History May Repeat Itself:
RSI Seen from a Previous AI Era

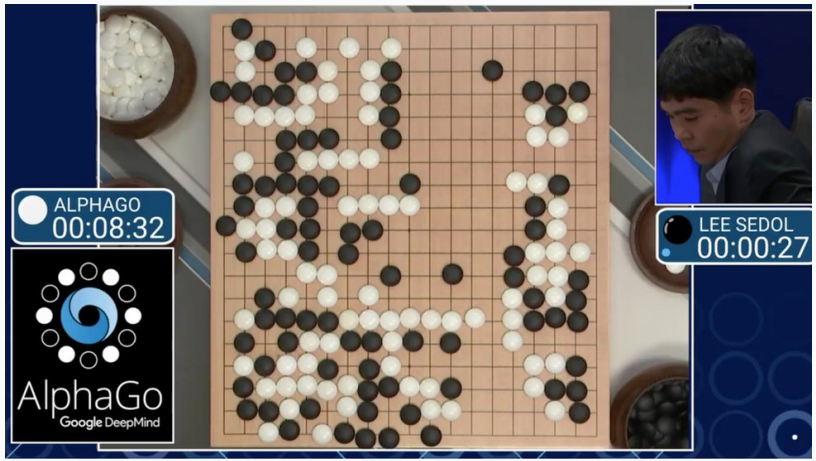
Yuandong Tian
Co-founder of a stealth startup
ex-Research Director in Meta (FAIR)

The Promise of Recursive Self-Improvement (RSI)

What will happen next??

Predicting the future by learning from the **history**

Self-Improving Systems back a decade ago



AlphaGo (2016)

(Self-play, with human knowledge)

AlphaZero (2018)

(Self-play, No human knowledge)

OpenGo (2018)

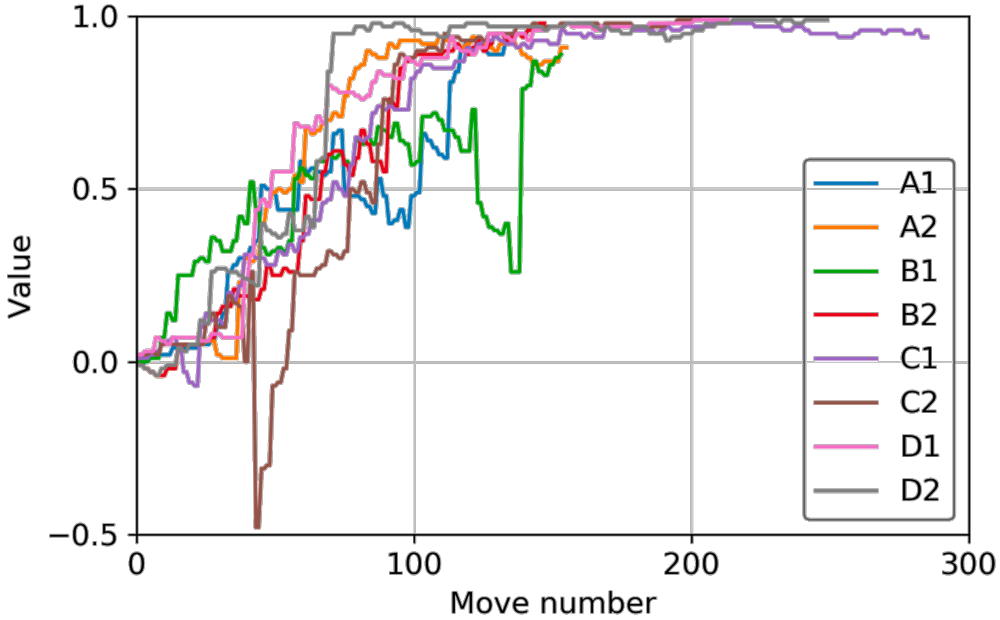
Vs top professional players

Name (rank)	ELO (world rank)	Result
Kim Ji-seok	3590 (#3)	5-0
Shin Jin-seo	3570 (#5)	5-0
Park Yeonghun	3481 (#23)	5-0
Choi Cheolhan	3466 (#30)	5-0

Single V100 GPU, 80k rollouts, 50 seconds
Offer **unlimited thinking time** for the players

Vs professional players

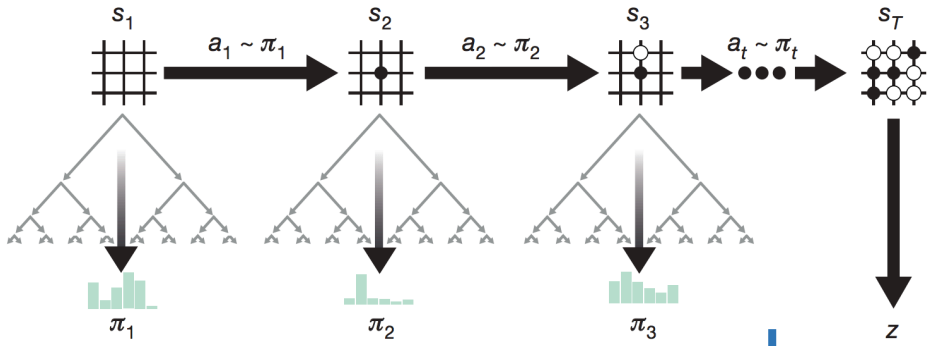
Single GPU, 2k rollouts, 27-0 against Taiwanese pros.



Trained on 2k GPUs for 2 weeks, No human knowledge
Evaluated with a single V100 GPU

[Y. Tian et al, ELF OpenGo: An Analysis and Open Reimplementation of AlphaZero, ICML 2019]

