Reasoning
LLMs, context and meaning
Programming

Louis ten Bosch

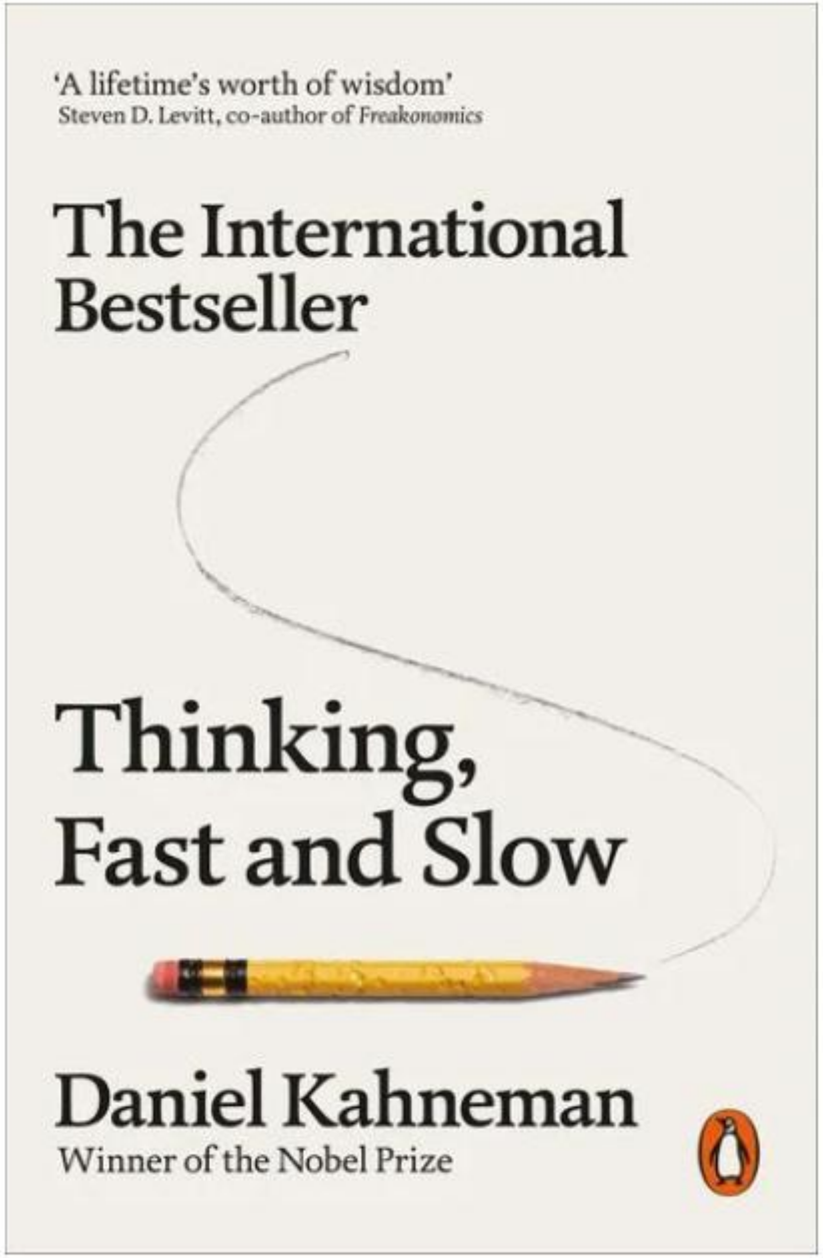
Large Reasoning Models

- LLMs trained on specialized text corpora
  - Containing argumentation and reasoning, “if A then B”
  - Challenge: reasoning in text can be very implicit.

- Specific domains:
  - Math, exact sciences
  - Health
  - Law, finance

Reasoning… small and large scale

- exciting development
- emergence of **hybrid models**
  - combine fast, low-cost “worker” models with slower, more thoughtful “planner” models
  - inspired by *Thinking, Fast and Slow*, the behavioral psychology classic by Daniel Kahneman.
- hybrid systems use lightweight models for routine tasks and heavier reasoning models for deeper problems.
- lends well to AI agent-based frameworks



‘A lifetime’s worth of wisdom’
Steven D. Levitt, co-author of *Freakonomics*

The International
Bestseller

Thinking,
Fast and Slow

Daniel Kahneman
Winner of the Nobel Prize

Current key limitations

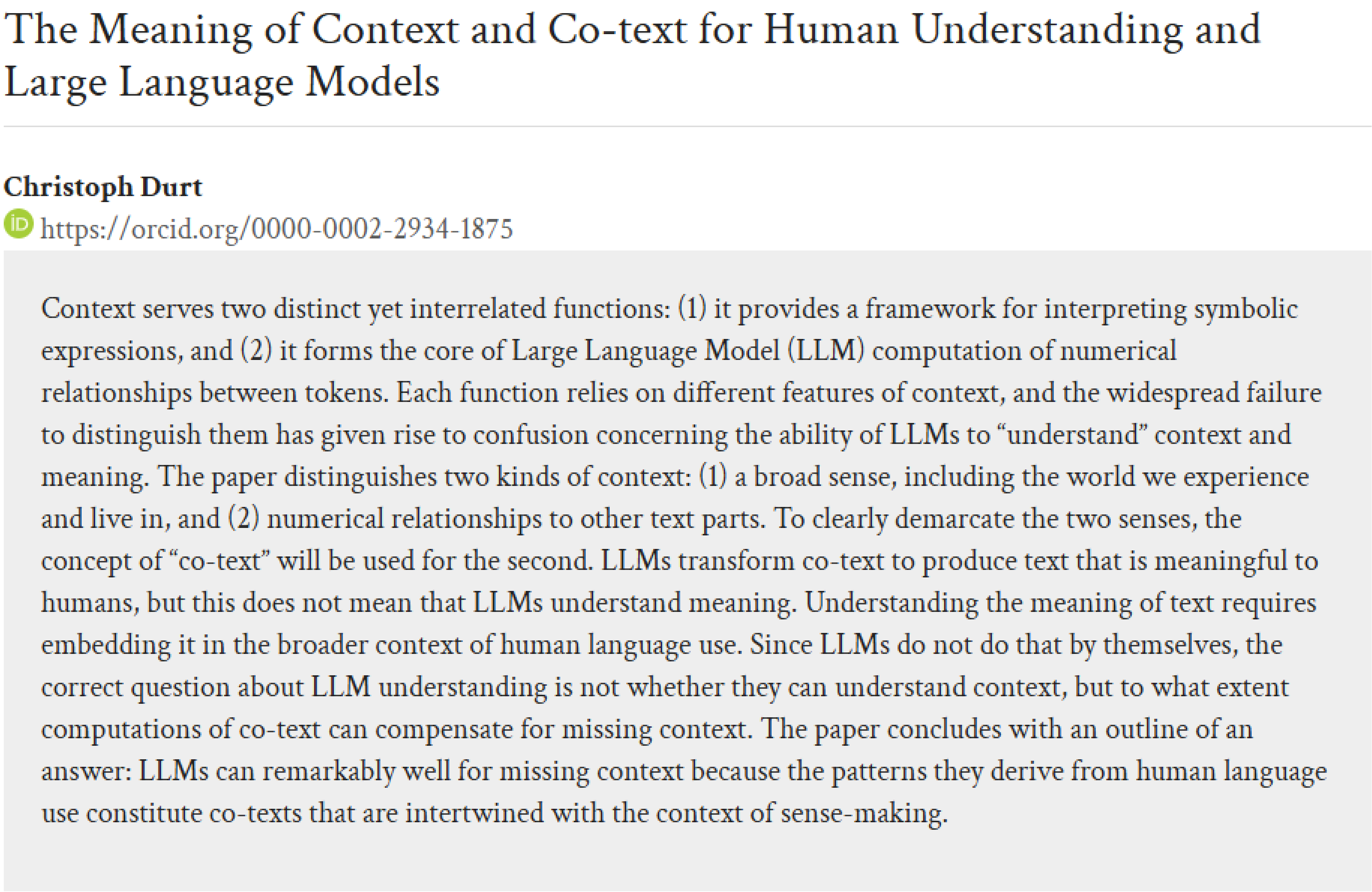
- **Causal reasoning**: These models often still struggle with cause-and-effect relationships, a critical aspect of true reasoning.
- **Increased hallucination risk**: Longer chains of inference can sometimes compound errors.
- **Opacity**: Even when showing their “thinking,” models might provide plausible but incorrect reasoning paths.

