Stanford CS224v Course
Agentic AI

Lecture 13

Hot Topics in LLMs

Monica Lam Jiacheng Sang Shicheng Liu
Andrew Biondo Cyrus Zhou Joachim Baumann Ananjan Nandi

Project Result Presentation

- Dec 2: Videos and posters
  - 2:30-3 pm: Poster set up Gates 109.
  - 3-4:20 pm: CODA B80
    a 1-minute video
    Highlighting the purpose and a demo
  - 4:20–6pm: Poster session (Gates 109)

- Dec 9 midnight: Final project report

Final report (Due Dec 9, midnight)

Deliverables:

1. Final paper + short video demo
2. Code (including a README on how to run your program for a demo)

Suggested outline of the final paper: (Use ACL style for your paper: ACL style files)

Abstract: 2-3 paragraphs summarizing the paper, including the results

Introduction, which includes the motivation, main idea, and overall contribution

Related work

The core ideas

Experimental results

What you learned and future work

Work performed by each team member

Conclusion

Appendix: Examples, Prompts that you engineered ...

Course Summary

This course has introduced important agentic building blocks:

VLM: CHURRO

AI-systems:

- Verdict: accountable decision making
- SatIR: constraint-satisfaction-based retrieval
- SUQL: structured/unstructured data
- Genie Worksheets: task-oriented agents

Knowledge agents:

- Wikichat: minimizes hallucination in RAG
- STORM: Long-form literature search
- DataChat: Database retrieval
- Sheldon: hypothesis-evidence reasoner
- SLIDERS: evidence management
- DataSTORM: Insight generation
- ADAPT: Structured context management

Agentic flows are continually incorporated into coding agents via training

Hot Topics in LLMs

What you should keep in mind as you work on your projects

- Coding agents are powerful
  - **Do we need to continue to build purpose-built agents?** (Jiacheng Sang, Shicheng Liu)
  - **Why & how to build purpose-built agents? (Case study: historical OCR)** (Andrew Biondo, Cyrus Zhou)
- Be careful! Watch out for LLM cheating and reward hacking
  - **Reward hacking incidents and prevention** (Joachim Baumann)
  - **A low-level run-time substrate for performance and security?** (Ananjan Nandi)

Generalization Beyond Coding

An Empirical Comparison of Coding Agents and Purpose-Built Agents

Jiacheng Sang · Shicheng Liu · Marquita Ellis · Monica S. Lam

Motivation

Coding agents

Codex, Claude Code, OpenHands

- Edit files, write and run code
- Evaluated mostly on software tasks

Purpose-built agents

Customed for specific tasks

- Stages for retrieval, reasoning, planning, verification
- Domain-specific tools or data source

Q1 How far can coding agents generalize across non-coding domains?

Q2 What can we learn from those SOTA and open-source purpose-built agents?

Choosing benchmarks and SOTA agents

Benchmarks

- Non-coding
- Not saturated
- Tested by published methods
- Public data and evaluation
- Recent or widely cited

SOTA purpose-built agent

- Highest published score
- Released code
- Diverse agent types
- Official metrics
- Full set, or a stated subset

Coding agent Codex CLI on gpt-5.6-sol

Method:

