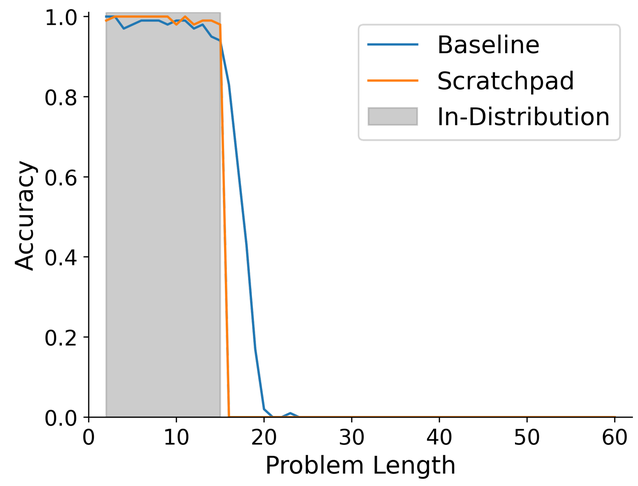
Code

Re-Tuning: Overcoming the Compositionality Limits of Large Language Models with Recursive Tuning

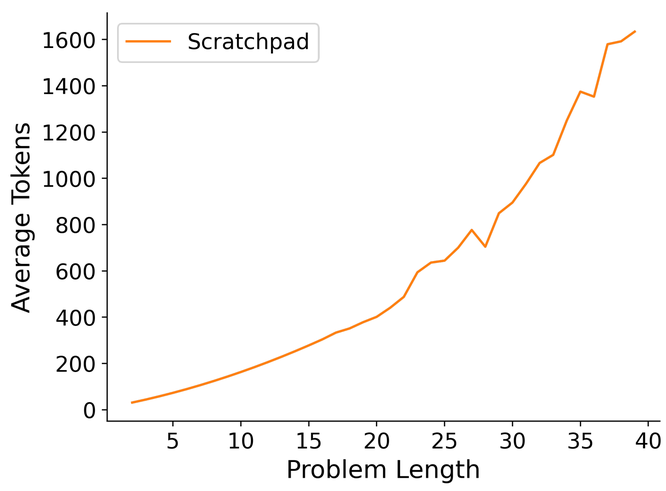
Eric Pasewark, Kyle Montgomery, Kefei Duan, Dawn Song, Chenguang Wang
Washington University in St. Louis & UC Berkeley

LLMs struggle to solve compositional tasks (e.g., integer addition), especially on problems longer than those seen during training.



Accuracy quickly falls to 0% on problems longer than those within the training distribution. The training distribution is represented by the grey-shaded region.

Challenge:
The number of tokens required to solve a compositional problem increases super-linearly with the problem length.



With existing methods, LLMs struggle to attend to the relevant context on longer sequences, resulting in poor performance. Moreover, LLMs have a fixed-sized context window. Extrapolating beyond this window results in lower accuracy.

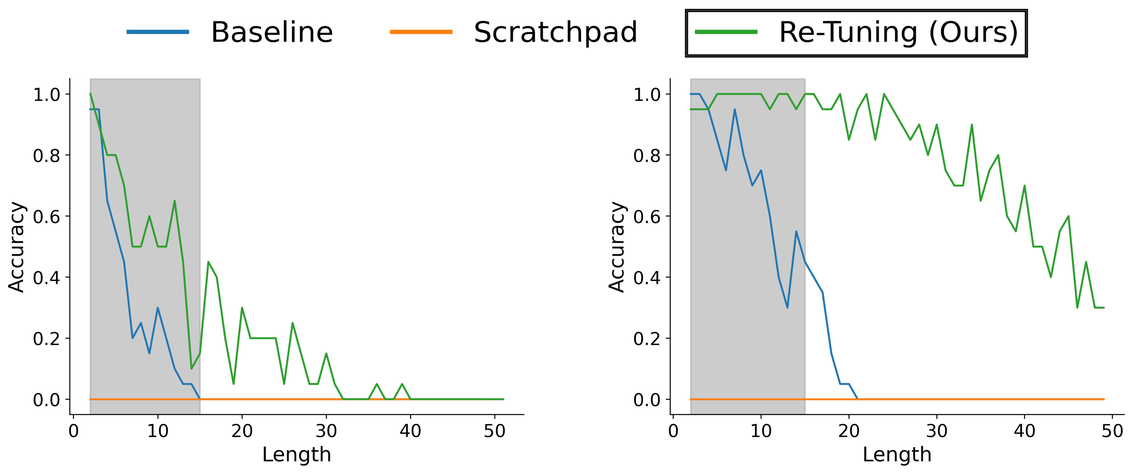
Re-Tuning directly leverages recursion during inference!

