ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

Prof. R. Wattenhofer

Scaling laws for test-time compute

In collaboration with SID.ai

For pretraining of Large Language Models (LLMs) the "Kaplan" [3] & "Chinchilla" [2] scaling laws were breakthroughs showing that we could precisely predict the test loss of an LLM with a simple power law relation after scaling up the number of parameters and dataset size through multiple orders of magnitude, as well as the optimal way to scale these values given fixed pretraining compute (≈ dataset size × model size).

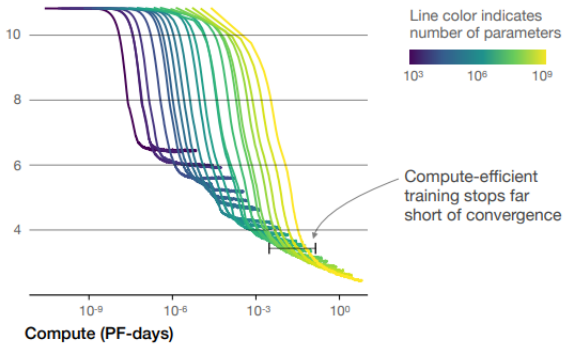

Figure 2 of Kaplan et al. [3]. y-axis is test loss. Each line is a separate LLM with similar architecture but parameter counts stretching across 6 OOMs.

With the post-training breakthroughs of "reasoning" models such as OpenAI o-series and DeepSeek-r1 [1], a natural question emerges: What are the scaling laws for this post-training regime? So far, work in this area [4, 6] has focused on the best-of-N approaches that were the preferred way to scale inference compute before reasoning post-training, as well as the initial analyses of performance as related to the test-time compute spend of a single model (see for example Figure 1 of Muennighoff et al. [5]).

Roughly, the cost of generating a reasoning chain-of-thought will be:

$$\text{cost} \approx \text{model size} \times \text{chain-of-thought length} + \alpha \text{ model size} \times (\text{chain-of-thought length})^2,$$

for some small α, due to the quadratic attention cost. There is some evidence ([7], Figure 12) that better base models use shorter chains of thought, thus presenting a tradeoff between model size and chain-of-thought length for a given performance threshold.

In this project, we will try to do a fine-grained analysis of how to trade off the model size & chain-of-thought length in various reasoning models. Doing so will require innovating to precisely control the chain-of-thought length and develop a measure of question hardness (if such a measure exists) that we can use to aggregate results.

Requirements: Strong programming skills & knowledge of RL. Weekly meetings will be scheduled to address questions, discuss progress, and brainstorm future ideas.

Interested? Contact:

- Sam Dauncey : sdauncey@ethz.ch, ETZ G61.1
- Maximilian-David Rumpf : max@sid.ai

Please attach a CV and transcripts.

References

[1] DeepSeek-AI et al. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. 2025. arXiv: 2501.12948 [cs.CL].

[2] Jordan Hoffmann et al. An empirical analysis of compute-optimal large language model training. In: Advances in Neural Information Processing Systems. Ed. by S. Koyejo et al. Vol. 35. Curran Associates, Inc., 2022, pp. 30016–30030.

[3] Jared Kaplan et al. Scaling Laws for Neural Language Models. 2020. arXiv: 2001.08361 [cs.LG].

[4] Noam Levi. A Simple Model of Inference Scaling Laws. In: *arXiv preprint arXiv:2410.16377* (2024).

[5] Niklas Muennighoff et al. s1: Simple test-time scaling. 2025. arXiv: 2501.19393 [cs.CL].

[6] Charlie Snell et al. Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters. 2024. arXiv: 2408.03314 [cs.LG].

[7] Weihao Zeng et al. SimpleRL-Zoo: Investigating and Taming Zero Reinforcement Learning for Open Base Models in the Wild. 2025. arXiv: 2503.18892 [cs.LG].

Project scaffold

Here is how we would approach the project. It is certainly not the best way, and if you see a better way, definitely tell us and change course!

Simple starter: Getting a handle on how performance interacts with model size and output length

- Take a dataset of problems (eg. MATH).
- Take a series of reasoning models from HuggingFace (eg. r1-distills) and evaluate on them drawing multiple samples per question (use quantization to squeeze the models onto our GPUs)
- Make a database of (prompt, completion, total context length, first solution position, model, compute cost)
- (?) Make an ELO-system to predict if a given model will get a given question correct.

Project: Using more fine-grained control of CoT length to get cleaner scaling laws

- Use RL and LoRA to adapt reasoning model to output in the following format (where the parts in square brackets [] and end of thought is forced):

  [prompt][you have n tokens]$\langle$ n tokens of CoT $\rangle$[The solution is:] $\langle$ answer $\rangle$
- See how the performance varies across multiple samples as k increases (this should look like the log-log plot on the $o1$ announcement or in $s1$ [5])
- Make a similar database as before and plot this for multiple model sizes and varying n for fixed question difficulty

Project Deliverables

We denote the following primary tasks mandatory (on the right side you find a rough estimate for the time that we allocate to the respective task):

- Literature research ($\star$)
- Write a report. ($\star\star$)
- Present your findings. ($\star$)

The Student's Duties

- One meeting per week with the advisors to discuss current matters.
- Regular check-ins into the provided *revision control system*.
- A final report in English, presenting work and results.
- A final presentation (15 min) of the work and results obtained in the project.