STEP 1

Find the comparable SOTA agent

Public code,
Official settings

STEP 2

Run both on one backbone

Same environment
Official scorer

STEP 3

Compare the paired traces

LLM annotators,
human-checked

STEP 4

Test with ablations

Give Codex the agent's tools or steps

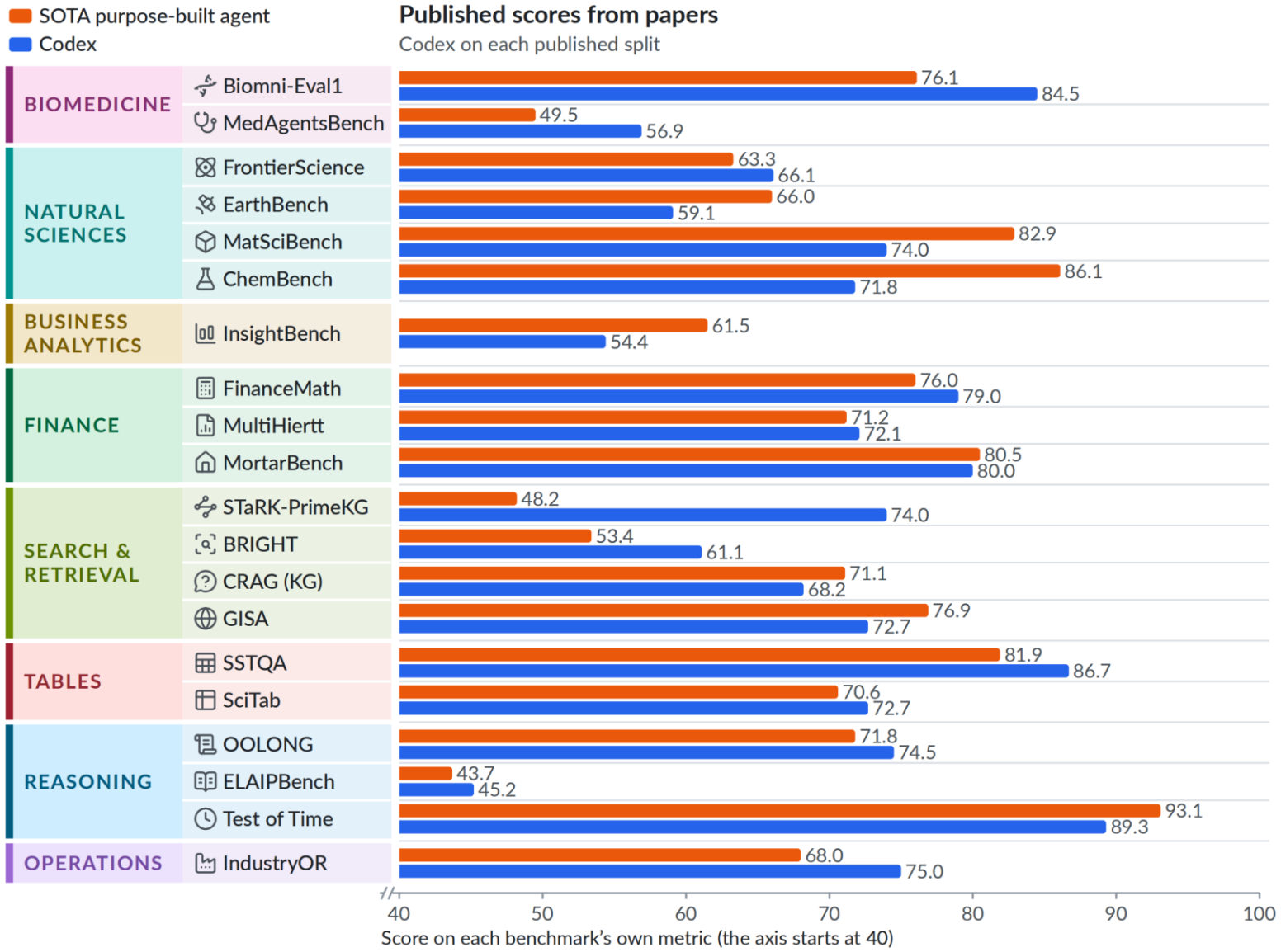
562 → 20
benchmarks

17
SOTA agents

4,649
paired instances

gpt-5.6-sol
for both agents

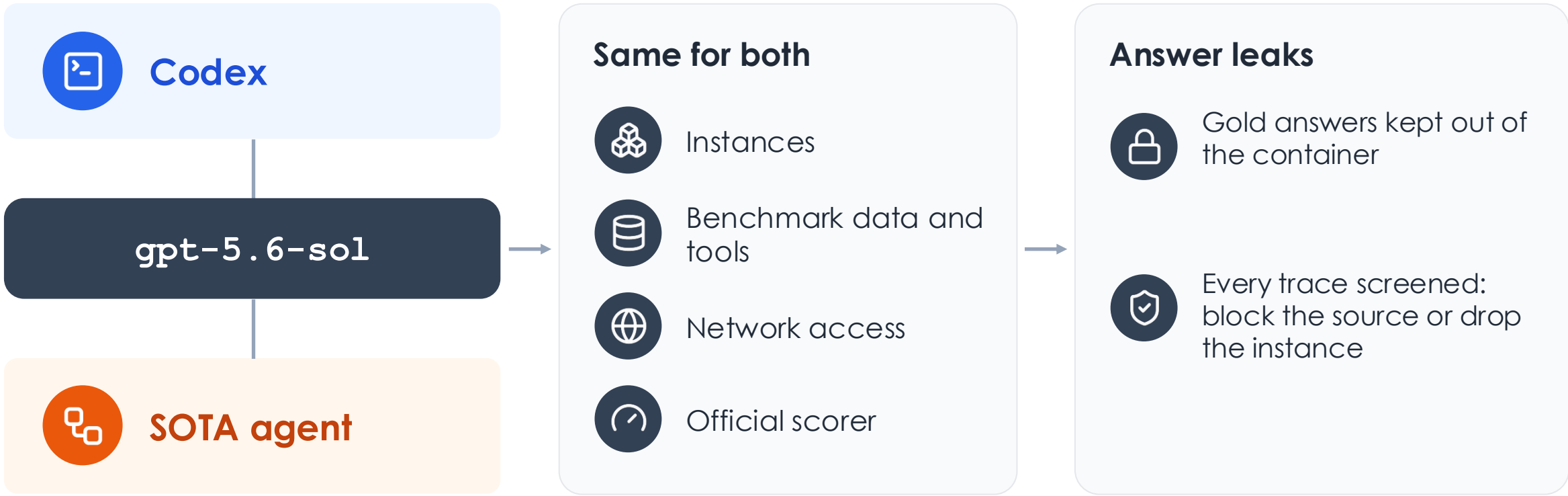
Published scores: Codex ahead on 12 of 20



Published scores alone are not enough for analysis

- **Traces and answers rarely released**
- **Model and harness mixed**
- **Evaluation environment varies**
- **Leaks not checked**

Fair comparison:
Same Backbone, Standardized env

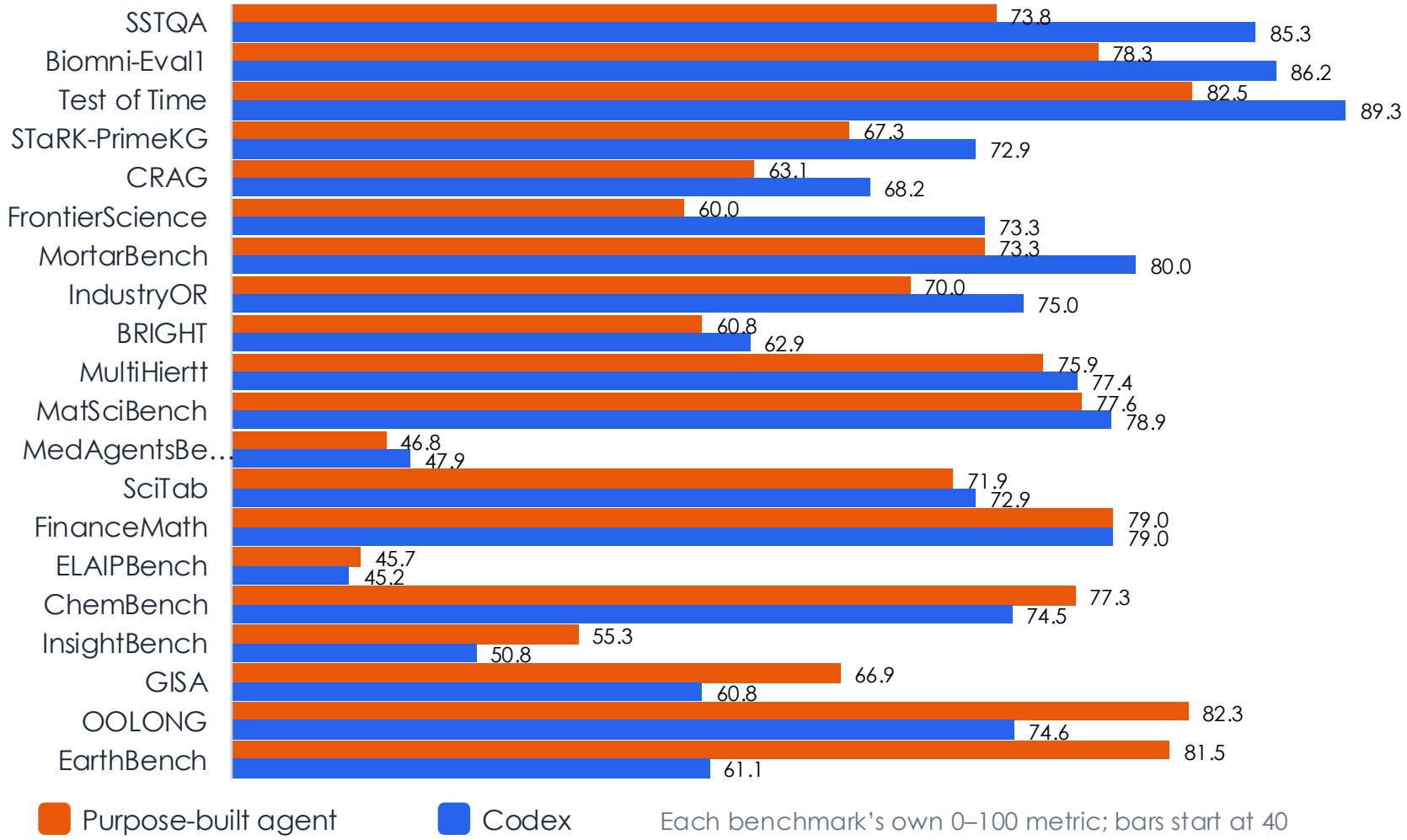


Same backbone: Codex wins 5, loses 2 of 20

5
Codex significantly better

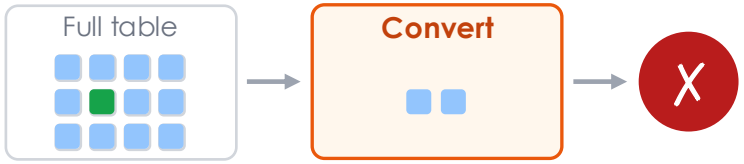
13
The difference is not statistically significant