AlphaZero



Generate Training data

Without human knowledge

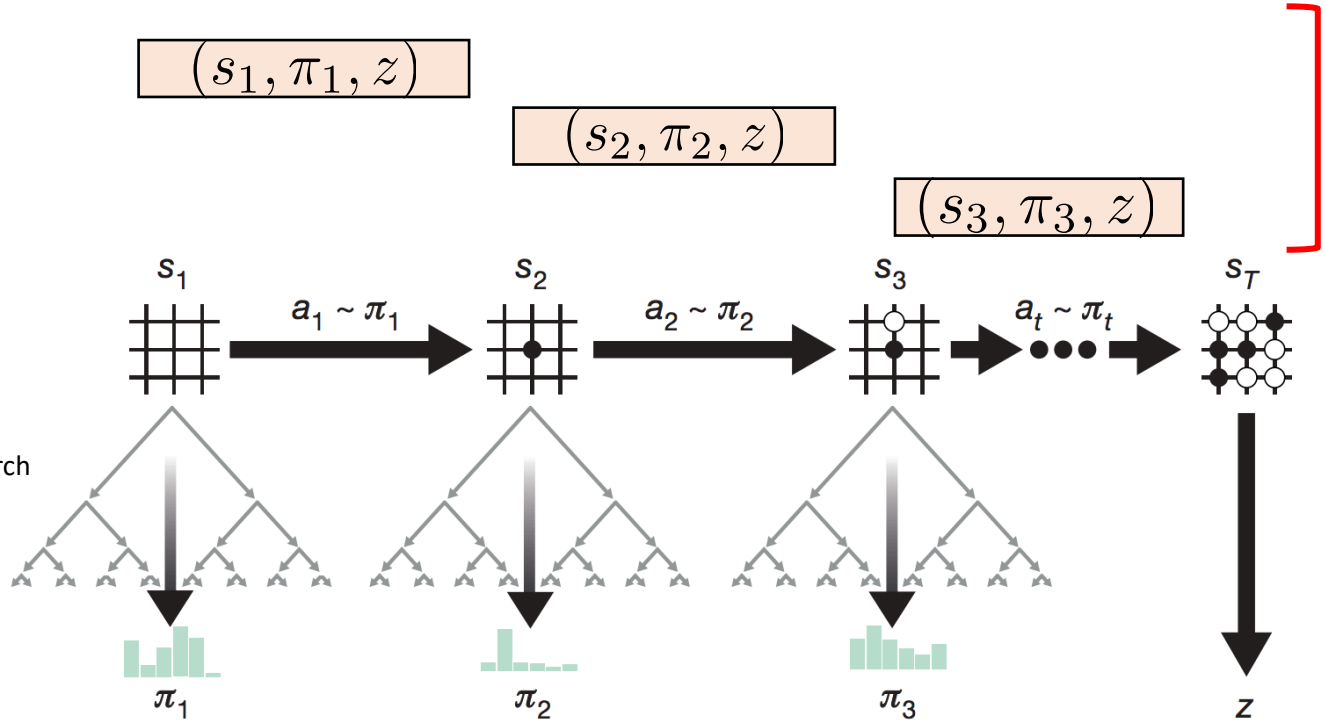
θ_i

Update Models

Self-Replays

[D. Silver et al, Mastering the game of Go without human knowledge, Nature 2017]

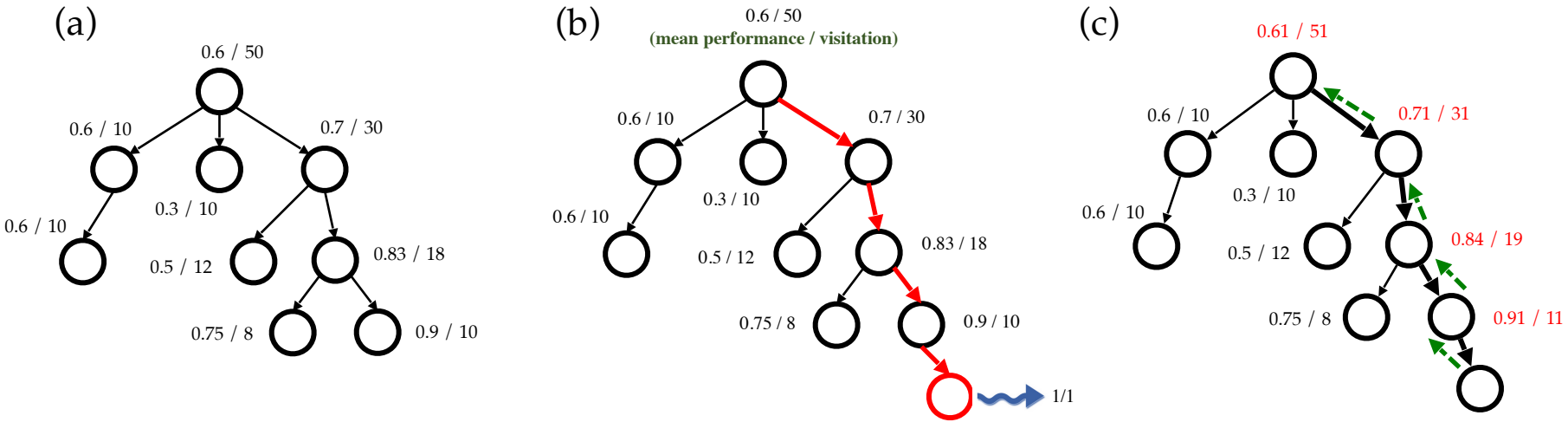
Generating Self-play Games



Model + Harness = Self-Improvement

Monte Carlo Tree Search (MCTS):

Search towards the good nodes while keeping exploration in mind



Call network models
(policy/value)

Policy/Value networks = Models

MCTS = Harness

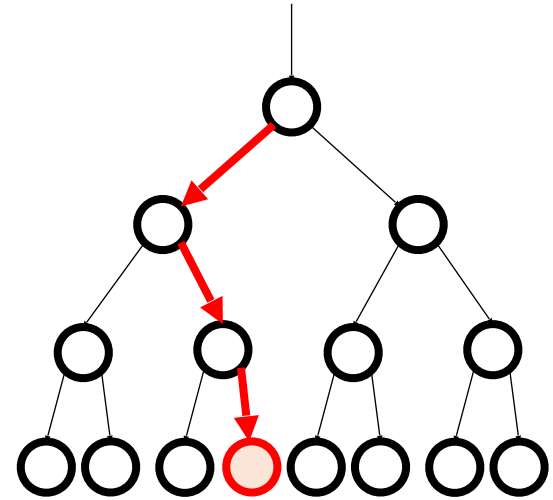
2026: What's the difference?

