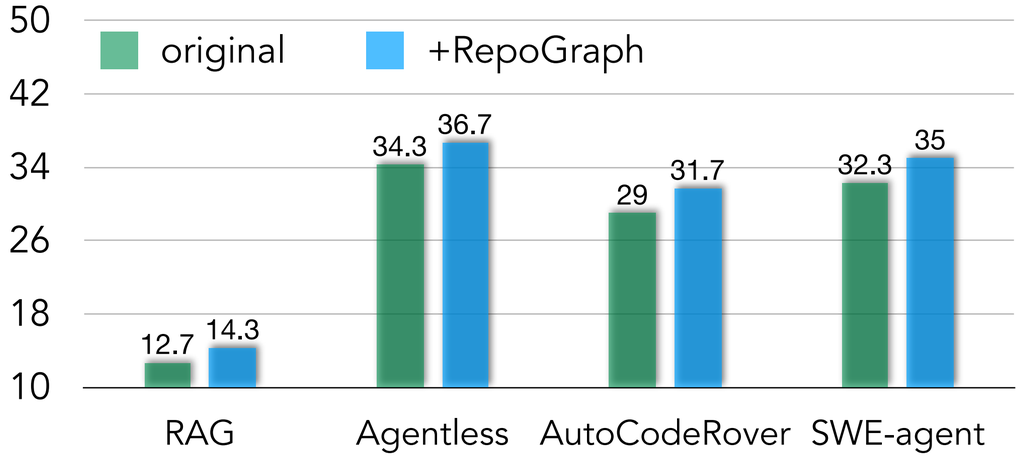
Localization coverage



File-level recall



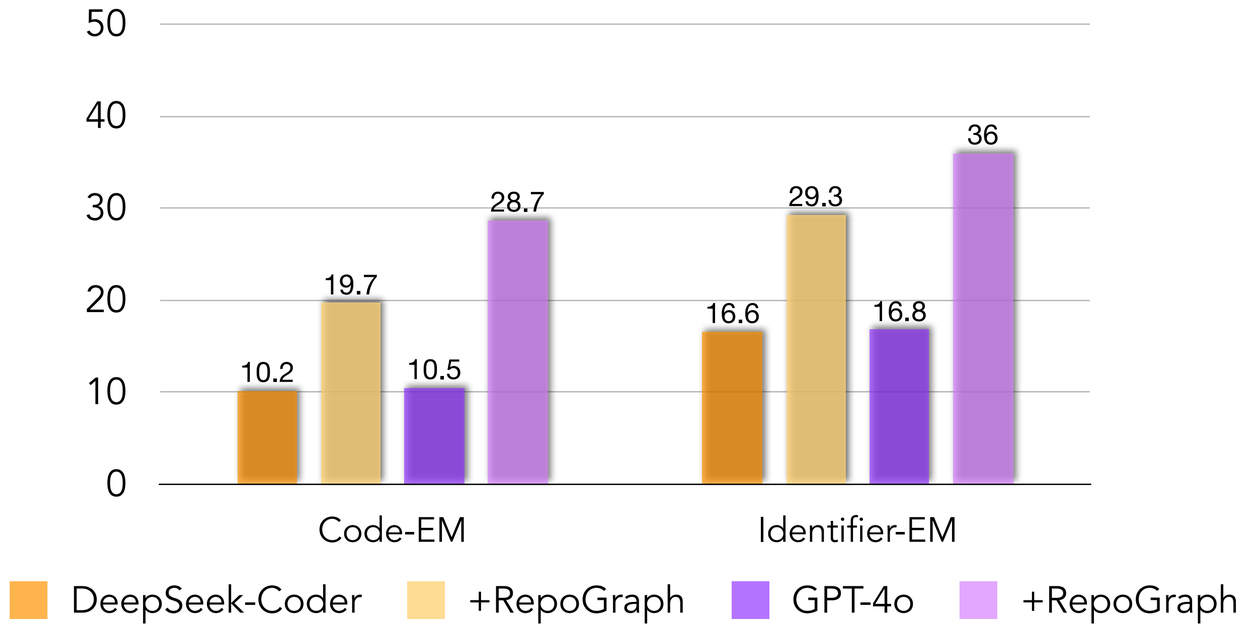
Function-level recall



Line-level recall

Recall improves at all granularities; the improvement at finer granularity is relatively smaller.

Transferability test



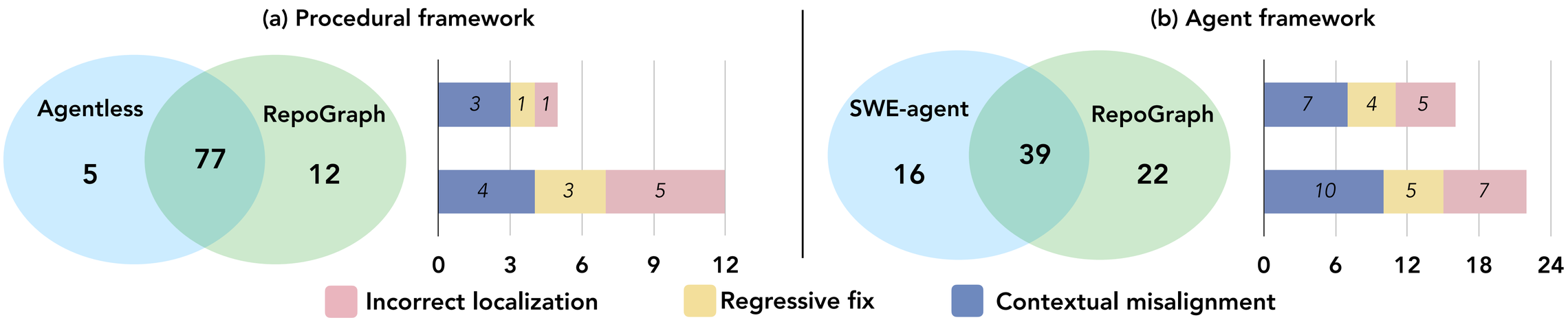
Experiments on CrossCodeEval, a repository-level code completion benchmark.

with the integration of the RepoGraph method, there is a substantial improvement across all metrics with both LLMs.

the improvement brought by RepoGraph is more pronounced when integrated with GPT-4o.

Error analysis

Proportion of all three error types decreases.



(i) **incorrect localization:** failure in accurately identifying code snippets

(ii) **contextual misalignment:** happens when the generated patch fails to align with the broader context of the codebase

(iii) **regressive fix:** introduces new issues in resolving the original issues

Improvement on agent frameworks are more complementary than procedural frameworks

Summary & Discussions

- Better structural representation for code repositories
  - Other file formats, such as docs/cofigs
  - Multi-granularity structures, such as folders
- Procedural v.s. agent frameworks
  - Exploration/choices on different scenarios

Thanks!

Q&A