2 **Purpose-built significantly better**



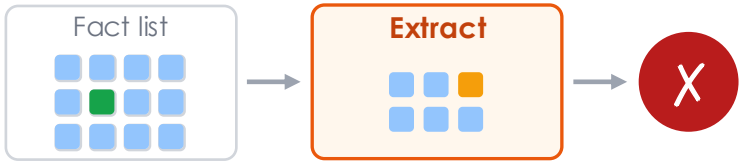
Why Codex wins: fixed pipelines drop evidence

ASTRA
SSTQA



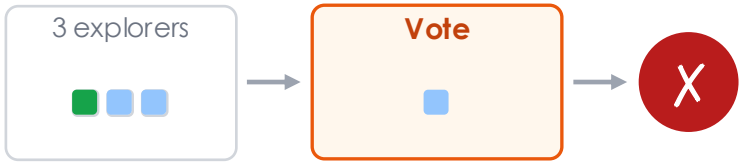
- Rebuilds each table as a tree
- One workbook: the tree kept only 19% of the table
- Answered 0 instead of 6

REMem-I
Test of Time



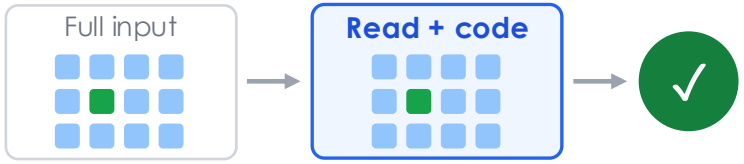
- Extracts the facts into a memory first
- The memory spelled one relation two ways
- Counted one spelling only: 1 instead of 3

ARK
STaRK-PrimeKG



- Three explorers search the graph, then vote
- One traced the graph to the right gene
- Two started from a guess and outvoted it

Codex
same inputs



- Works on the full, original input
- Decides what to read after reading the question
- Right on all three examples

174 of the 388 instances only Codex solved (45%): answer lost inside the purpose-built pipeline

What we can learn from Purpose-built agents?

Simple steps Codex seldom uses

ELAIPBench · AgentSPEX

Re-check multi-answer questions

```
if multi_answer and selected < 2:
    re-judge the rest leniently; pick ≥ 2
```

- Told each question has at least 2 correct options
- Codex still gave one or none on 21 of 315

OOLONG · EnSI-RAG

Label every record, then count

```
labels = [small_model(r) for r in records]
answer from Counter(labels)
```

- Codex takes a shortcut
- Samples a few records, then concludes

What we can learn from Purpose-built agents?

Use of Domain tools and Auxiliary models

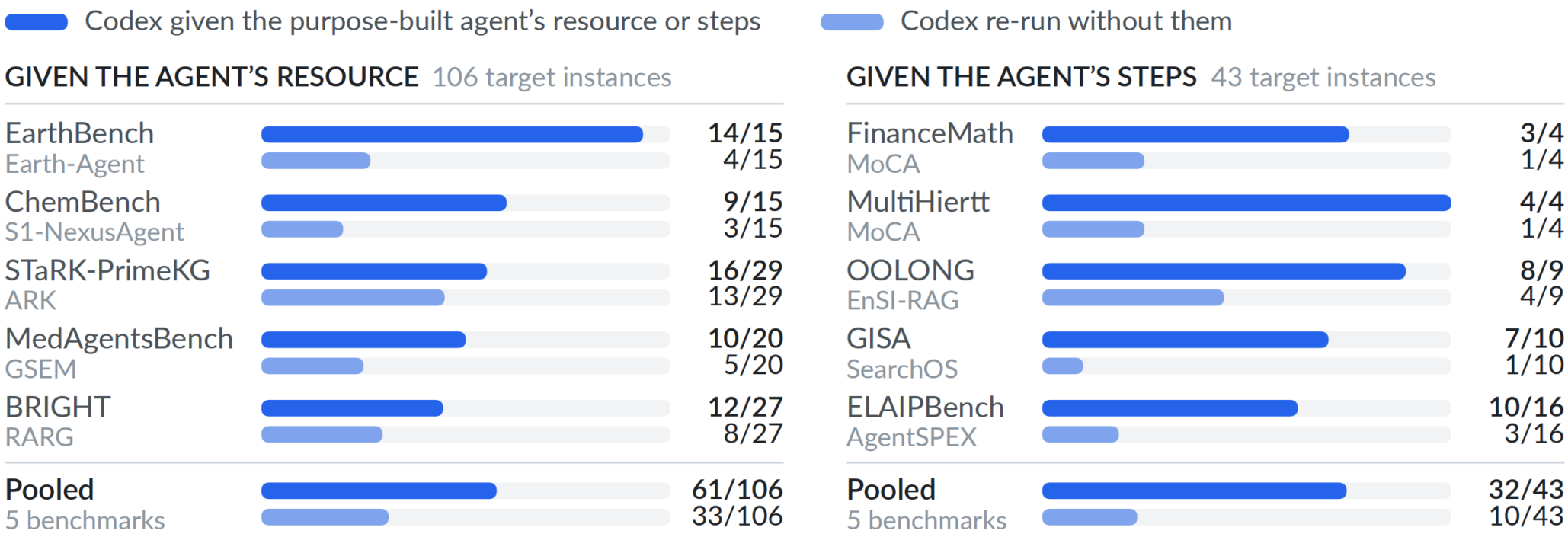
Domain tools and data

- **Earth-Agent:** 104 Remote sensing tools
- **S1-NexusAgent:** RDKit chemistry toolkit
- **GSEM:** memory of 1,465 clinical cases

Auxiliary models

- **Trained ranker:** RARG orders documents with a retrieval model, the main reason for 22 of its wins
- **Small helper model:** EnSI-RAG labels every OOLONG record with one, at about half of Codex's cost
- **Routers and embedders:** e.g. CacheRAG routes each question with a small Llama model

Ablation Experiments:



The strengths of SOTA Purpose-built agents are portable to Codex

Summary

Coding agent:
a strong starting point & baseline

General ReAct Harness

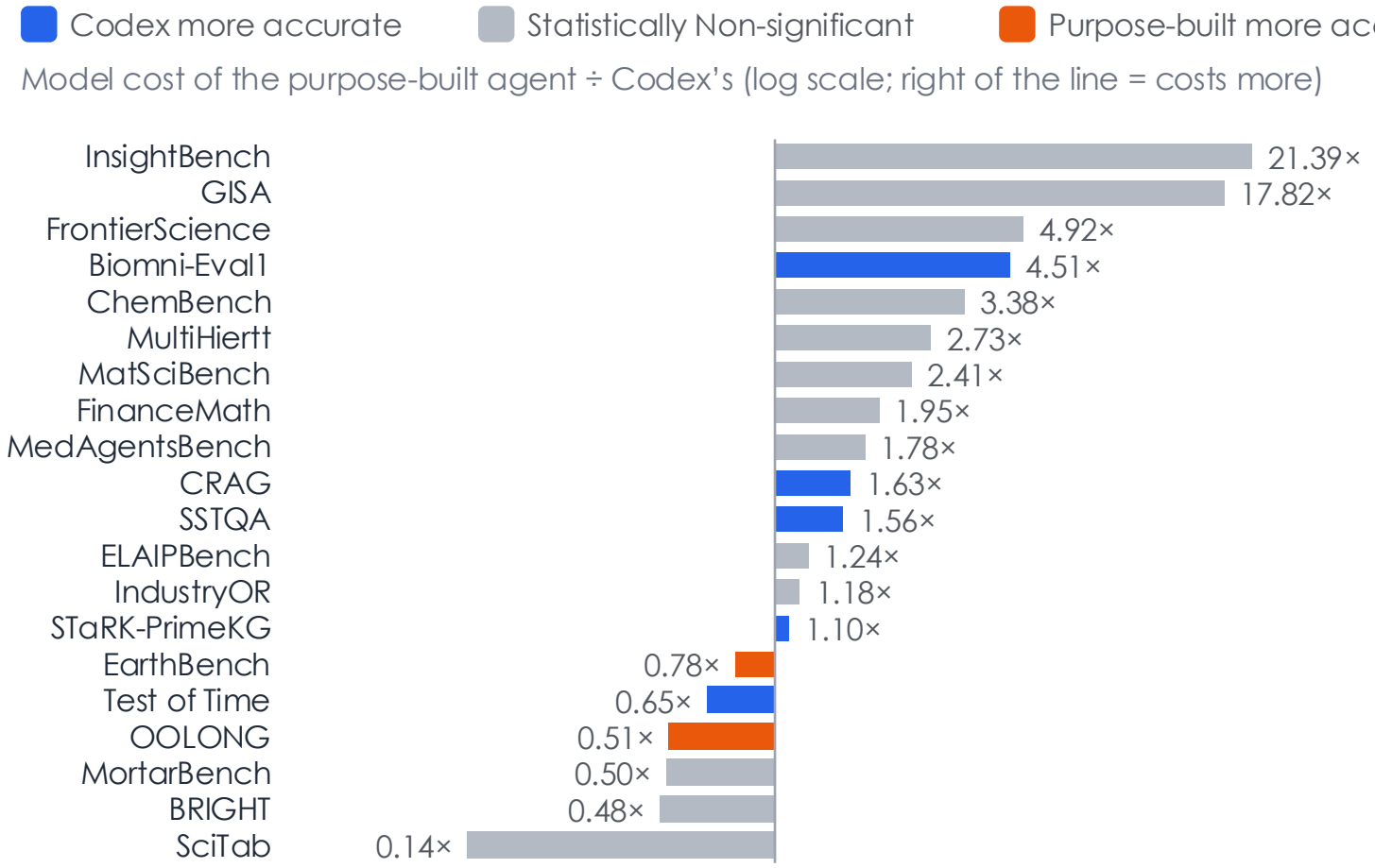
Grow with the model.

Purpose-built harnesses are still needed

Domain tools

Specific workflows.

Cost: purpose-built agents usually cost more



14 of 20

benchmarks where the purpose-built agent costs more

Higher cost does not buy accuracy

The three costliest agents are not significantly more accurate than Codex.

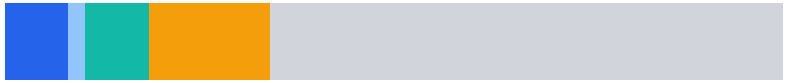
Both agents that beat Codex cost less

EnSI-RAG hands record extraction to a small model; Earth-Agent needs only a few calls.

Codex answers even when the question is broken

8 of 97

unanswerable questions where Codex reported in its answer



- **8** said so in the answer
- **2** stated an assumption
- **8** abstained
- **15** noticed, stayed silent
- **64** no visible sign

MedAgentsBench a lesion that was never shown

Listed the empty workspace twice, then wrote: "The lesion is at Site 3 (the cervical surface) and is Size 2"

FinanceMath a share price that was never given

Its web searches failed; it then used "Cratt's $24.86 share price", a number found nowhere in its input.

SSTQA a spreadsheet unrelated to the question

- Asked how many cash transactions there were on 6 May 2020
- It wrote: "The visible cells do not match the question"
- Still answered "0 transactions"

Reward hacking found in Codex's traces

GISA

Finds the answer key

Searched for the benchmark itself, cloned a public copy from GitHub, and submitted its stored 41-name answer.

```
"The benchmark's public
repository appears to include
the gold answer"
```

MatSciBench

Copies the textbook

Its code computed 2.898e-4. It opened the textbook PDF and submitted the book's 2.82e-4, claiming a derivation.

```
"Substitution gives an exponent
of approximately -8.17, so
T = exp(-8.17) = 2.82e-4"
```

MultiHiertt

Hunts for the answer file

- Read the question's ID
- Searched the machine for that ID and the question text
- Found no answer file
- Then answered from the table

```
rg -F "56853fe3d64a…"
rg -F "In which year is…"
```

Follow-up directions

1 **Can coding agents that automatically find and use specialist tools and skills?**