LLMs, context and meaning

A complex topic!

Does an LLM ‘understand’ something?

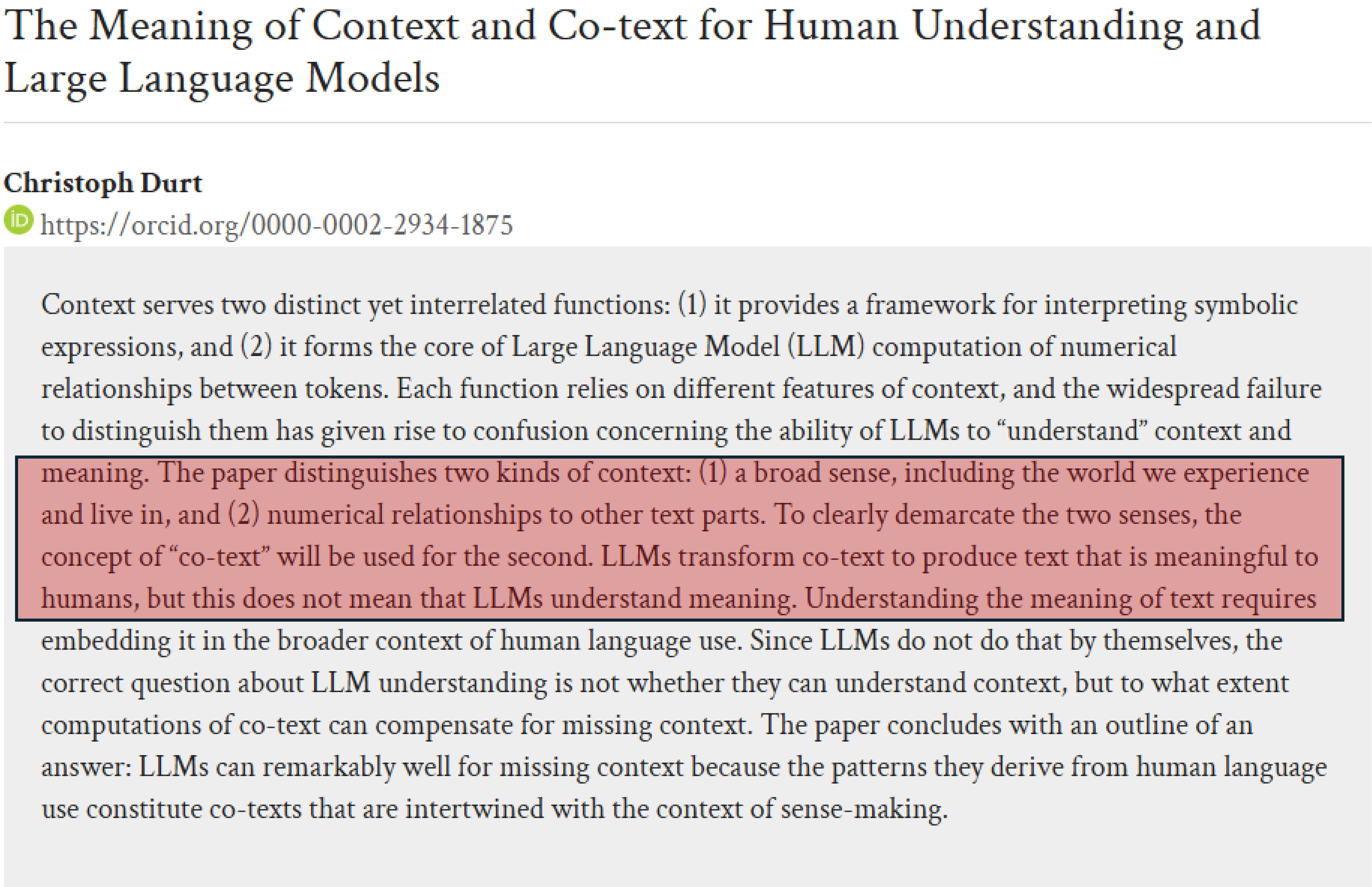
- context
- co-text
- Human understanding versus LLMs
  - Cristoph Durt (2025). The Meaning of Context and Co-text for Human Understanding and Large Language Models. https://journals.ub.uni-koeln.de/index.php/phidi/article/view/11666



The Meaning of Context and Co-text for Human Understanding and Large Language Models

Christoph Durt
https://orcid.org/0000-0002-2934-1875

Context serves two distinct yet interrelated functions: (1) it provides a framework for interpreting symbolic expressions, and (2) it forms the core of Large Language Model (LLM) computation of numerical relationships between tokens. Each function relies on different features of context, and the widespread failure to distinguish them has given rise to confusion concerning the ability of LLMs to "understand" context and meaning. The paper distinguishes two kinds of context: (1) a broad sense, including the world we experience and live in, and (2) numerical relationships to other text parts. To clearly demarcate the two senses, the concept of "co-text" will be used for the second. LLMs transform co-text to produce text that is meaningful to humans, but this does not mean that LLMs understand meaning. Understanding the meaning of text requires embedding it in the broader context of human language use. Since LLMs do not do that by themselves, the correct question about LLM understanding is not whether they can understand context, but to what extent computations of co-text can compensate for missing context. The paper concludes with an outline of an answer: LLMs can remarkably well for missing context because the patterns they derive from human language use constitute co-texts that are intertwined with the context of sense-making.



The Meaning of Context and Co-text for Human Understanding and Large Language Models

Christoph Durt
https://orcid.org/0000-0002-2934-1875

Context serves two distinct yet interrelated functions: (1) it provides a framework for interpreting symbolic expressions, and (2) it forms the core of Large Language Model (LLM) computation of numerical relationships between tokens. Each function relies on different features of context, and the widespread failure to distinguish them has given rise to confusion concerning the ability of LLMs to "understand" context and meaning. The paper distinguishes two kinds of context: (1) a broad sense, including the world we experience and live in, and (2) numerical relationships to other text parts. To clearly demarcate the two senses, the concept of "co-text" will be used for the second. LLMs transform co-text to produce text that is meaningful to humans, but this does not mean that LLMs understand meaning. Understanding the meaning of text requires embedding it in the broader context of human language use. Since LLMs do not do that by themselves, the correct question about LLM understanding is not whether they can understand context, but to what extent computations of co-text can compensate for missing context. The paper concludes with an outline of an answer: LLMs can remarkably well for missing context because the patterns they derive from human language use constitute co-texts that are intertwined with the context of sense-making.

Durt (2025): co-text and context

- Durt’s paper distinguishes two kinds of context: \
  - (1) a broad sense, including the world we experience and live in, and
  - (2) numerical relationships to other text parts.

- To clearly demarcate the two senses, Durt uses the term “co-text” for the second.

Durt (2025): co-text and context

- LLMs transform co-text to produce text that is meaningful to humans, but this does not mean that LLMs understand meaning

- Understanding the meaning of text requires embedding it in the broader context of human language use.

Durt (2025): co-text and context

- Since LLMs do not do that by themselves, the correct question about LLM understanding is not whether they can understand context, but to what extent computations of co-text can compensate for missing context.

- Durt's paper concludes with an outline of an answer: LLMs do remarkably well for missing context because the patterns they derive from human language use constitute co-texts

Context

Context serves two distinct yet interrelated functions:

- (1) it provides a framework for interpreting symbolic expressions, and
- (2) it forms the core of Large Language Model (LLM) computation of numerical relationships between tokens.

