o3:
Generating data from multiple sampling for self-improvement + Path Ahead

Disclaimer: Contains some speculation about o3 architecture and training procedure

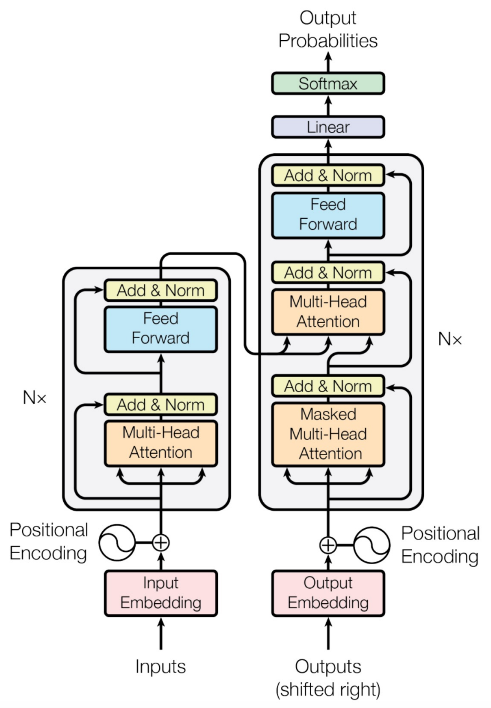
Presented by:
John Tan Chong Min
23/30 Dec 2024

Deep Learning is Data-Driven

- Deep learning systems, like Transformers, require data in order to be trained well
- In the past, deep learning was done mainly via **Supervised Learning**, that uses expert human labels to train a model to predict these labels
- Now, instead of using expert labels, we can use Web Data and do **Self-Supervised Learning** by predicting the next token given an existing sequence of tokens

Web Data is almost used up

Web Data



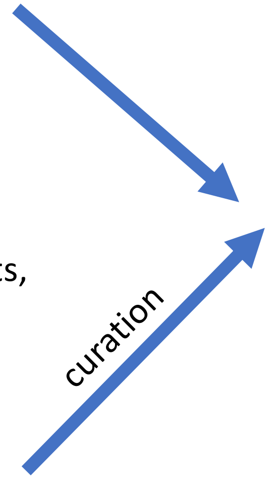
Transformer

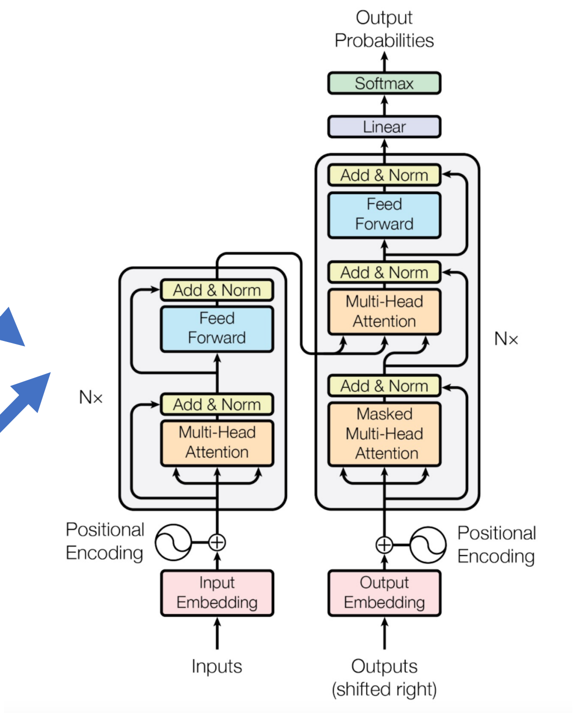
- Scrape web data
- Tokenise text
- Predict next token
  - The cat is on <token> -> the
  - The cat is on the <token> -> mat
- + Fine-tune on target data, human preferences

How to get more quality data?

Expert Data (e.g. PhD students, domain-specific data)

Web Data



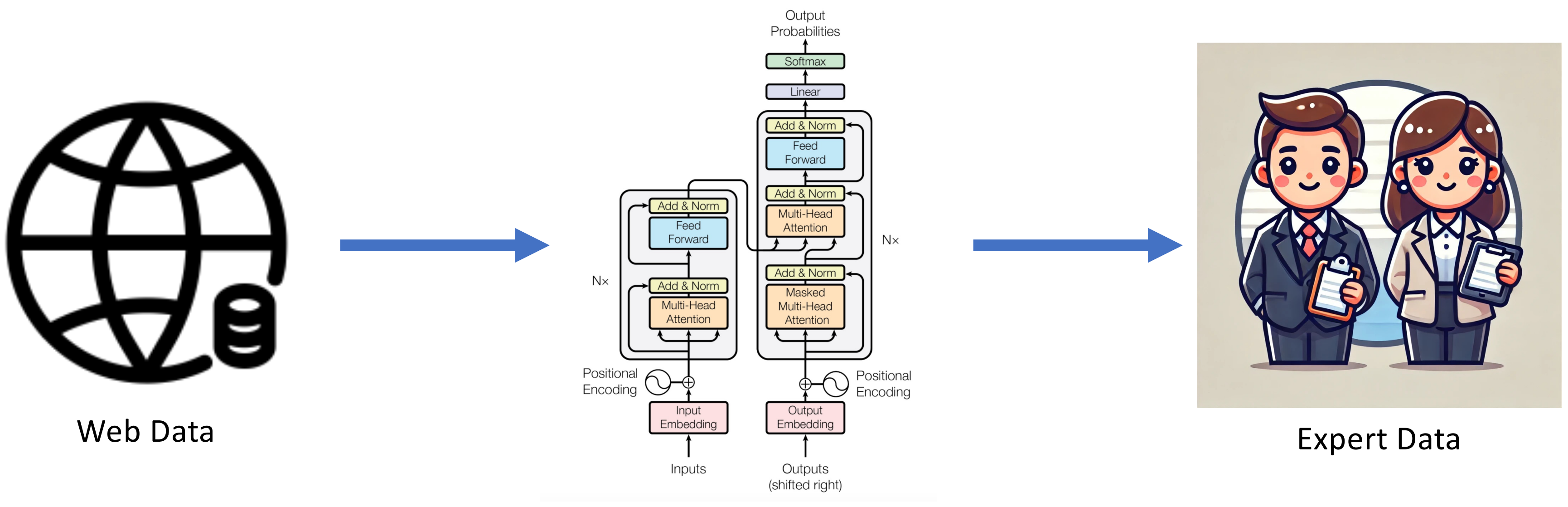


Transformer

- Data quality is very important
- Using data that is of high quality can help with better output quality, e.g. more curation of data in Llama 3 compared to Llama 2
- Get also get high quality data from expert humans
  - But time consuming!

Can the model generate its own data?

- Can we use existing models to generate high quality data?
  - E.g. TinyStories uses GPT-3.5/GPT-4 with prompting to generate target data