Algorithm 1 RecursiveGenerate Pseudocode

```
function RecursiveGenerate(model, tokenizer, prompt)
    context ← Generate(model, tokenizer, prompt)
    while ContainsUnexecutedCall(context) do
        call ← ExtractCall(context)
        result ← RecursiveGenerate(model, tokenizer, call)
        context ← context + result
        context ← Generate(model, tokenizer, context)
    end while
    return context
end function
```

During inference, Re-tuning involves (1) recursively decreasing the size of the problem, (2) solving the base case(s), and (3) passing the solutions up the recursive stack, solving subproblems of increasing complexity along the way. Each subproblem is solved in its own context, enabling the LLM to better attend to the relevant information.

Re-Tuning is significantly more sample-efficient than existing methods.



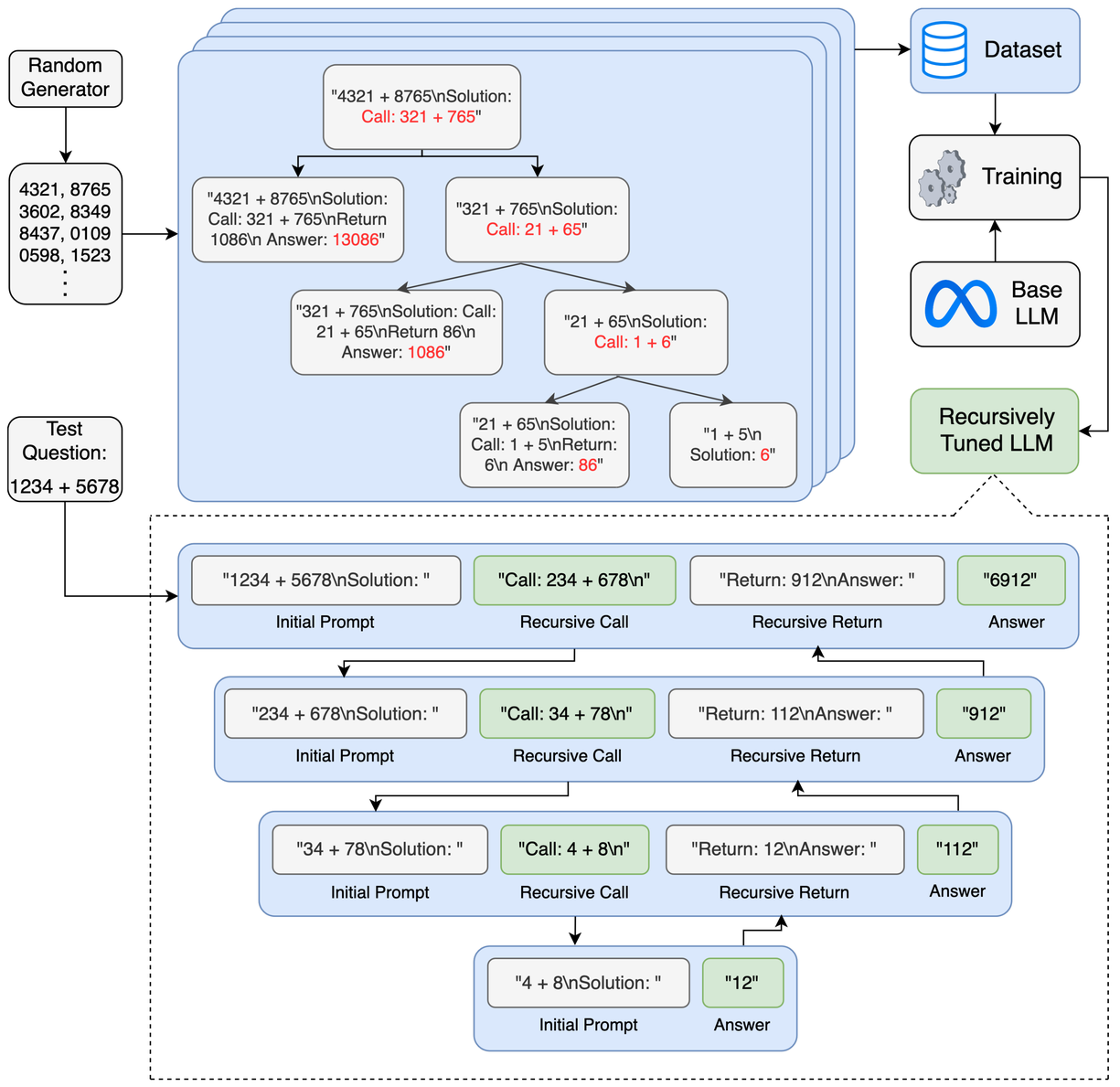
(a) 10 examples per problem length.