Large Language Models

Abundant world knowledge

Diverse Thinking patterns

Understanding of concepts

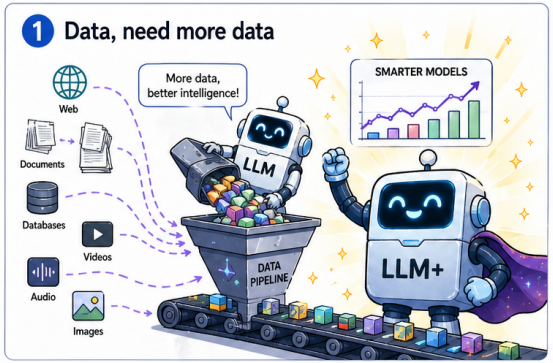


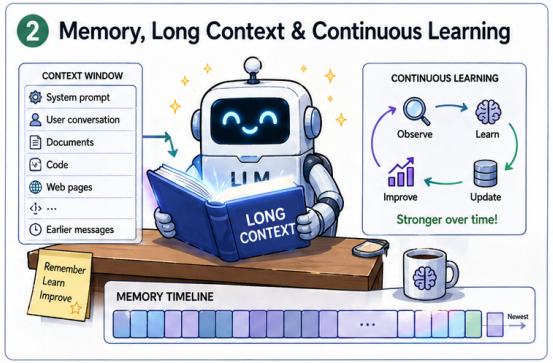
❌ Human Knowledge / Heuristics

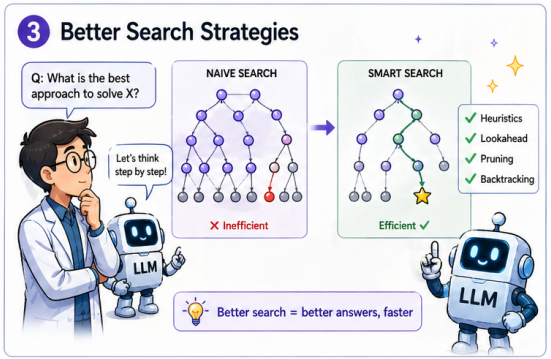
❌ Machine learned models

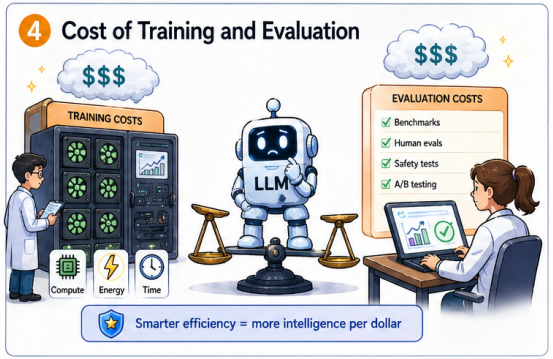
✅ Model with General knowledge

Challenges in Recursive Self-improvement







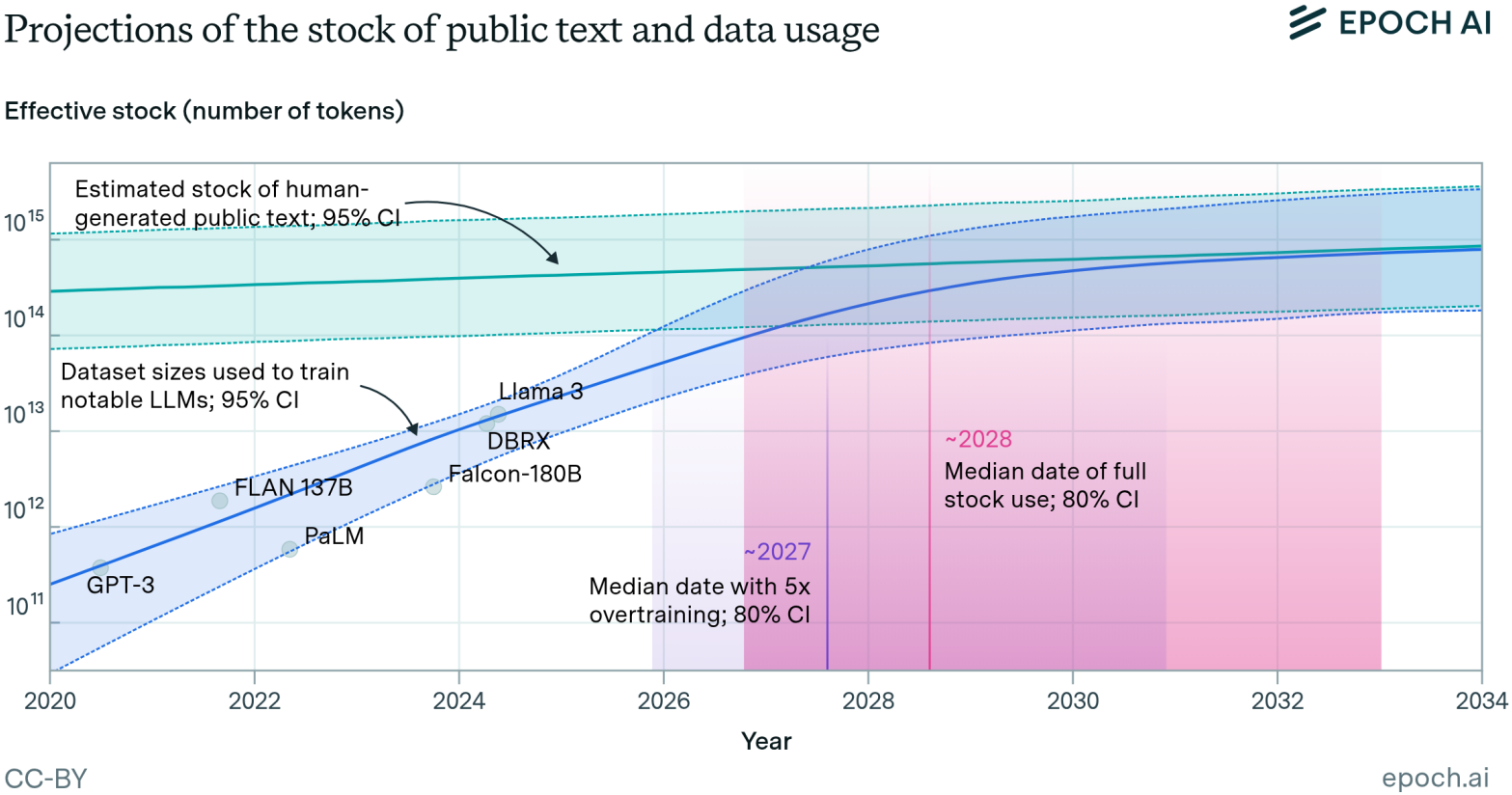


A large-scale RSI system with:

✅ detailed engineering design

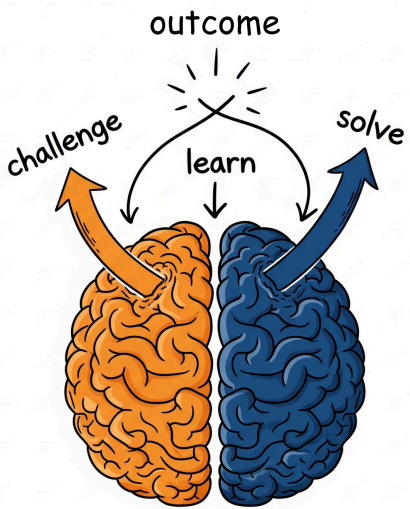
✅ research breakthroughs

1 Data, data and More data

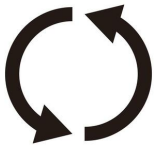


1 Data, data and More data

Generating Synthetic Environment/Data Distribution on the Fly



Solver: Learning using RL on the current environments



Challenger: Generate hard environment on the fly

$$\min_{\pi_{\text{Ch}}} \max_{\pi_{\text{Sol}}} \mathbb{E}_{\text{q}\sim\pi_{\text{Ch}},\text{a}\sim\pi_{\text{Sol}}}[R(\text{q},\text{a})]$$

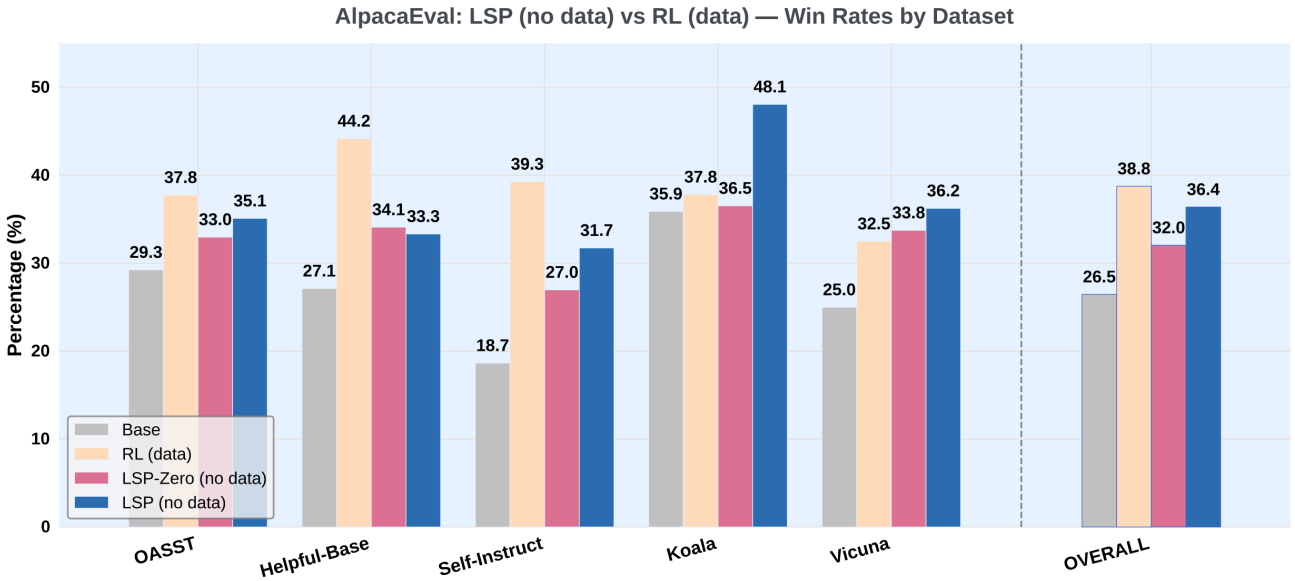
Proper regularization is needed to avoid reward hacking

(e.g., adding quality metrics to make it general-sum game)

[J. Kuba, Language Self-Play For Data-Free Training]

1 Data, data and More data

Generating Synthetic Environment/Data on the Fly



LSP-Zero: Zero-sum formulation

(no regularization)

LSP: General-sum formulation

(With regularization)

[J. Kuba, Language Self-Play For Data-Free Training]

1 Data, data and More data

Generating Synthetic Environment/Data on the Fly

Box 4: Challenger-Generated Prompts

500 iterations. Create a treasure map on a deserted island using a piece of paper, a pen, and two different rocks.

