Who improves the improver?

Meta^n: Recursive Self-Improvement through Emergent Depth

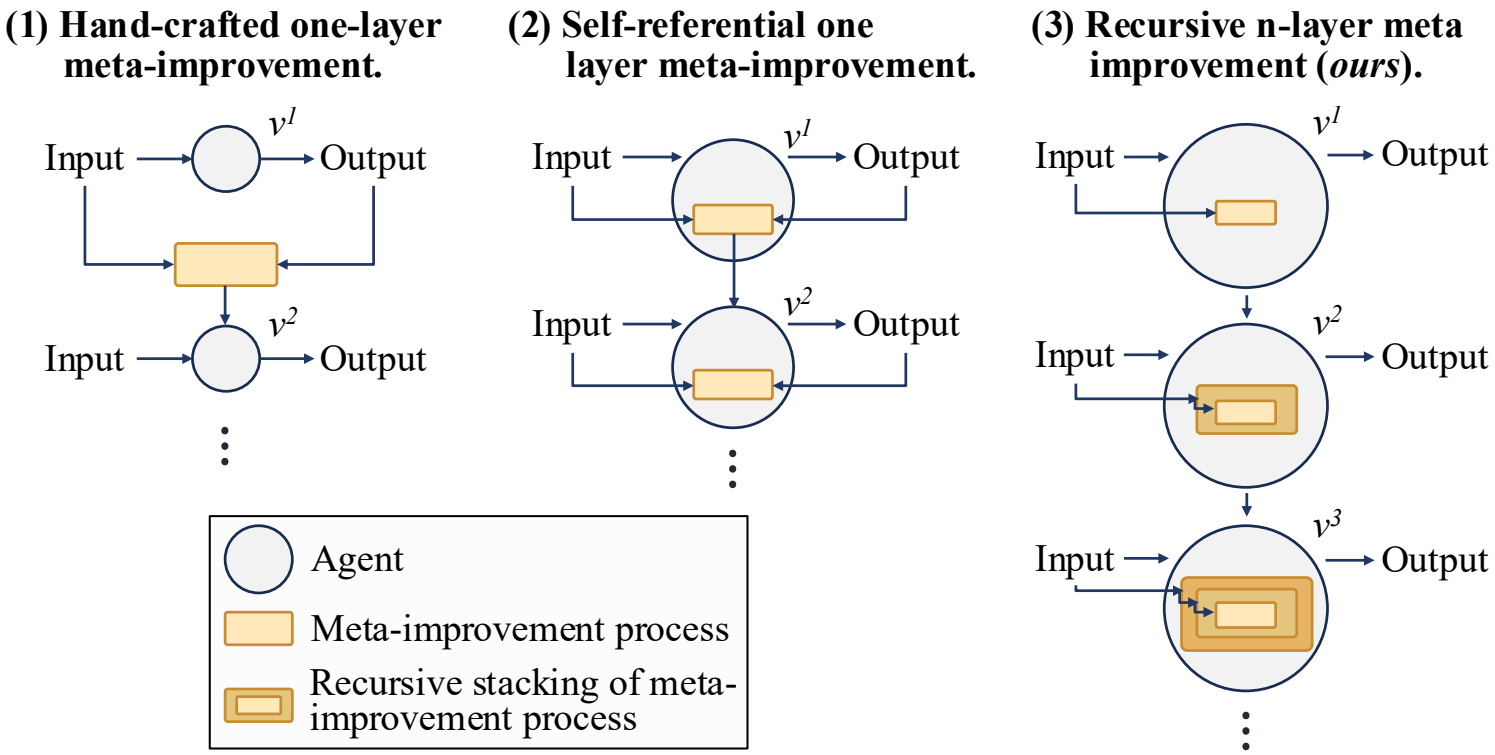
Zae Myung Kim[1], Young-Jun Lee[1], Seungyeon Jwa[2], Dongyeop Kang[1]

[1]University of Minnesota [2]Seoul National University





Today, depth costs stability



Realized meta-depth

1 — **Hand-crafted**
FunSearch, AlphaEvolve, ADAS. The search loop never changes.

