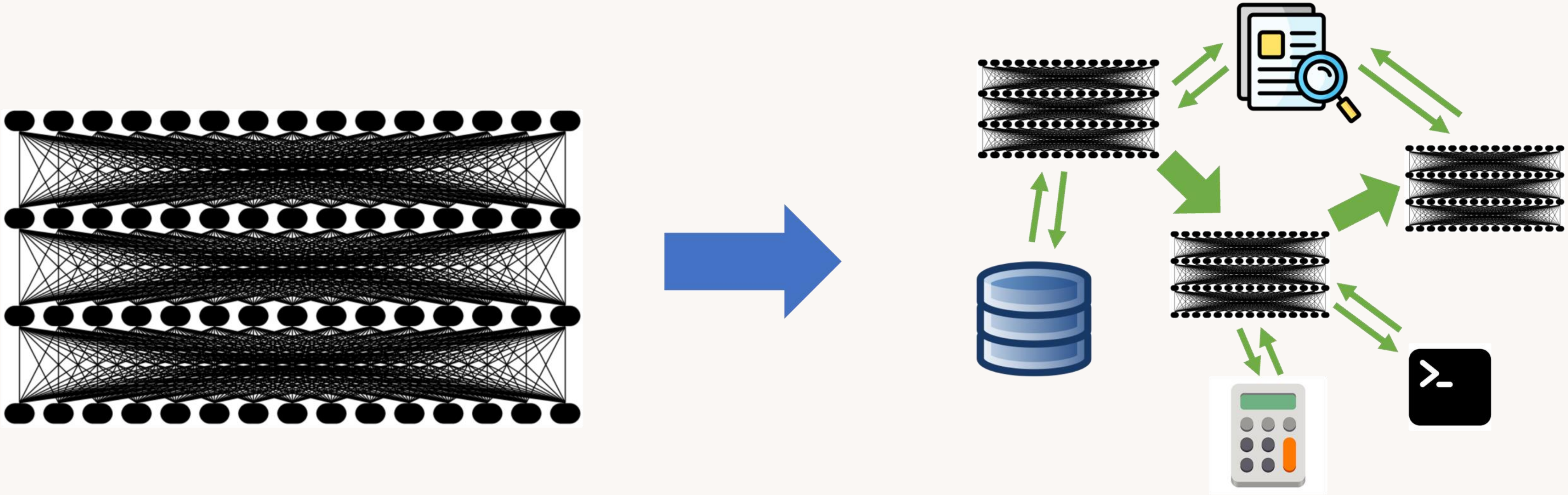
Are More LM Calls All You Need? Towards the Scaling Properties of Compound AI Systems

