EnCompass

Enhancing Agent Programming with Search Over Program Execution Paths

Speaker: Zhening Li
Authors: Zhening Li, Armando Solar-Lezama, Yisong Yue, Stephan Zheng

asari ai

NeurIPS 2025

EnCompass is a programming framework for adding **inference-time strategies** to **AI agents**.

sampling, refinement, backtracking, tree search, etc.

any program that calls LLMs to solve subtasks

EnCompass enables experimenting with different **inference-time strategies** *without modifying the underlying agent source code.*

Zhening Li, Armando Solar-Lezama, Yisong Yue and Stephan Zheng. "EnCompass: Enhancing Agent Programming with Search Over Program Execution Paths." NeurIPS, 2025.

Here’s a program that makes LLM calls to solve tasks in an escape room:

```python
def solve_escape_room():
    cipher = solve_cipher()
    logic = solve_logic(cipher)
    riddle = solve_riddle(cipher, logic)
    code = solve_combination(cipher, logic, riddle)
    open_door(code)
```

Zhening Li, Armando Solar-Lezama, Yisong Yue and Stephan Zheng. “EnCompass: Enhancing Agent Programming with Search Over Program Execution Paths.” NeurIPS, 2025.

But LLMs make mistakes — let’s use inference-time strategies like resampling and backtracking.

```
def solve_escape_room():
    # Try the cipher multiple times
    cipher_solutions = []
    for attempt in range(N):
        candidate = llm.solve_cipher()
        score = verify_cipher(candidate)
        cipher_solutions.append((candidate, score))

    best_cipher, best_cipher_score = max(cipher_solutions, key=lambda x: x[1])

    # Now try the logic puzzle multiple times
    ...
    best_logic, best_logic_score = max(logic_solutions, key=lambda x: x[1])
    if best_logic_score == 0:
        # Backtrack to attempt cipher again
        candidate = llm.solve_cipher()
        ...
    ...
```

The inference-time strategy is *hardcoded* into the workflow.

❌ readable
❌ modular
❌ flexible
❌ scalable

Zhening Li, Armando Solar-Lezama, Yisong Yue and Stephan Zheng. “EnCompass: Enhancing Agent Programming with Search Over Program Execution Paths.” NeurIPS, 2025.

The dream:

1. Annotate the steps that we may resample or backtrack to
2. Annotate information used by the resampling/backtracking strategy
3. Resampling/backtracking happens automatically at runtime!

The inference-time strategy is *separated* from the workflow.

✅ readable
✅ modular
✅ flexible
✅ scalable

Zhening Li, Armando Solar-Lezama, Yisong Yue and Stephan Zheng. “EnCompass: Enhancing Agent Programming with Search Over Program Execution Paths.” NeurIPS, 2025.

EnCompass: Separate inference-time strategies from the workflow

```
def solve_escape_room():

    cipher = solve_cipher()


    logic = solve_logic(cipher)


    riddle = solve_riddle(cipher, logic)


    code = solve_combination(cipher, logic, riddle)
    success = open_door(code)

solve_escape_room()
```

Zhening Li, Armando Solar-Lezama, Yisong Yue and Stephan Zheng. “EnCompass: Enhancing Agent Programming with Search Over Program Execution Paths.” NeurIPS, 2025.

EnCompass: Separate inference-time strategies from the workflow

```
@encompass.compile
def solve_escape_room():
    branchpoint()
    cipher = solve_cipher()
    record_score(verify_cipher(cipher))
    branchpoint()
    logic = solve_logic(cipher)
    record_score(verify_logic(logic))
    branchpoint()
    riddle = solve_riddle(cipher, logic)
    record_score(verify_riddle(riddle))
    branchpoint()
    code = solve_combination(cipher, logic, riddle)
    success = open_door(code)
    record_score(success)
solve_escape_room().search(search_algo, **search_config)
```

Zhening Li, Armando Solar-Lezama, Yisong Yue and Stephan Zheng. “EnCompass: Enhancing Agent Programming with Search Over Program Execution Paths.” NeurIPS, 2025.