≈2.5 — **Self-editing**
Gödel Agent, DGM. A frozen driver keeps it stable.

n — **Meta^n**
Fixed improver, growing input. Depth set by convergence.

Rewrite the improver and it can break itself. Freeze it and depth stops.

Fix the improver, recurse on its input

Ω never changes

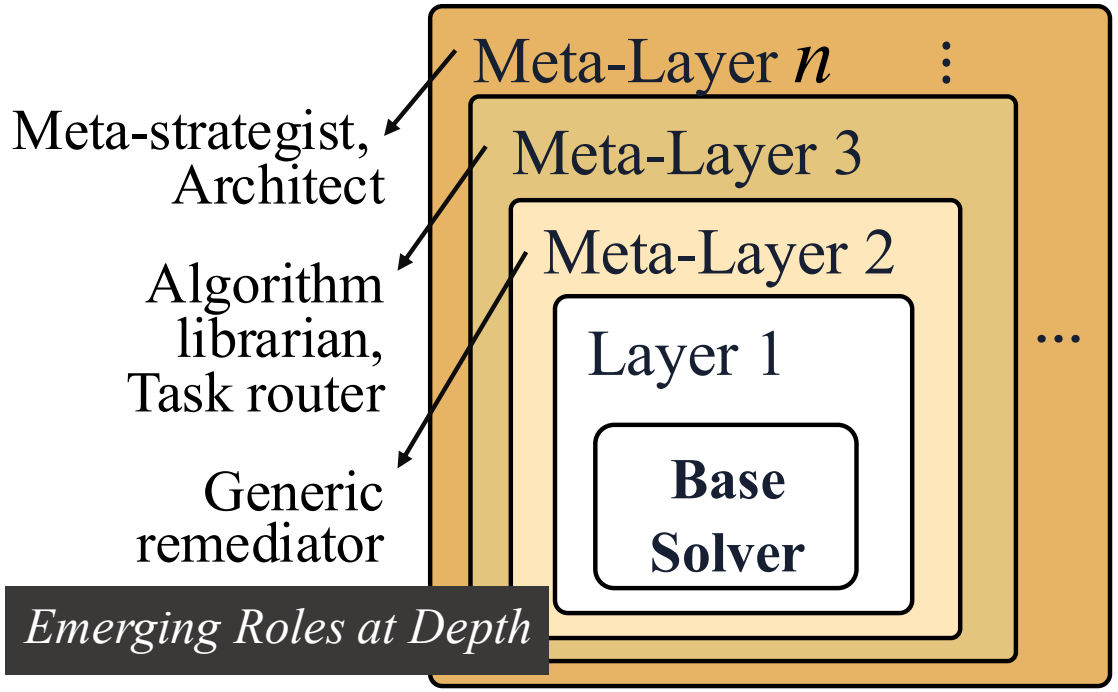
The same prompt at every depth and benchmark.

Its input grows

Each layer reads the traces and the code of the stack below.

