The Document Manipulation Benchmark

On the limits of agentic document manipulation

Adrian de Wynter
Microsoft & The University of York

adewynter@microsoft.com

Microsoft

The Problem

There's two, actually:

1. There is a **major** disconnect between an LLM's theoretical capabilities, and their real-world impact.
    - For example, some problems which cannot be 'solved', LLMs can do reasonably well (e.g. PARITY).
    - We haven't characterised agentic systems well, either.

2. Evaluation is broken!
    - We tend to evaluate *really easy* (and artificial) *tasks.*
    - Benchmarks tend to get contaminated a little too fast for my taste
    - No further comments.

Reza Davari

Saurabh Kumar Pandey

Vishwas Suryanarayanan

Kartik Fnu

Shruthi Ramesh

Utkarsh Garg

Lukasz Koprowski

Si-Qing Chen

Afra Mashhadi*

*Also with UW

Microsoft

Content

- Mathematical background
  - On the limits of theory: they don't mix that well
  - Agentic complexity
- Aside: the problem of evaluation
- The Document Manipulation Benchmark (TDMB)
  - Why document manipulation?
  - Building a robust benchmark
- Preliminary results
  - Agentic complexity in TDMB
- Future work

Content

- Mathematical background
  - On the limits of theory: they don't mix that well
  - Agentic complexity
- Aside: the problem of evaluation
- The Document Manipulation Benchmark (TDMB)
  - Why document manipulation?
  - Building a robust benchmark
- Preliminary results
  - Agentic complexity in TDMB
- Future work

Not all agents are the same

- Many proposed complexity definitions focus on budget, time, etc, not on whether they will ever be able to **do** something.

Observe, for example, how an LLM *could* be a partial recursive function:

$$LLM(x_{i+1}) = \bigoplus_{j \in \{1,\ldots i\}} LLM(x_j)$$

But this is useless since it always writes to stdout.

Adding functionality (a tool) helps, but: (1) it factors in reliability; and (2) solves the *one* problem.

How *can* we measure all of this?

Motivation: LLM limitations (I)

- Problems can be categorised in various ways.

- Formal languages under the Chomsky hierarchy are defined by the complexity of the grammar generating them.

- Circuits are another, categorised based on the minimum size of a circuit computing a function (restricted by the type of gate)

$$\mathsf{NC}^0 \subseteq \mathsf{AC}^0 \subseteq \mathsf{AC}^0[m] \subseteq \mathsf{ACC} \subseteq \mathsf{TC}^0 \subseteq \mathsf{NC}^1 \subseteq \mathsf{AC}^1 \subseteq \mathsf{NC}^2 \subseteq \mathsf{AC}^2 \subseteq \cdots \subseteq \mathsf{P/poly}.$$

Motivation: LLM limitations (II)

- A simple LLM cannot solve things an agent can.

LLMs *alone* can only solve things somewhere in between AC^0 and TC^0.

Proof is left to the reader, but here's a nice motivating example:

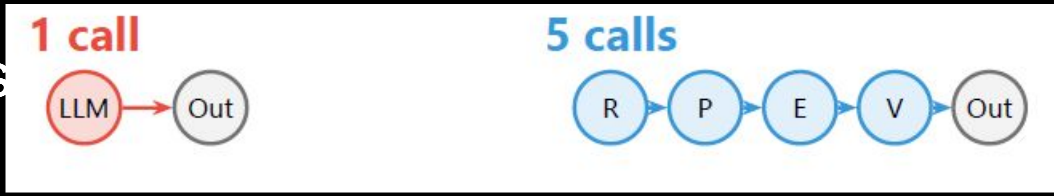
Let L be an LLM with knowledge up to 2025. Consider the following query:

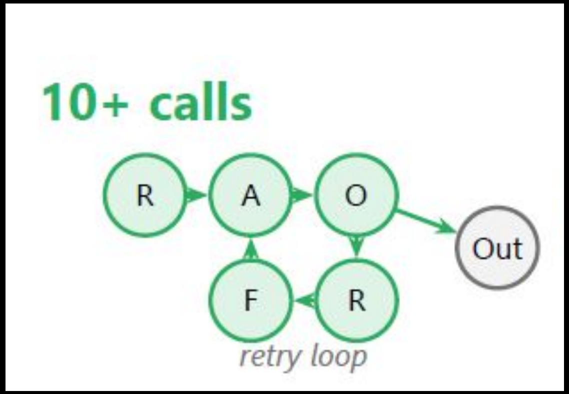
What are the projected earnings for 2027, based on a linear regression model on the earnings from 2022-2026?