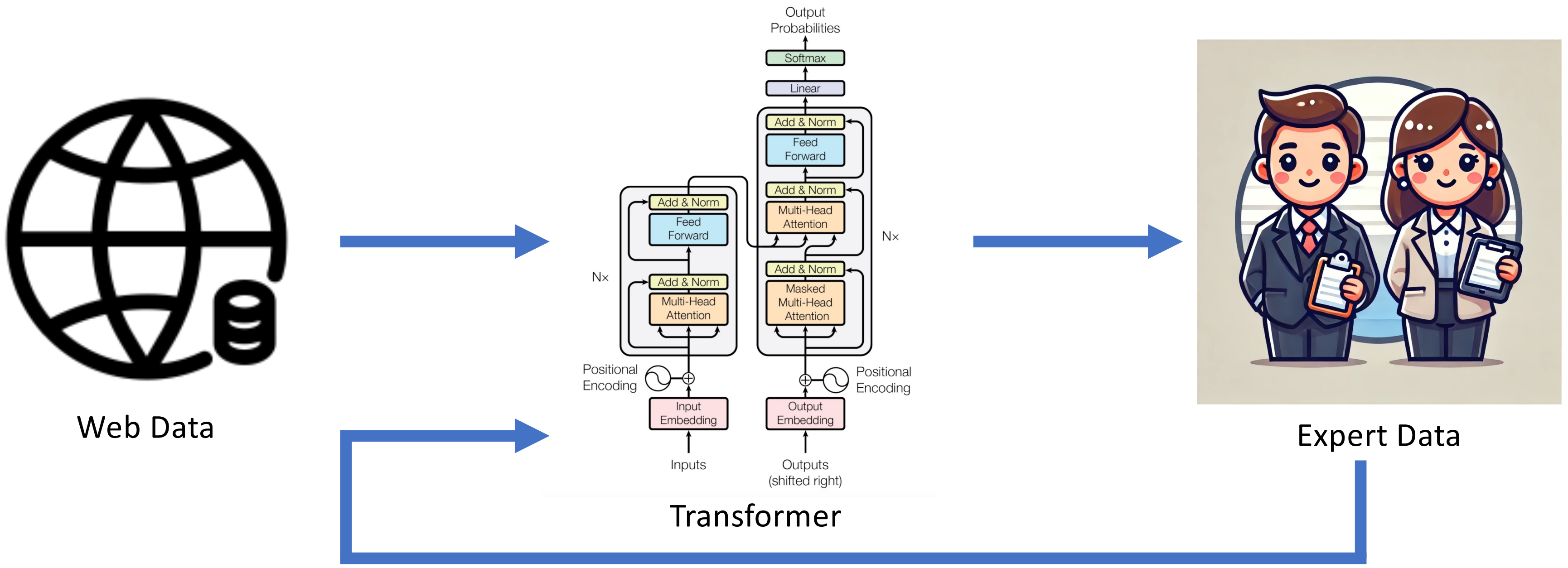
Web Data

Transformer

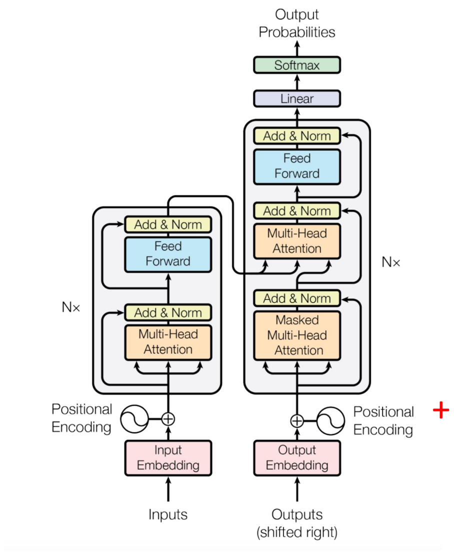
Expert Data

Can the model self-improve with better and better data?

- My experience (finetuning GPT2 in LLMs for Bio. Hackathon 2023) tells me that improvement will come at a cost of losing generalisability -> but can improve to better fit target distribution
- May be good for some targeted use cases



Zooming in on Expert Data Generation



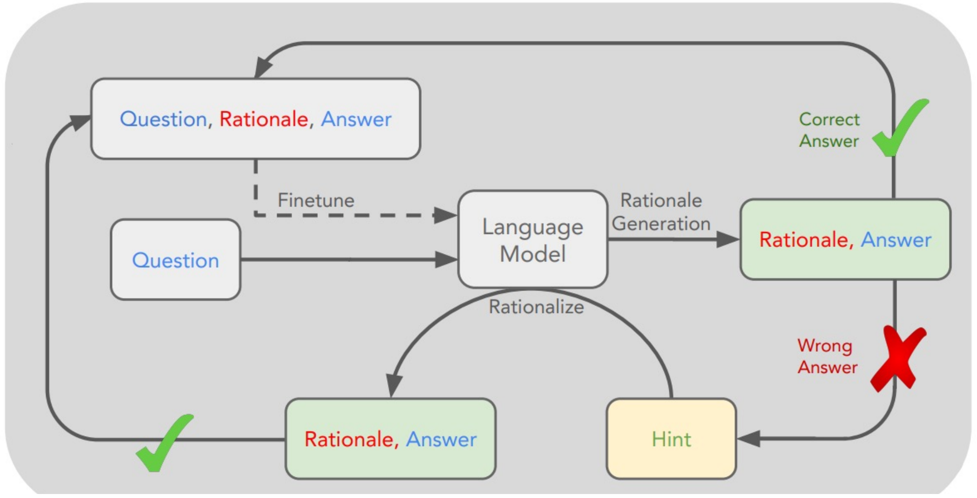
Prompt
+ Add Reasoning
+ Multiple Sampling
+ Filter
by Ground Truth

Expert Data

- How would we know whether the output is considered "expert data" or correct data?
- We will need alignment with the ground truth
- Question to ponder: **What if there is no ground truth?**

Self-taught Reasoner (STaR): Get better data by fine-tuning with reasoning traces

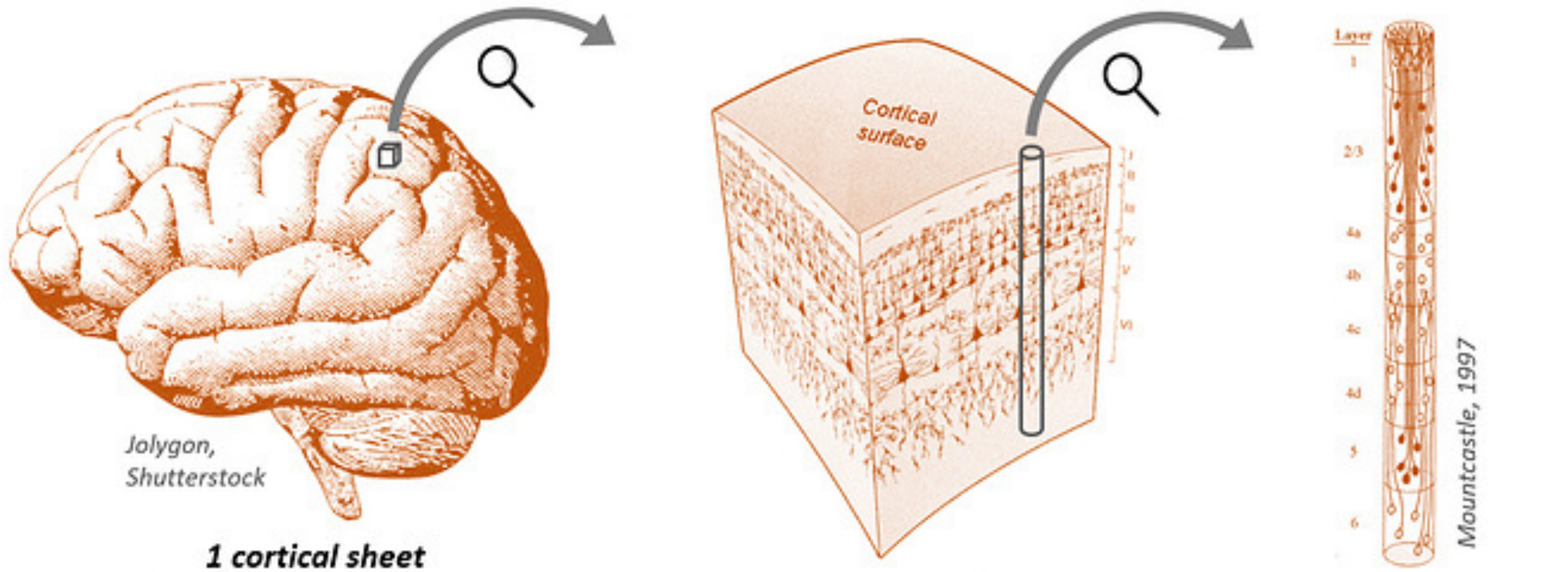
- LLMs get the right sequence more often with Chain of Thought to condition the next few tokens with previous reasoning/context
- Why not imbue such reasoning into the expert data?



```
Q: What can be used
to carry a small dog?
Answer Choices:
(a) swimming pool
(b) basket
(c) dog show
(d) backyard
(e) own home
A: The answer must be
something that can be
used to carry a small
dog. Baskets are
designed to hold things.
Therefore, the answer
is basket (b).
```

Figure 1: An overview of STaR and a STaR-generated rationale on CommonsenseQA. We indicate the fine-tuning outer loop with a dashed line. The questions and ground truth answers are expected to be present in the dataset, while the rationales are generated using STaR.