Lingjiao Chen

Joint work with

Jared Davis, Boris Hanin, Peter Bailis, Ion Stoica, Matei Zaharia, James Zou

Increasingly More AI Results by Compound Systems



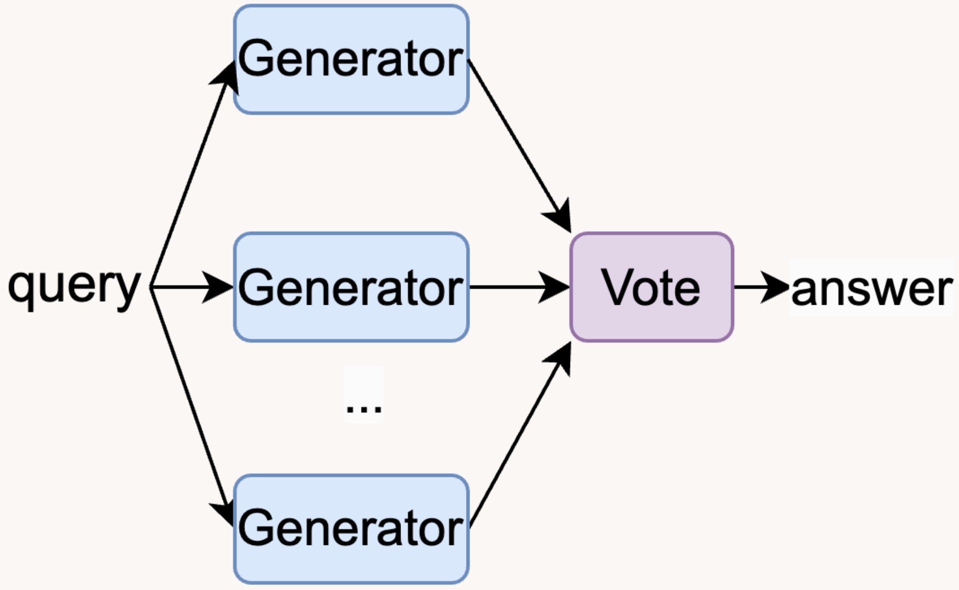
monolithic models

compound AI systems

- Key shift: From 1 model call to many model calls
- Examples: AlphaCode 2, AlphaGeometry, Gemini’s CoT@32, MedPrompt, …

But How Do Compound AI Systems Scale?

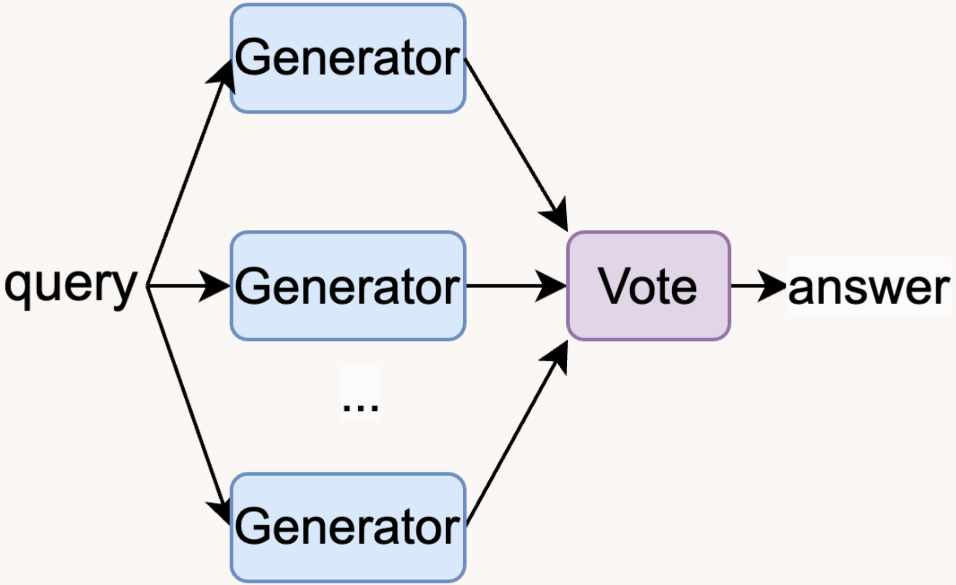
Our Focus : Two Compound AI Systems (1/2)



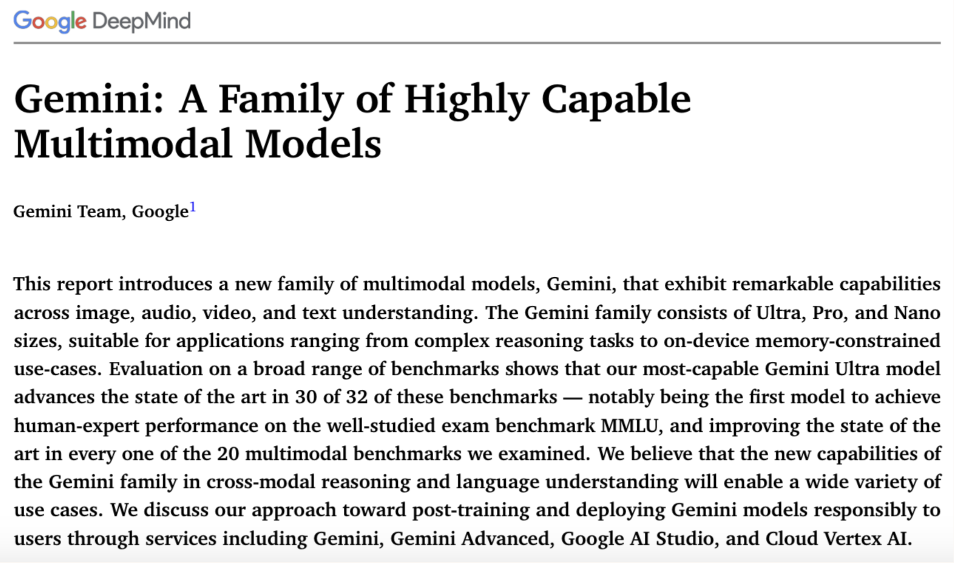
Vote

➢ Easy to understand and thus commonly used

Our Focus : Two Compound AI Systems (1/2)