Hint: Computing A^k and inverting it are both in TC^1

Note: finer-grained measures are also needed

- These help us understand capabilities, even when they do not provide technological leaps
- E.g., reasoning depth:
- Deeper graphs solve harder problems, but eventually reach a plateau
  - **1-call**: single-shot edit (naïve)
  - **3-5 calls**: inspect, plan, edit, verify
  - **10+ calls**: multi-step ReAct/actor-critic loops
- Reasoning depth is distinct from tool call as it governs *how deeply can an agent reason about a problem, not how many actions it can take.*





R = Read **P** = Plan **E** = Edit **V** = Verify
A = Act **O** = Observe **F** = Fix

Microsoft

A proposed agentic complexity measure

Type	Formal	Example (for a domain)
Type-0	A function takes an input, modifies it based on internal logic, and returns the string	A naïve LLM call
Type-1	A deterministic program helps retrieve the input; the function modifies it, and uses a program to write it to memory.	An LLM taking in XML/HTML, producing text, piped to a *file*
Type-2	Type 1, but with the ability to write and call functions beyond what it 'knows' it can do.	Agent calls specialised APIs
Type-3	Type 2 + the ability to reduce to Type-1 contextually	Agents call *any* potential API for a given domain
Type-4	Every component of the system's calls is automated, including the ability to reduce to Type-2 or Type-0	---
Type-5	Everything is fully optimised.*	--

Microsoft

A proposed agentic complexity measure

Type	Formal	Example (for a domain)
Type-0	A function takes an input, modifies it based on internal logic, and returns the string	A naïve LLM call
Type-1	A deterministic program helps retrieve the input; the function modifies it, and uses a program to write it to memory.	An LLM taking in XML/HTML, producing text, piped to a *file*
Type-2	Type 1, but with the ability to write and call functions beyond what it 'knows' it can do.	**Agent calls specialised APIs**
Type-3	Type 2 + the ability to reduce to Type-1 contextually	**Agents call *any* potential API for a given domain**
Type-4	Every component of the system's calls is automated, including the ability to reduce to Type-2 or Type-0	---
Type-5	Everything is fully optimised.*	--

Content

- Mathematical background
  - On the limits of theory: they don't mix that well
  - Agentic complexity
- Aside: the problem of evaluation
- The Document Manipulation Benchmark (TDMB)
  - Why document manipulation?
  - Building a robust benchmark
- Preliminary results
  - Agentic complexity in TDMB
- Future work

The problem with evaluation

- Everyone cheats

Thank you!

Any questions?

The problem with evaluation

- Everyone cheats
- If you cheat, you don't really know how well you're doing
- There's also no guarantee the results are realistic in any sense of the word
- Running a serious, stat-sig benchmark where one can say 'my confidence bound is' costs money and time and water.

Corpus	Dataset ↑	Train split	Dev split	Test split	Source
ChatGPT	ACE05	Suspicious	Suspicious	Suspicious	
RedPajama	AESLC			Suspicious	Paper
The Pile	AESLC			45.5% Contaminated	Paper
OSCAR	AESLC			Suspicious	Paper
C4	AESLC			1.6% Contaminated	Paper
GPT-3.5	AG News	Clean		Clean	Paper
GPT-4	AG News	Contaminated		Contaminated	Paper
GLaM	ANLI R1		96.2% Contaminated		Paper
FLAN	ANLI R1		98.6% Contaminated		Paper
GPT-3	ANLI R1			20.0% Contaminated	Paper
GLaM	ANLI R2		96.8% Contaminated		Paper
FLAN	ANLI R2		97.9% Contaminated		Paper
GPT-3	ANLI R2			18.0% Contaminated	Paper
GLaM	ANLI R3		40.7% Contaminated		Paper
FLAN	ANLI R3		40.2% Contaminated		Paper
GPT-3	ANLI R3			16.0% Contaminated	Paper

Content

- Mathematical background
  - On the limits of theory: they don't mix that well
  - Agentic complexity
  - Aside: the problem of evaluation
- The Document Manipulation Benchmark (TDMB)
  - Why document manipulation?
  - Building a robust benchmark
- Preliminary results
  - Agentic complexity in TDMB
- Future work

Microsoft

What is a realistic, widespread, commonly-used task?