1000 iterations. Generate a set of instructions that an 5-year-old can follow to build a simple bridge using only 10 wooden blocks, a piece of string, and scissors, without a template or any external aid, within a 10-minute time frame.

1500 iterations. Write a 2048-line assembly code for a subtracting two 16-bit numbers stored in two consecutive 32-bit registers. The numbers should be stored in the registers and the result should be automatically saved in a new register.

[J. Kuba, Language Self-Play For Data-Free Training]

2️⃣ Memory, Long Context and Continuous Learning

Claude Code Memory Architecture

All memories stored as transparent, auditable markdown files

User Memory

Role, preferences, expertise, knowledge

Feedback Memory

Corrections, validations, approach guidance

Project Memory

Goals, initiatives, deadlines, active work

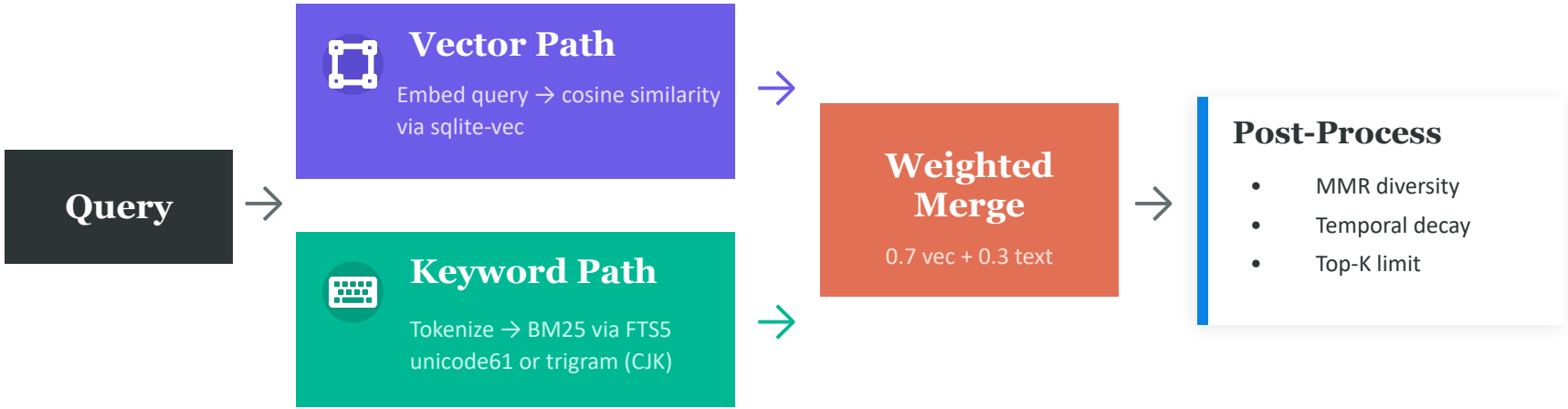
Reference Memory

Pointers to external systems and tools

2️⃣ Memory, Long Context and Continuous Learning

OpenClaw Hybrid Search from Memory

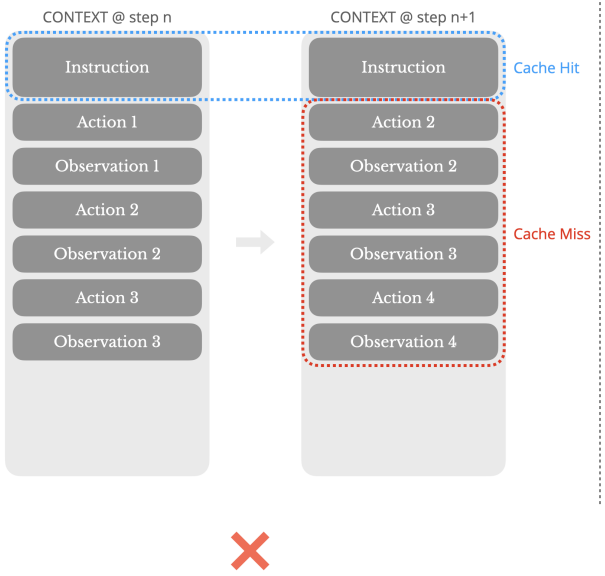
From user query to ranked results in a single call

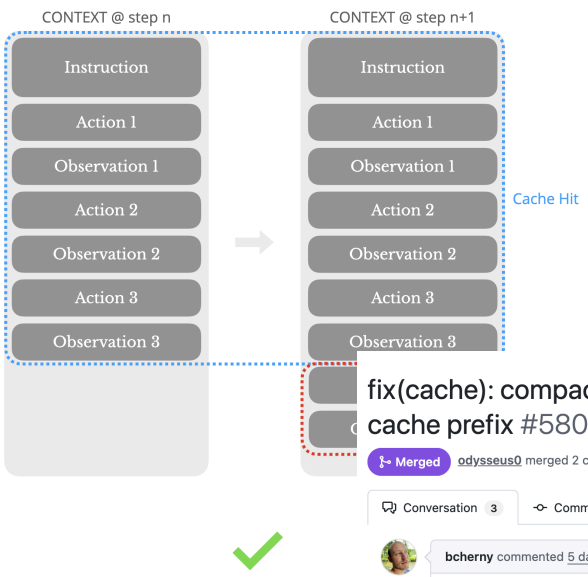


Both paths run in parallel against a SQLite database — vector via sqlite-vec extension, keywords via FTS5 full-text index

Cost and Latency in Context Design

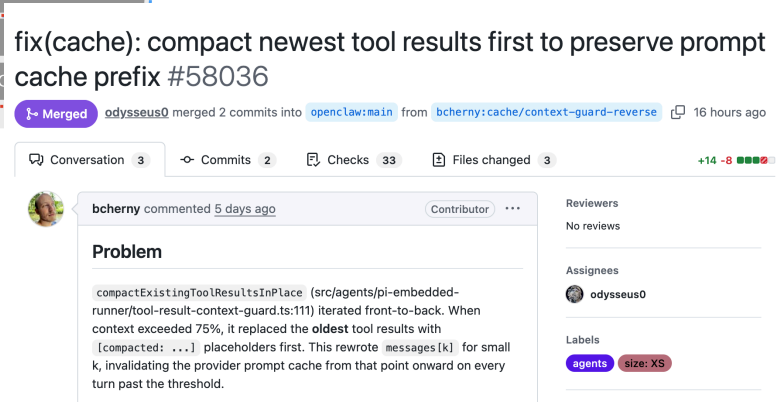
If you change the content at the beginning of KV cache, all subsequent content will have cache miss —> **High cost and latency!**





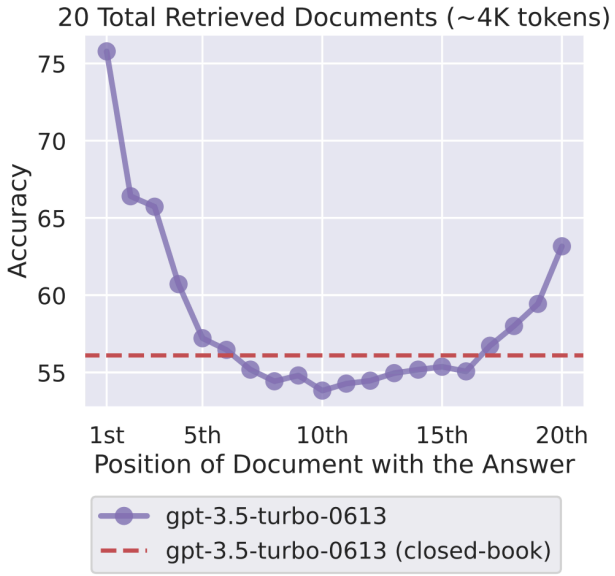
Keep your prompt prefix stable.

