Optimizing LLM Test-Time Compute Involves Solving a Meta-RL Problem

AUTHORS

Amrith Setlur, Yuxiao Qu, Matthew Yang, Lunjun Zhang, Virginia Smith, Aviral Kumar

AFFILIATIONS

MLD CMU, University of Toronto

PUBLISHED

January 8, 2025

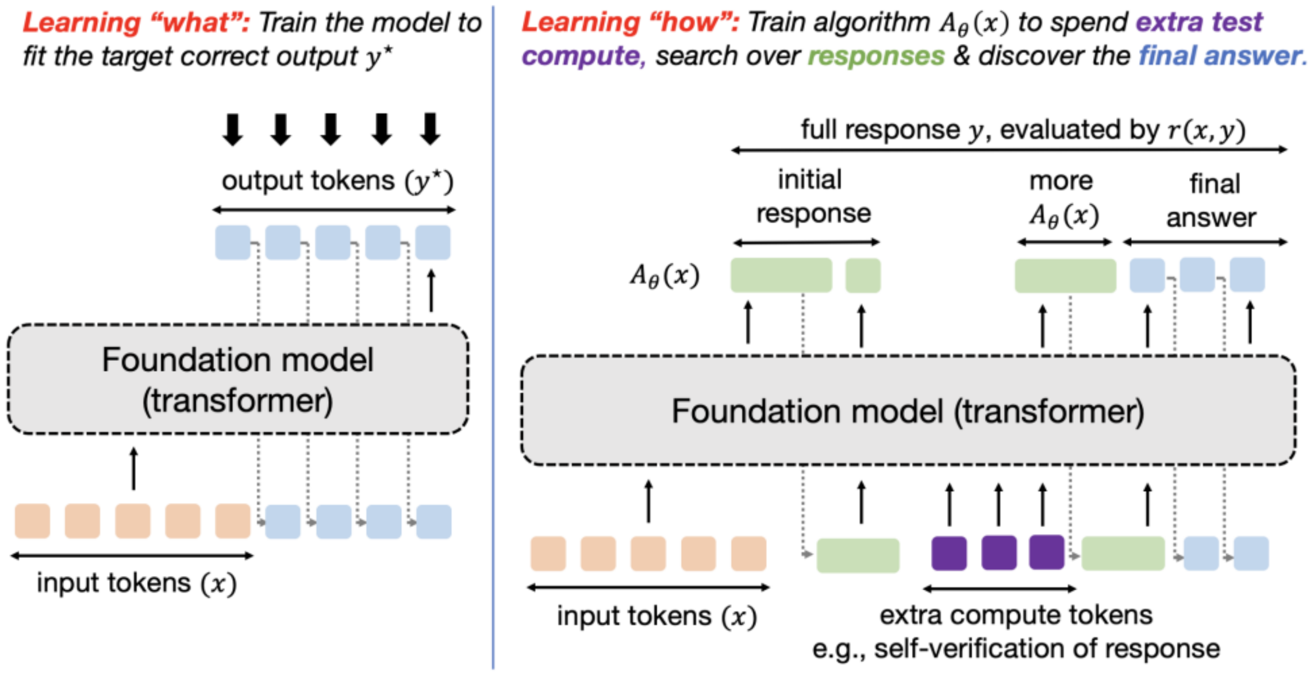
Learning "what": Train the model to fit the target correct output y^*
output tokens (y^*)
Foundation model (transformer)
input tokens (x)
Learning "how": Train algorithm $A_\theta(x)$ to spend extra test compute, search over responses & discover the final answer.
full response y, evaluated by $r(x, y)$
initial response
more $A_\theta(x)$
final answer
$A_\theta(x)$
Foundation model (transformer)
input tokens (x)
extra compute tokens
e.g., self-verification of response

***Figure 1**: Training models to optimize test-time compute and learn "how to discover" correct responses, as opposed to the traditional learning paradigm of learning "what answer" to output.*

The major strategy to improve large language models (LLMs) thus far has been to use more and more high-quality data for supervised fine-tuning (SFT) or reinforcement learning (RL). Unfortunately, it seems this form of scaling will soon hit a wall, with the scaling laws for pre-training plateauing, and with reports that high-quality text data for training maybe exhausted by 2028, particularly for more difficult tasks, like solving reasoning problems which seems to require scaling current data by about 100x to see