EnCompass: Separate inference-time strategies from the workflow

```
@encompass.compile
def solve_escape_room():
    branchpoint()
    cipher = solve_cipher()
    record_score(verify_cipher(cipher))
    branchpoint()
    logic = solve_logic(cipher)
    record_score(verify_logic(logic))
    branchpoint()
    riddle = solve_riddle(cipher, logic)
    record_score(verify_riddle(riddle))
    branchpoint()
    code = solve_combination(cipher, logic, riddle)
    success = open_door(code)
    record_score(success)
solve_escape_room().search(search_algo, **search_config)
```

Zhening Li, Armando Solar-Lezama, Yisong Yue and Stephan Zheng. “EnCompass: Enhancing Agent Programming with Search Over Program Execution Paths.” NeurIPS, 2025.

EnCompass: Separate inference-time strategies from the workflow

```python
@encompass.compile
def solve_escape_room():
    branchpoint()
    cipher = solve_cipher()
    record_score(verify_cipher(cipher))
    branchpoint()
    logic = solve_logic(cipher)
    record_score(verify_logic(logic))
    branchpoint()
    riddle = solve_riddle(cipher, logic)
    record_score(verify_riddle(riddle))
    branchpoint()
    code = solve_combination(cipher, logic, riddle)
    success = open_door(code)
    record_score(success)
solve_escape_room().search(search_algo, **search_config)
```

Zhening Li, Armando Solar-Lezama, Yisong Yue and Stephan Zheng. “EnCompass: Enhancing Agent Programming with Search Over Program Execution Paths.” NeurIPS, 2025.

EnCompass: Separate inference-time strategies from the workflow

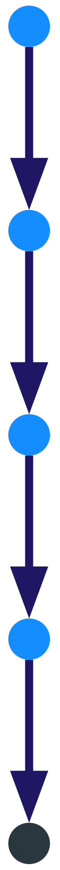
```
@encompass.compile
def solve_escape_room():
    branchpoint()
    cipher = solve_cipher()
    record_score(verify_cipher(cipher))
    branchpoint()
    logic = solve_logic(cipher)
    record_score(verify_logic(logic))
    branchpoint()
    riddle = solve_riddle(cipher, logic)
    record_score(verify_riddle(riddle))
    branchpoint()
    code = solve_combination(cipher, logic, riddle)
    success = open_door(code)
    record_score(success)
solve_escape_room().search(search_algo, **search_config)
```

Inference time strategies as *search over program execution paths*

Zhening Li, Armando Solar-Lezama, Yisong Yue and Stephan Zheng. “EnCompass: Enhancing Agent Programming with Search Over Program Execution Paths.” NeurIPS, 2025.

Inference time strategies as search over program execution paths

```
@encompass.compile
def solve_escape_room():
    branchpoint()
    cipher = solve_cipher()
    record_score(verify_cipher(cipher))
    branchpoint()
    logic = solve_logic(cipher)
    record_score(verify_logic(logic))
    branchpoint()
    riddle = solve_riddle(cipher, logic)
    record_score(verify_riddle(riddle))
    branchpoint()
    code = solve_combination(cipher, logic, riddle)
    success = open_door(code)
    record_score(success)
solve_escape_room().search(search_algo, **search_config)
```