Minicolumns are plentiful in humans



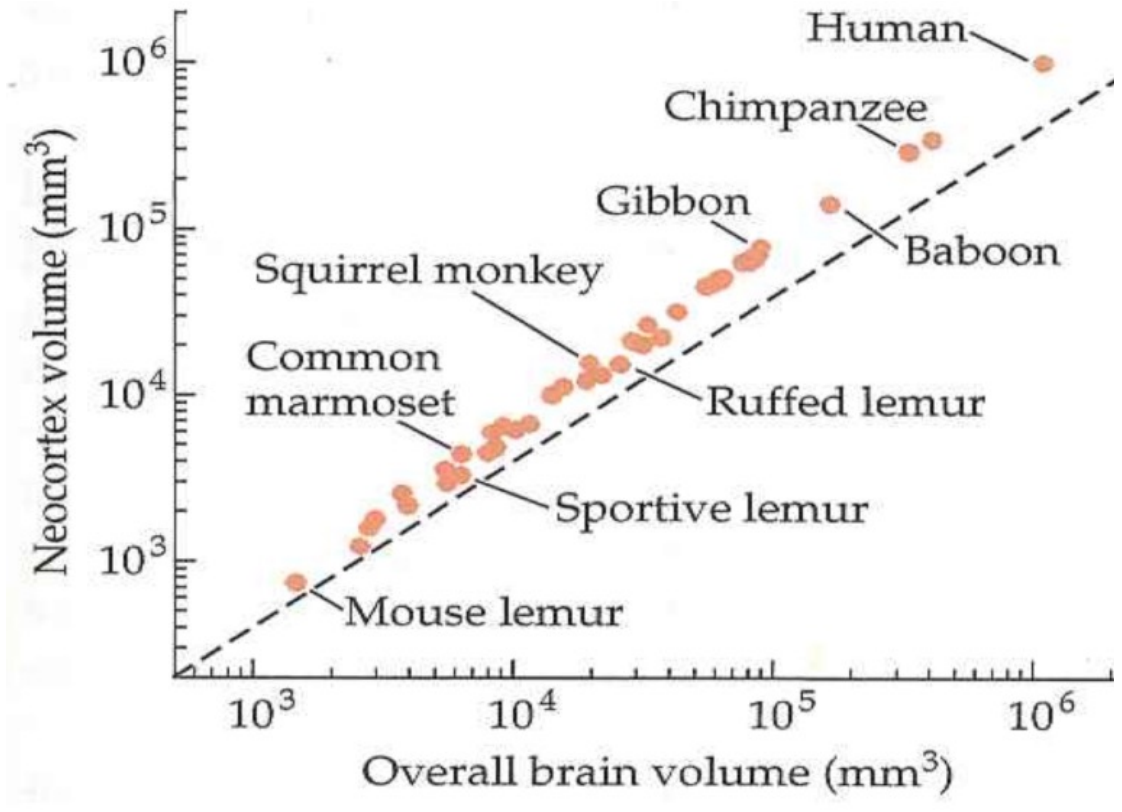
1 cortical sheet
2 million macrocolumns
200 million minicolumns
20 billion neurons

1 macrocolumn
100 minicolumns
10.000 neurons

1 minicolumn
100 neurons

Numenta

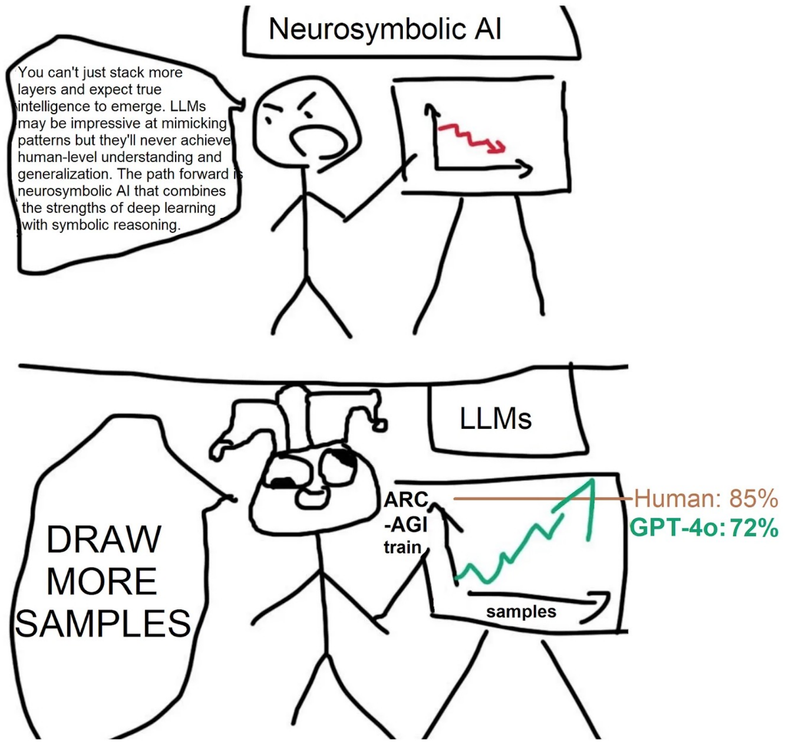
Perhaps intelligence can be gained with multiple sampling?

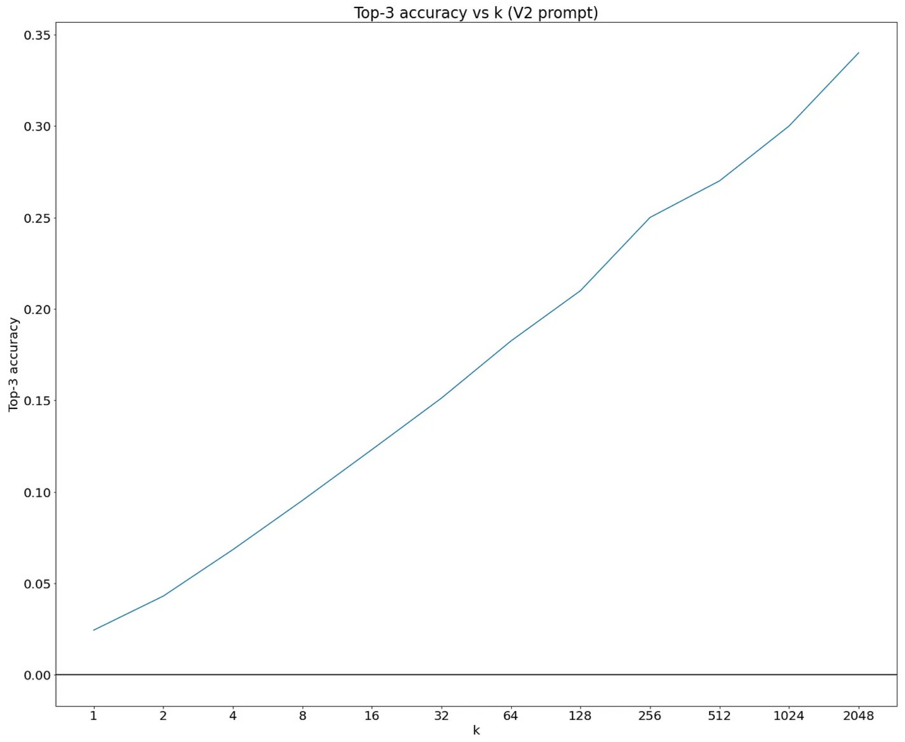


Toward the quantification of cognition. Richard Granger. 2020.

Ryan Greenblatt on ARC: Benefits of Multiple Sampling

- Sampling more times can lead to correct answer being found (if model is capable of generating it)


Neurosymbolic AI
You can't just stack more layers and expect true intelligence to emerge. LLMs may be impressive at mimicking patterns but they'll never achieve human-level understanding and generalization. The path forward is neurosymbolic AI that combines the strengths of deep learning with symbolic reasoning.
LLMs
DRAW MORE SAMPLES
ARC -AGI train
Human: 85%
GPT-4o: 72%
samples



https://redwoodresearch.substack.com/p/getting-50-sota-on-arc-agi-with-gpt

How to generate diverse samples?