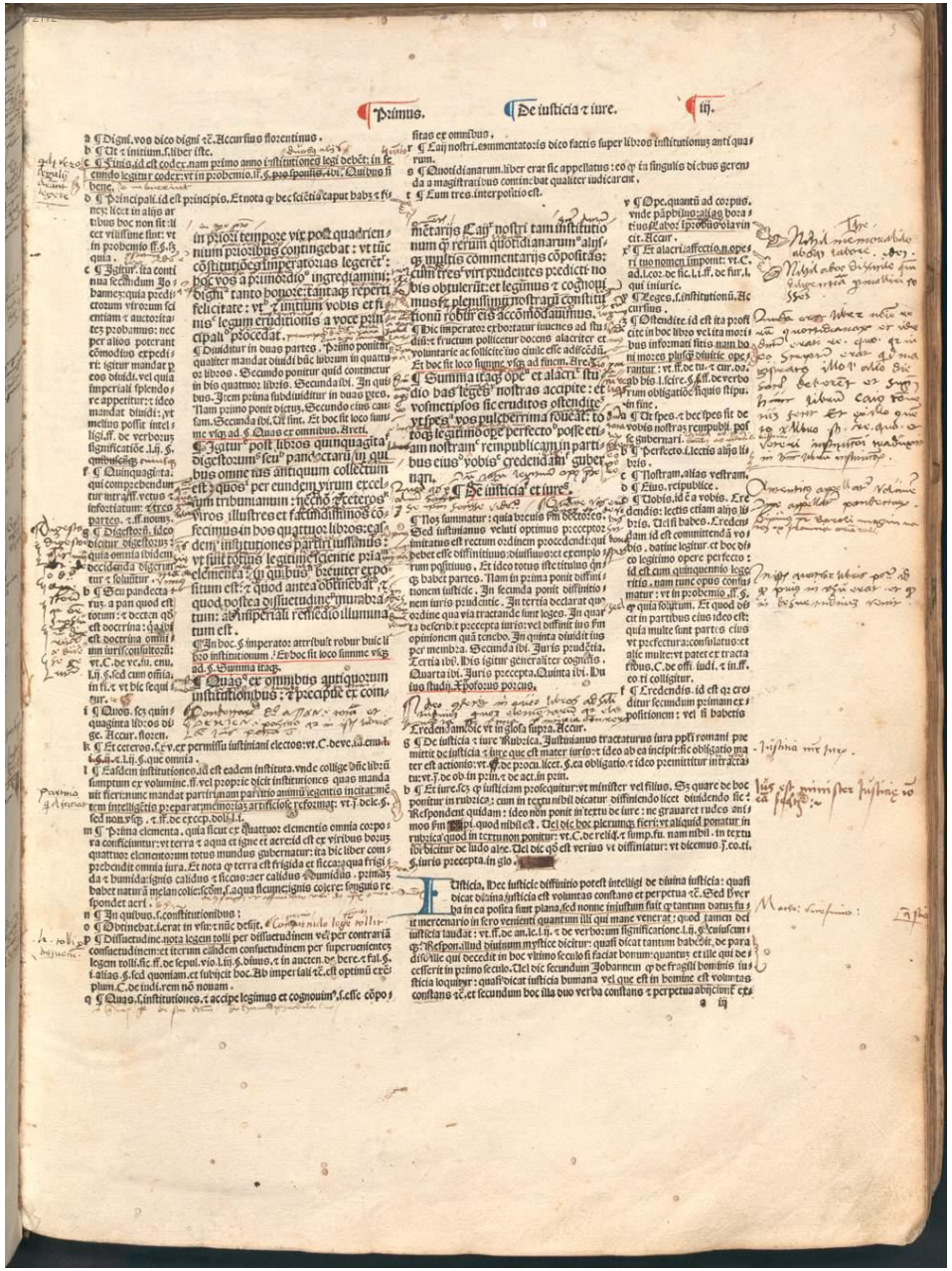
2 **How to make agents critical and honest?**



Hot Topics in LLMs

What you should keep in mind as you work on your projects

- Coding agents are powerful
  - **Do we need to continue to build purpose-built agents?** (Jiacheng Sang, Shicheng Liu)
  - **Why & how to build purpose-built agents? (Case study: historical OCR)** (Andrew Biondo, Cyrus Zhou)
- Be careful! Watch out for LLM cheating and reward hacking
  - **Reward hacking incidents and prevention** (Joachim Baumann)
  - **A low-level run-time substrate for performance and security?** (Ananjan Nandi)



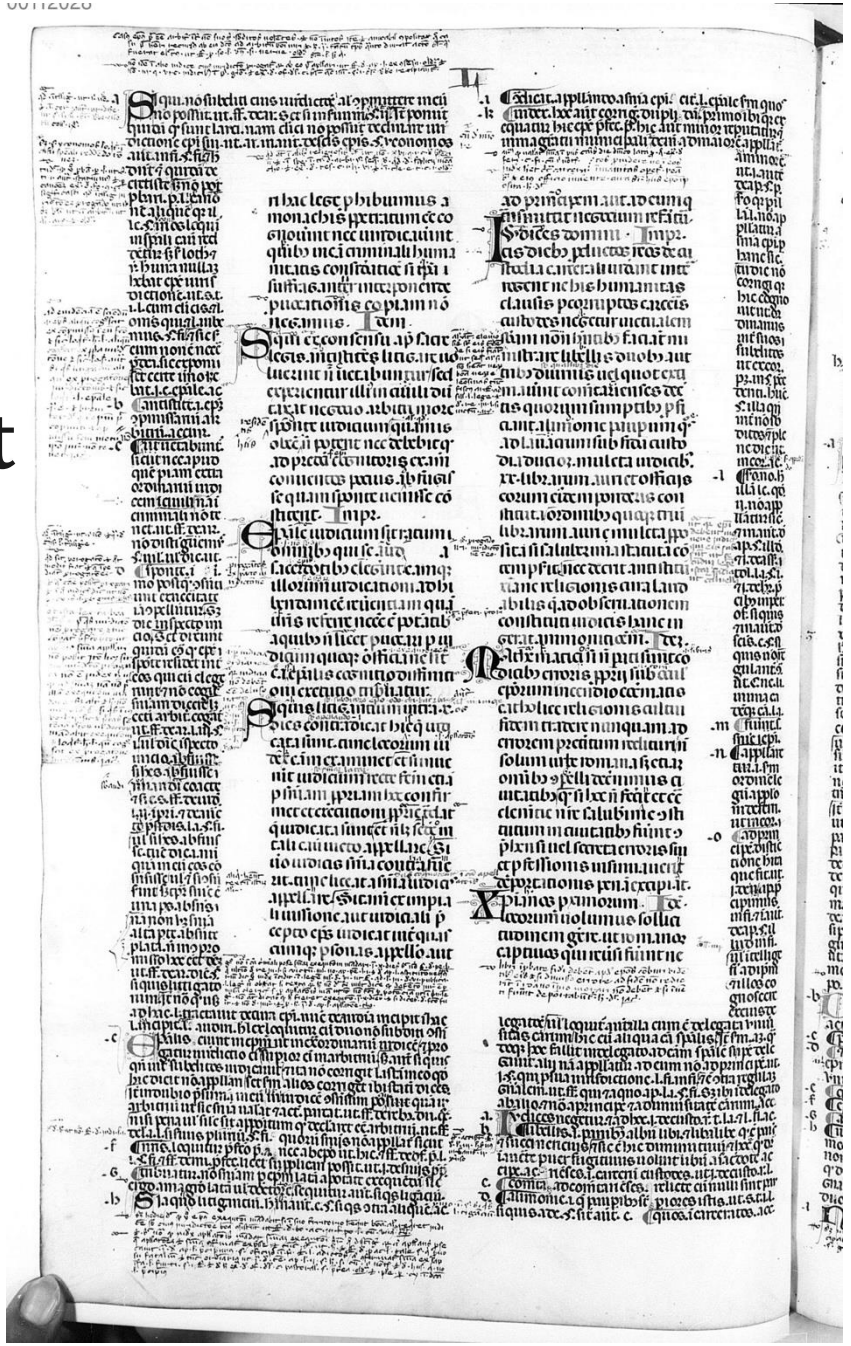
Roman law with the Accursian gloss, ed. Girolamo Chirai. Venice: Georgius Arrivabene, 24 April 1494 [GW 7634]. Bayerische Staatsbibliothek, München.