Word-processor document manipulation!

1. Second-most used file type on the internet (after *.pdf)
2. **<u>Really</u>** ugly and complicated
3. It has to factor in user intent, which in turn adds a layer of complexity over NLU
4. (nice) HTML is context-free!

What is a realistic, widespread, commonly-used task?

Word-processor document manipulation!

1. Second-most used file type on the internet (after *.pdf)
2. **<u>Really</u>** ugly and complicated
3. It has to factor in user intent, which in turn adds a layer of complexity over NLU
4. (nice) HTML is context-free!

*This means LLMs **cannot** solve all HTML problems (or XML, or whatever)*

What is a realistic, widespread, commonly-used task?

Word-processor document manipulation!

1. Second-most used file type on the internet (after *.pdf)
2. **Really** ugly and complicated
3. It has to factor in user intent, which in turn adds a layer of complexity over NLU
4. (nice) HTML is context-free!

*This means LLMs **cannot** solve all HTML problems (or XML, or whatever)*

OR CAN THEY

It really is ugly

THE ROSE FAMILY

The rose is a rose,
And was always a rose.
But the theory now goes
That the apple 's a rose,
And the pear is, and so 's
The plum, I suppose.
The dear only knows
What will next prove a rose.
You, of course, are a rose—
But were always a rose.


```
rsidRPr="0049288B"><w:rPr><w:rFonts w:ascii="Times New Roman" w:hAnsi="Times
New Roman" w:cs="Times New Roman"/><w:sz w:val="32"/><w:szCs w:val="32"/><w:
lang w:val="en-GB"/></w:rPr><w:t>The rose is a rose,</w:t></w:r><w:r w:rsidRPr
="0049288B"><w:rPr><w:rFonts w:ascii="Times New Roman" w:hAnsi="Times New
Roman" w:cs="Times New Roman"/><w:sz w:val="32"/><w:szCs w:val="32"/><w:lang w
:val="en-GB"/></w:rPr><w:br/><w:t>And was always a rose.</w:t></w:r><w:r w:
rsidRPr="0049288B"><w:rPr><w:rFonts w:ascii="Times New Roman" w:hAnsi="Times
New Roman" w:cs="Times New Roman"/><w:sz w:val="32"/><w:szCs w:val="32"/><w:
lang w:val="en-GB"/></w:rPr><w:br/><w:t>But the theory now goes</w:t></w:r><w:
r w:rsidRPr="0049288B"><w:rPr><w:rFonts w:ascii="Times New Roman" w:hAnsi="
Times New Roman" w:cs="Times New Roman"/><w:sz w:val="32"/><w:szCs w:val="32"
/><w:lang w:val="en-GB"/></w:rPr><w:br/><w:t>That the apple 's a rose,</w:t></
w:r><w:r w:rsidRPr="0049288B"><w:rPr><w:rFonts w:ascii="Times New Roman" w:
hAnsi="Times New Roman" w:cs="Times New Roman"/><w:sz w:val="32"/><w:szCs w:
val="32"/><w:lang w:val="en-GB"/></w:rPr><w:br/><w:t>And the pear is, and so
's</w:t></w:r><w:r w:rsidRPr="0049288B"><w:rPr><w:rFonts w:ascii="Times New
Roman" w:hAnsi="Times New Roman" w:cs="Times New Roman"/><w:sz w:val="32"/><w:
szCs w:val="32"/><w:lang w:val="en-GB"/></w:rPr><w:br/><w:t>The plum, I
suppose.</w:t></w:r><w:r w:rsidRPr="0049288B"><w:rPr><w:rFonts w:ascii="Times
New Roman" w:hAnsi="Times New Roman" w:cs="Times New Roman"/><w:sz w:val="32"
/><w:szCs w:val="32"/><w:lang w:val="en-GB"/></w:rPr><w:br/><w:t>The dear
only knows</w:t></w:r><w:r w:rsidRPr="0049288B"><w:rPr><w:rFonts w:ascii="
Times New Roman" w:hAnsi="Times New Roman" w:cs="Times New Roman"/><w:sz w:val
="32"/><w:szCs w:val="32"/><w:lang w:val="en-GB"/></w:rPr><w:br/><w:t>What
will next prove a rose.</w:t></w:r><w:r w:rsidRPr="0049288B"><w:rPr><w:rFonts
w:ascii="Times New Roman" w:hAnsi="Times New Roman" w:cs="Times New Roman"/><w
:sz w:val="32"/><w:szCs w:val="32"/><w:lang w:val="en-GB"/></w:rPr><w:br/><w:t
>You, of course, are a rose—</w:t></w:r><w:r w:rsidRPr="0049288B"><w:rPr><w:
rFonts w:ascii="Times New Roman" w:hAnsi="Times New Roman" w:cs="Times New
```

The Document Manipulation Benchmark

A benchmark performing k-fold cross-validation over a *private* dataset.

