Lessons Learned: A Multi-Agent Framework for Code LLMs to Learn and Improve

Yuanzhe Liu[1], Ryan Deng[2], Tim Kaler[2], Xuhao Chen[2,3], Charles E. Leiserson[2], Yao Ma[1], Jie Chen[4]

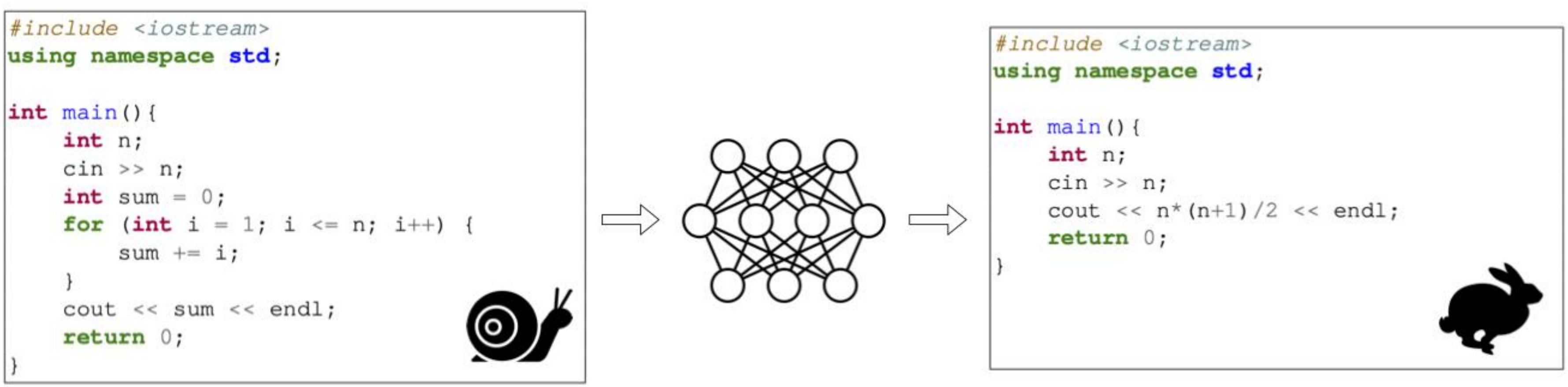
[1]Rensselaer Polytechnic Institute [2]Massachusetts Institute of Technology

[3]Michigan State University [4]MIT-IBM Watson AI Lab, IBM Research

NeurIPS 2025

Motivation

- Code Optimization is less explored.
- Task of code optimization: `f(slower-code) = faster-code`
- Slower code is
  - Compilable & Correct

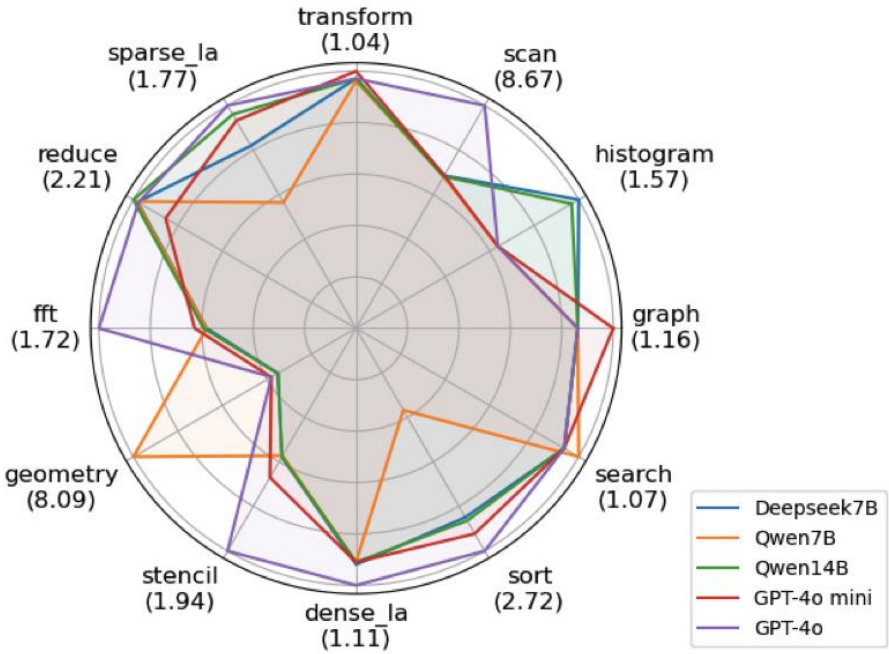


(a) Slower Code. (b) Faster Code.