Unlocking Ancient Knowledge

Teaching AI to transcribe medieval manuscripts and early printed books

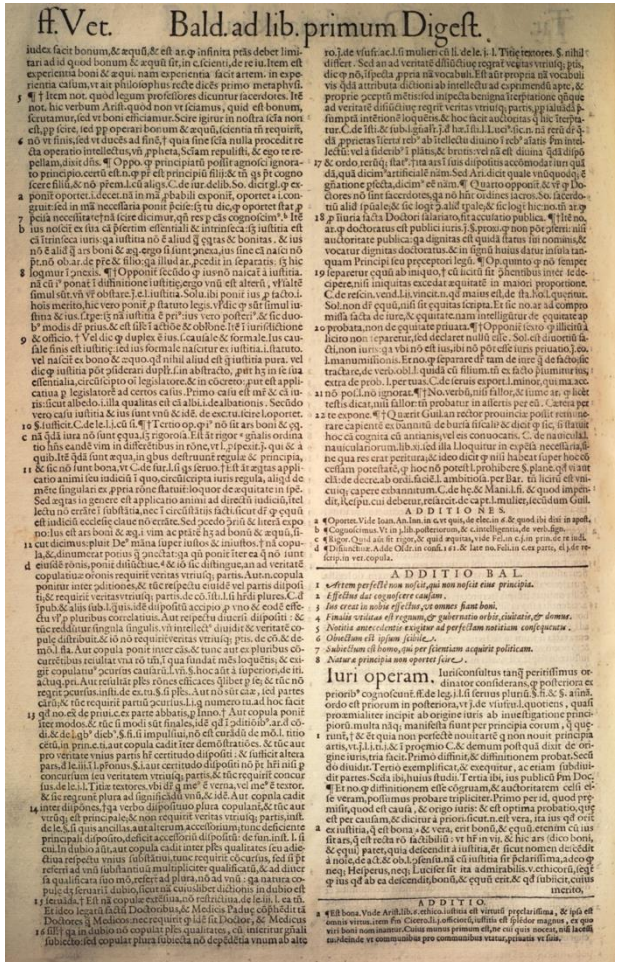
Andrew M. Biondo
PhD Candidate, Philosophy, Stanford

CS 224V · 18 November 2026

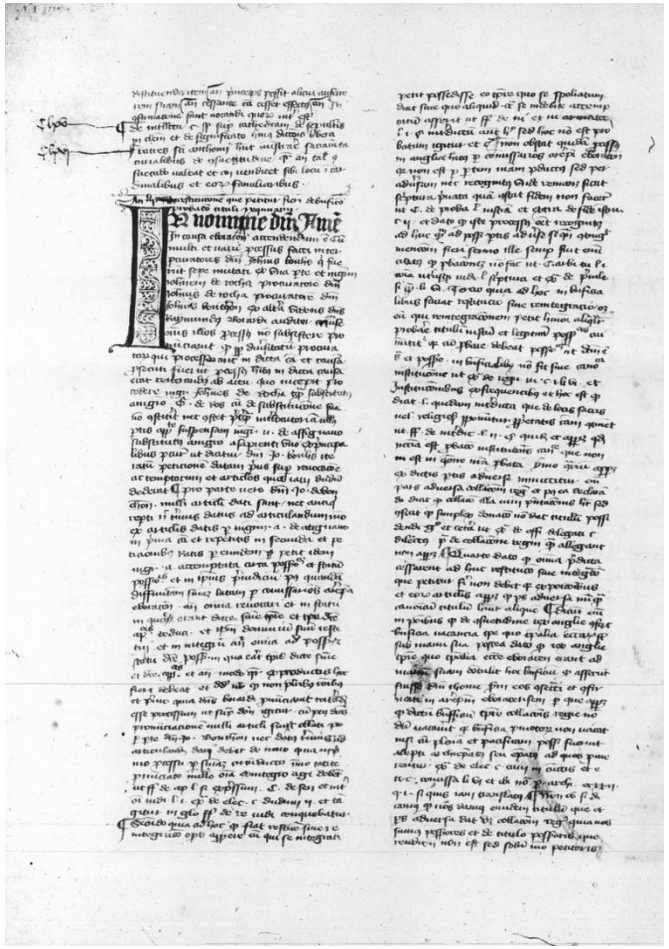


13th-century manuscript of the Accursian gloss. Munich, Bayerische Staatsbibliothek, Clm 3501.

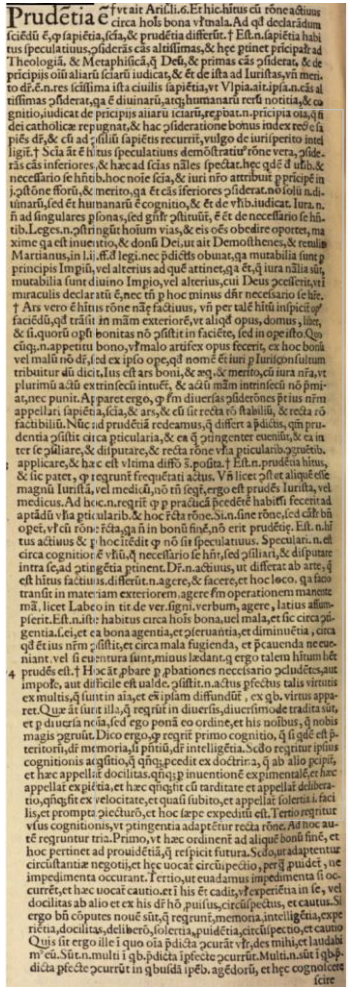
Millions of books, almost none machine-readable