LLMs versus LRMs

Aspect	Large language models (LLMs)	Large reasoning models (LRMs)
Core function	Learn patterns in data to generate fluent, human-like text	Extend LLMs to solve problems requiring logical reasoning and contextual understanding
Use cases	Content generation, language translation, summarization, user queries	Math problem solving, interpreting ambiguous clinical data, complex decision-making
Reasoning ability	Limited structured reasoning; may struggle with multi-step logic or ambiguity	Trained to apply consistent reasoning steps; outputs are more logical and verifiable
Performance	Fast response times; optimized for scalability and speed	Slower response times; requires more processing to reason through problems step by step

How large reasoning models (LRMs) work

Large reasoning models (LRMs) use a combination of training methods and prompt strategies.

- Training on enriched datasets
  - LRMs are trained on datasets that include not just language patterns but also examples designed to teach reasoning... This helps the model learn both the correct outputs and the reasoning steps needed to reach them.
- Reinforcement learning (RL)
  - the model is rewarded for correct or logically consistent answers and penalized for incorrect ones. This helps reinforce desirable reasoning patterns.

How large reasoning models (LRMs) work

- Human feedback (RLHF)
  - human reviewers guide and refine the model’s outputs
  - nuanced reasoning strategies that align with domain expertise — especially valuable in fields like healthcare and finance
- Prompt engineering
  - help guide the model through multi-step tasks or layered questions. Too generic prompts may not trigger reasoning behavior
  - Chain-of-thought (CoT) prompting: CoT prompting is a method that explicitly encourages the model to break a problem into smaller steps and explore multiple possible reasoning paths.

Types of reasoning in large reasoning models

Four main types

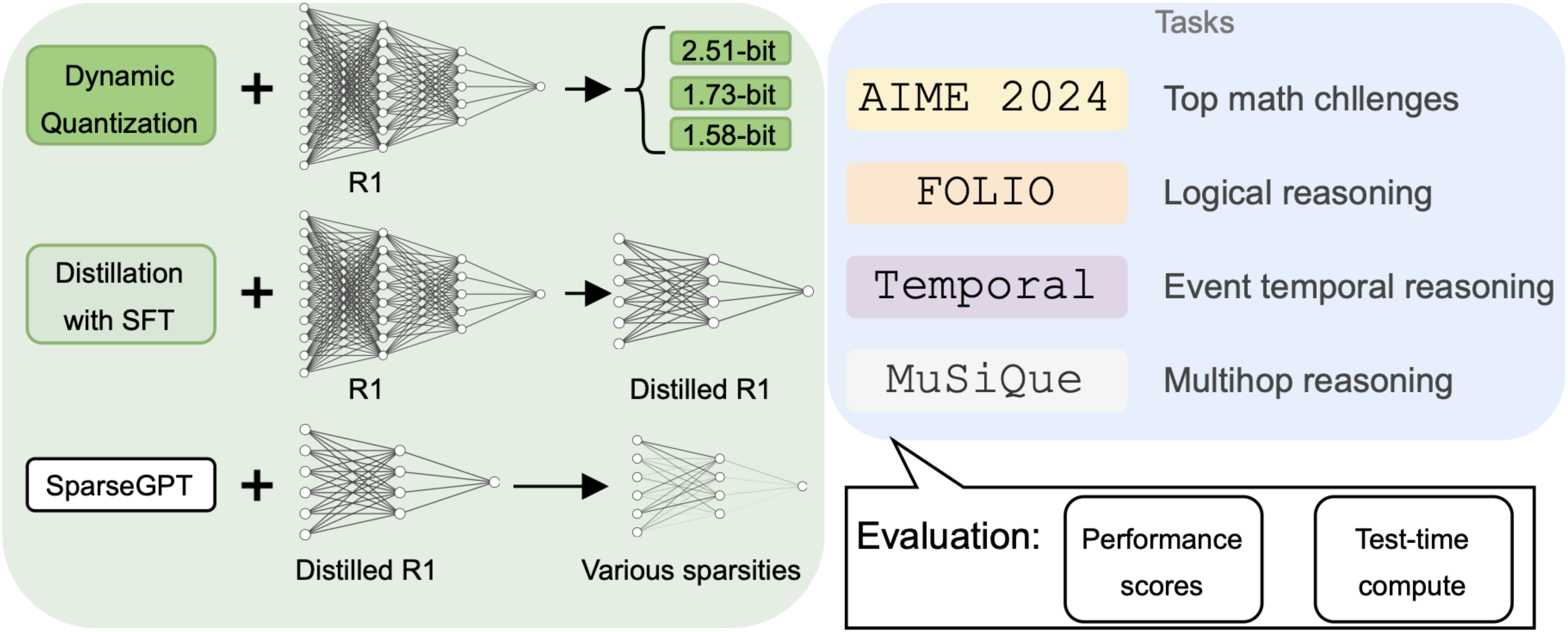
- Deductive reasoning
  - Deductive reasoning applies general rules to specific cases to reach logically certain conclusions. It is best suited for tasks that require strict adherence to established rules or facts — also known as *top-down reasoning*.
  - Models leveraging deductive reasoning excel at structured, rule-based tasks, as they prioritize logical consistency in generating outputs. This makes them valuable in high-accuracy domains, such as regulatory compliance checks or medical protocol analysis.
- Inductive reasoning
  - Inductive reasoning draws general conclusions from specific observations in training data. LRMs identify patterns and trends to generalize across new, unseen inputs.
  - Inductive reasoning is less rigid than deductive reasoning and supports probabilistic predictions rather than guaranteed outcomes. It is especially useful in dynamic, data-rich scenarios — such as fraud detection — but may yield less reliable results

Types of reasoning in large reasoning models

- Abductive reasoning
  - Abductive reasoning involves inferring the most likely explanation based on available evidence – particularly useful when data is incomplete or uncertain.
  - LRMs using abductive reasoning generate hypotheses rather than definitive answers.
  - This approach is useful in domains like medical diagnostics, where timely, plausible interpretations of ambiguous symptoms are often needed.
- Analogical reasoning
  - Analogical reasoning involves identifying similarities between different situations or datasets and applying insights from one context to another.
  - LRMs may recognize relational patterns across examples and transfer learned associations to novel inputs.