Shypula, Alexander, et al. "Learning performance-improving code edits." *arXiv preprint arXiv:2302.07867* (2023).

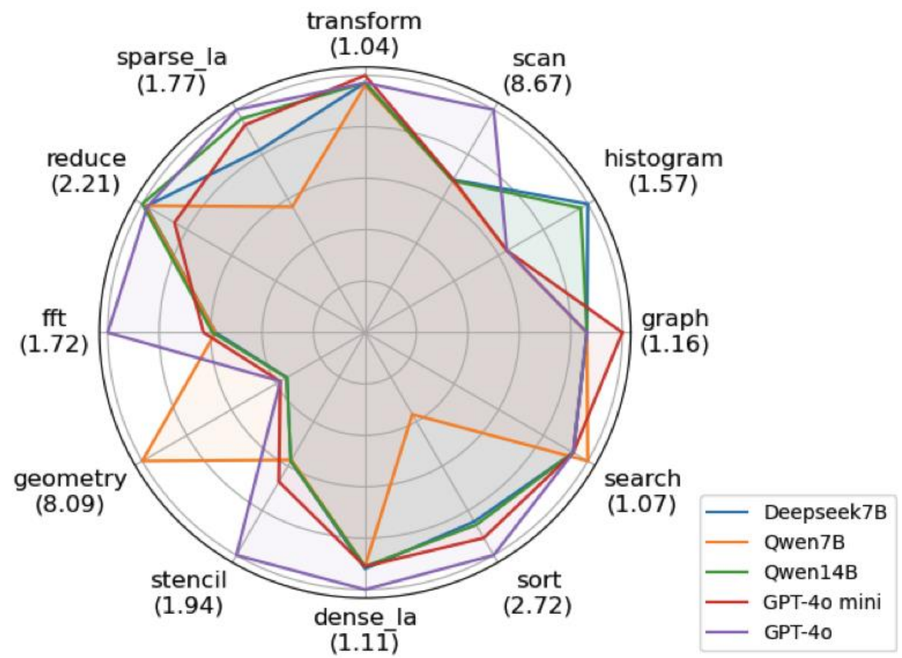
Our findings

- On ParEval benchmark, no one LLM performs the best on all problems



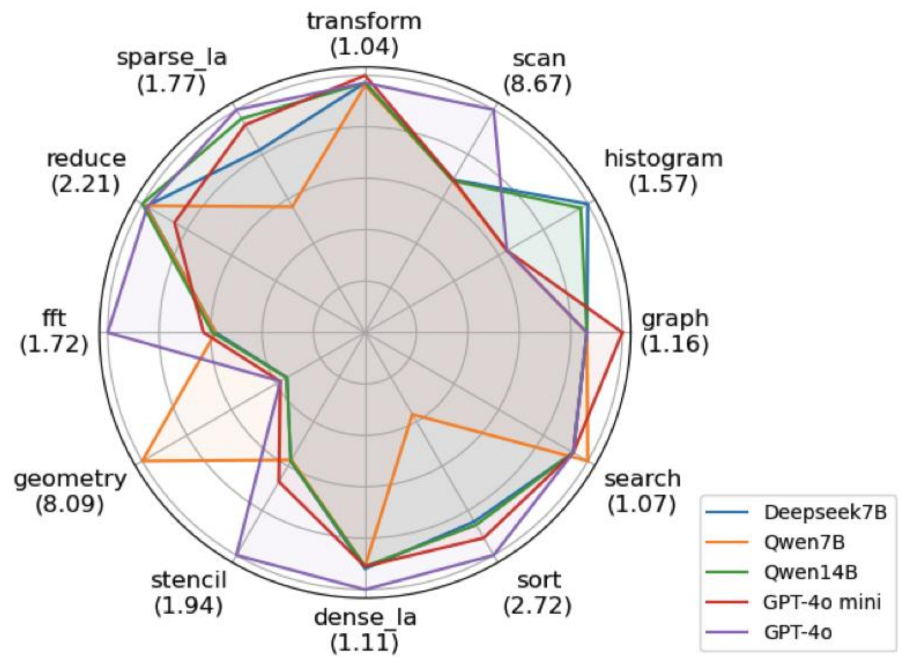
Our findings

- On ParEval benchmark, no one LLM performs the best on all problems
- GPT-4o is the overall winner.