Vote

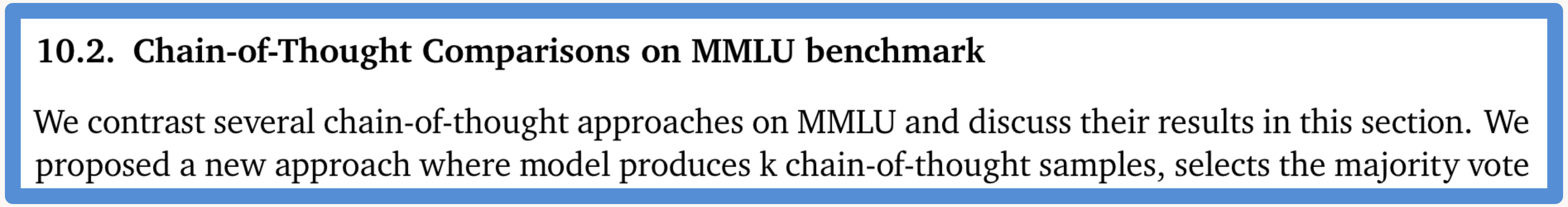


Google DeepMind

Gemini: A Family of Highly Capable Multimodal Models

Gemini Team, Google[1]

This report introduces a new family of multimodal models, Gemini, that exhibit remarkable capabilities across image, audio, video, and text understanding. The Gemini family consists of Ultra, Pro, and Nano sizes, suitable for applications ranging from complex reasoning tasks to on-device memory-constrained use-cases. Evaluation on a broad range of benchmarks shows that our most-capable Gemini Ultra model advances the state of the art in 30 of 32 of these benchmarks — notably being the first model to achieve human-expert performance on the well-studied exam benchmark MMLU, and improving the state of the art in every one of the 20 multimodal benchmarks we examined. We believe that the new capabilities of the Gemini family in cross-modal reasoning and language understanding will enable a wide variety of use cases. We discuss our approach toward post-training and deploying Gemini models responsibly to users through services including Gemini, Gemini Advanced, Google AI Studio, and Cloud Vertex AI.