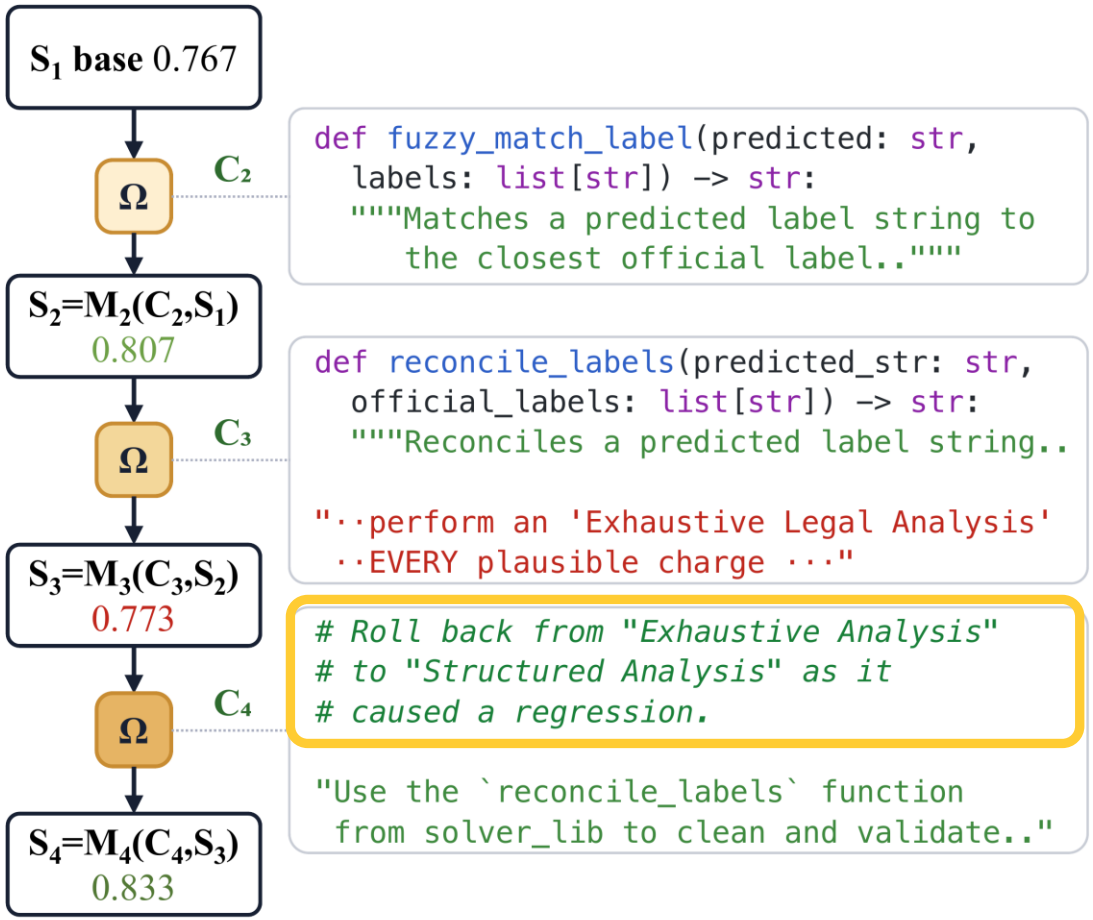
Depth is not chosen

It stops at convergence, after 3 to 6 layers.



The improver stays fixed. What it reads does not.

A layer that undoes a mistake



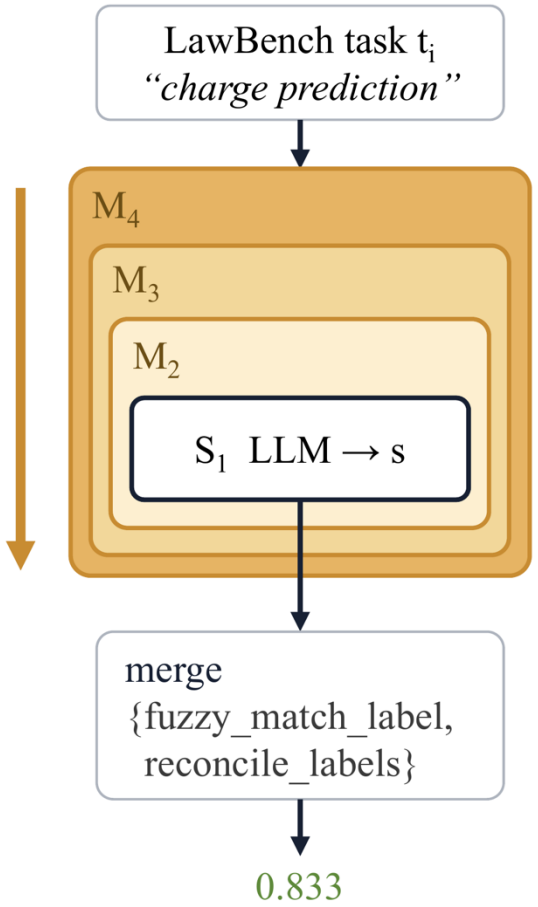

```python
def fuzzy_match_label(predicted: str,
  labels: list[str]) -> str:
  """Matches a predicted label string to
     the closest official label.."""
```

```python
def reconcile_labels(predicted_str: str,
  official_labels: list[str]) -> str:
  """Reconciles a predicted label string..

"··perform an 'Exhaustive Legal Analysis'
 ··EVERY plausible charge ···"
```

```python
# Roll back from "Exhaustive Analysis"
# to "Structured Analysis" as it
# caused a regression.

"Use the `reconcile_labels` function
 from solver_lib to clean and validate.."
```