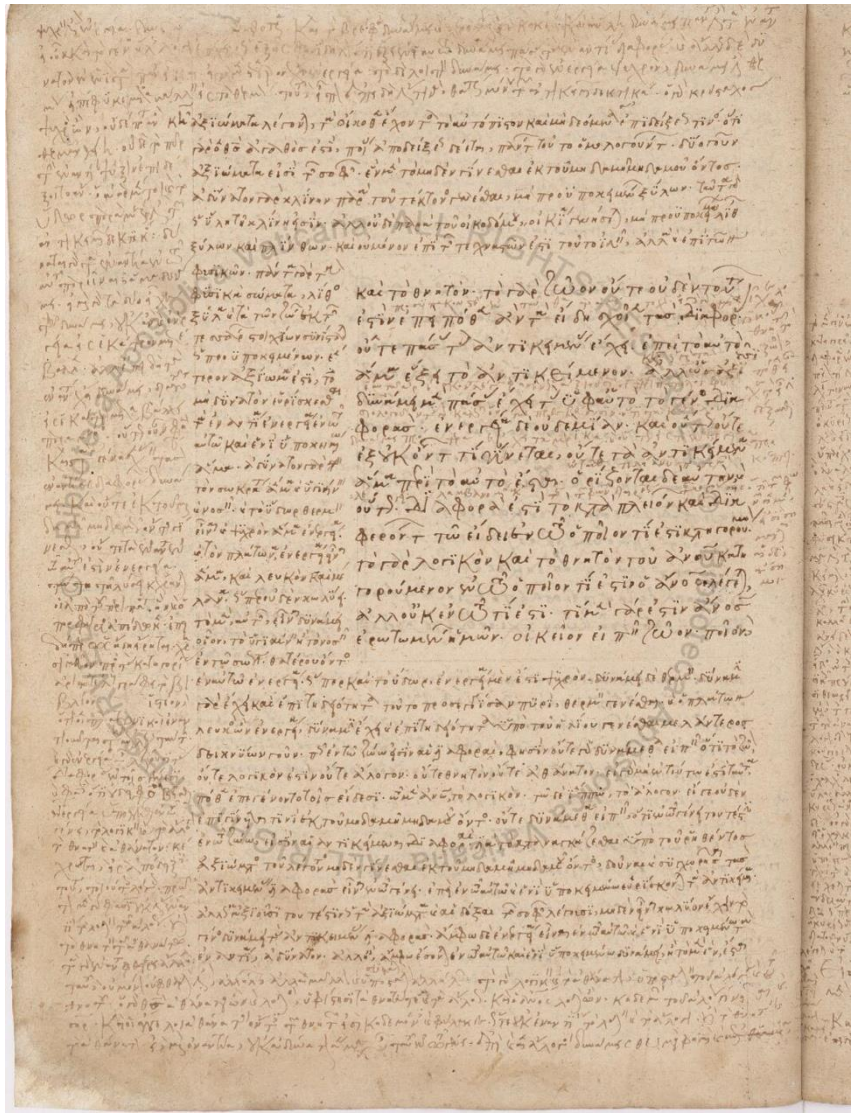
Baldus de Ubaldis on the Digest. 16th-century print.



Oldradus da Ponte, Consilia. Bordeaux, Bibl. municipale, ms. 404.



Bartolus, Tractatus testimoniorum. Venice, 1570.



Aristotle, Prior Analytics, with commentary. Vat. gr. 244, f. 20v.

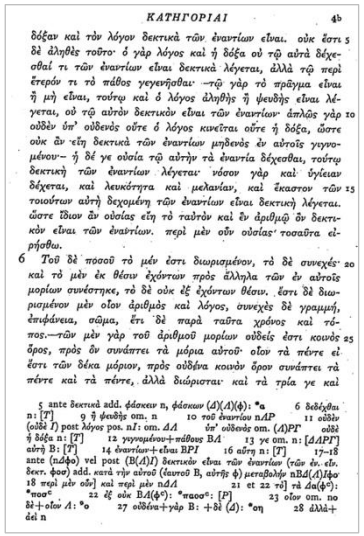
Photographed, online and *out of copyright* (oftentimes a photograph itself can be copyrighted, though this use is arguably transformative). But a photograph is not text: nothing in it can be searched, quoted, compared or translated until someone transcribes it.

Images: BSB München; Bordeaux, Bibl. municipale; Biblioteca Apostolica Vaticana.

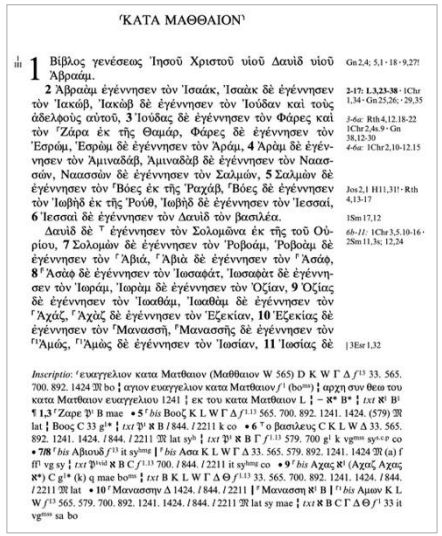
The goal for a scholar: critical editions

Census → **Transcribe** → Collate → Constitute → Publish

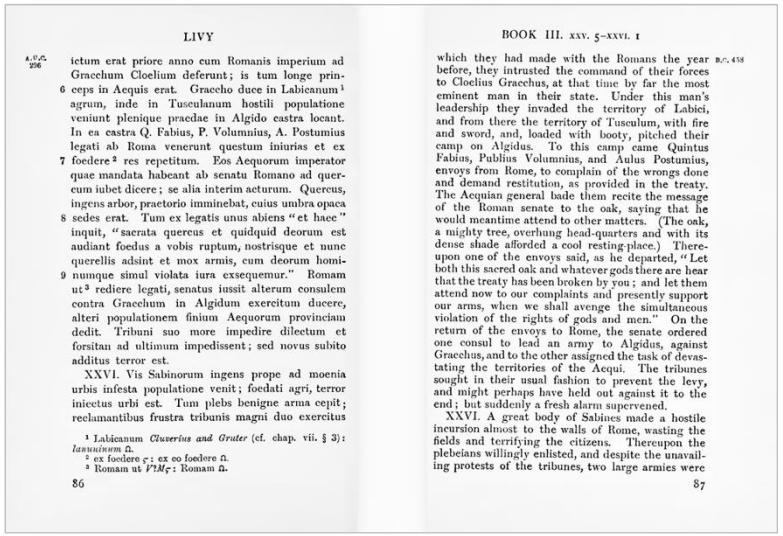
A critical edition is the standard text of a work, rebuilt by comparing every surviving copy. One edition can take a scholar decades. Every later step depends on the transcription, so it has to be right to the letter. The transcription stage is a bottleneck.



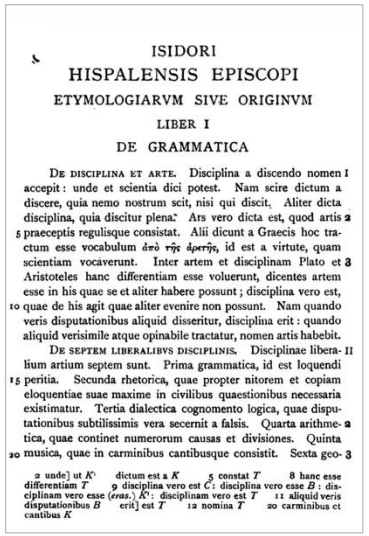
Oxford Classical Text: Aristotle, Categories