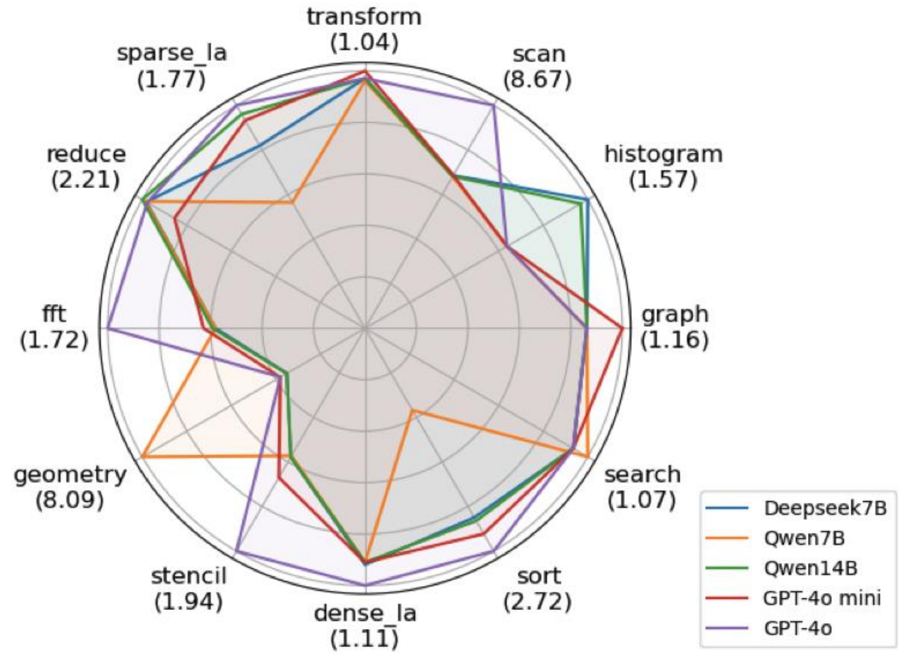
Our findings

- On ParEval benchmark, no one LLM performs the best on all problems
- GPT-4o is the overall winner.
- E.g. on “Histogram” Deepseek7B and Qwen14B outperform GPT-4o by 1.6 ×



Our findings

- On ParEval benchmark, no one LLM performs the best on all problems
- GPT-4o is the overall winner.
- E.g. on "Histogram" Deepseek7B and Qwen14B outperform GPT-4o by 1.6×
- Different LLM presents distinct capabilities.



Usage of Multiple LLMs

- How to use multiple agents to solve a coding problem?

Usage of Multiple LLMs

- How to use multiple agents to solve a coding problem?
  - We advocate the concept of *lessons*

Usage of Multiple LLMs

- How to use multiple agents to solve a coding problem?
  - We advocate the concept of *lessons*
  - Such *lessons* are summarized by LLM agents, learned by others, so they can collectively improve the code performance.

Usage of Multiple LLMs

- How to use multiple agents to solve a coding problem?
  - We advocate the concept of *lessons*
  - Such *lessons* are summarized by LLM agents, learned by others, so they can collectively improve the code performance.

Original code

```
for (int i = 0; i < n; ++i)
  for (int j = 0; j < n; ++j)
    for (int k = 0; k < n; ++k)
      C[i][j] += A[i][k] * B[k][j];
```

Naive implementation of matrix multiplication $C = AB$.

Improved code, round 1

```
for (int i = 0; i < n; ++i)
  for (int k = 0; k < n; ++k)
    for (int j = 0; j < n; ++j)
      C[i][j] += A[i][k] * B[k][j];
```

Lesson: Reordering loops improves cache locality and increases performance. The order of `(i,k,j)` out of 6 different permutations often performs the best, because of how caches work.