Depth 2

0.807 +0.040

a label-matching helper

Depth 3

0.773 −0.034

a second helper and a directive

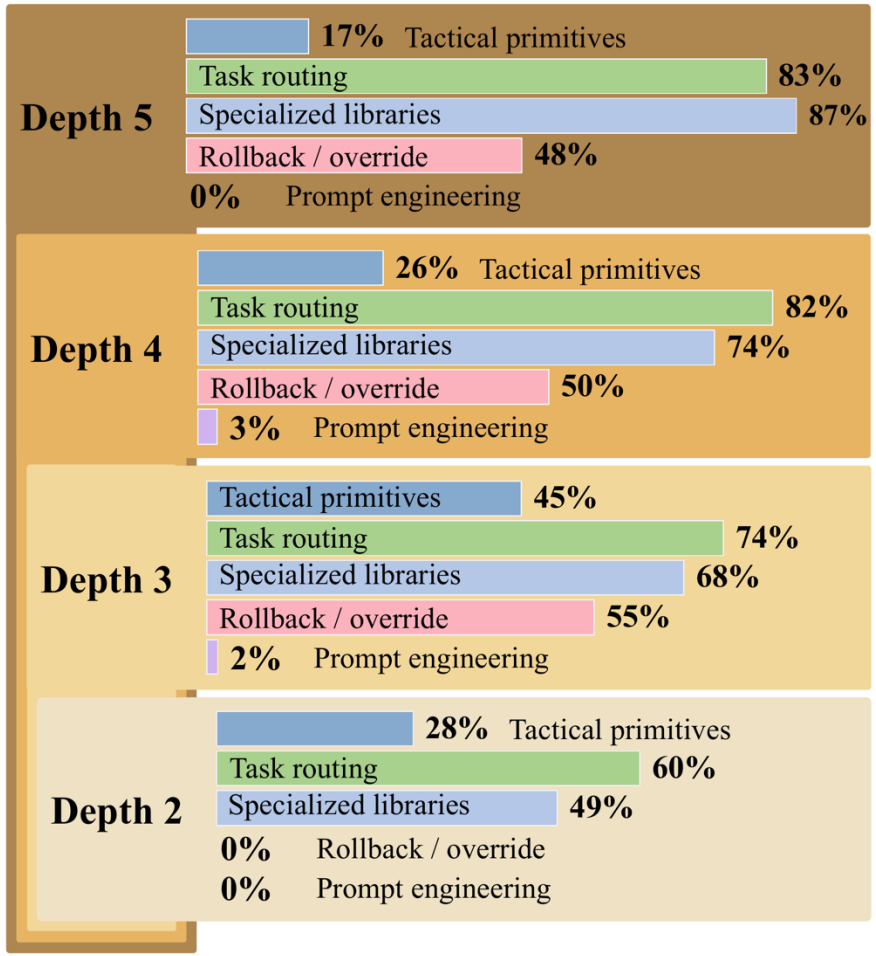
Depth 4

0.833 +0.060

rolls back, keeps the helper

Nobody told it to roll back. It read the code and found the cause.

Roles emerge that nobody prompted



0% to 55%

Rollback / override
Absent at depth 2, in over half of emissions at depth 3.

45% to 17%

Tactical primitives
Peak at depth 3, then fade by depth 5.