- Obtain Diverse Samples:
  - (Optional) Get model to output differently by conditioning with different instructions
  - With each instruction, output many times with a high temperature

- Improve generation diversity by having more ground truth data to train model
  - Synthetic data helps
  - Having multiple diverse solution trajectories to solve one problem in ground truth data can be useful

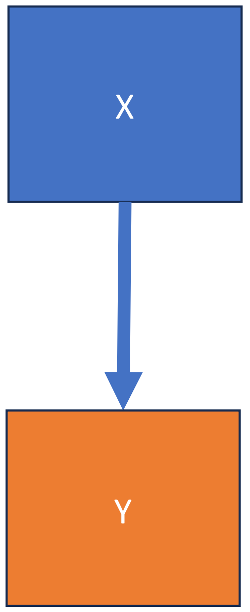
Improving chance of correct generation

- Train-time consolidation: Sample diversely, only train on correct trajectories (use ground truth, or use some scoring function)
    - **Question to ponder: Should we use some form of tree search to selectively generate parts of the trajectory based on some heuristic score instead of the full generation?**

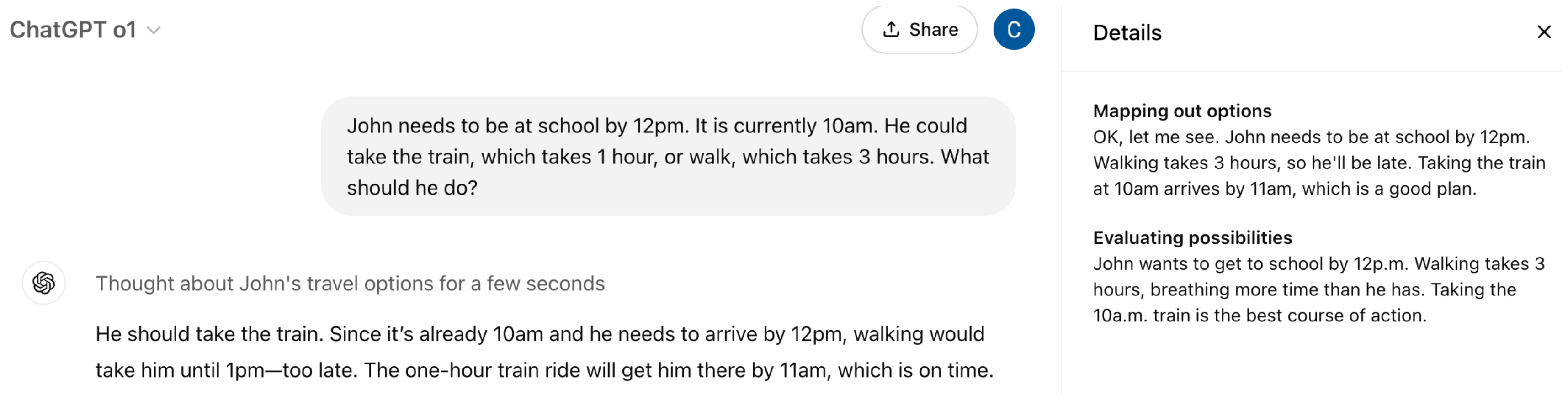
- Test-time consolidation: Use most frequent answer from samples, or rerank with a learned scoring function (https://openai.com/index/learning-to-reason-with-llms/)

How fine-tuning can be curriculum learning

- Let’s say initially you can solve problem type X
- You have a problem type Y that has problem type X as a prerequisite
- You have a better chance of solving problem type Y if you already have the reasoning traces to solve problem type X
- Now you have solved problem type Y, you can use it as a stepping stone to solve problems that has problem type Y as prerequisite



o1/o3: Chain of Thought via Subtasks



John needs to be at school by 12pm. It is currently 10am. He could take the train, which takes 1 hour, or walk, which takes 3 hours. What should he do?

Thought about John's travel options for a few seconds

He should take the train. Since it's already 10am and he needs to arrive by 12pm, walking would take him until 1pm—too late. The one-hour train ride will get him there by 11am, which is on time.

Mapping out options
OK, let me see. John needs to be at school by 12pm. Walking takes 3 hours, so he'll be late. Taking the train at 10am arrives by 11am, which is a good plan.

Evaluating possibilities
John wants to get to school by 12p.m. Walking takes 3 hours, breathing more time than he has. Taking the 10a.m. train is the best course of action.

- Instead of doing the whole reasoning at one go, break it down into subtasks so that the model can learn how to break down task into subtasks, and write details for each subtask

- This is still done as one large output chunk, and enables the model to do backtracking, reflection, questioning assumptions etc.

Limitations of LLM-only reasoning

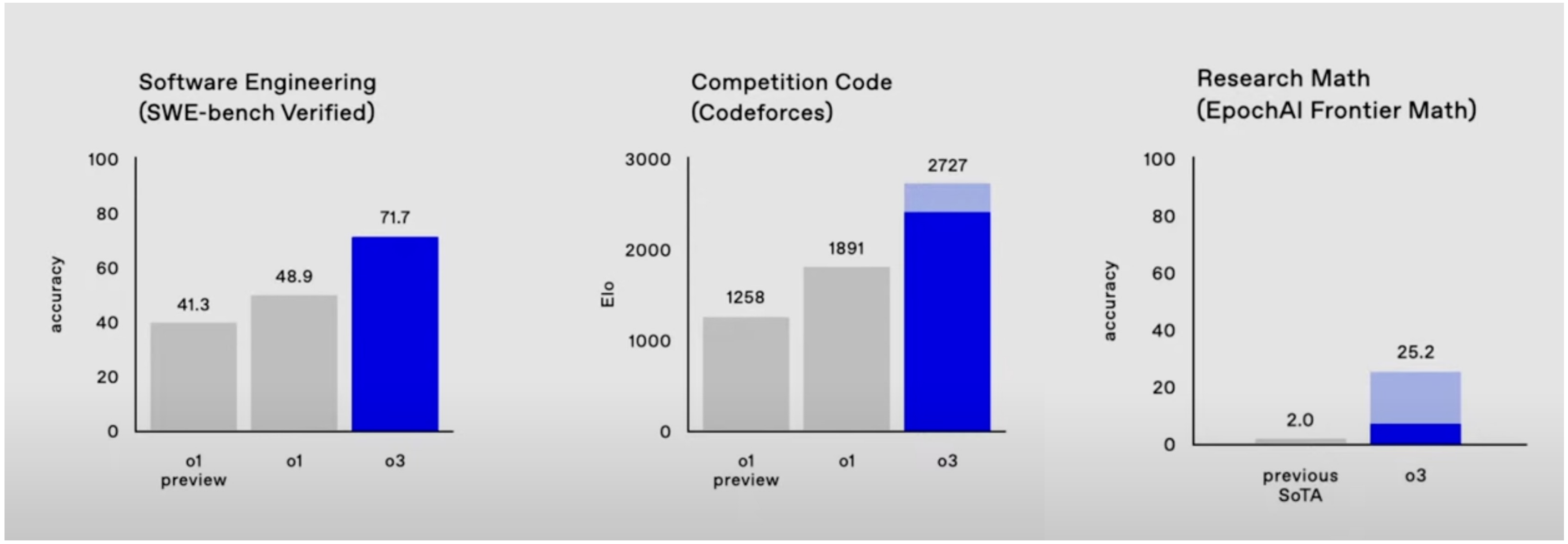
- Structured logical processing is not robust
- External tool use not natively integrated
- Verifier is not robust
- **Question to ponder: Should we use an agentic approach instead of one model doing everything?**

o3 Results

Is it AGI?

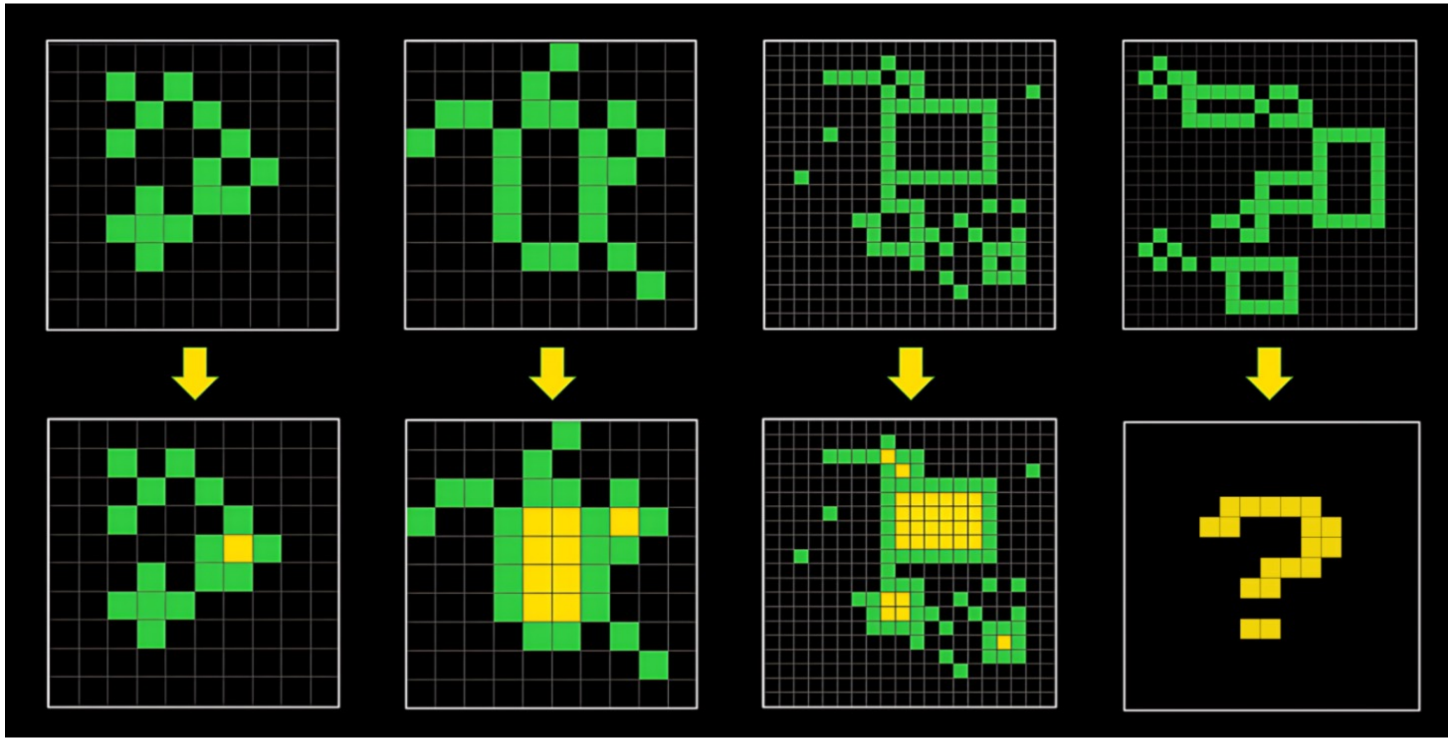
o3 – Impressive benchmark scores (Light blue – more compute)

