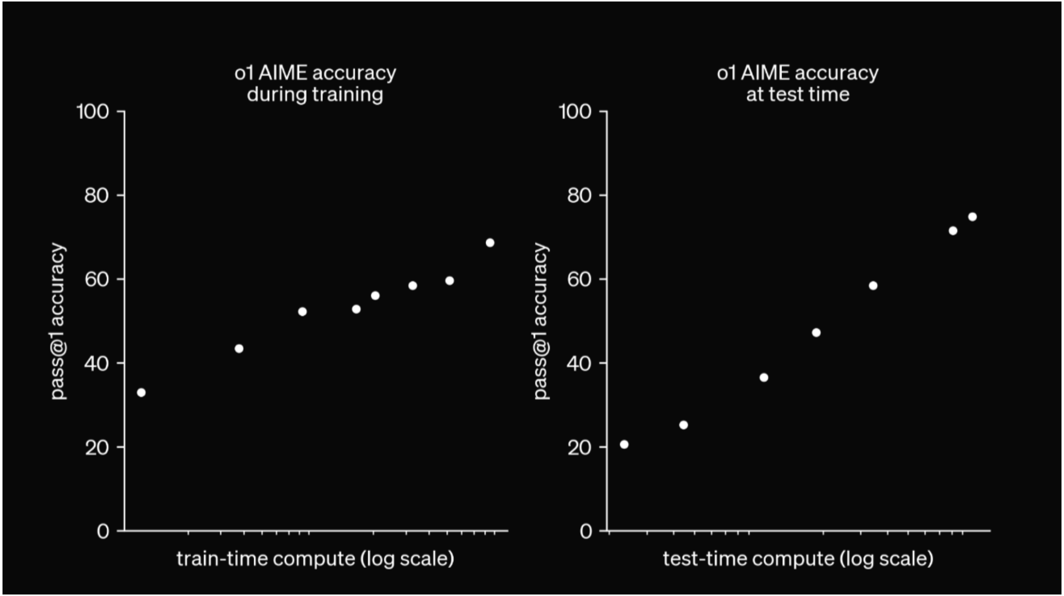
Speculations on Test-Time Scaling

Sasha Rush Daniel Ritter

Cornell

Outline


o1 AIME accuracy
during training
pass@1 accuracy
100
80
60
40
20
0
train-time compute (log scale)
o1 AIME accuracy
at test time
pass@1 accuracy
100
80
60
40
20
0
test-time compute (log scale)

Context

- LLM (2018-2024) driven by training scaling
- Speculation: Benefit of static data running out

Implication

- Breakthrough in large-scale RL Training
-

What have we seen?

- Public demo model
- Strong result in constrained domains.

This Talk

- Survey of the public literature
- Synthesis of discussions with expert
- Gossip and hearsay

Thanks

Lewis Tunstall, Edward Beeching, Aviral Kumar, Charlie Snell, Michael Hassid, Yoav Artzi, Risab Agarwal, Kanishk Gandhi, Wenting Zhao, Yuntian Deng, Nathan Lambert

What we know

Our large-scale **reinforcement learning algorithm** teaches the model how to think productively using its **chain of thought** in a highly **data-efficient** training process.

What we know

- RL - Signal from verifiable problems
- CoT - “Thinking” occurs in token stream
- Data Efficient - Fixed set of good problems

From Gossip

- Single final model
- Not learned from expert examples
-

Chain of Thought

o1 learns to hone its chain of thought and refine the strategies it uses. It learns to recognize and **correct its mistakes**. It learns to **break down tricky steps** into simpler ones. It learns to try a **different approach** when the current one isn't working.

Review: Chain of Thought

-

Planning

Backtracking

Strategies

Summary

- Solves problems by very long CoT
- CoT includes “thinking” (search / planning)
- Core novelty: Inducing this behavior

Notation - Test-Time (No learning yet!)