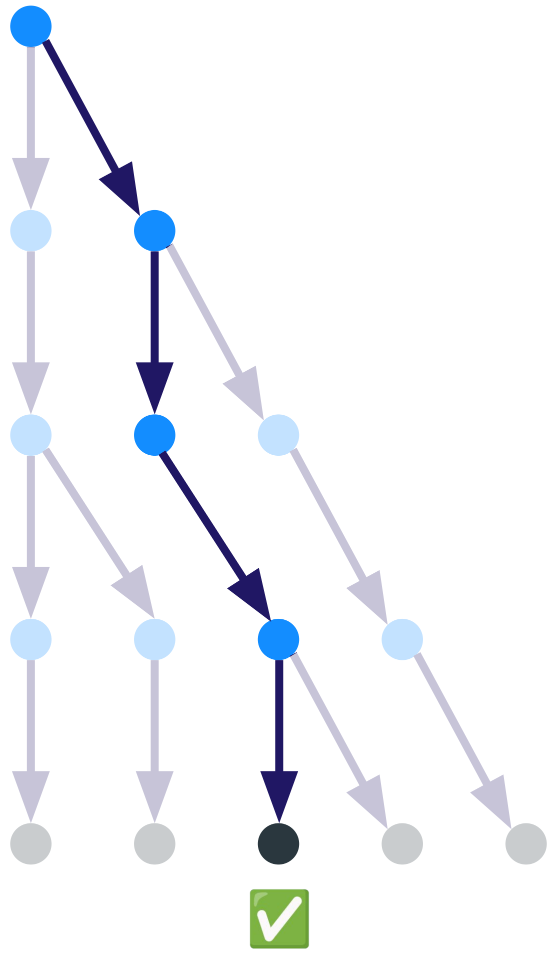


Zhening Li, Armando Solar-Lezama, Yisong Yue and Stephan Zheng. “EnCompass: Enhancing Agent Programming with Search Over Program Execution Paths.” NeurIPS, 2025.

Inference time strategies as search over program execution paths

```
@encompass.compile
def solve_escape_room():
    branchpoint()
    cipher = solve_cipher()
    record_score(verify_cipher(cipher))
    branchpoint()
    logic = solve_logic(cipher)
    record_score(verify_logic(logic))
    branchpoint()
    riddle = solve_riddle(cipher, logic)
    record_score(verify_riddle(riddle))
    branchpoint()
    code = solve_combination(cipher, logic, riddle)
    success = open_door(code)
    record_score(success)
solve_escape_room().search(search_algo, **search_config)
```

← maximize recorded score



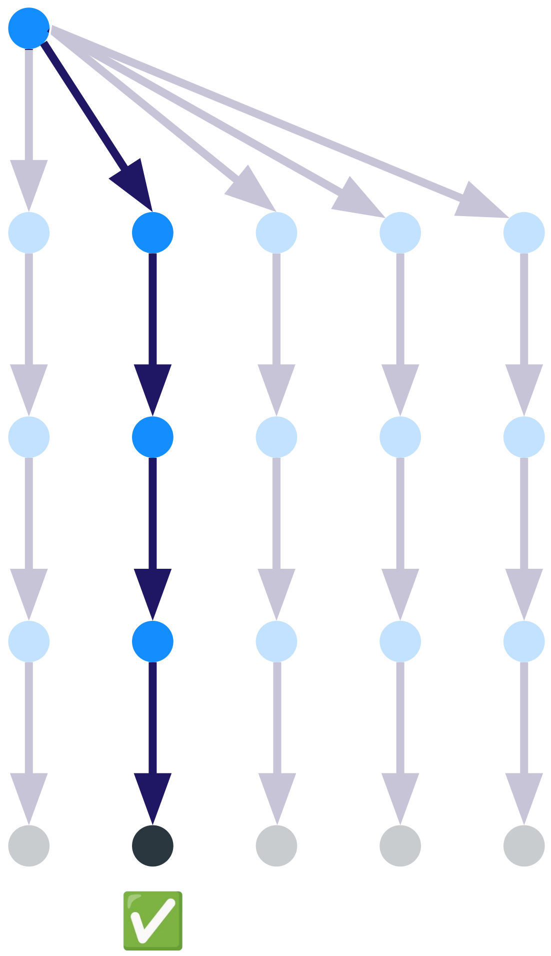
Zhening Li, Armando Solar-Lezama, Yisong Yue and Stephan Zheng. “EnCompass: Enhancing Agent Programming with Search Over Program Execution Paths.” NeurIPS, 2025.

Inference time strategies as search over program execution paths

```
@encompass.compile
def solve_escape_room():
    branchpoint()
    cipher = solve_cipher()
    record_score(verify_cipher(cipher))
    branchpoint()
    logic = solve_logic(cipher)
    record_score(verify_logic(logic))
    branchpoint()
    riddle = solve_riddle(cipher, logic)
    record_score(verify_riddle(riddle))
    branchpoint()
    code = solve_combination(cipher, logic, riddle)
    success = open_door(code)
    record_score(success)
solve_escape_room().search("sampling", num_rollouts=5)
```