Make your context append-only.

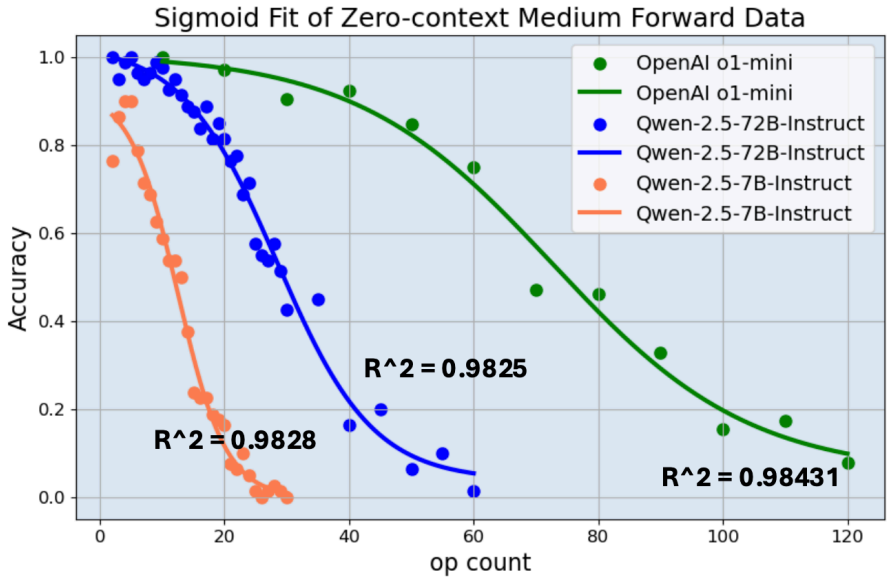


https://manus.im/blog/Context-Engineering-for-AI-Agents-Lessons-from-Building-Manus

Long-lasting Issues in Long Context



Liu et al., "Lost in the Middle" (2023)

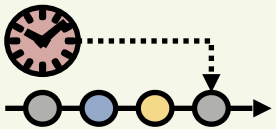


Y. Zhou et al., “GSM-Infinite” (2025)

Long-lasting Issues in Long Context

Memory Retrieval

Recall



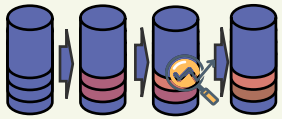
e.g, What is the value of object X at step Y?

Causal Inference

e.g, What's the pre-condition of action Z execution?

Memory Evolution

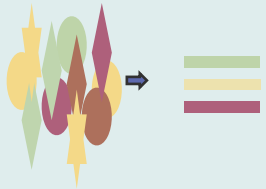
State-Updating



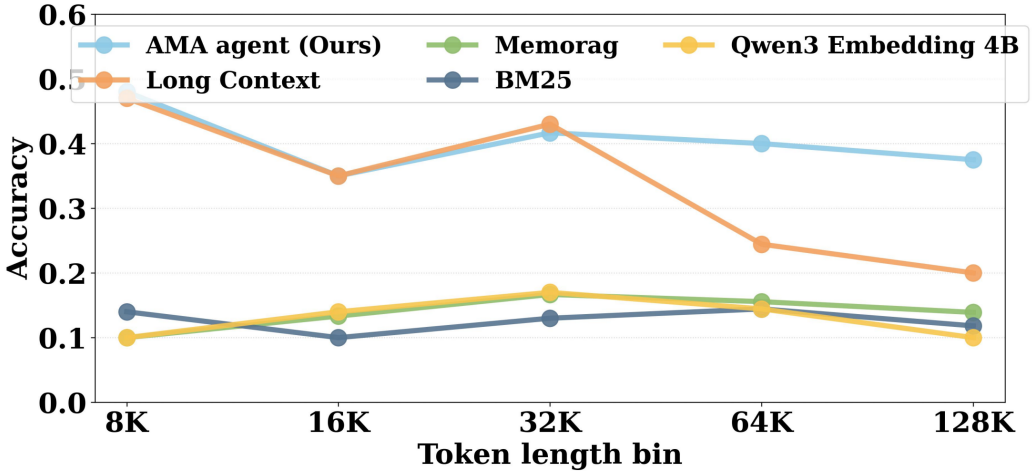
e.g, What is the current hidden state of Object X at step Y?

Memory Condensation

State Abstraction



e.g, What is the key observation between step X to Y?

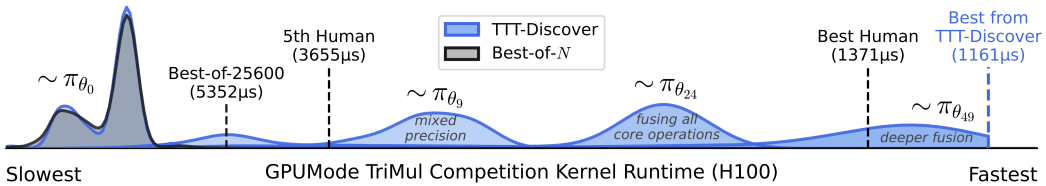


[Y. Zhao et al, AMA-Bench: Evaluating Long-Horizon Memory for Agentic Applications]

Continuous Learning

TTT-Discover

	Mathematics Erdős' Min. Overlap (↓)	**Kernel Eng. (TriMul)** A100 (↓)	H100 (↓)	**Algorithms (AtCoder)** Heuristic Contest 39 (↑)	**Biology** Denoising (↑)
Best Human	0.380927 [20]	4531 µs	1371 µs	566,997 [56]	0.64
Prev. Best AI	0.380924 [50]	N/A	N/A	558,026 [37]	N/A
TTT-Discover	**0.380876**	**2198 µs**	**1161 µs**	**567,062**	**0.71**



$$J_\beta(\theta) = \mathbb{E}_{s\sim \text{reuse}(\mathcal{H})}\left[\log \mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[e^{\beta(s)R(s,a)}\right]\right]$$

Fine-tune a model that favors maximal rewards

(Want maximal performance rather than average performance)

The fine-tuned model becomes specific to the task (loss of diversity)

[M. Yuksekgonul et al, Learning to Discover at Test Time]

Back to Games

"Grandmaster-Level" Chess Without Search

Agent	Train	Search	Input	Tournament Elo	Lichess Elo vs. Bots	Lichess Elo vs. Humans	Puzzle Acc. (%)
9M Transformer (ours)	SL		FEN	2025 (±18)	2054	-	88.9
136M Transformer (ours)	SL		FEN	2259 (±16)	2156	Blitz game only	94.5
270M Transformer (ours)	SL		FEN	2299 (±15)	2299	2895	95.4
GPT-3.5-turbo-instruct	SSL		PGN	-	1755	-	66.5
AlphaZero (policy net only)	RL		PGN	1777 (±25)	-	-	56.1
AlphaZero (value net only) ~23M parameters	RL		PGN	1992 (±19)	-	-	82.0
AlphaZero (400 MCTS sim.)	RL	✓	PGN	2470 (±16)	-	-	95.6
Leela Chess Zero (policy net only)	RL		PGN	2292 (±16)	2224	-	88.6
Leela Chess Zero (value net only)	RL		PGN	2418 (±16)	2318	-	95.9
Leela Chess Zero (400 MCTS sim.)	RL	✓	PGN	2858 (±20)	2620	-	99.6
Stockfish 16 (50ms per move) [oracle]	SL	✓	FEN + Moves	2711 (±18)	2713	-	99.8
Stockfish 16 (1.5s per board)	SL	✓	FEN + Moves	2935 (±23)	2940	-	100.0