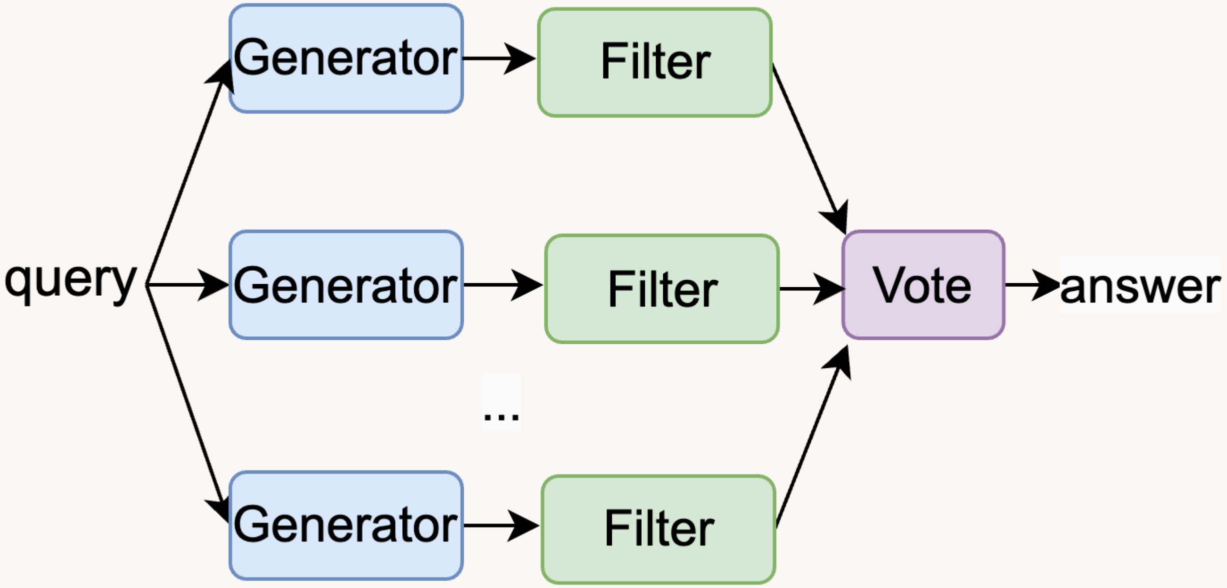


10.2. Chain-of-Thought Comparisons on MMLU benchmark

We contrast several chain-of-thought approaches on MMLU and discuss their results in this section. We proposed a new approach where model produces k chain-of-thought samples, selects the majority vote

- ➢ Easy to understand and thus commonly used
- ➢ Example: Google Gemini's CoT@32 strategy (slightly more complex)

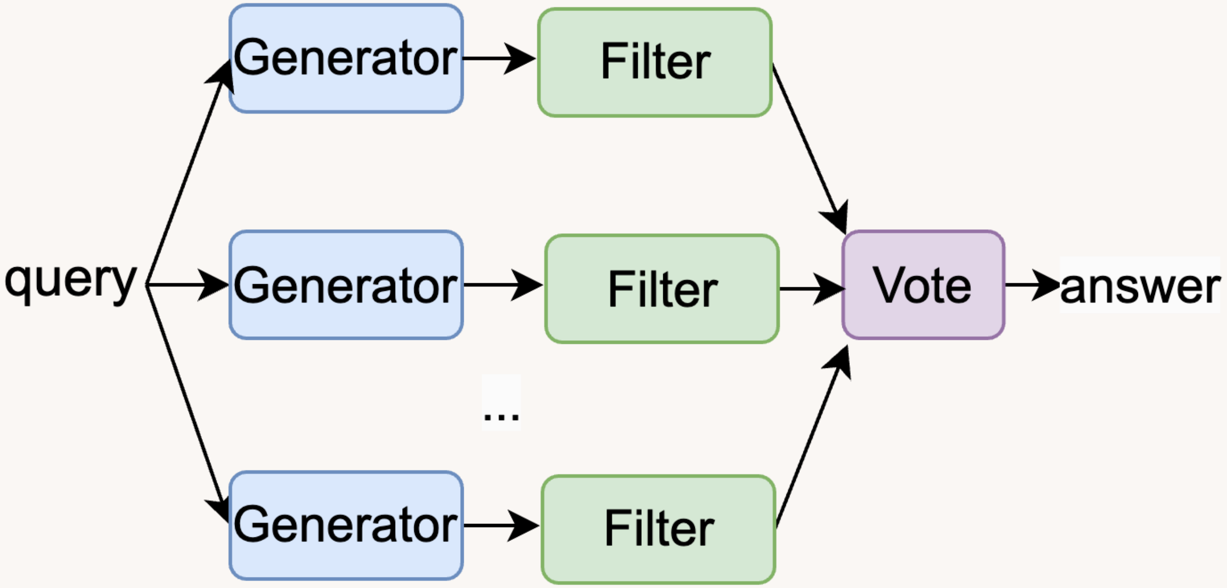
Our Focus : Two Compound AI Systems (2/2)