← maximize recorded score

Global best-of-*N*



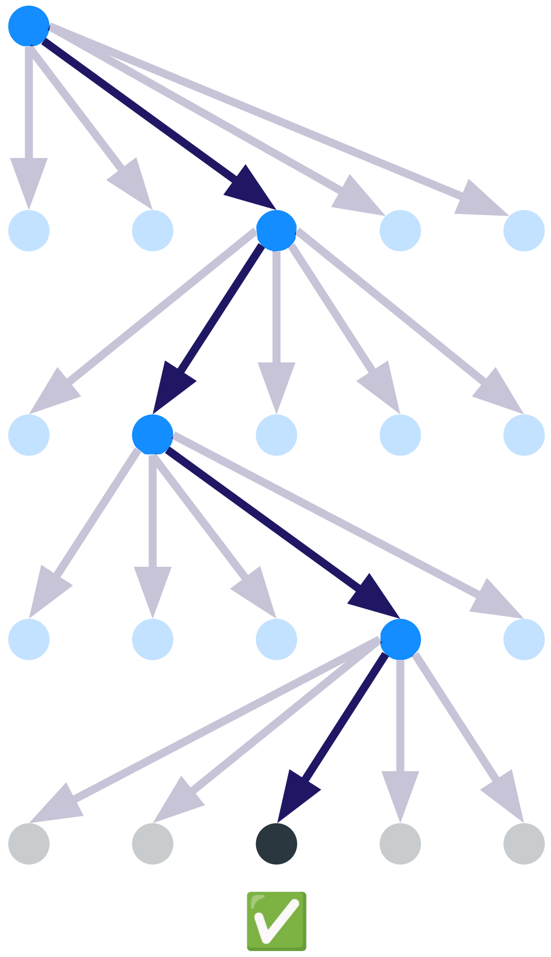
Zhening Li, Armando Solar-Lezama, Yisong Yue and Stephan Zheng. “EnCompass: Enhancing Agent Programming with Search Over Program Execution Paths.” NeurIPS, 2025.

Inference time strategies as search over program execution paths

```
@encompass.compile
def solve_escape_room():
    branchpoint()
    cipher = solve_cipher()
    record_score(verify_cipher(cipher))
    branchpoint()
    logic = solve_logic(cipher)
    record_score(verify_logic(logic))
    branchpoint()
    riddle = solve_riddle(cipher, logic)
    record_score(verify_riddle(riddle))
    branchpoint()
    code = solve_combination(cipher, logic, riddle)
    success = open_door(code)
    record_score(success)
solve_escape_room().search("beam", beam_width=1, branching=5)
```

← maximize recorded score

Local best-of-*N*



Zhening Li, Armando Solar-Lezama, Yisong Yue and Stephan Zheng. "EnCompass: Enhancing Agent Programming with Search Over Program Execution Paths." NeurIPS, 2025.

Inference time strategies as search over program execution paths

```
@encompass.compile
def solve_escape_room():
    branchpoint()
```

Local best-of-*N*

```
solve_escape_room().search("beam", beam_width=1, branching=5)
```

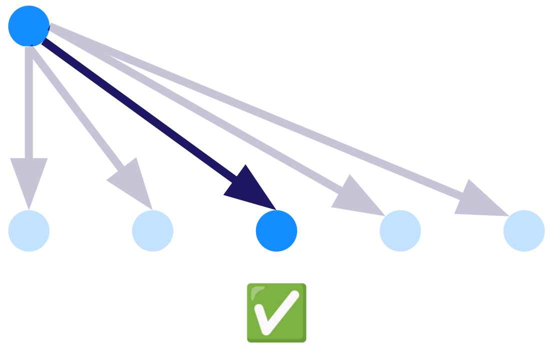
Zhening Li, Armando Solar-Lezama, Yisong Yue and Stephan Zheng. “EnCompass: Enhancing Agent Programming with Search Over Program Execution Paths.” NeurIPS, 2025.