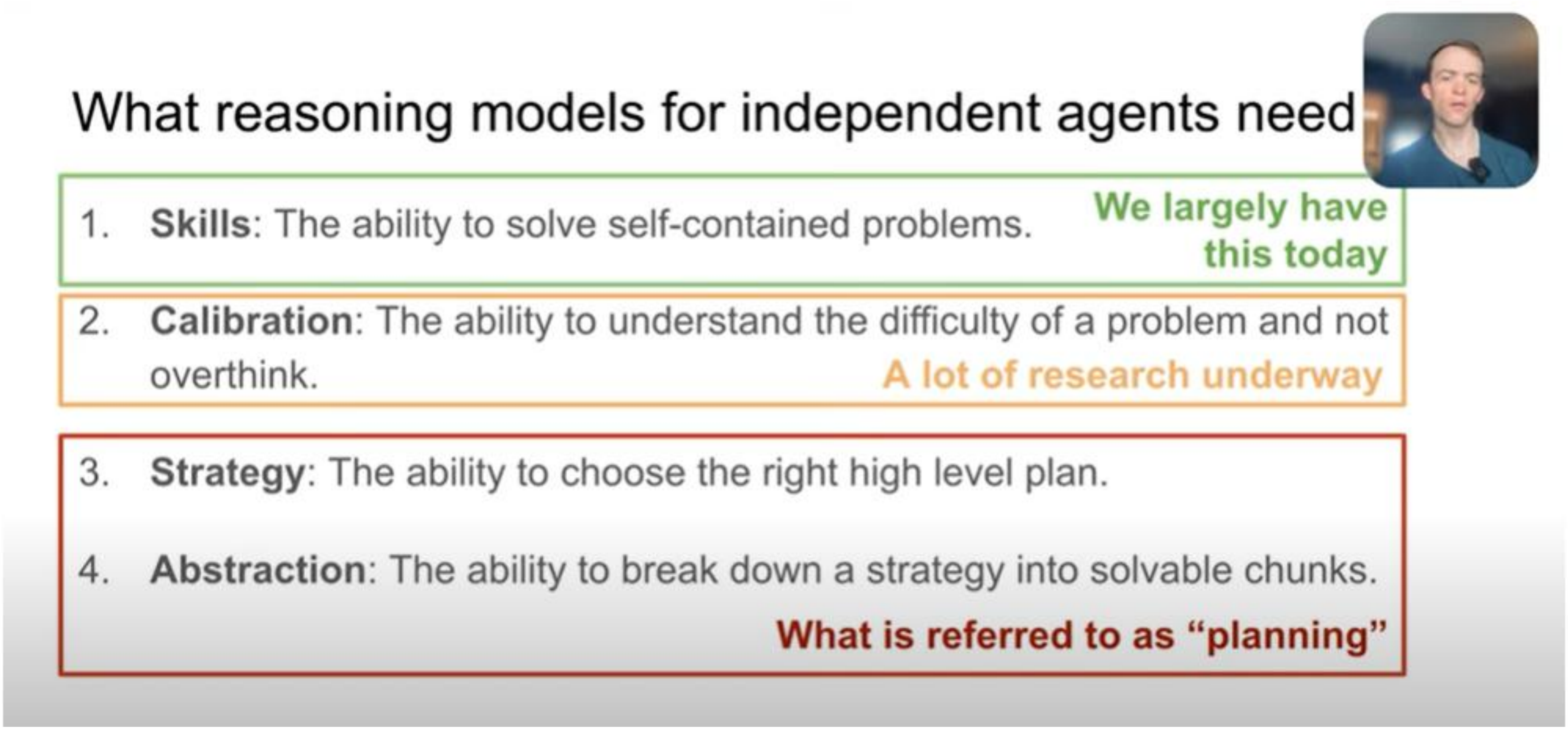
Benchmarking

- Very dynamic field, see e.g., https://arxiv.org/abs/2504.02010 (May 2025)
    - benchmarks compressed DeepSeek-R1 models, using quantization, distillation, and pruning methods.
- Reasoning datasets:
    - AIME 2024, FOLIO, Temporal Sequences of BIG-Bench Hard, and MuSiQue, ranging from mathematical to multihop reasoning
- Models:
    - 2.51-, 1.73-, and 1.58-bit R1 (models that adopt dynamic quantization)
    - distilled R1 models that are based on LLaMA or Qwen and run SparseGPT on them to obtain various sparsity levels
- Assessment
    - performance scores, computing time, number of tokens spent on each question
    - effect of parameter count on knowledge memorization
    - effect of parameter count on reasoning capability


Dynamic Quantization
+
R1
2.51-bit
1.73-bit
1.58-bit
Distillation with SFT
+
R1
Distilled R1
SparseGPT
+
Distilled R1
Various sparsities
Tasks
AIME 2024
Top math chllenges
FOLIO
Logical reasoning
Temporal
Event temporal reasoning
MuSiQue
Multihop reasoning
Evaluation:
Performance scores
Test-time compute

More about reasoning

- https://www.youtube.com/watch?v=DZFEX3ppflk
- Nathan Lambert



What reasoning models for independent agents need

1. **Skills**: The ability to solve self-contained problems.

We largely have this today

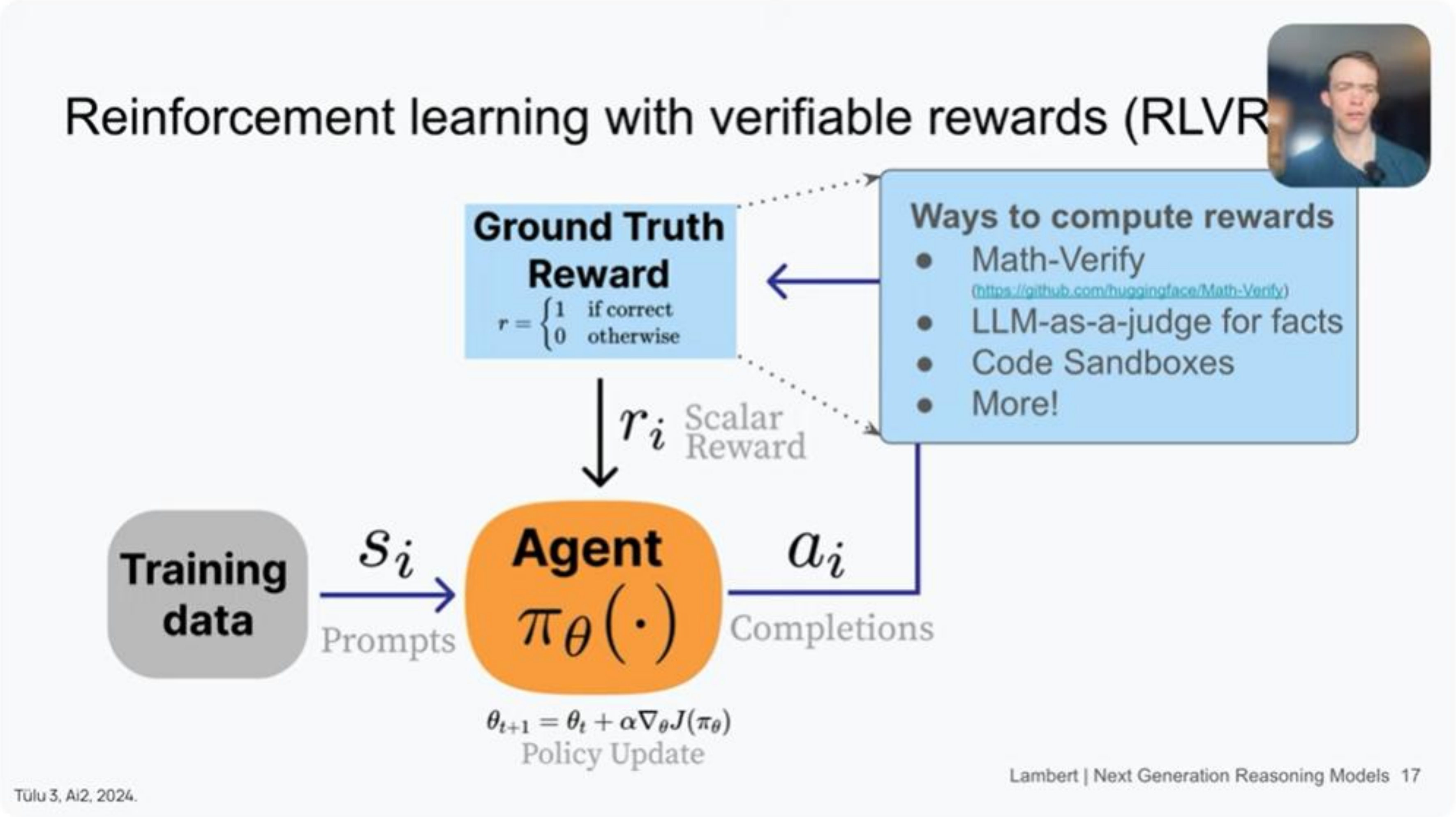
2. **Calibration**: The ability to understand the difficulty of a problem and not overthink.

A lot of research underway

3. **Strategy**: The ability to choose the right high level plan.

4. **Abstraction**: The ability to break down a strategy into solvable chunks.