- Exact details of compute not specified by OpenAI



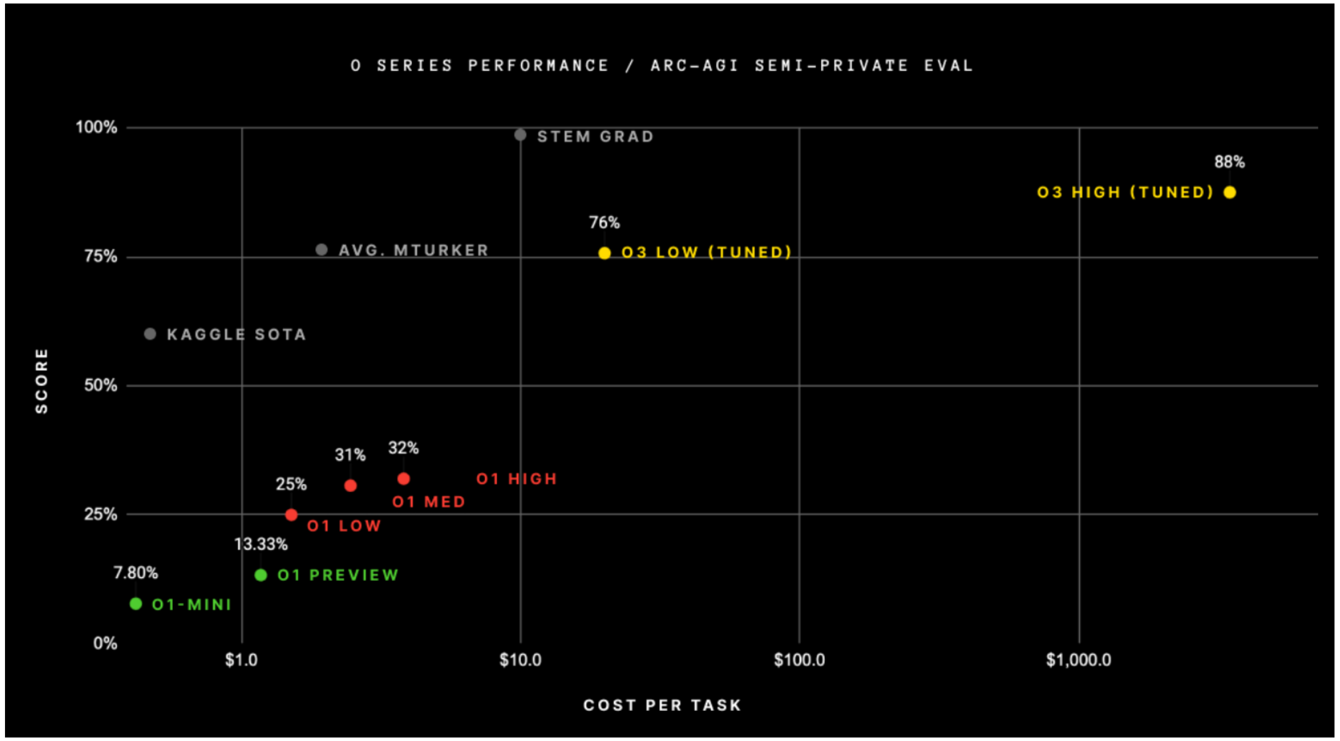
https://www.youtube.com/watch?v=SKBG1sqdyIU&t=826s

ARC-AGI



https://arcprize.org/arc

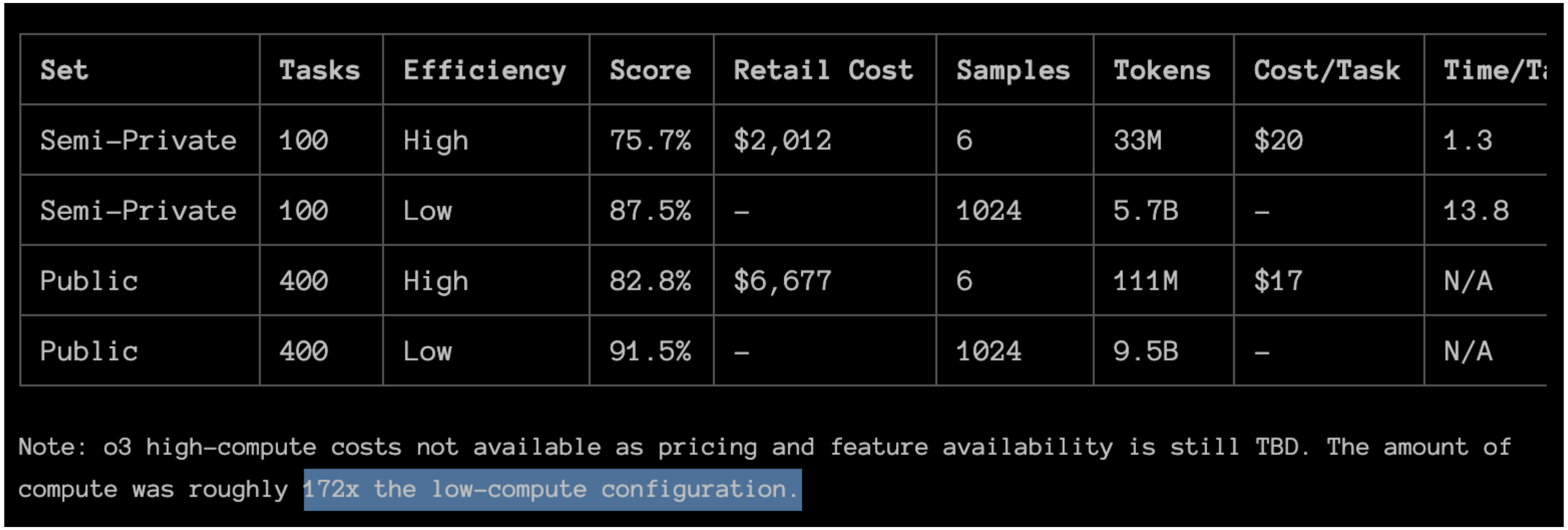
o3 can potentially solve >85% of ARC with enough compute



https://arcprize.org/blog/oai-o3-pub-breakthrough

Performance appears transferable to semi-private test set

- Note o3 is likely just text-based generation without calling any functions, which could mean better scores if we could do neurosymbolic integration



Set	Tasks	Efficiency	Score	Retail Cost	Samples	Tokens	Cost/Task	Time/T
Semi-Private	100	High	75.7%	$2,012	6	33M	$20	1.3
Semi-Private	100	Low	87.5%	-	1024	5.7B	-	13.8
Public	400	High	82.8%	$6,677	6	111M	$17	N/A
Public	400	Low	91.5%	-	1024	9.5B	-	N/A

Note: o3 high-compute costs not available as pricing and feature availability is still TBD. The amount of compute was roughly 172x the low-compute configuration.

https://arcprize.org/blog/oai-o3-pub-breakthrough

Is it AGI?