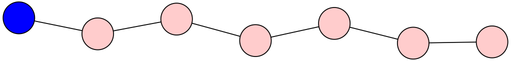
- x - the problem specification
- $z \in \mathcal{S}^T$ - the chain of thought (CoT)
- $y \in \mathcal{Y}$ - the final answer
- $p(y|x) = \mathbb{E}_{z \sim p(z|x)} p(y|x, z)$ - model

Warm-up: Ancestral Sampling

- [?](z | x)

- [?](y | x, z=)

$|\tilde{z}|$ amount of test-time compute



Warm-up: Monte-Carlo (Self-Consistency)

- [?](z | x)

- [?](y | x,)

Pick $y = \tilde{y}$

Warm-up: Beam Search

- [?](z | x)

- [?](y | x,)

z $\in$S^T - the chain of thought

y - the response

$p(y|x) = E_{z \sim p(x|z)} p(y|x, z)$ - model

Warm up: Rejection Sampling

- [?](z | x)
- [?](y | x,)

Warm up: Monte-Carlo Roll-Outs

Start at z

- [?](z | x, z)

- [?](y | x,)

Goal: Learning

- $\max_{theta} \sum \log p(y|x;\theta)$

- Intractable expectation over latent CoT

Outline

Introduction

The Clues

Guess and Check

Guided Search

Full AlphaZero

Learning to Search

Something Wild

The Suspects

- Guess + Check
- Guided Search
- AlphaZero
- Learn to Search
- Wildcard

A Note About Names

- Many different communities
- Names conflict and overlap with past methods
- This talk: First explain, then discuss names

Offline / Online?

- Each approach has two variants
- I will describe offline/batch variant
- Companies have complex internal RL optimizers to make online variant works

Informal: Guess + Check

- Sample N CoTs
- Check if successful
- Train on good ones

Formalization: EM

- $$\max_{theta} \sum \log p(y|x;\theta) = \sum \log E_z p(y,z|x)$$

- E-Step: Compute $p(z|y,x) \propto (Ver(y))p(z|x)$

- M-Step: Fit $p(y,z|x)$

  Hard EM

Offline

- Batch servers to sample
- Check if successful
- Train on good ones

Online

- Sample N CoTs
- Check if successful
- Train on good ones

Terminology

- STaR
- ReST
- ReST-EM
- Filtered Rejection Sampling
- Best-of-N

Why might this be right?

- Extremely simple and scalable
- Good baseline in past work

Why might this be wrong?

- No evidence this learns to correct, plan
- Well-explored in literature with marginal gains

Alternative

- Can we improve upon the process of finding adequate CoTs?

Informal: Guided Search

- Sample several next steps for CoT
- Check with a guide model for which to pursue
- Continue to the end
- Train on good ones

Warm-up: Beam Search with Roll-Outs

- [?](z | x)

- $\tilde{y}$[?]$(y|x, \tilde{z})$

Where does the guide come from?

PRM/RollOuts

- Point 1
- Point 2
- Point 3

Why might this be right?

- Major demonstrated RL result
-

Why might this be wrong?

- Does not inject into CoT
- Relatively complex to scale

Alternative

- Can we improve on the search?

Reminder: AlphaZero

Informal: AlphaZero

- Search for best solution with model
- Collect the best CoT
- Train on good ones

Formalized: Expert Iteration

-

Why might this be right?

- Major demonstrated RL result
-

Why might this be wrong?

- Does not inject into CoT
- Relatively complex to scale

Alternative

- Can we get the CoT to search?

Challenge

Informal: Learning to Correct

- Find optimal paths
- Adjust to add mistakes
-

Formalized: Stream of Search

-
-
-

Formalized: Advantage

-
-
-

Why might this be right?

-
-

Why might this be wrong?

-
-

No Supervision

-

MuZero

-

Reference I

[Brown et al., 2024] Brown, B., Juravsky, J., Ehrlich, R., Clark, R., Le, Q. V., Ré, C., and Mirhoseini, A. (2024).

Large language monkeys: Scaling inference compute with repeated sampling.

arXiv [cs.LG].

[Gandhi et al., 2024] Gandhi, K., Lee, D., Grand, G., Liu, M., Cheng, W., Sharma, A., and Goodman, N. D. (2024).

Stream of search (SoS): Learning to search in language.

arXiv [cs.LG].

Reference II

[Silver et al., 2017] Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., and Hassabis, D. (2017).

Mastering chess and shogi by self-play with a general reinforcement learning algorithm.

arXiv [cs.AI].

Reference III

[Uesato et al., 2022] Uesato, J., Kushman, N., Kumar, R., Song, F., Siegel, N., Wang, L., Creswell, A., Irving, G., and Higgins, I. (2022).

Solving math word problems with process- and outcome-based feedback.

arXiv [cs.LG].