What is referred to as “planning”



Reinforcement learning with verifiable rewards (RLVR

Ground Truth Reward

$r = \begin{cases} 1 & \text{if correct} \\ 0 & \text{otherwise} \end{cases}$

Ways to compute rewards

- Math-Verify (https://github.com/huggingface/Math-Verify)
- LLM-as-a-judge for facts
- Code Sandboxes
- More!

r_i Scalar Reward

Training data

s_i Prompts

Agent $\pi_\theta(\cdot)$

a_i Completions

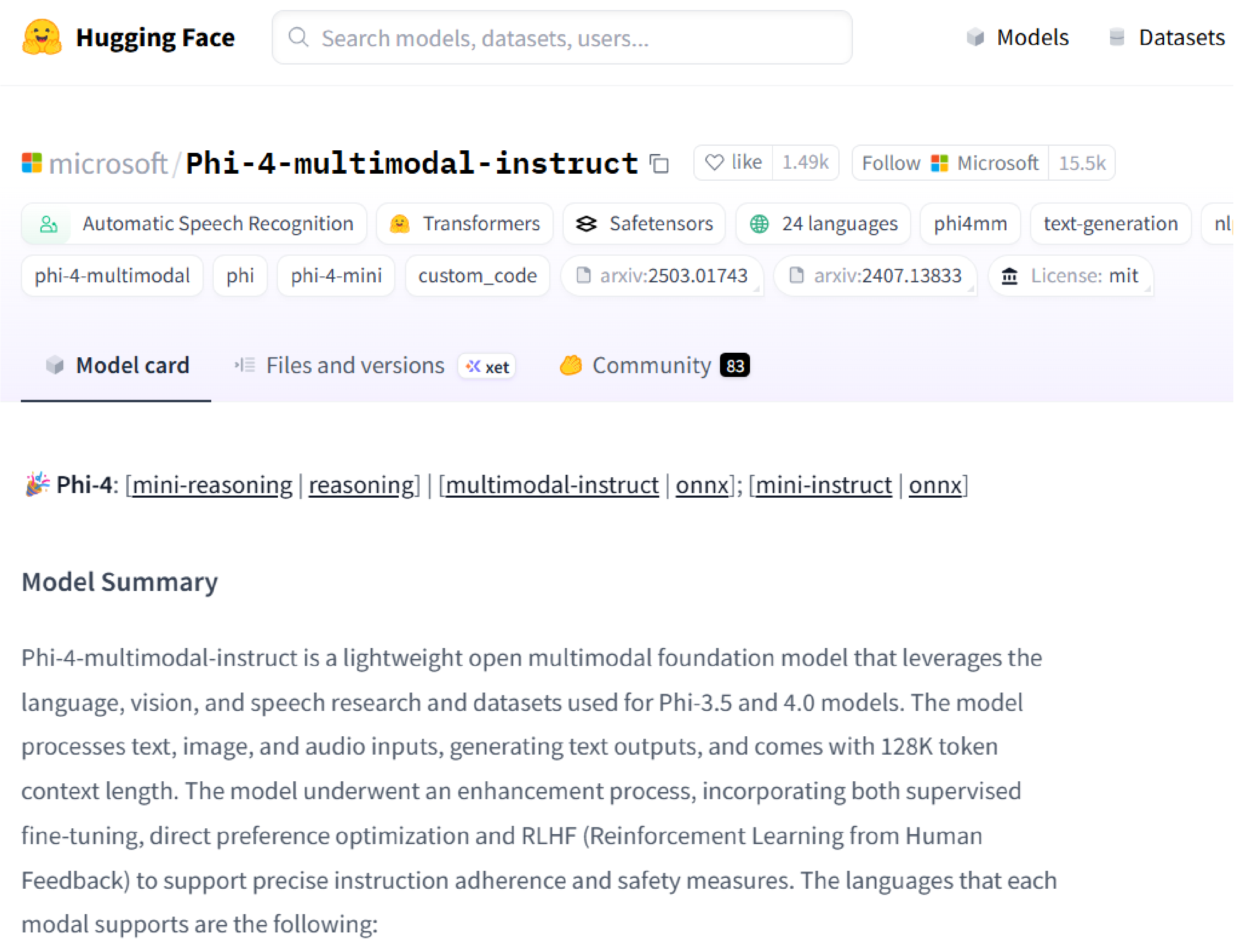
$\theta_{t+1} = \theta_t + \alpha \nabla_\theta J(\pi_\theta)$

Policy Update

Tülu 3, Ai2, 2024.

Lambert | Next Generation Reasoning Models 17

Traits of next generation reasoning models



🎉 **Phi-4**: [mini-reasoning | reasoning] | [multimodal-instruct | onnx]; [mini-instruct | onnx]

Model Summary

Phi-4-multimodal-instruct is a lightweight open multimodal foundation model that leverages the language, vision, and speech research and datasets used for Phi-3.5 and 4.0 models. The model processes text, image, and audio inputs, generating text outputs, and comes with 128K token context length. The model underwent an enhancement process, incorporating both supervised fine-tuning, direct preference optimization and RLHF (Reinforcement Learning from Human Feedback) to support precise instruction adherence and safety measures. The languages that each modal supports are the following:

Dynamic field: Phi-4-MM is one of the recent models

- lightweight open multimodal foundation model that leverages the language, vision, and speech research and datasets used for Phi-3.5 and 4.0 models
- processes text, image, and audio inputs, generating text outputs, and comes with 128K token context length
- supervised fine-tuning, direct preference optimization and RLHF (Reinforcement Learning from Human Feedback)

What comes next?

- the landscape is likely to fragment, not consolidate.
    - rather than a one-size-fits-all universal reasoning model, expect a wave of **domain-specific, task-optimized reasoning agents**
        - legal workflows, scientific research, financial forecasting
    - niche models will likely outperform general-purpose systems in their domain
- This trend mirrors broader shifts in AI
    - from monolithic to modular
    - from general to specific
    - from centralized to customizable

What about programming languages…

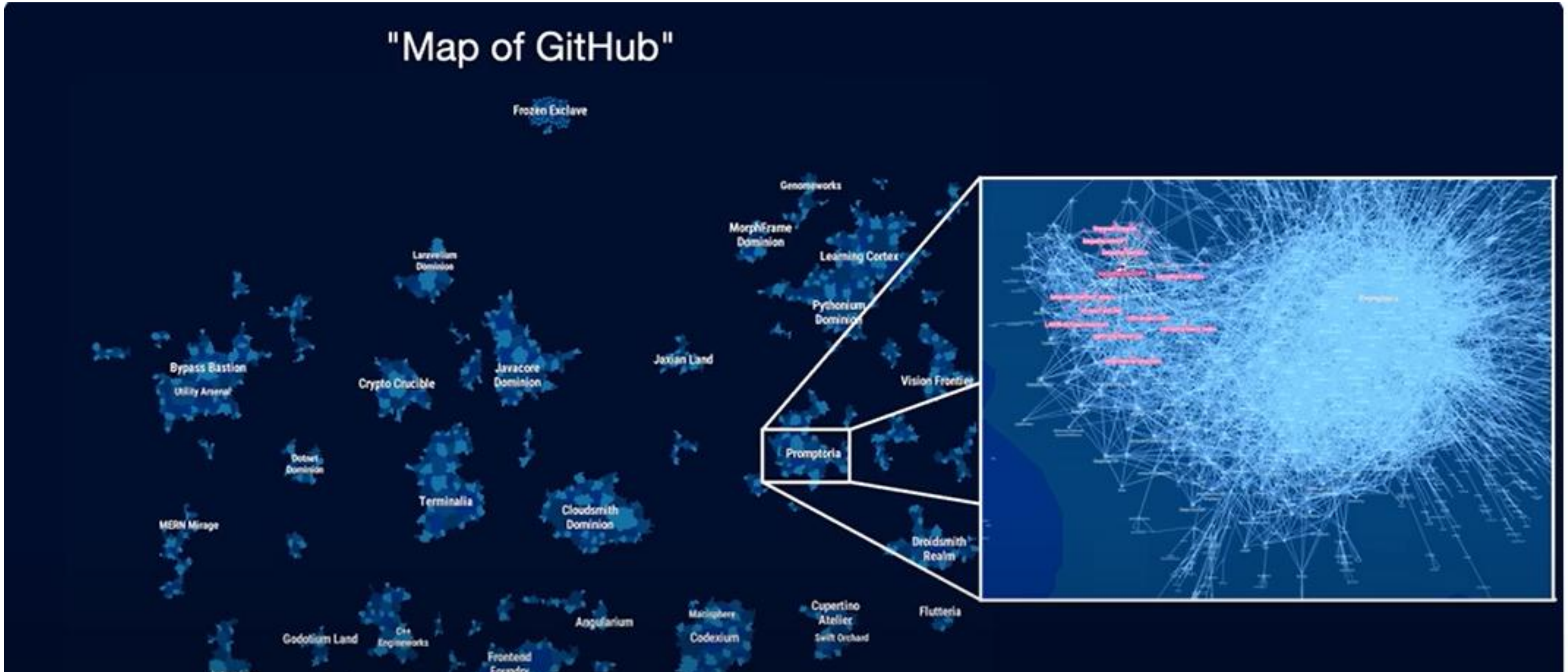
- Andrej Karpathy's keynote on June 17, 2025 at AI Startup School in San Francisco