- Measures **docx, doc, odt,…** manipulation over realistic intents
- Access to (most of the) judges and the sample data is open

Input:

$$(\text{document}, \text{prompt(s)}, \text{grounding})$$

Output:

$$(\text{modified})\ \text{document}$$

No test set is seen twice (ever), and it is blind.

- Scoring is given as an average + standard deviation + CI over runs.

Core Concepts

- Two types of data available:
  - All-Purpose (AP) Prompts
  - Radioactive dataset
- Everything is **handcrafted** and as realistic as possible.
- 12 languages (each 1:1 with the same English subset):

  zh-Hans-cn, zh-Hant-tw,
  sl-SI, hi-IN it-IT, ie-IE,
  ja-JP, ko-KO, es-ES, es-MX
  th-TH, fy-NL

AP Prompts

- Starting prompts are human-written
- They are grammarised based on some appropriate Σ.

```
"Description": "Add page numbers to the {position}",
```

- At every call c, the ith prompt of type t requiring features k is matched to human-annotated documents having k, and sampled from there:

$$ {}^{c}_{i}p_{t,s} \sim \{d \mid d \in \mathbf{D}, k \in d\} \times \left\{q_{j,\cdot} \mid q_{j,\cdot} \in \mathbf{P}, k \in q_{j,\cdot}, j = t\right\} \times \{s \sim \Sigma\} $$

This leads to about 100k entries.

Note that $\{{}^{j}_{i}p_{t,s}\}_{i=1,\ldots,n} \sim \mathcal{AP}$ (i.e., prompts at any c are $\mathcal{AP}$- distributed)

Radioactive

- Fully-handcrafted I/O multiturn dataset.
  - This means also creating the ground truth by executing the prompts
- Prompt type needs to be difficult and not necessarily present in AP
  - Multi-hop reasoning, doing and undoing changes, etc.
- Never change across runs
- About 25% the size of AP
- Used to detect p-hacking and to calibrate The Doctor (the judging mechanism)

The Doctor

- Asserts for the easy cases
  - Verifiable, deterministic, easy peasy
- Ensemble of six VLMs for the difficult cases
  - Human-calibrated, binaryish score: $J_i\colon \mathbf{D} \times \mathbf{P} \times \mathbf{D} \to \{0,1\}$
  - Boosted:

$$\mathrm{score} \;=\; \mathrm{maj}\left(\left\langle \begin{bmatrix} \mathrm{w}_1 \\ \cdots \\ \mathrm{w}_6 \end{bmatrix}, \begin{bmatrix} \mathrm{J}_1(\mathrm{d},\mathrm{p},\mathrm{g}) \\ \cdots \\ \mathrm{J}_6(\mathrm{d},\mathrm{p},\mathrm{g}) \end{bmatrix} \right\rangle\right)$$

1. The weight vector is renormalised in case of any failures.
2. Calls to at least three judges must succeed; two from high-quality judges.

Why x-val?

- Let
    1. $S := \Sigma_i^n x_i$ for i.i.d x_i
    2. $\mu = \mathbb{E}[S]$ (and thus $\mu = np$ if $\mathbb{E}[x_i] = p$)

So:

$$Pr[| \mathrm{S} \; - \; \mu| > \delta \mathrm{n}] \leq 2e^{-2n\delta^2}$$

And let's fix some α and let $\delta = 1/10$.

It then follows that to get $2e^{-2n/100} < \alpha$, you need $n > 50log(\alpha/2)$.

So the mean error rate on S is within 1/10 of the 'true' error rate on some S' s.t. $S \subset S'$.

This could still be very large, but average out all errors and you get an estimate of true error provided condition (1) holds.

Why single-blind?