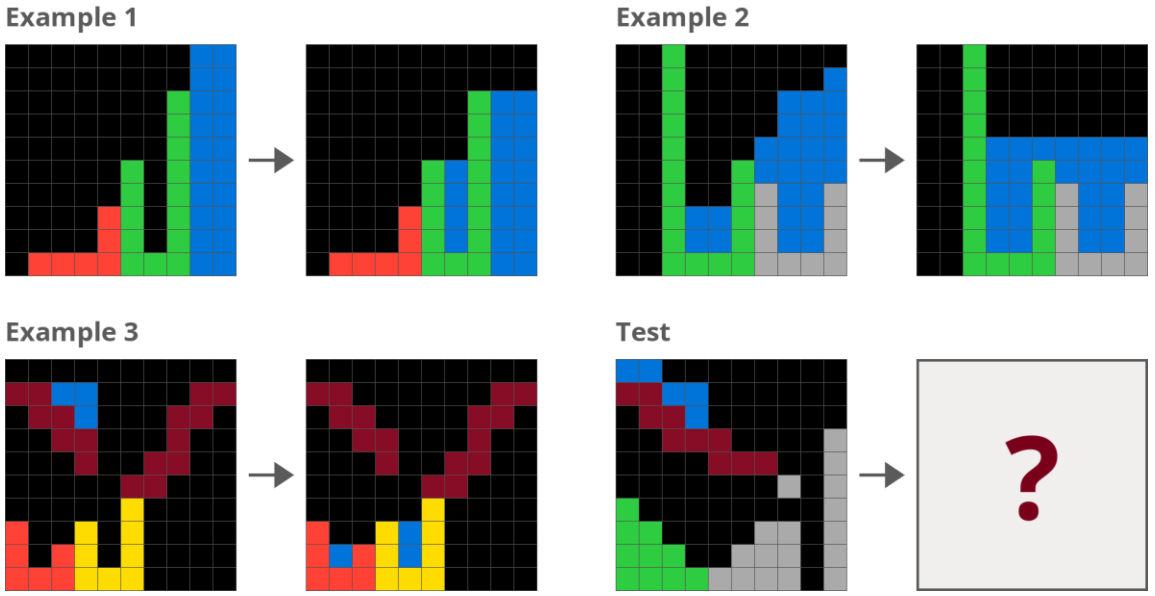
49% to 87%

Specialized libraries
Keep growing from depth 2 to depth 5.

288 Ω emissions on CO-Bench, AlphaEvolve Math and SR. Bars count labels on which both LLM raters agree.

Same operation, different place in the stack, different job.

Same model, six times the best baseline

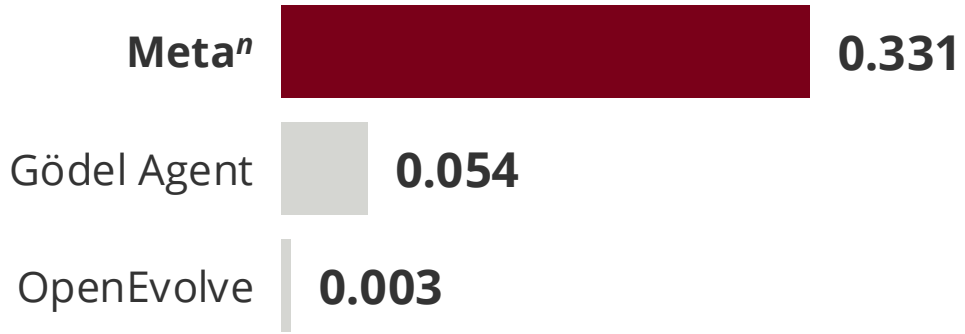


An ARC-AGI-2 task (public evaluation set, 28a6681f). Infer the rule from three examples, then fill in the test grid. The benchmark is built to resist skill memorization.

0.331

Meta^n on ARC-AGI-2

GPT-5.2, dev score, archive-best



8 benchmark families, 2 backbones, ahead of every baseline we ran.

72% of the gain is one string of text



Share of the recursion gain on CO-Bench, Gemma 4 31B-IT

code library

recursion machinery

Open question

What happens when a stronger model sits on top of a weaker one?

Come find me at the poster.

Code

Zae's homepage