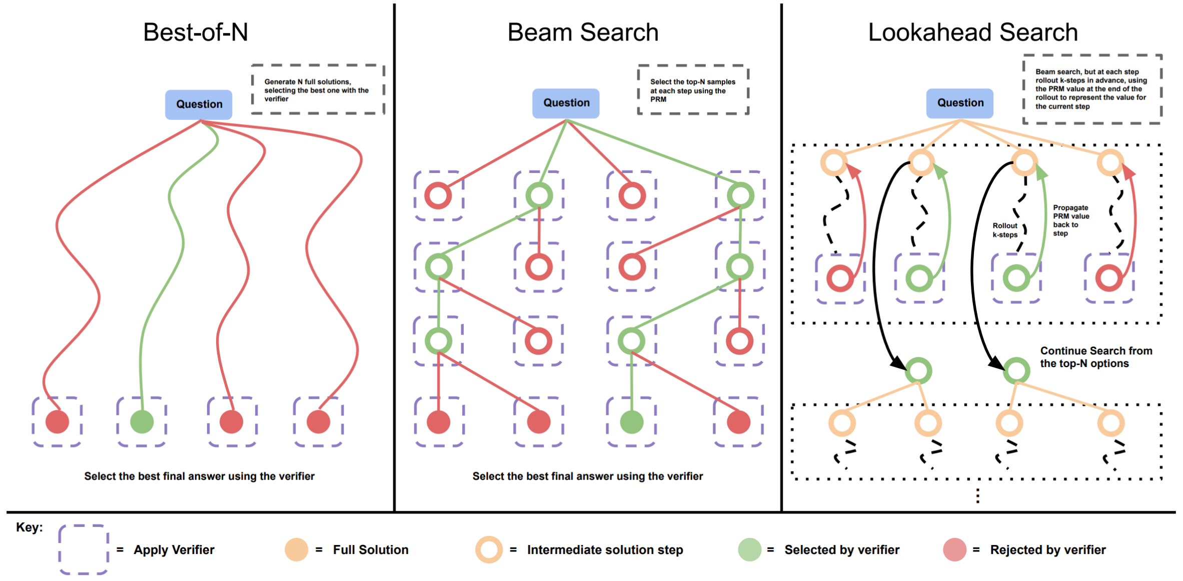
- AGI is a loaded term – but here we just take it that it is as good as an average human being at general tasks
- In that case, for ARC-AGI, it has attained this definition
- But is this performance generalisable across all tasks? I don’t think so
- If we are able to use the o1/o3 method to learn arbitrary tasks by fine-tuning on the correct trajectories, it would be good
  - Method hinges upon being able to **generate a correct trajectory**, which may not happen
  - (Comparison) AlphaGo/AlphaZero has self-play, which means there is a learning signal even when losing -> can o1/o3 get a learning signal when they do not solve the problem?

Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters

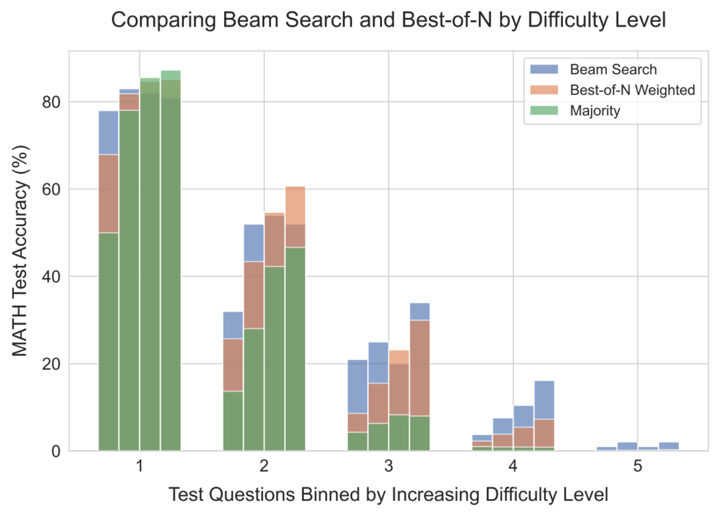
Charlie Snell, Jaehoon Lee, Kelvin Xu, Aviral Kumar

Google DeepMind

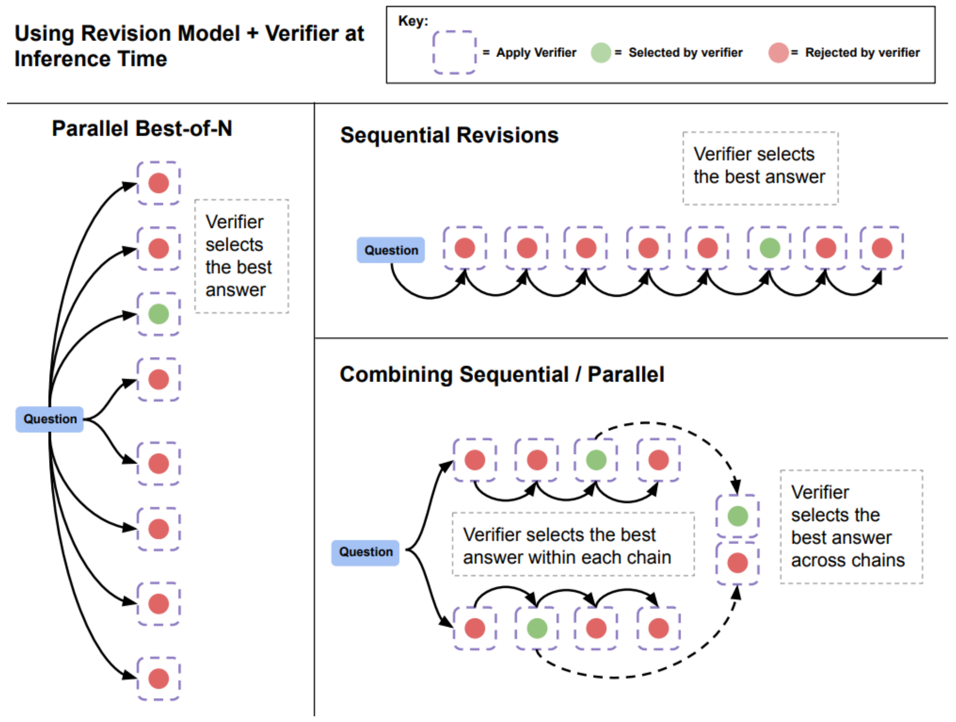
Tree Search: Yes or No?

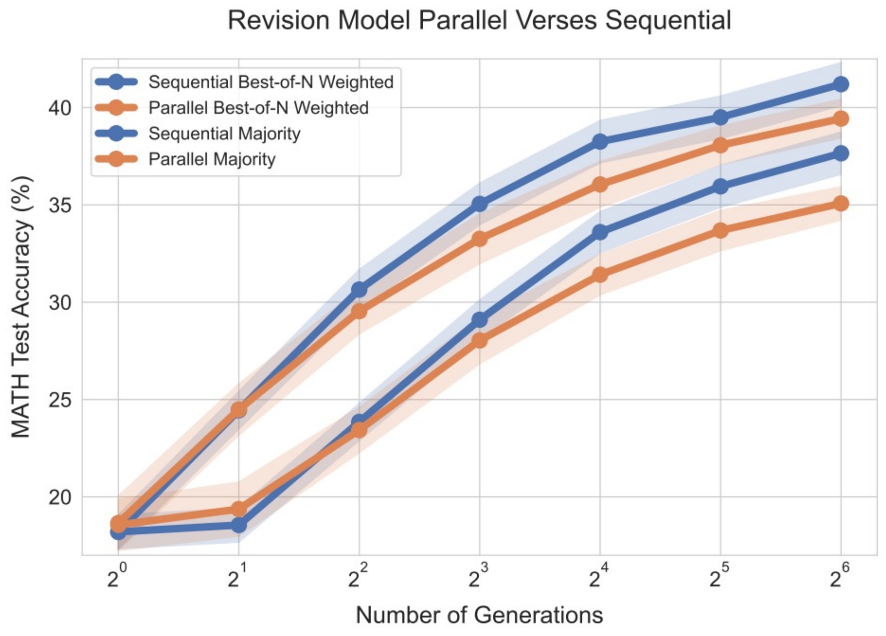


Best-of-N
Question
Generate N full solutions, selecting the best one with the verifier
Select the best final answer using the verifier
Beam Search
Question
Select the top-N samples at each step using the PRM
Select the best final answer using the verifier
Lookahead Search
Question
Beam search, but at each step rollout k-steps in advance, using the PRM value at the end of the rollout to represent the value for the current step
Rollout k-steps
Propagate PRM value back to step
Continue Search from the top-N options
Key:
= Apply Verifier
= Full Solution
= Intermediate solution step
= Selected by verifier
= Rejected by verifier



Sequential or Parallel?