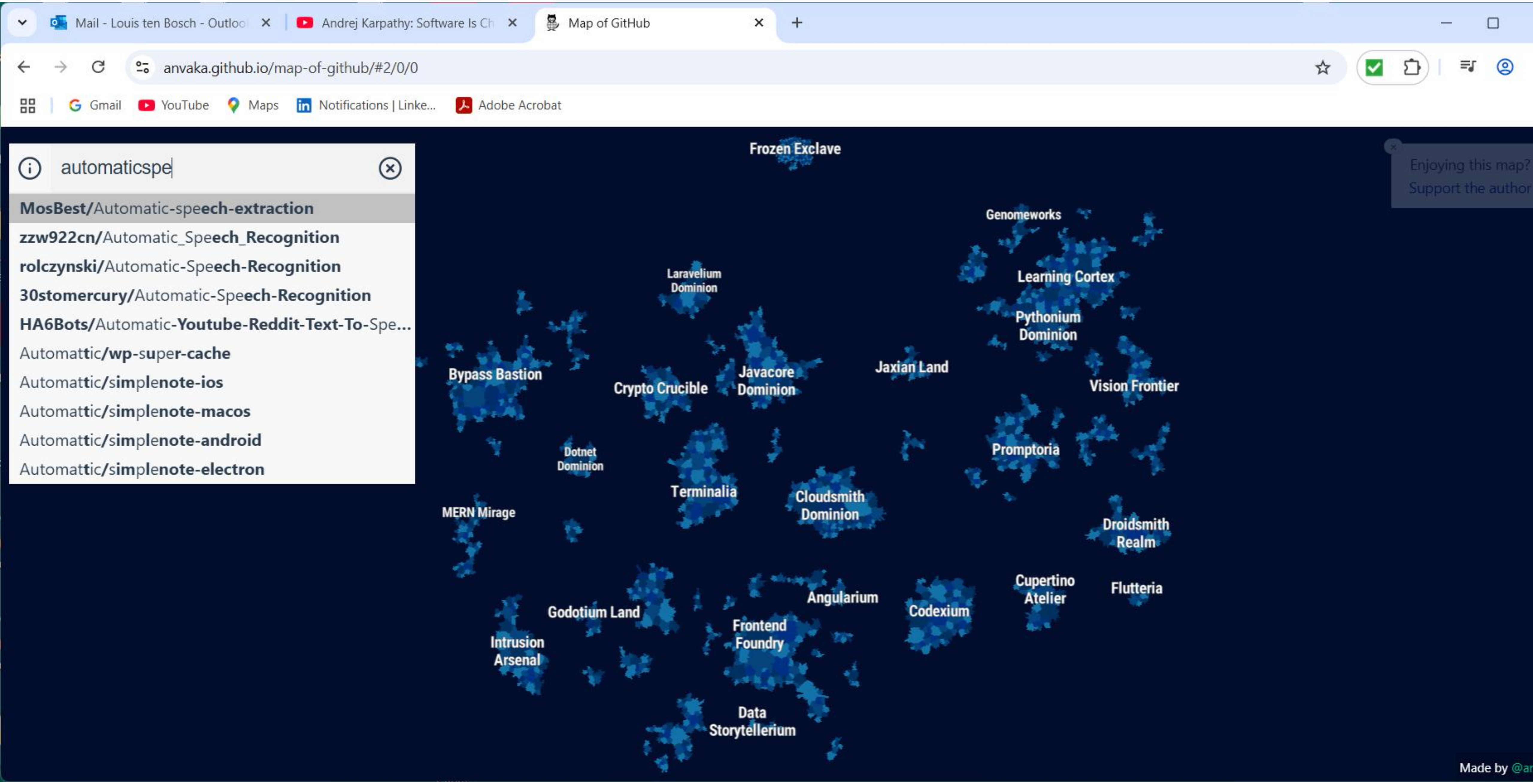
- Software 1.0, software 2.0, software 3.0

- https://www.youtube.com/watch?v=LCEmiRjPEtQ

Software is changing rapidly

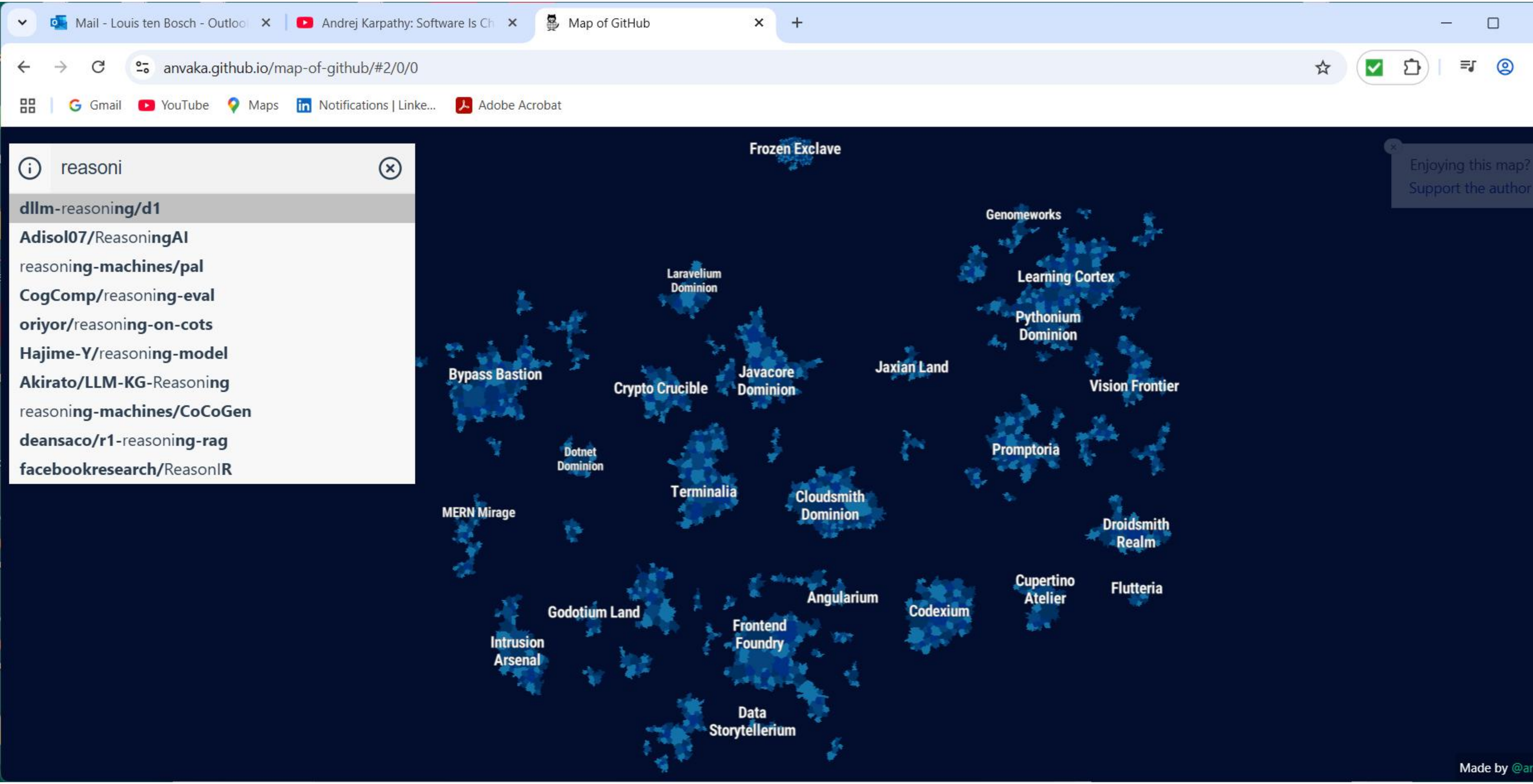
- “Map of Github”