Usage of Multiple LLMs: **LessonL**

How do agents work together in practice by *lesson*?

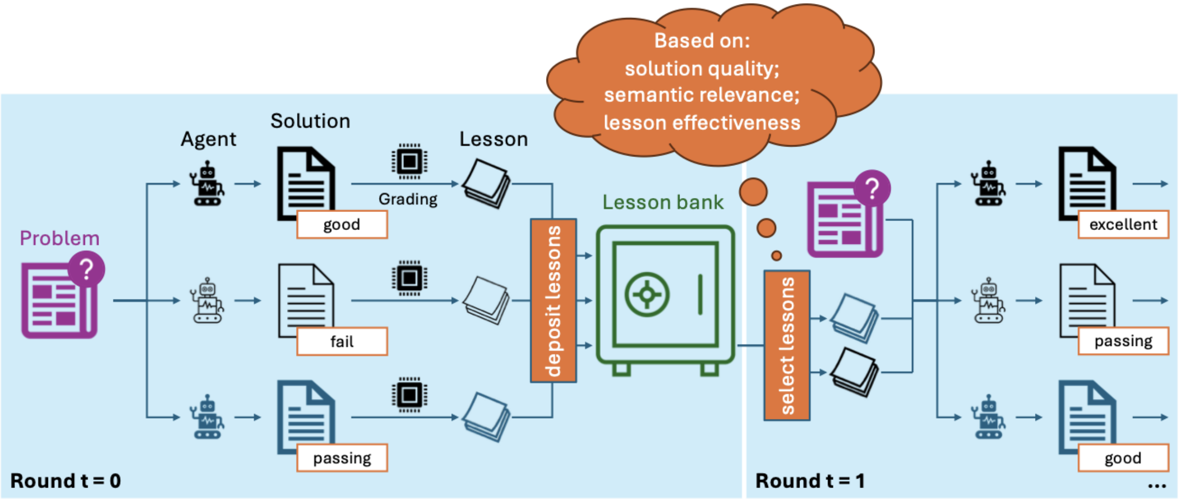


Figure 2: The LessonL framework (which may repeat multiple rounds).

Usage of Multiple LLMs: **LessonL**

How do agents work together in practice by *lesson*?

- Lesson Solicitation
  - under different scenarios

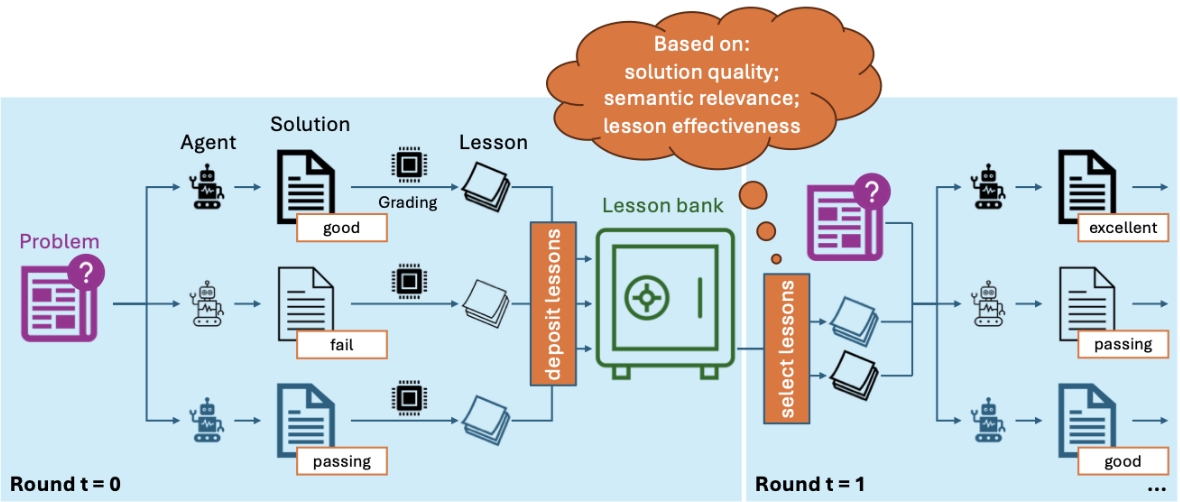


Figure 2: The LessonL framework (which may repeat multiple rounds).