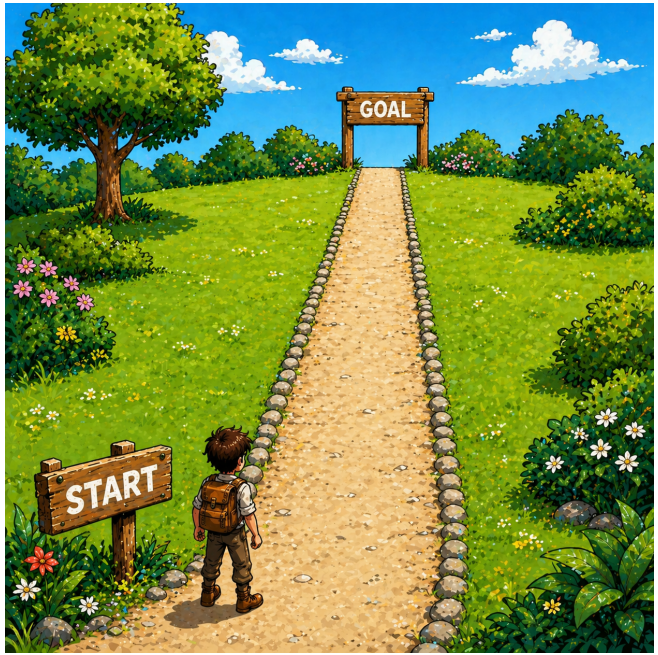
With Search (harness) >> Without search

Maybe Agent Harness is still needed even if the model is strong

[A. Ruoss, et al, Amortized Planning with Large-Scale Transformers: A Case Study on Chess]

3️⃣ Better Search Strategies

No good representation



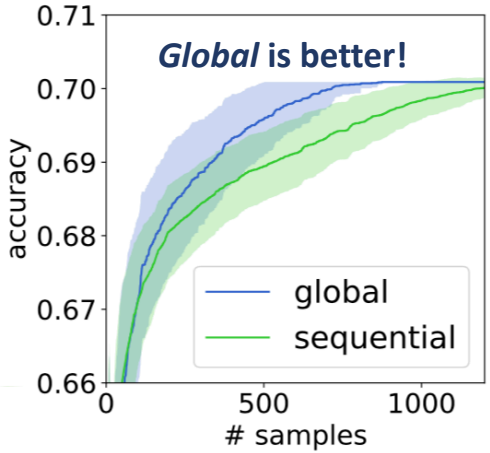
With good representation

3 Better Search Strategies

How to represent the "Action Space"

Depth = {1, 2, 3, 4, 5}
Channels = {32, 64}
KernelSize = {3x3, 5x5}

1364 networks.



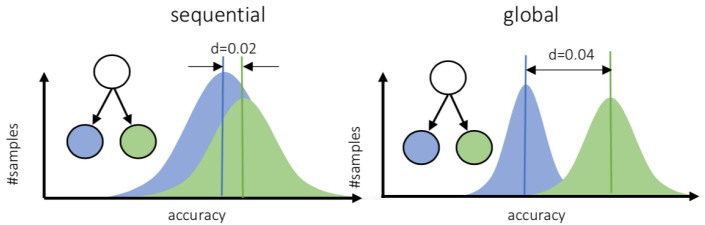
Goal (Neural Architecture Search)

Find the network
with the best accuracy using fewest trials.

Representation of action space

Sequential = { add a layer, set K, set C }

Global = { Set depth, set all K, set all C }

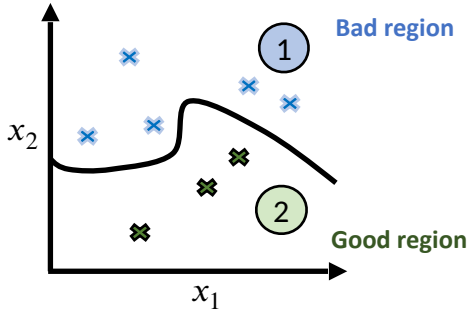


[L. Wang, R. Fonseca, Y. Tian, Learning Search Space Partition for Black-box Optimization using Monte Carlo Tree Search, NeurIPS 2020]
[L. Wang, S. Xie, T. Li, R. Fonseca, Y. Tian, Sample-Efficient Neural Architecture Search by Learning Action Space, TPAMI 2021]

3 Better Search Strategies

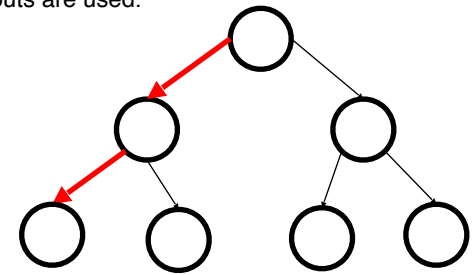
Latent Space Monte Carlo Tree Search (LaMCTS)

(a) Learn the action space.

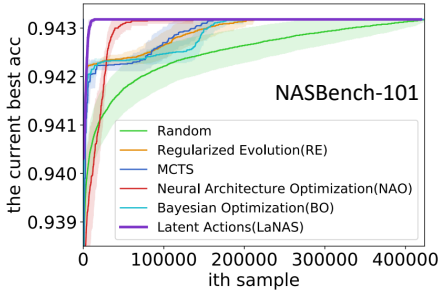


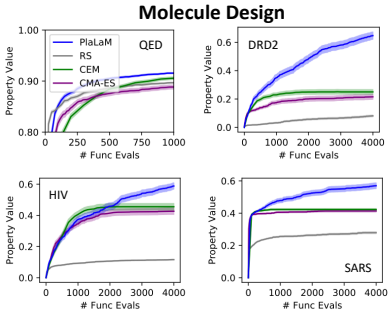
Recursive self-improvement loop

(b) Search using learned action space until a fixed #rollouts are used.

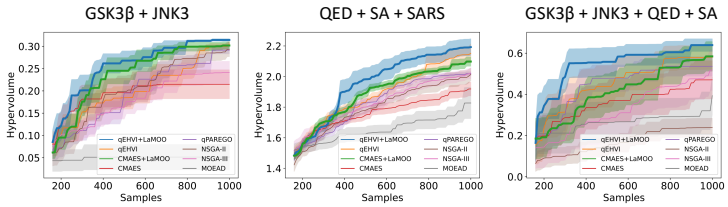


Getting the true quality $f(x)$ for the solution x





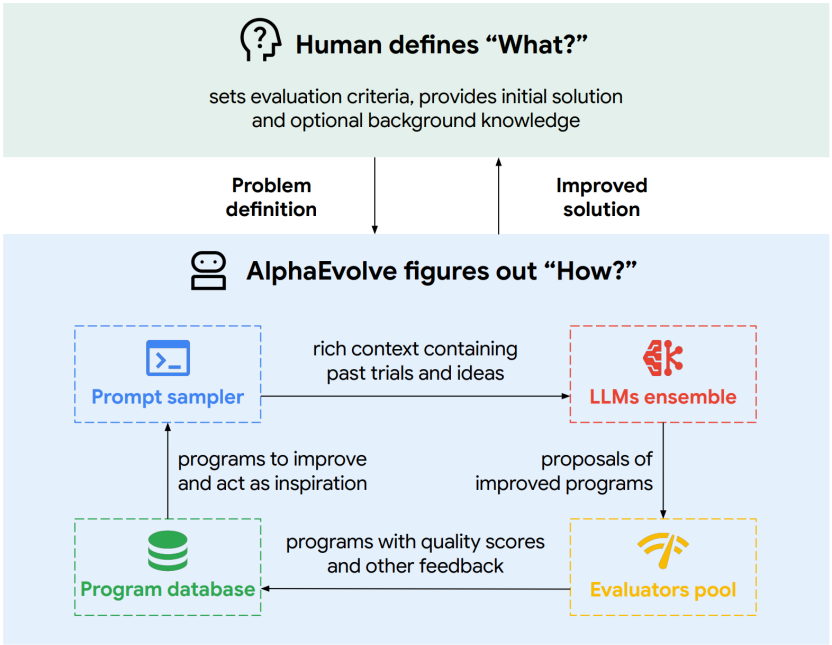
Latent representation learned from unlabeled molecule dataset (1.8M molecules)