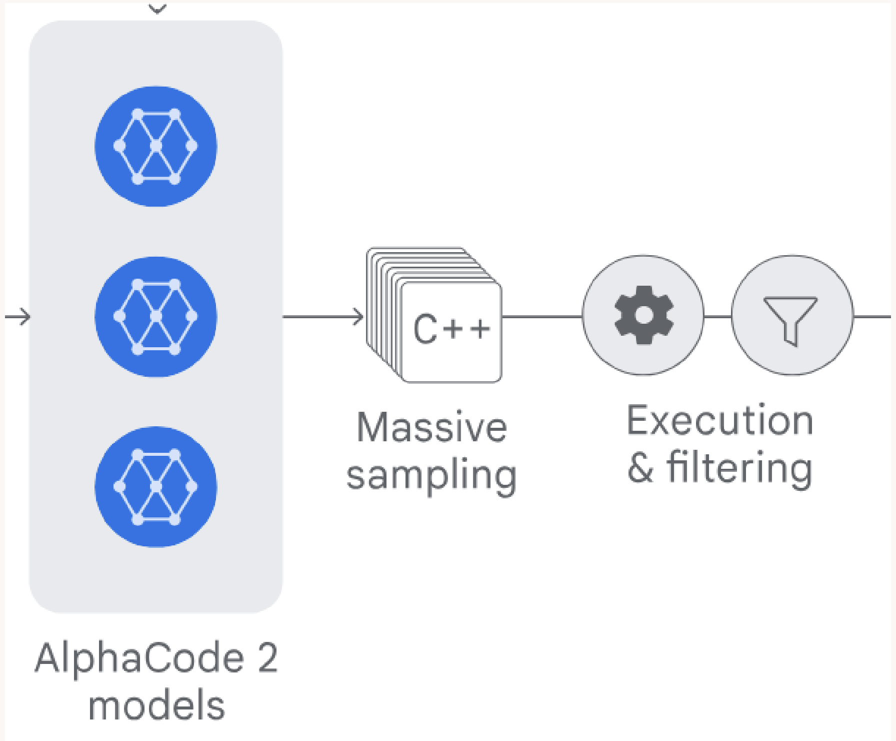
Filter-Vote

- ➢ Easy to understand and thus commonly used

Our Focus : Two Compound AI Systems (2/2)

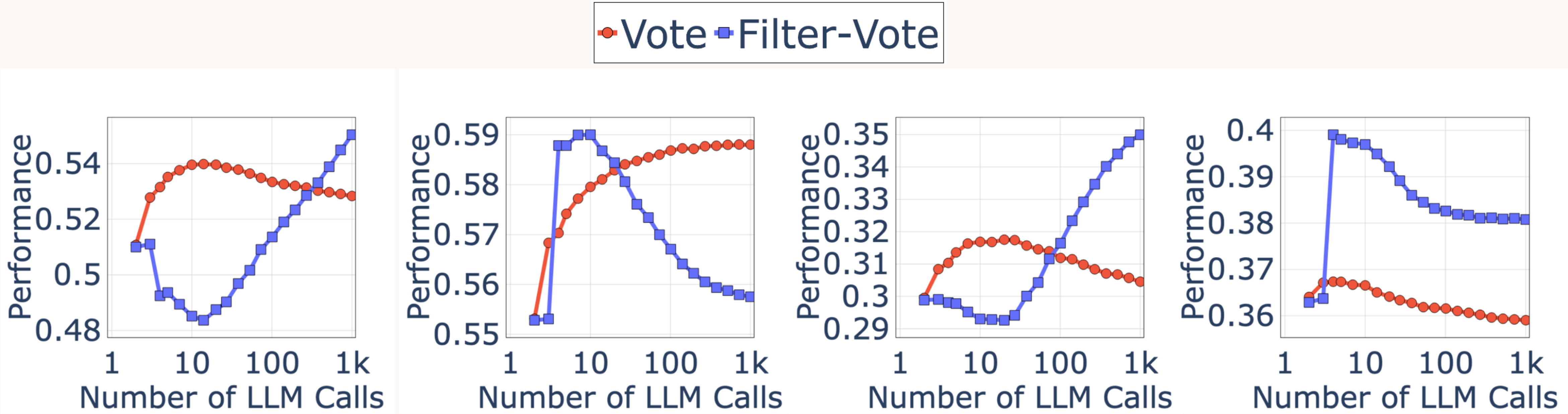


Filter-Vote



- ➢ Easy to understand and thus commonly used
- ➢ Example: AlphaCode 2 for code generation (slightly more complex)