Using Revision Model + Verifier at Inference Time
Key:
= Apply Verifier
= Selected by verifier
= Rejected by verifier
Parallel Best-of-N
Question
Verifier selects the best answer
Sequential Revisions
Question
Verifier selects the best answer
Combining Sequential / Parallel
Question
Verifier selects the best answer within each chain
Verifier selects the best answer across chains

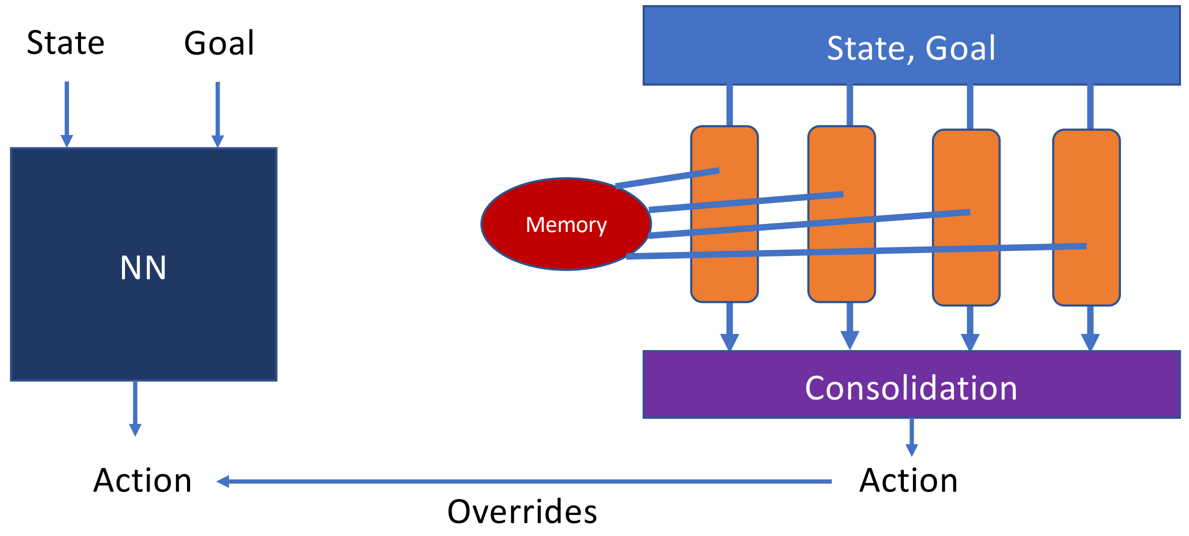


Path ahead

Way ahead: Adaptive Benchmarks, Fast learning with External Memory and Fine-Tuning for Longer-Term Learning

- Fine-tuning usually takes a long while to get the intended effects, and may overwrite previously learned information (catastrophic forgetting)
- Changing external memory is much faster for learning, and can adapt faster to environmental changes
- Most benchmarks used to evaluate LLMs are still very **static** – they don't change with the environment input
    - We need more challenging benchmarks that test LLM's response to changing ground truths, e.g. calling APIs that have a chance of failure, navigating in an environment that keeps changing

Two Networks – Fast and Slow



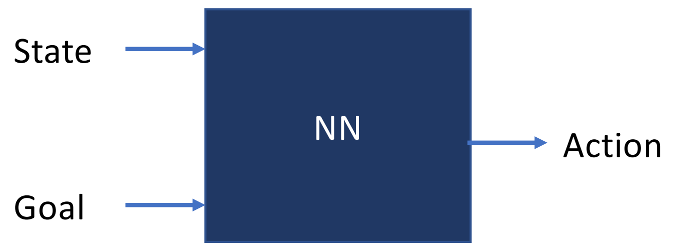
Neural Networks: Fast retrieval, slow learning

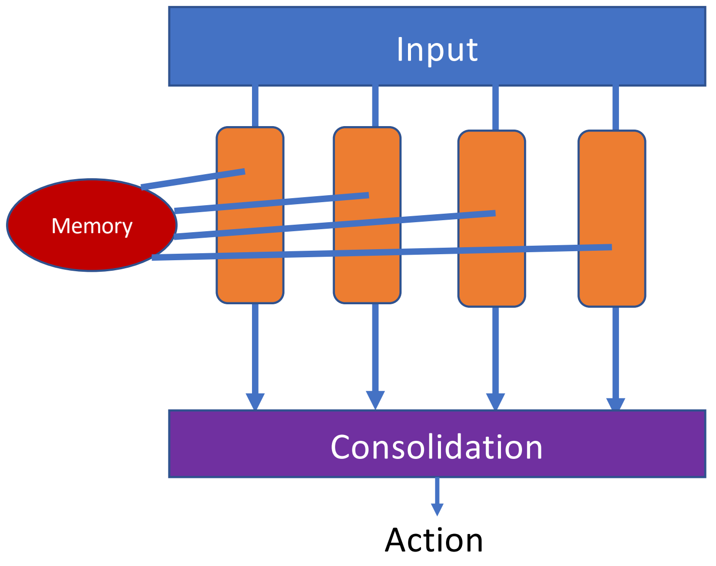
External Memory: Slow retrieval, fast learning

Learning, Fast and Slow. John and Motani. 2023

Two Networks – Fast and Slow

- **Fast (System 1):** Neural network predicts best initial action given a goal
- **Slow (System 2):** Memory retrieval network to form possible trajectories to goal state



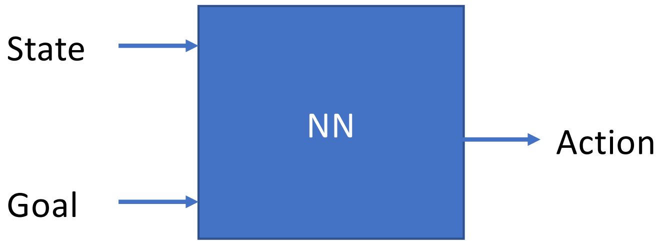


Fast Goal-Directed Neural Network

- Fast Neural Network Update

(at each time step)

- **Previous states replay**
- **Future states replay (only if trajectory found**



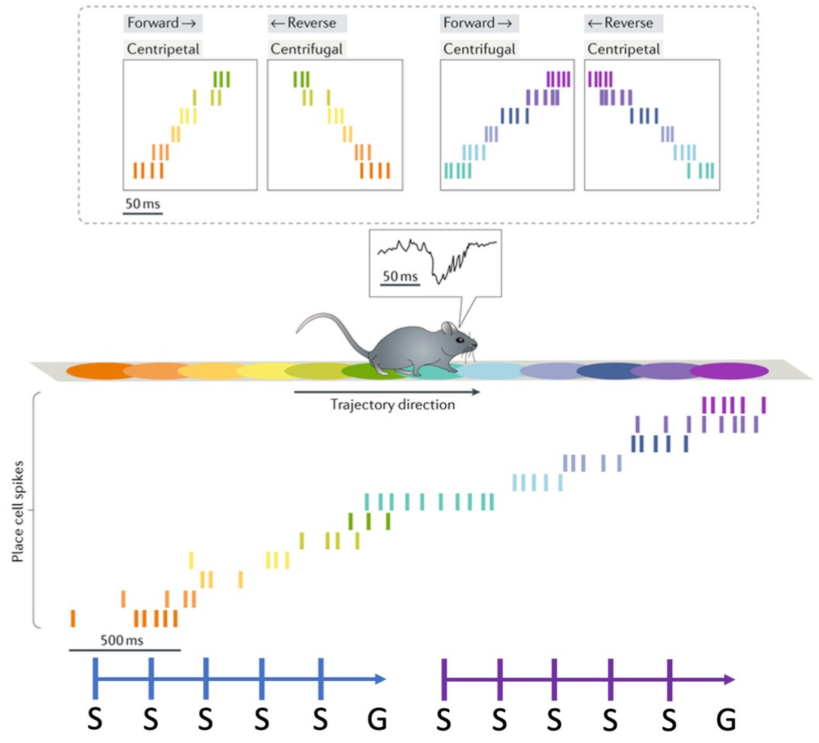
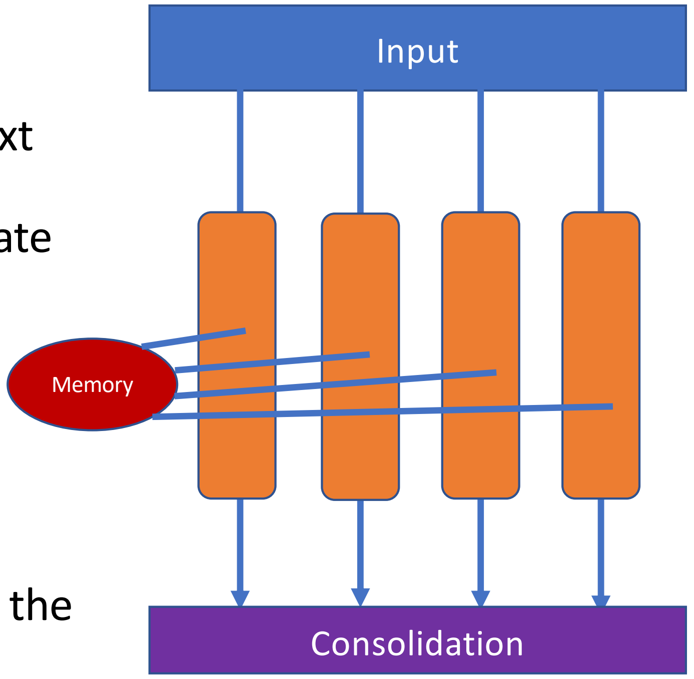