- Case study: RSA. Consider a system where you have a plaintext x.
- You obfuscate x with some function (generally multiplying it by some number k s.t. $GCD(N, k) = 1$ and taking the RSA mod N).
  - Encryption:
    - RSA: $f(x) \equiv_N x^e$
    - Obfuscated: $f(x) \equiv_N (xk)^e$
  - Decryption. Let $c = f(x)$:
    - RSA: $c^d \equiv_N = x$
    - Obfuscated: $c^d \equiv_N = \left((xk^{-1})^e\right)^d = x \mod \mathrm{N}$ – an attacker would need to know r^{-1}
- TL;DR: an attacker trying to extract information will only have access to the 'cyphertext' (learn the random test set) but cannot extract the 'message' (the full dataset) since it doesn't know r (the pairing function).
- They also get throttled, and there's radioactivity

Content

- Mathematical background
  - On the limits of theory: they don't mix that well
  - Agentic complexity
- Aside: the problem of evaluation
- The Document Manipulation Benchmark (TDMB)
  - Why document manipulation?
  - Building a robust benchmark
- Preliminary results
  - Agentic complexity in TDMB
- Future work

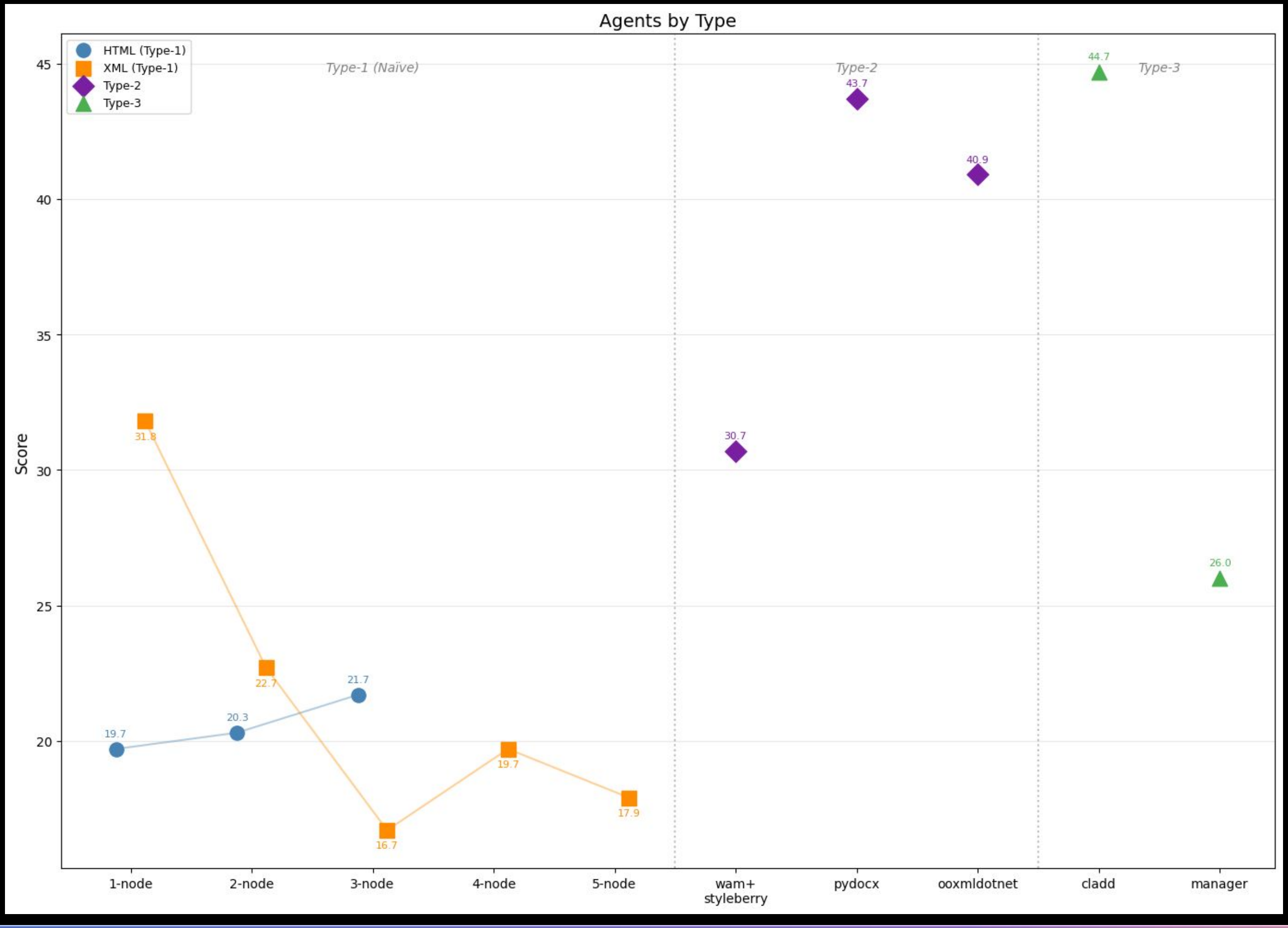
Preliminary results (tool access)