Our Finding: The Non-monotonic Behavior

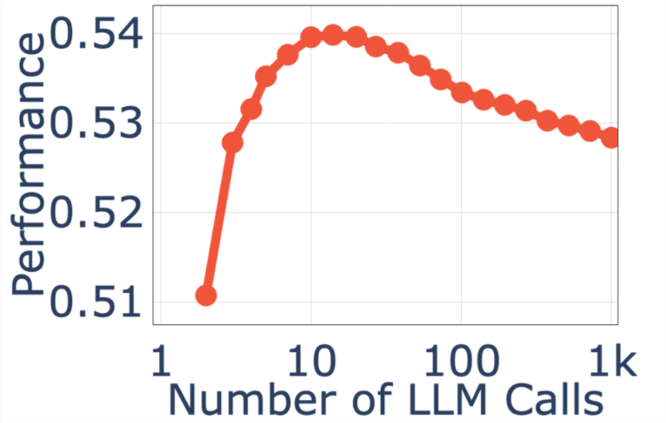


MMLU PHYSICS TRUTHFULQA GPQA AVERITEC

- ➢ As More LLM calls are invoked, the performance can
- ➢ (i) increase then decrease, or (ii) decrease and then increase (!)

Why Does the Non-monotonic Behavior Occur?

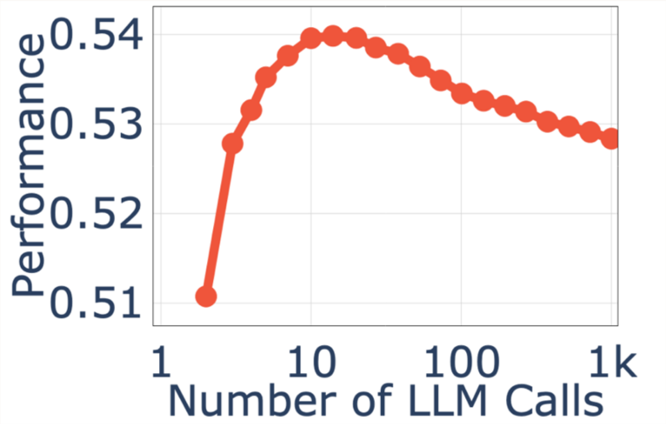
Our Analysis: Query Difficulty-based Explanation



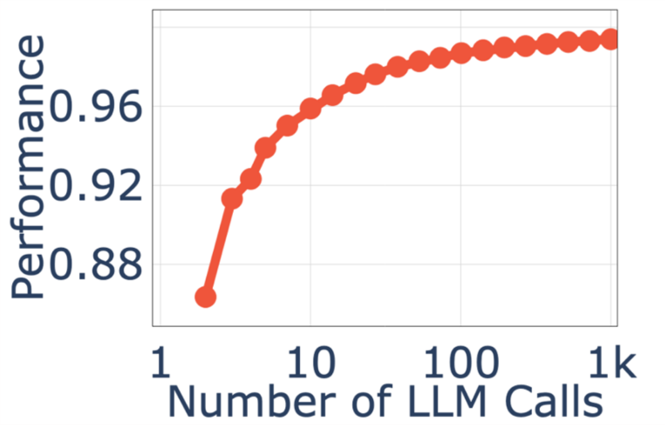
Overall (100%)

Strategy	Vote
Task	MMLU PHYSICS

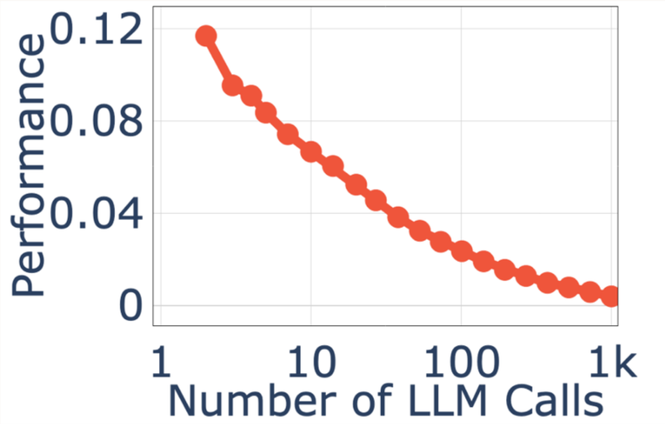
Our Analysis: Query Difficulty-based Explanation



Overall (100%)



Easy (53%)