Figure extracted from Joo, H. R., & Frank, L. M. (2018). The hippocampal sharp wave-ripple in memory retrieval for immediate use and consolidation. Nature reviews. Neuroscience, 19(12), 744–757. https://doi.org/10.1038/s41583-018-0077-1

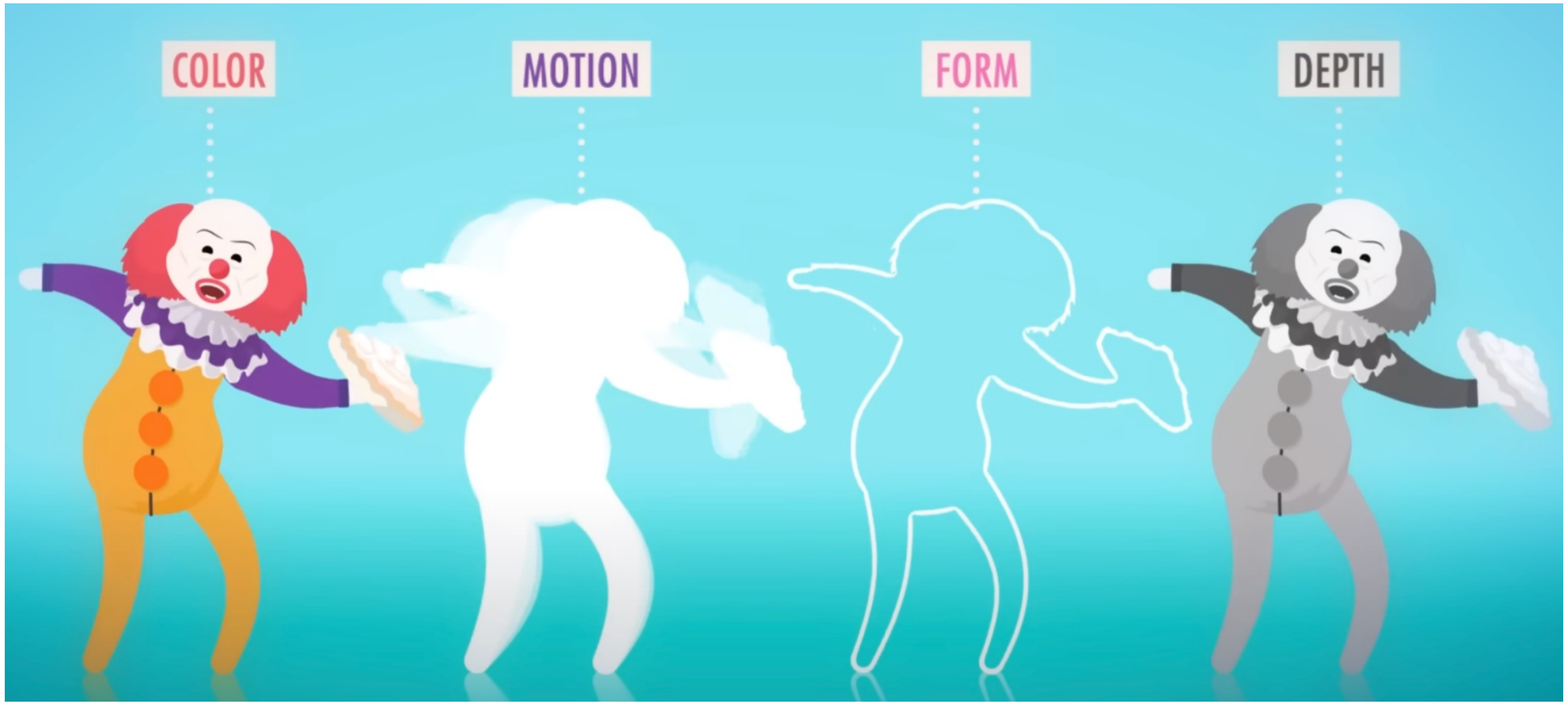
Slow Memory Retrieval Network

- Uses parallel processing with *B* branches
- Each parallel branch is like a minicolumn in the neocortex
- Takes the starting state and then reference memory for next state
- If more than one match, randomly pick one for the next state
- If next state is goal state, break
- Continue with next state as the key to reference memory
- Repeat until *D* lookahead timesteps

- All parallel branches will come back with a response
- See which branch has shortest trajectory to goal state, use the first action



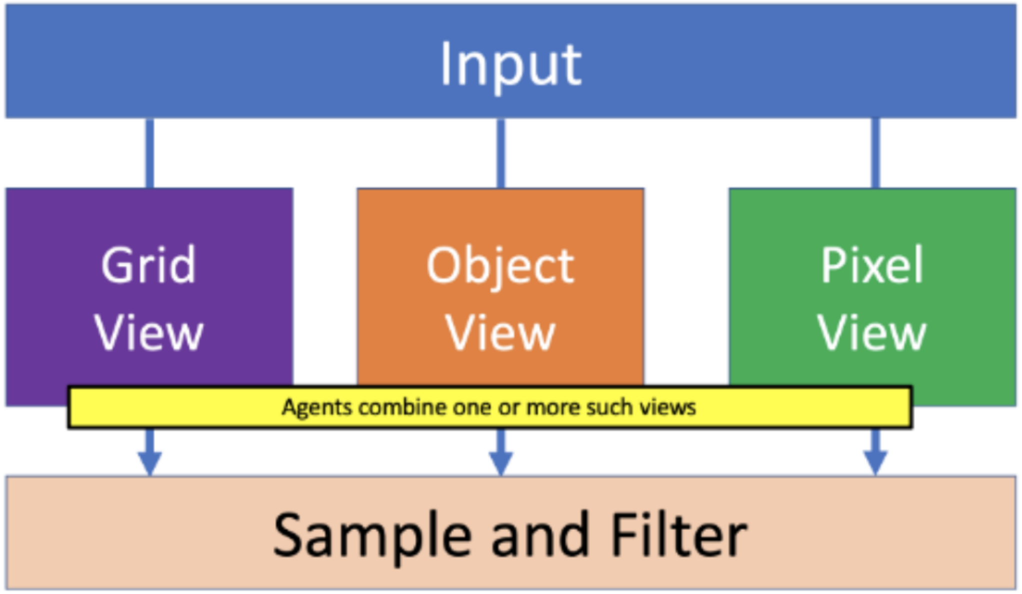
Vision has multiple abstraction spaces



Sensation and Perception: Crash Course Psychology #5
https://www.youtube.com/watch?v=unWnZvXJH2o

Reducing Training Samples: Imbuing multiple abstraction spaces / neurosymbolic integration

- Each agent views the grid differently, like grid, object, or pixel level
- Generate multiple potential **Python programs**
- Filter solutions by those which pass training examples
- 20 views * 3 samples = 60 for each problem
- **50 out of 111 public training set tasks solved (45%)**



LLMs as a System of Multiple Expert Agents. John and Motani. 2023.

Questions to Ponder

- How do we fine-tune with reasoning traces if all traces do not give the right ground truth answer?
- How do we self-augment the training set / learn the scoring function if we do not have a ground truth?
- Should we use some form of tree search to selectively generate parts of the trajectory based on some heuristic score, or just do full generation?
- Should we use an agentic approach instead of one LLM model doing everything?
- Should we imbue more bias into the preprocessing of the model, to reduce samples required for training?