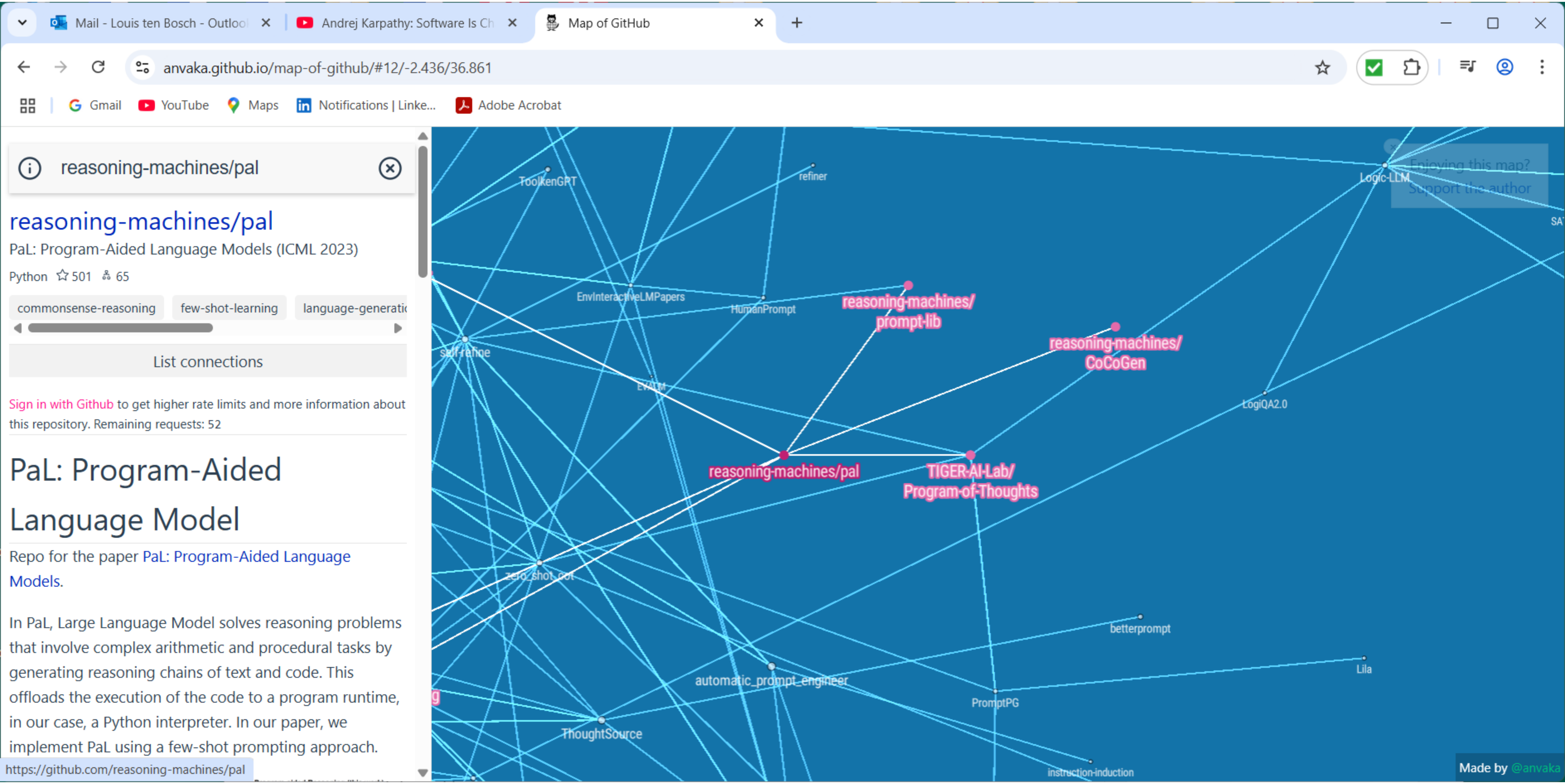
automaticspe

- MosBest/Automatic-speech-extraction
- zzw922cn/Automatic_Speech_Recognition
- rolczynski/Automatic-Speech-Recognition
- 30stomercury/Automatic-Speech-Recognition
- HA6Bots/Automatic-Youtube-Reddit-Text-To-Spe...
- Automattic/wp-super-cache
- Automattic/simplenote-ios
- Automattic/simplenote-macos
- Automattic/simplenote-android
- Automattic/simplenote-electron



reasoni

dllm-reasoning/d1
Adisol07/ReasoningAI
reasoning-machines/pal
CogComp/reasoning-eval
oriyor/reasoning-on-cots
Hajime-Y/reasoning-model
Akirato/LLM-KG-Reasoning
reasoning-machines/CoCoGen
deansaco/r1-reasoning-rag
facebookresearch/ReasonIR



anvaka.github.io/map-of-github/#12/-2.436/36.861

reasoning-machines/pal

reasoning-machines/pal

PaL: Program-Aided Language Models (ICML 2023)

Python ☆ 501 65

commonsense-reasoning few-shot-learning language-generati

List connections

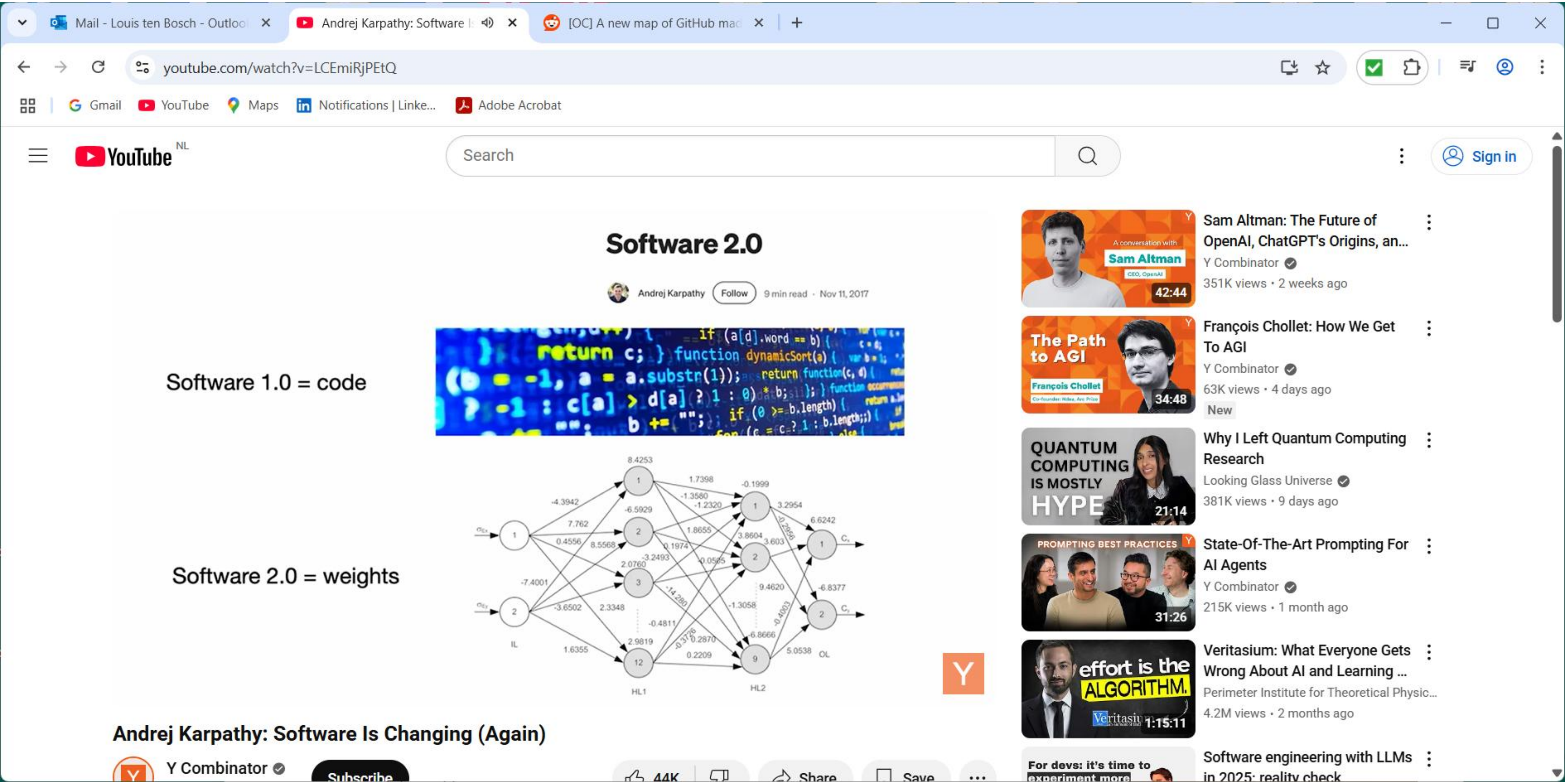
Sign in with Github to get higher rate limits and more information about this repository. Remaining requests: 52