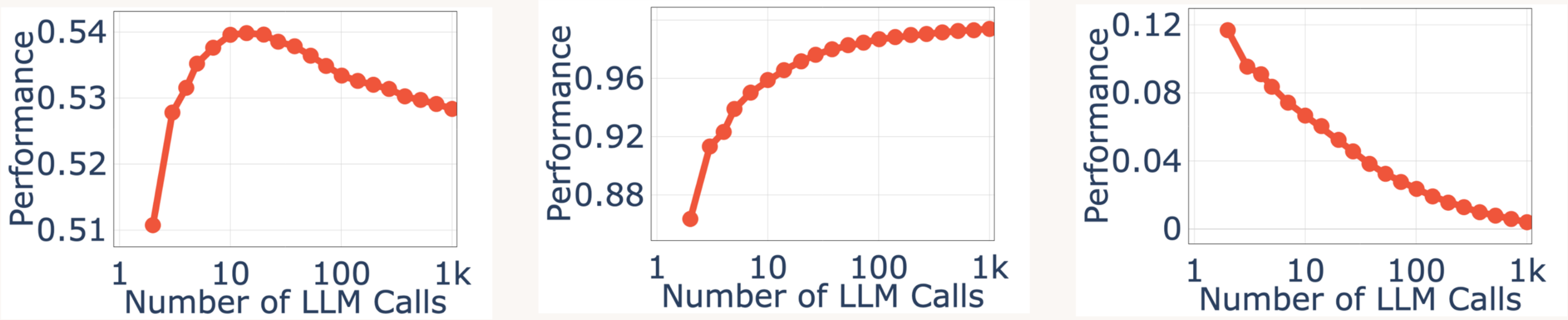


Difficult (47%)

Performance breakdown

Strategy	Vote
Task	MMLU PHYSICS

Our Analysis: Query Difficulty-based Explanation



Overall (100%)

Easy (53%)

Difficult (47%)

Performance breakdown

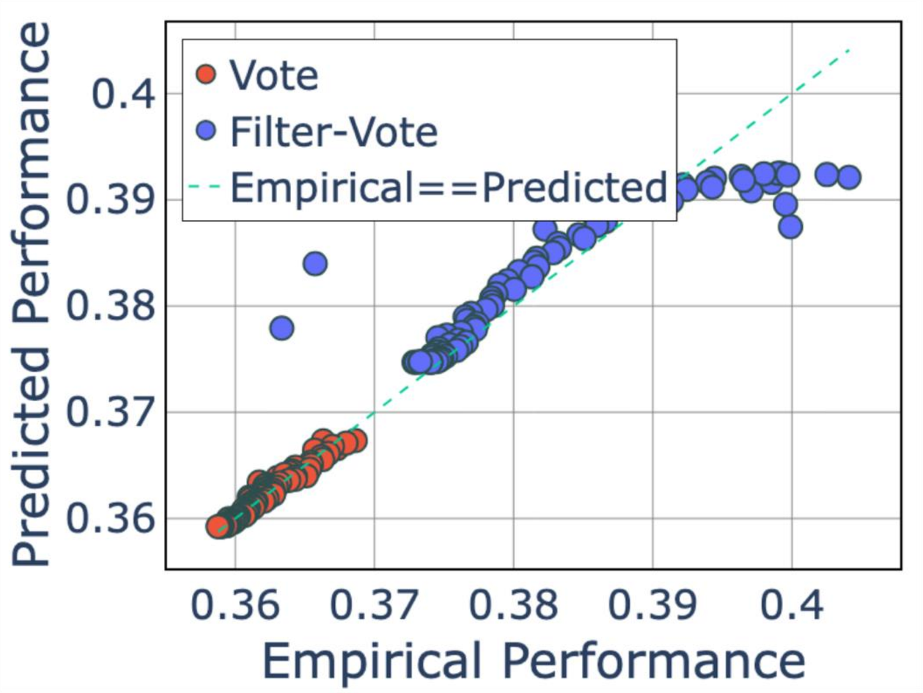
Strategy	Vote
Task	MMLU PHYSICS

➢ More LLM calls: better on easy queries, but worse on difficult queries!

More in Our Paper

- Difficulty-based explanation
  - The formal notion
  - A rigorous analysis
  - Concrete examples

- How to predict the scaling properties



Takeaway Message

Perf of Vote and Filter-Vote is non-monotonic in # LLM calls

The diversity of query difficulty explains this

Heuristics can predict the optimal # of LLM calls

Link to project website