Usage of Multiple LLMs: **LessonL**

How do agents work together in practice by *lesson*?

- Lesson Solicitation
  - under different scenarios
- Lesson Banking and Selection
  - reduce context length

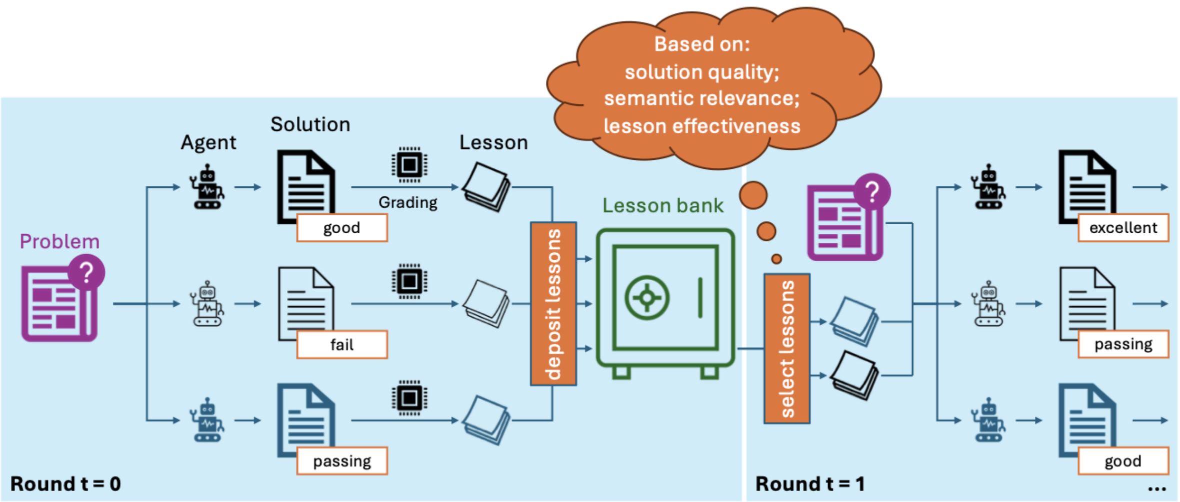


Figure 2: The LessonL framework (which may repeat multiple rounds).

Usage of Multiple LLMs: **LessonL**

How do agents work together in practice by *lesson*?

- Lesson Solicitation
  - under different scenarios
- Lesson Banking and Selection
  - reduce context length
- Effectiveness Adjustment factor **f**:
  - adjust selection dynamically

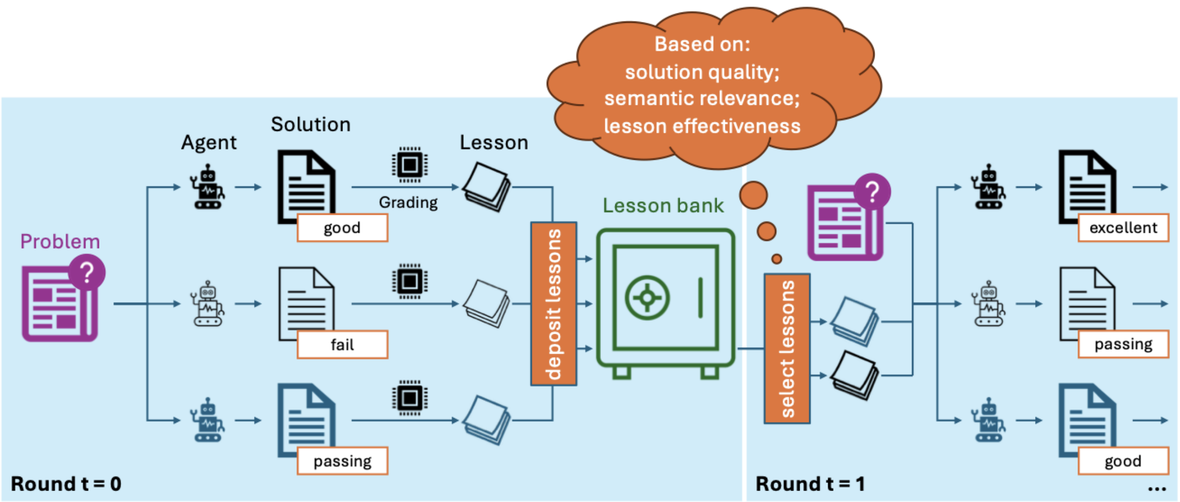


Figure 2: The LessonL framework (which may repeat multiple rounds).