Inference time strategies as search over program execution paths

```
@encompass.compile
def solve_escape_room():
    branchpoint()
    cipher = solve_cipher()
    record_score(verify_cipher(cipher))
    branchpoint()
```

maximize
recorded score

Local best-of-*N*



```
solve_escape_room().search("beam", beam_width=1, branching=5)
```

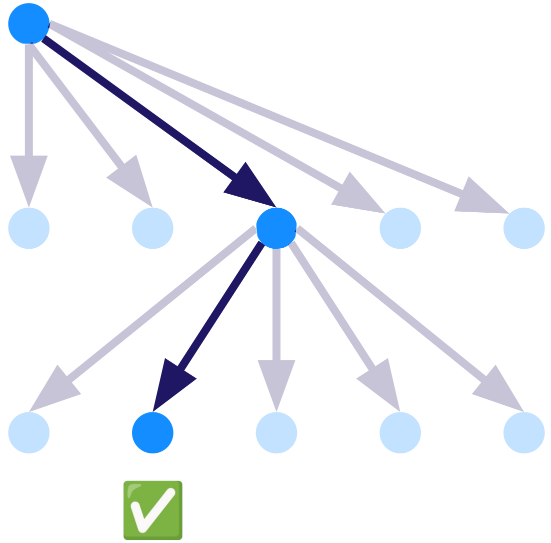
Zhening Li, Armando Solar-Lezama, Yisong Yue and Stephan Zheng. “EnCompass: Enhancing Agent Programming with Search Over Program Execution Paths.” NeurIPS, 2025.

Inference time strategies as search over program execution paths

```
@encompass.compile
def solve_escape_room():
    branchpoint()
    cipher = solve_cipher()
    record_score(verify_cipher(cipher))
    branchpoint()
    logic = solve_logic(cipher)
    record_score(verify_logic(logic)) ←
    branchpoint()
```

maximize
recorded score

Local best-of-*N*



```
solve_escape_room().search("beam", beam_width=1, branching=5)
```

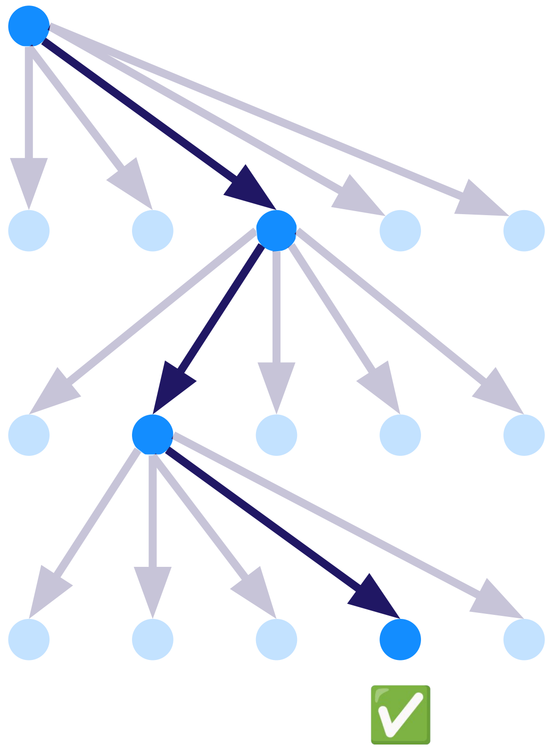
Zhening Li, Armando Solar-Lezama, Yisong Yue and Stephan Zheng. “EnCompass: Enhancing Agent Programming with Search Over Program Execution Paths.” NeurIPS, 2025.

Inference time strategies as search over program execution paths

```
@encompass.compile
def solve_escape_room():
    branchpoint()
    cipher = solve_cipher()
    record_score(verify_cipher(cipher))
    branchpoint()
    logic = solve_logic(cipher)
    record_score(verify_logic(logic))
    branchpoint()
    riddle = solve_riddle(cipher, logic)
    record_score(verify_riddle(riddle))
    branchpoint()
```