PaL: Program-Aided Language Model

Repo for the paper PaL: Program-Aided Language Models.

In PaL, Large Language Model solves reasoning problems that involve complex arithmetic and procedural tasks by generating reasoning chains of text and code. This offloads the execution of the code to a program runtime, in our case, a Python interpreter. In our paper, we implement PaL using a few-shot prompting approach.

https://github.com/reasoning-machines/pal



Mail - Louis ten Bosch - Outloo
Andrej Karpathy: Software I
[OC] A new map of GitHub mac
youtube.com/watch?v=LCEmiRjPEtQ
Gmail
YouTube
Maps
Notifications | Linke...
Adobe Acrobat
YouTube NL
Search
Sign in

Software 2.0

Andrej Karpathy Follow 9 min read · Nov 11, 2017

Software 1.0 = code

Software 2.0 = weights

Andrej Karpathy: Software Is Changing (Again)

Y Combinator

Sam Altman: The Future of OpenAI, ChatGPT's Origins, an...
Y Combinator
351K views • 2 weeks ago

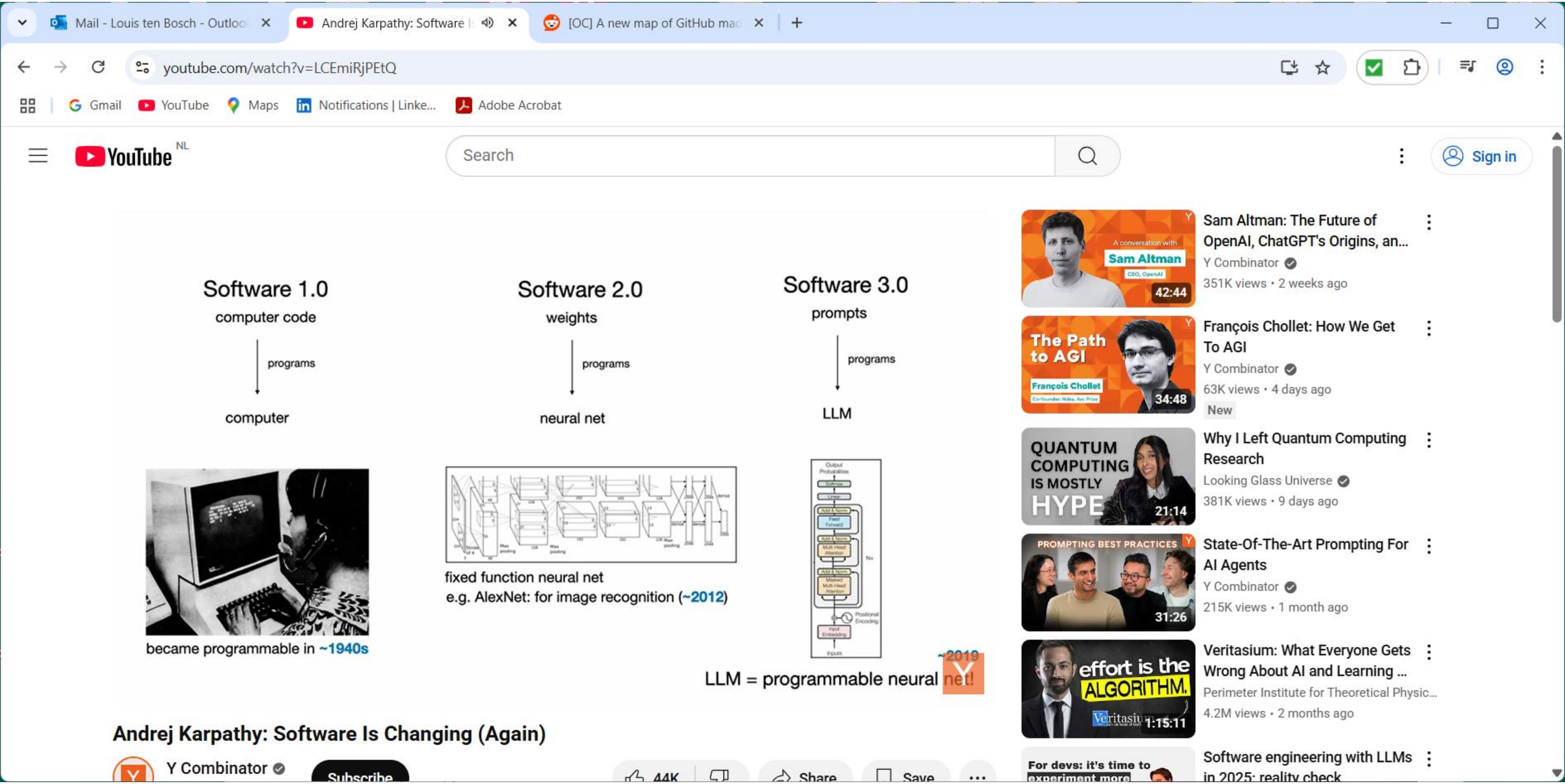
François Chollet: How We Get To AGI
Y Combinator
63K views • 4 days ago
New

Why I Left Quantum Computing Research
Looking Glass Universe
381K views • 9 days ago

State-Of-The-Art Prompting For AI Agents
Y Combinator
215K views • 1 month ago

Veritasium: What Everyone Gets Wrong About AI and Learning ...
Perimeter Institute for Theoretical Physic...
4.2M views • 2 months ago

Software engineering with LLMs in 2025: reality check



Mail - Louis ten Bosch - Outlook
Andrej Karpathy: Software I
[OC] A new map of GitHub mac
youtube.com/watch?v=LCEmiRjPEtQ
Gmail
YouTube
Maps
Notifications | Linke...
Adobe Acrobat
YouTube NL
Search
Sign in

Software 1.0

computer code

programs

computer

became programmable in ~1940s

Software 2.0

weights

programs

neural net

fixed function neural net
e.g. AlexNet: for image recognition (~2012)

Software 3.0

prompts

programs

LLM

~2019

LLM = programmable neural net!

Andrej Karpathy: Software Is Changing (Again)

Y Combinator

Sam Altman: The Future of OpenAI, ChatGPT's Origins, an...
Y Combinator
351K views • 2 weeks ago
42:44

François Chollet: How We Get To AGI
Y Combinator
63K views • 4 days ago
New
34:48

Why I Left Quantum Computing Research
Looking Glass Universe
381K views • 9 days ago
21:14

State-Of-The-Art Prompting For AI Agents
Y Combinator
215K views • 1 month ago
31:26

Veritasium: What Everyone Gets Wrong About AI and Learning ...
Perimeter Institute for Theoretical Physic...
4.2M views • 2 months ago
1:15:11

Software engineering with LLMs in 2025: reality check

A conversation with Sam Altman CEO, OpenAI
The Path to AGI
François Chollet
QUANTUM COMPUTING IS MOSTLY HYPE
PROMPTING BEST PRACTICES
effort is the ALGORITHM.
Veritasium
For devs: it's time to experiment more