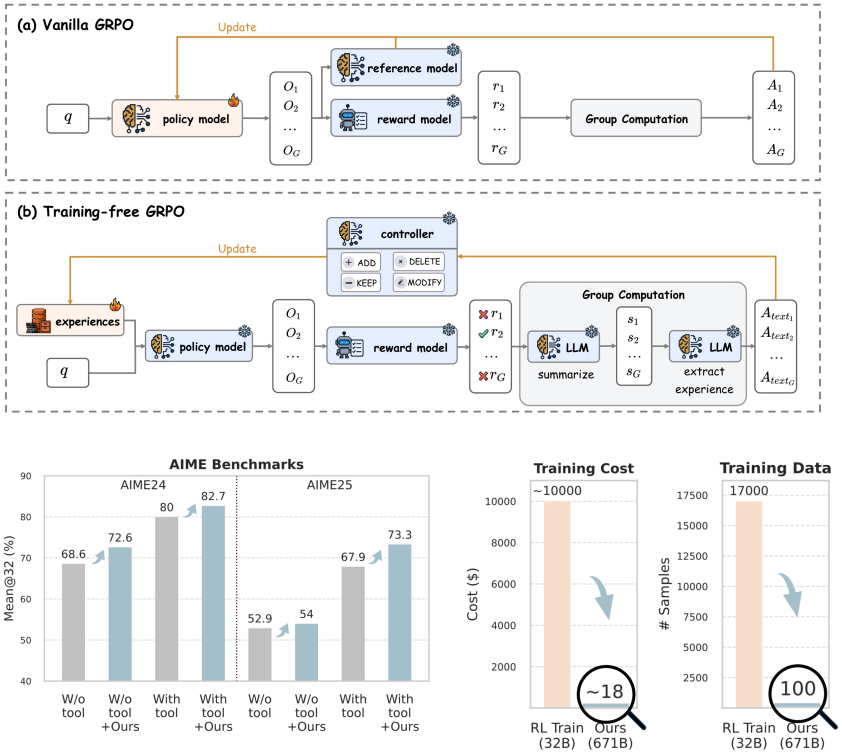
[L. Wang, R. Fonseca, Y. Tian, Learning Search Space Partition for Black-box Optimization using Monte Carlo Tree Search, NeurIPS 2020]
[L. Wang, S. Xie, T. Li, R. Fonseca, Y. Tian, Sample-Efficient Neural Architecture Search by Learning Action Space, TPAMI 2021]
[K. Yang, T. Zhang, ... Y. Tian, Learning Space Partitions for Path Planning, NeurIPS 2021]
[Y. Zhao, ..., Y. Tian, Multi-objective Optimization by Learning Space Partitions, ICLR 2022]

3 Better Search Strategies

Is Reinforcement Learning (RL) the best strategy?

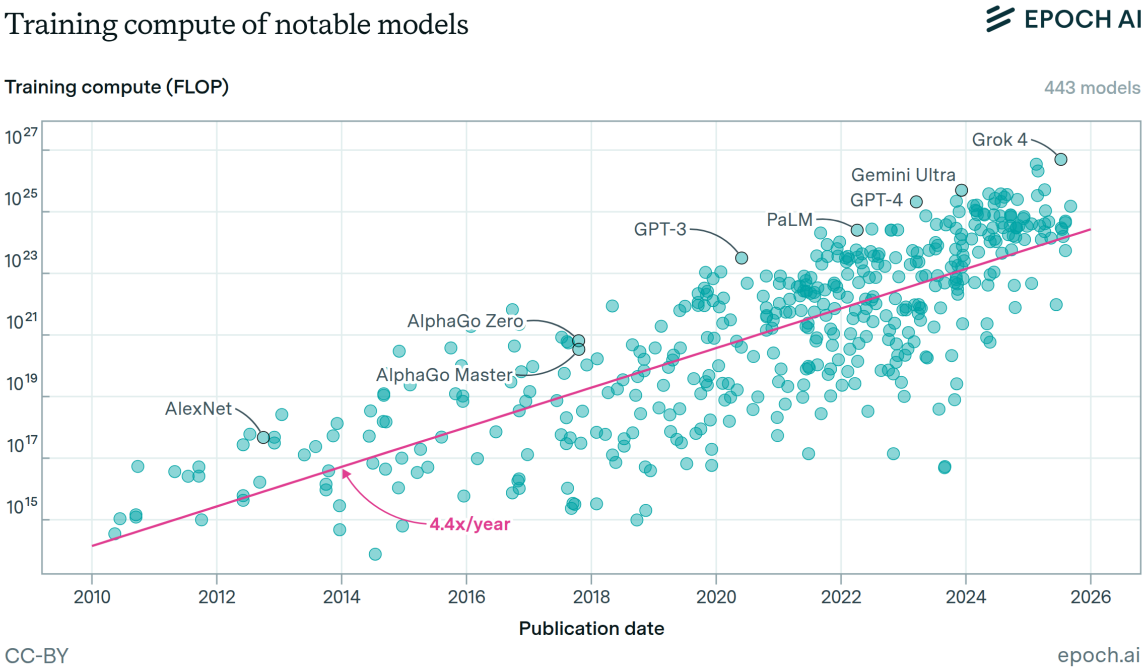


[A. Novikov et al, AlphaEvolve: A coding agent for scientific and algorithmic discovery]



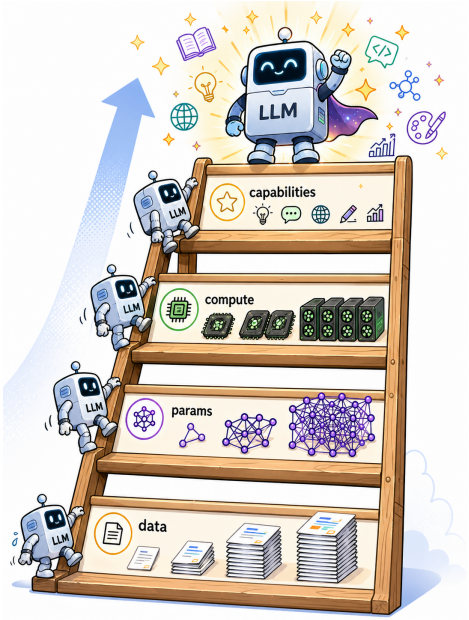
[Y. Cai et al, Training-Free Group Relative Policy Optimization]

4 Cost of Training and Evaluation

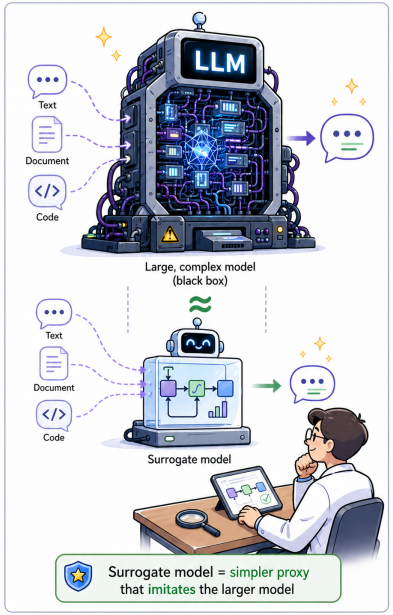


What's the solution if we want to do recursive self-improvement?