(b) 50 examples per problem length.

Re-Tuning requires significantly fewer training samples than existing methods to reach the same level of performance.

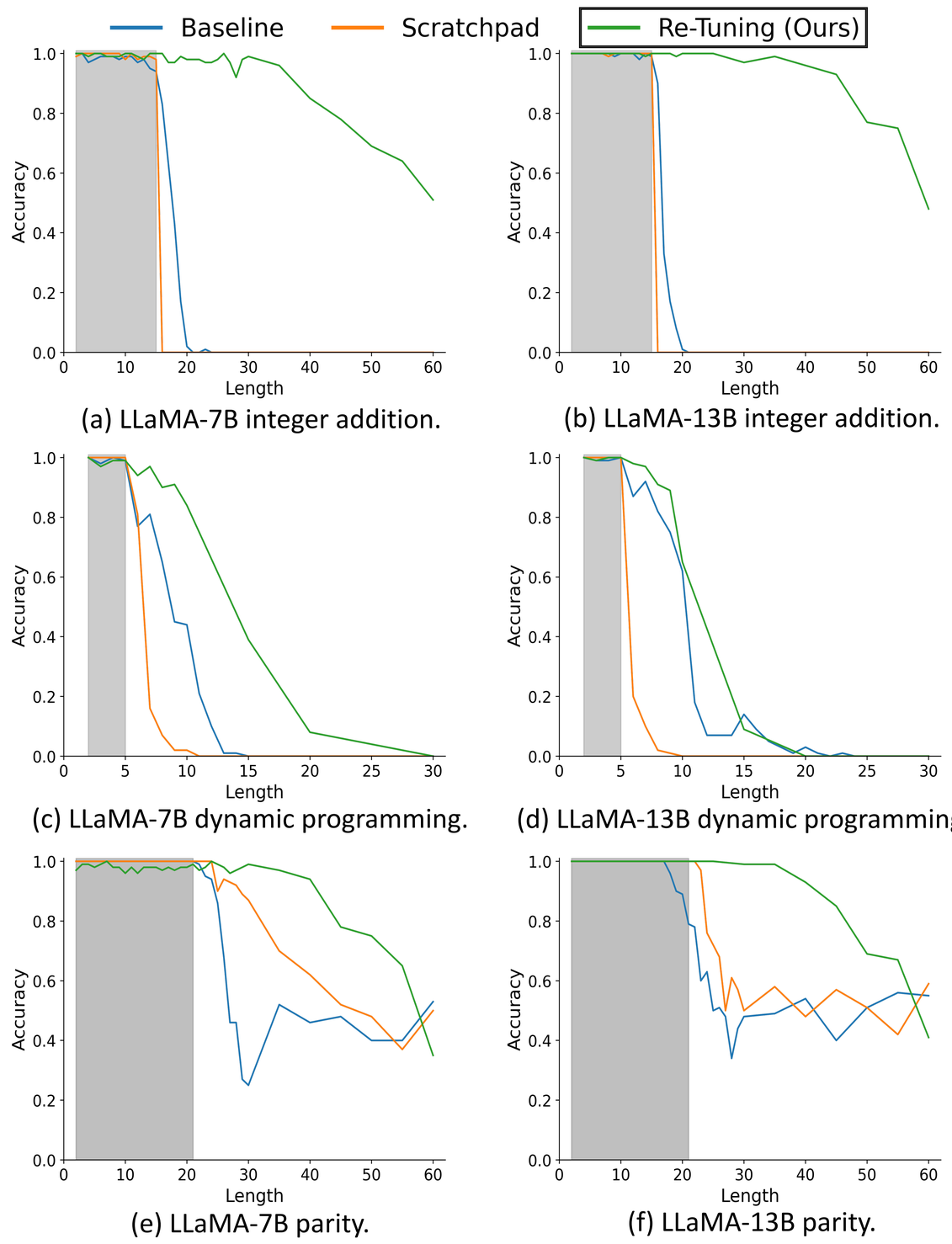
Our Method: Re-Tuning

We design a training and inference pipeline that exploits the recursive nature of compositional problems to improve accuracy and efficiency.



In short, Re-Tuning tunes LLMs to leverage recursion to better solve compositional problems with a higher degree of accuracy.

Re-Tuning exhibits significantly better performance across three representative compositional tasks, especially on longer problems.



(a) LLaMA-7B integer addition.

(b) LLaMA-13B integer addition.

(c) LLaMA-7B dynamic programming.

(d) LLaMA-13B dynamic programming.

(e) LLaMA-7B parity.

(f) LLaMA-13B parity.

Notably, on average, Re-Tuning enables 30%-40% accuracy improvements over existing methods on LLaMA-7B and LLaMA-13B across three representative compositional problems.