Library-wise

1. Type-1 (No tools)
2. Type-2, (ooxmldotnet only, pydocx only)
3. Type-2.5: both tools available.

Note:

4. The jump from no tools to one tool
5. The marginal value of combining two under an orchestrator.



Microsoft

In Sum

Agentic workflows can solve a lot than single LLMs cannot do.

Complexity does scale with respect to (tool) abilities—but **this can only take you so far.**

You need **two** measures of complexity:

1. Ability-based complexity for understanding current and frontier capabilities, and
2. Expressive-power complexity for true 'into the unknown' work

It is feasible to build a benchmark with guarantees of correctness

In Sum

Agentic workflows can solve a lot than single LLMs cannot do.

Complexity does scale with respect to (tool) abilities—but **this can only take you so far.**

You need **two** measures of complexity:

1. Ability-based complexity for understanding current and frontier capabilities, and
2. Expressive-power complexity for true ‘into the unknown’ work

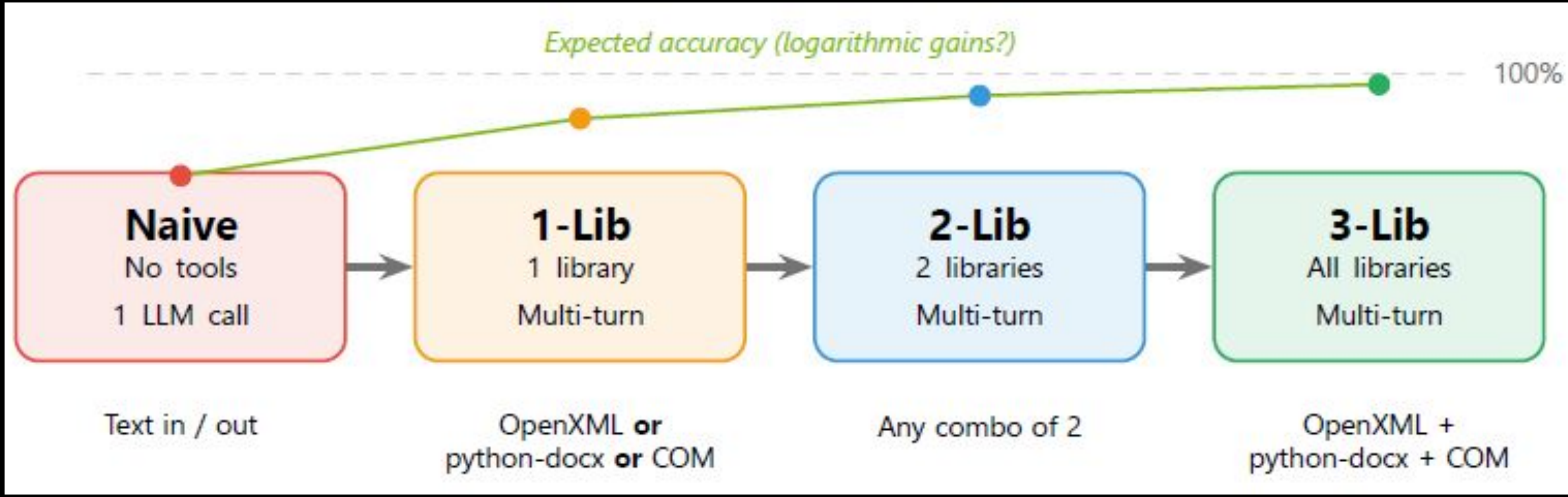
It is feasible to build a benchmark with guarantees of correctness