maximize
recorded score

Local best-of-*N*



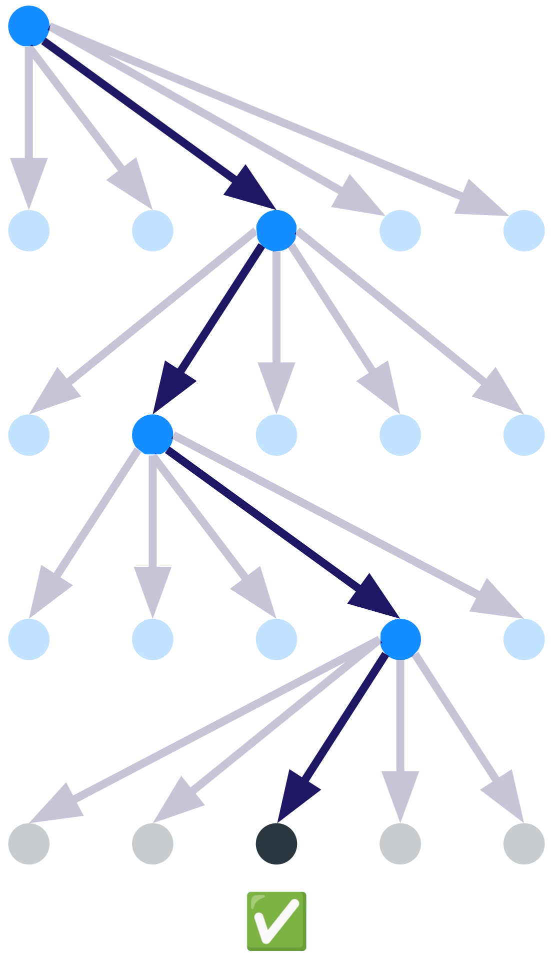
```
solve_escape_room().search("beam", beam_width=1, branching=5)
```

Zhening Li, Armando Solar-Lezama, Yisong Yue and Stephan Zheng. “EnCompass: Enhancing Agent Programming with Search Over Program Execution Paths.” NeurIPS, 2025.

Inference time strategies as search over program execution paths

```
@encompass.compile
def solve_escape_room():
    branchpoint()
    cipher = solve_cipher()
    record_score(verify_cipher(cipher))
    branchpoint()
    logic = solve_logic(cipher)
    record_score(verify_logic(logic))
    branchpoint()
    riddle = solve_riddle(cipher, logic)
    record_score(verify_riddle(riddle))
    branchpoint()
    code = solve_combination(cipher, logic, riddle)
    success = open_door(code)
    record_score(success)  ←— maximize recorded score
solve_escape_room().search("beam", beam_width=1, branching=5)
```