Usage of Multiple LLMs: **LessonL**

How do agents work together in practice by *lesson*?

- Lesson Solicitation
  - under different scenarios
- Lesson Banking and Selection
  - reduce context length
- Effectiveness Adjustment factor **f**:
  - adjust selection dynamically

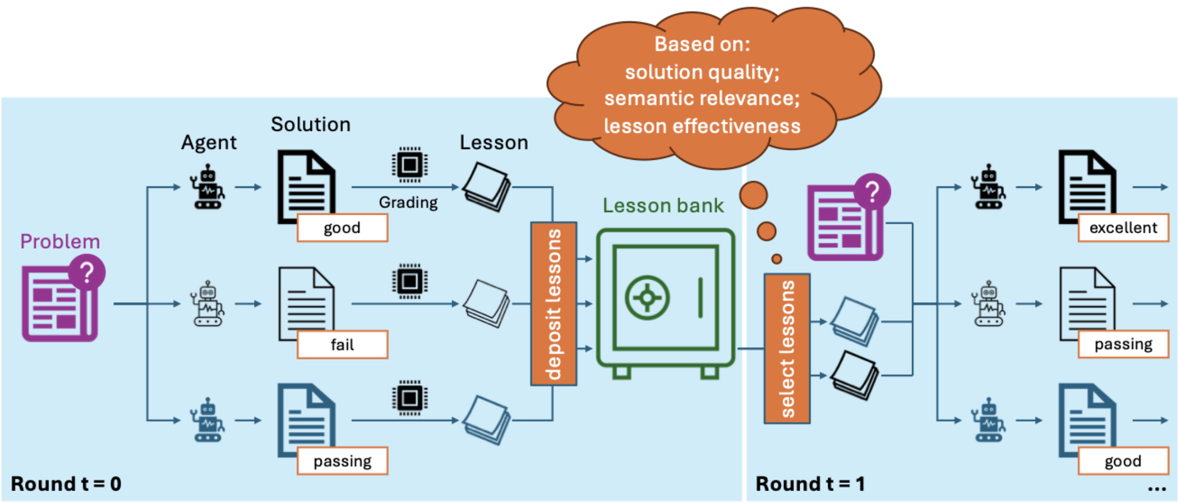


Figure 2: The LessonL framework (which may repeat multiple rounds).

Results on Code Optimization

ParEval	Serial mode			OpenMP mode		
	Correct	$> 2x$	Speedup	Correct	$> 2x$	Speedup
GPT-4o	0.80 ± 0.00	0.16 ± 0.03	1.72 ± 0.11	0.73 ± 0.05	0.58 ± 0.05	2.93 ± 0.30
OpenAI o3	0.77 ± 0.02	$\mathbf{0.23 \pm 0.04}$	$\mathbf{2.21 \pm 0.16}$	0.72 ± 0.03	0.58 ± 0.03	$\mathbf{3.55 \pm 0.27}$
MapCoder	$\underline{0.88 \pm 0.02}$	0.15 ± 0.02	1.85 ± 0.08	$\underline{0.83 \pm 0.05}$	0.58 ± 0.02	3.43 ± 0.17
LessonL	$\mathbf{0.91 \pm 0.02}$	$\underline{0.21 \pm 0.01}$	$\underline{2.16 \pm 0.11}$	$\mathbf{0.86 \pm 0.01}$	$\mathbf{0.62 \pm 0.02}$	$\underline{3.46 \pm 0.03}$

LessonL models:

- deepseek-coder-7b-instruct-v1.5
- Qwen2.5-Coder-7B-Instruct
- Qwen2.5-Coder-14B-Instruct

Thanks for listening!

NeurIPS 2025 Exhibition Hall C,D,E

Thu 4 Dec 4:30 p.m. – 7:30 p.m. PDT

paper

code