We are not done! Lots of sciency enhancements happening

Appendix

Agent Complexity

Tool Access



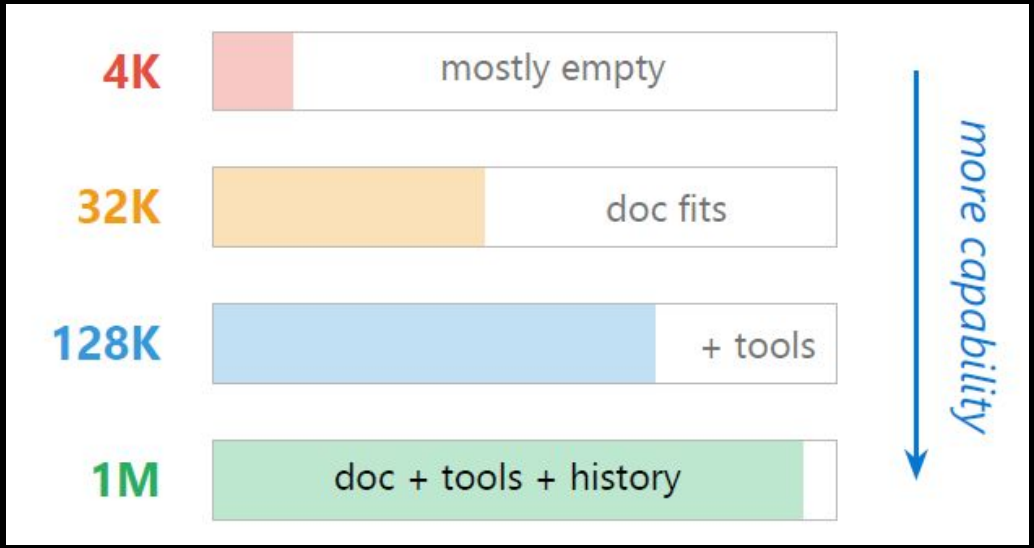
Expected accuracy gains with successive addition of tools/libraries for an agent

Agent Complexity

Context Budget

- Context Budget is essentially the agent's **working memory**
  - **Small context**: a few paragraphs, loses track of full document
  - **Medium context**: holds the document but not its history
  - **Large context**: full document + tool results + multi-turn reasoning

Practical constraint: A 40+ page document can consume 100K+ tokens just as input, which leaves little room for tool results and reasoning in smaller context windows.

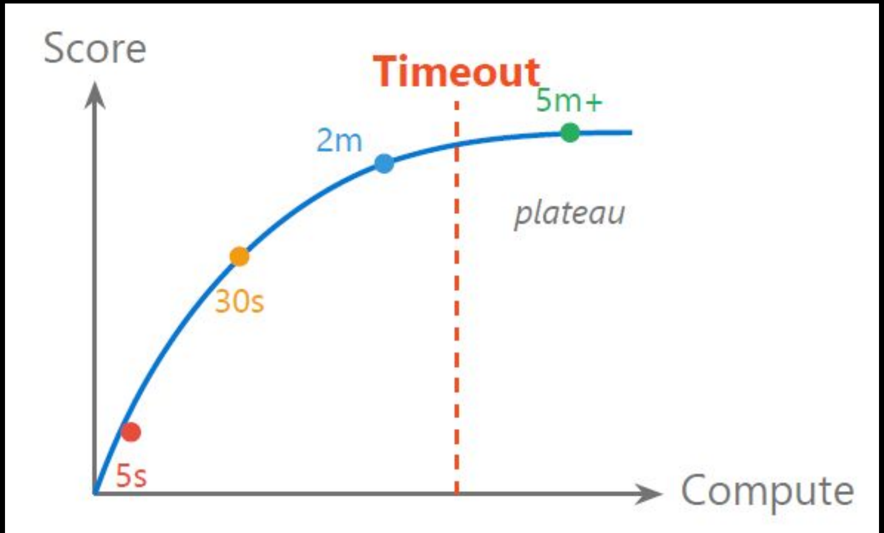


- Context budget constrains how much *information* the agent can hold at once, not how many steps it can take
- Context is the *bandwidth* of each reasoning step.

Agent Complexity

Compute Budget

- Describes what the agent *consumes*: the total computational resources (CPU, GPU, wall-clock time) spent executing a task.
- It is a resource the agent actively manages through its choices
- If the edit does not complete within the compute/time window, it counts as a **failure** regardless of partial progress - *this mirrors real deployment constraints where users expect responses within seconds to minutes.*



Kolmogorov Complexity

-

Microsoft