Local best-of-*N*

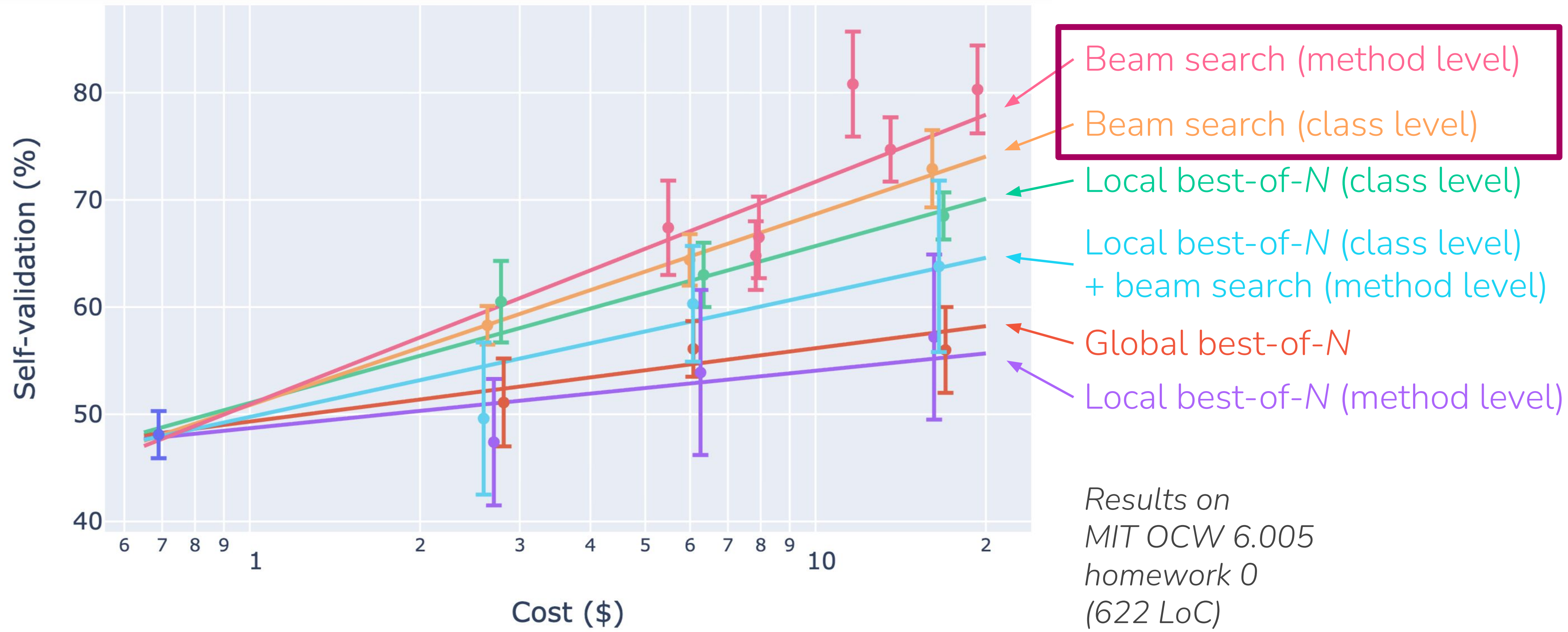


Zhening Li, Armando Solar-Lezama, Yisong Yue and Stephan Zheng. “EnCompass: Enhancing Agent Programming with Search Over Program Execution Paths.” NeurIPS, 2025.

Case Study: Java → Python repository translation agent

Agent with 597 lines of code (not including helper/utility functions, etc.)

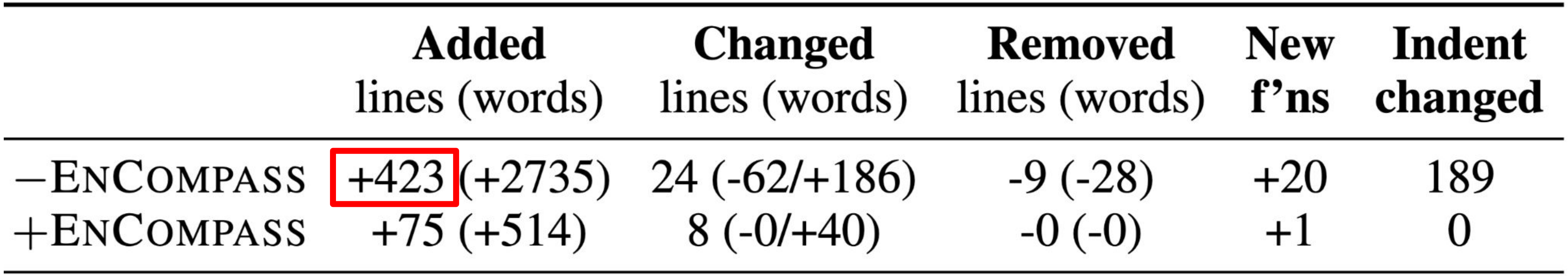
- Iterates through each class and method in Java repo, translating methods one by one



Results on
MIT OCW 6.005
homework 0
(622 LoC)

Zhening Li, Armando Solar-Lezama, Yisong Yue and Stephan Zheng. "EnCompass: Enhancing Agent Programming with Search Over Program Execution Paths." NeurIPS, 2025.

Case Study: Java → Python repository translation agent



	Added lines (words)	Changed lines (words)	Removed lines (words)	New f'ns	Indent changed
−EnCompass	+423 (+2735)	24 (-62/+186)	-9 (-28)	+20	189
+EnCompass	+75 (+514)	8 (-0/+40)	-0 (-0)	+1	0

Zhening Li, Armando Solar-Lezama, Yisong Yue and Stephan Zheng. “EnCompass: Enhancing Agent Programming with Search Over Program Execution Paths.” NeurIPS, 2025.

Key takeaways

- EnCompass separates the *overlaying inference-time strategy* from the *underlying workflow*
- This separation of concerns enables independent experimentation of each component
- This facilitates the discovery of inference-time strategies that scale better

Zhening Li, Armando Solar-Lezama, Yisong Yue and Stephan Zheng. “EnCompass: Enhancing Agent Programming with Search Over Program Execution Paths.” NeurIPS, 2025.