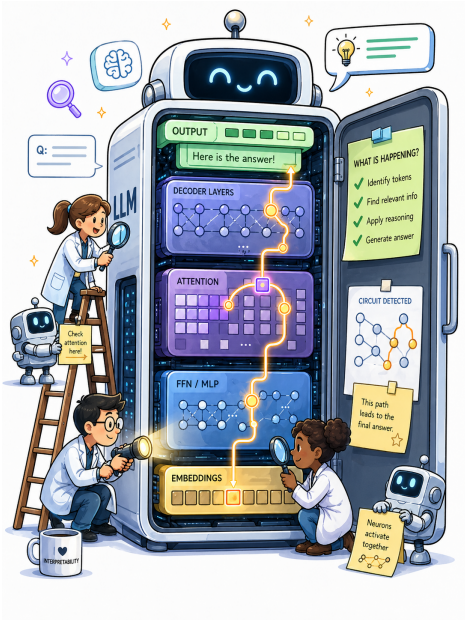
4 Cost of Training and Evaluation



Scaling ladders



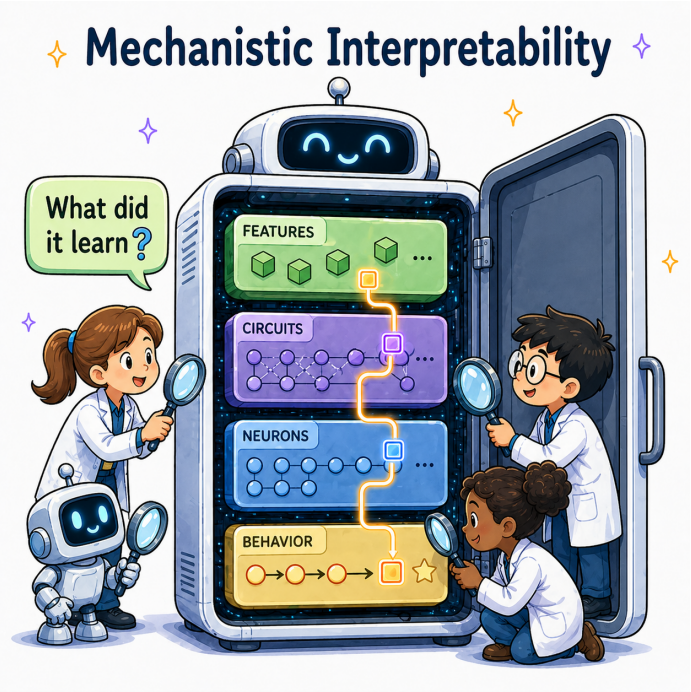
Surrogate models

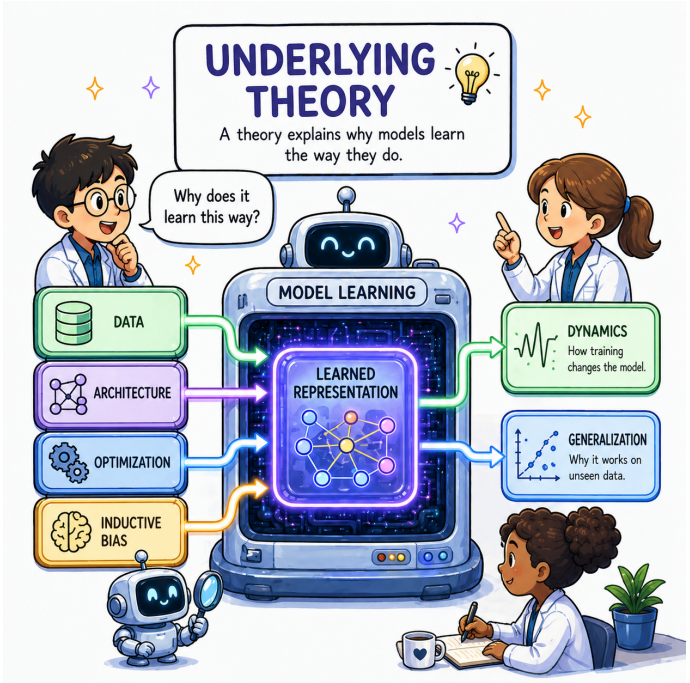


Interpretability

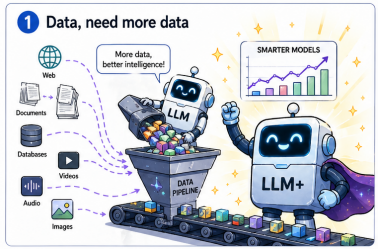
(Open the blackbox)

4 Cost of Training and Evaluation



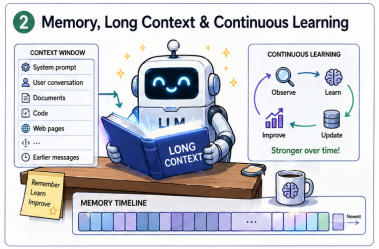


Summary



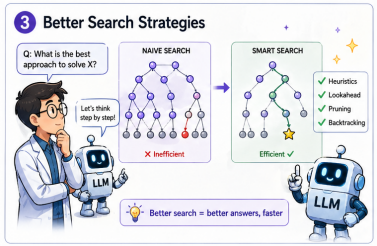
More synthetic data

Environment generation



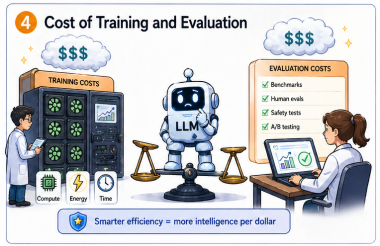
Detailed engineering design

Update weights on the fly



Pay attention to the representation

Is RL the best strategy?

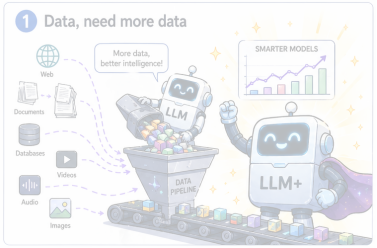


Scaling Ladders

Surrogate Models

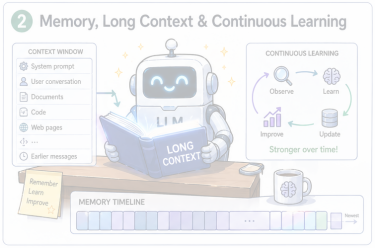
Interpretability

Summary



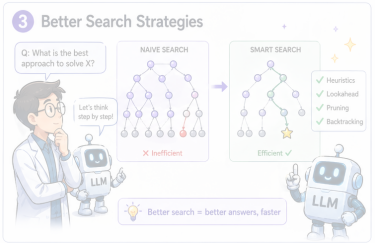
More synthetic data

Environment generation



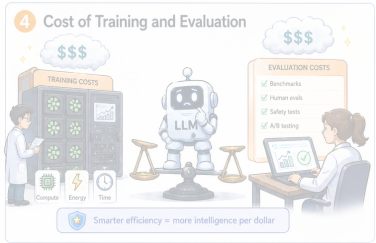
Detailed engineering design

Update weights on the fly



Pay attention to the representation

Is RL the best strategy?



Scaling Ladders

Surrogate Models

Interpretability

Can the research ideas presented in this talk automatically discovered by AI?

Thanks!