**Reasoning with o1**

o1 is the new series of reasoning model

* Uses “chain of thought” to explore all possible paths and verify its answers.
* Requires less context and prompting in order to produce comprehensive and thoughtful outputs.

O1 – Model for complex tasks that require broad general knowledge

O1-mini – A faster reasoning model tailored to coding, math and science.

Completion Tokens

Can now be broken into 2 distinct categories:

* Reasoning tokens
* Output tokens

Reasoning tokens are not passed from one turn to the next.

A diagram of a process

Description automatically generated

Need to consider reasoning tokens in the cost, and calculation of context limit.

Two key findings that lead to o1

* The more reinforcement learning done in post-training process, the more accurate the model got. The more we allow the model to think at inference time, the sharper increase is the accuracy.
* Teach the model to verify via Consensus/Majority voting.
  + Generate a bunch of solutions and ask the LLM to choose the most common one. Can be thought of as like sampling at low temperature
  + Consensus flatlines before 100 samples, hence doesn’t need a huge amount of samples to realize the performance improvement.

Benchmarks of o1

A graph of different colored bars

Description automatically generated with medium confidence

A group of green and orange bars

Description automatically generated

How does o1 works?

* It uses large-scale RL to generate a chain of thought (CoT) [Wei et al. NeurIPS-2022] before answering
* CoT is longer and high-quality than what is attained via prompting
* CoT contains behavior like:
  + Error correction
  + Trying multiple strategies
  + Breaking down problems into smaller steps
* Example CoTs on the research blog post: <http://openai.com/index/learning-to-reason-with-llms>

Generator-Verifier Gap

* For some problems, verifying a good solution is easier than generating one
  + Many puzzles, such as Sudoku
  + Math
  + Programming
* Examples where verification isn’t much easier
  + Information retrieval
  + Image recognition
* When a generator-verifier gap exists and we have a good verifier, we can **spend more compute on inference to achieve better performance**.

Where might o1 be used?

* Data Analysis: Interpreting complex datasets (genome sequencing results in biology) and performing advanced statistical reasoning.
* Mathematical Problem-solving: Deriving solutions or proofs for challenging mathematical questions or in physics theory.
* Experimental design: Proposing experimental setups in chemistry to test novel reactions or interpreting complicated physics experiments’ outcomes
* Scientific Coding: Writing and debugging specialized code for computational fluid dynamics models or astrophysics simulations
* Biological and chemical reasoning: Solving advanced biology or chemistry questions that require deep domain knowledge
* Algorithm Development: Aiding in creating or optimizing algorithms for data analysis workflows in computational neuroscience or bioinformatics.
* Literature Synthesis: Reasoning across multiple research papers to form coherent conclusions in interdisciplinary fields such as systems biology.

**Prompting o1**

1. Simple & Direct

Write prompts that are straightforward and concise. Direct instructions yield the best results with the o1 models.

1. No explicit CoT required

You can skip step-by-step (CoT) reasoning prompts. The o1 models can infer and execute these itself without detailed breakdowns.

1. Structure

Break complex prompts into sections using delimiters like markdown, XML, tags, or quotes.

This structured format enhances model accuracy – and simplifies your own troubleshooting.

1. Show rather than tell

Rather than using excessive explanation, give a contextual example to give the model understanding of the broad domain of your task.