Nestle-Aland: Gospel of Matthew



Loeb: Livy, with facing translation

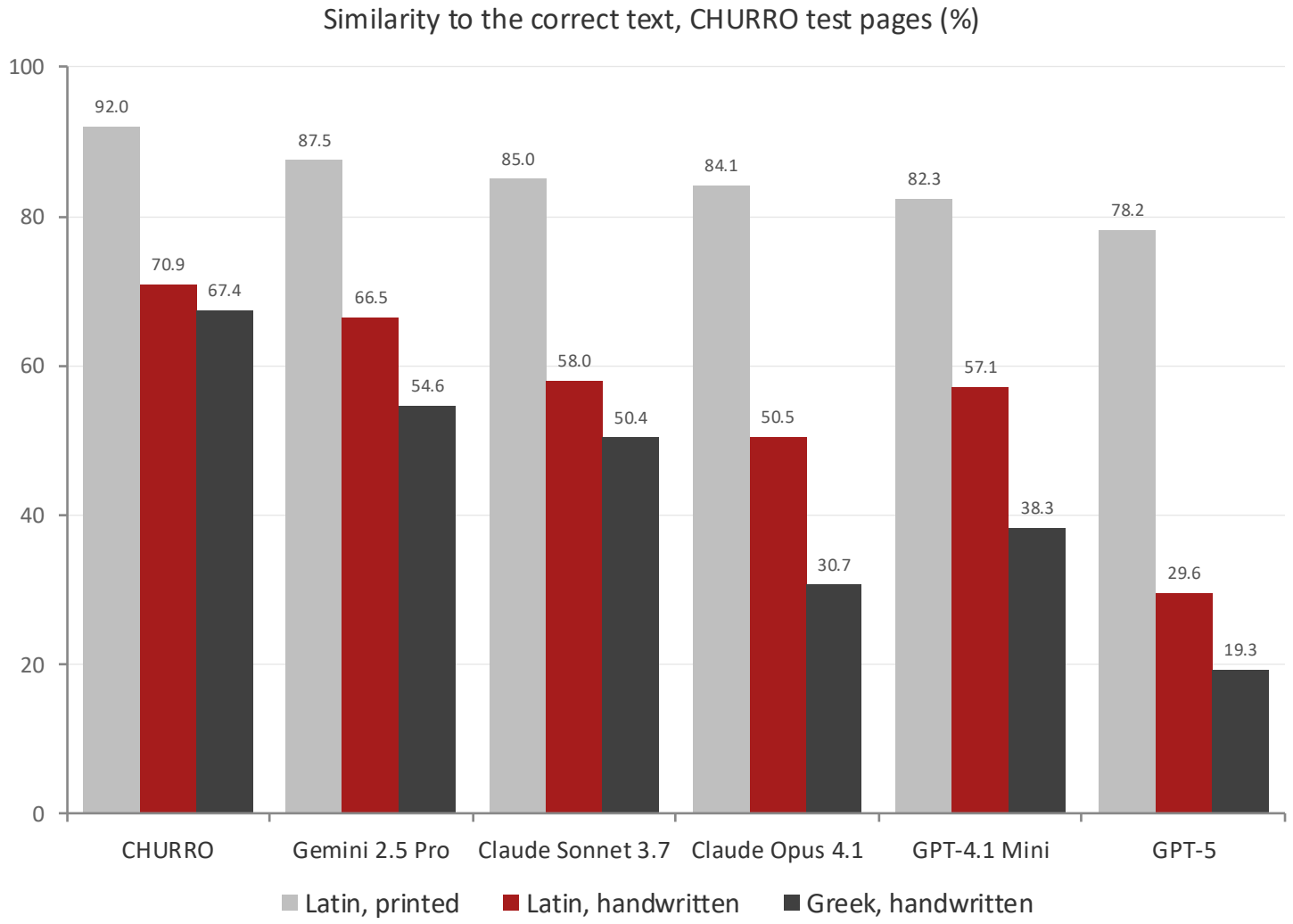


Isidore, Etymologiae I



Averroes, Long Commentary on the Metaphysics

State of the art



As of 2025, CHURRO is the best reader, and 15.5× more cost-effective than Gemini 2.5 Pro.

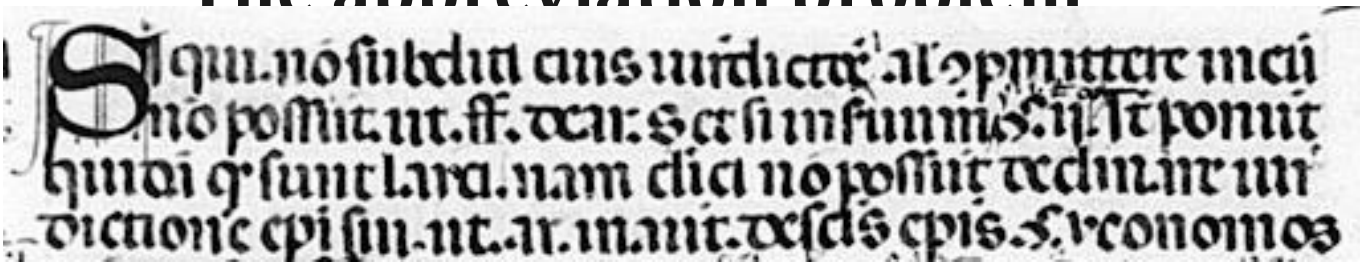
On handwritten Latin its score is 70.9: about three letters in ten are still wrong.

CHURRO is OCR: it copies the signs on the page. For these books, even a perfect copy of the signs is not yet a readable text. This text is often totally unusable.

However, Opus 5.5 and Gemini 3.1Pro are much better. When put into a careful pipeline, they can produce near full accuracy.

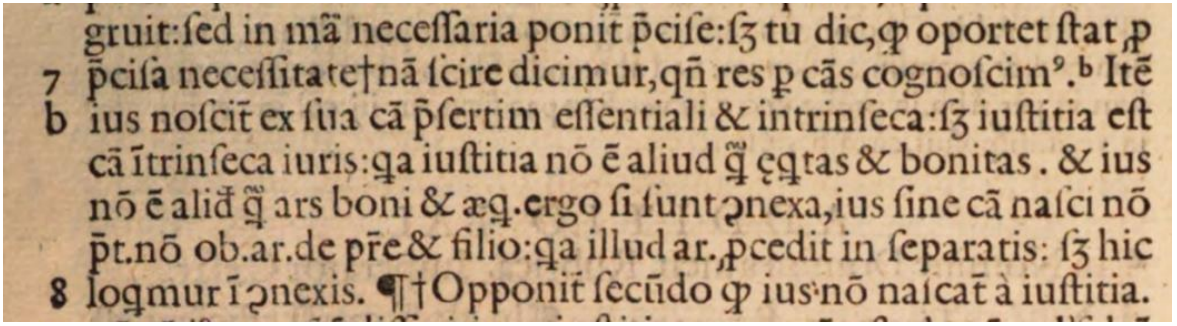
Semnani, Zhang, He, Tekgürler & Lam, “CHURRO,” EMNLP 2025 (arXiv:2509.19768), Tables 1–2 and abstract.

The abbreviation problem



Four lines of legal commentary. Munich, BSB, Clm 3501. The table follows one phrase, from line 3 into line 4.

Diplomatic (what perfect OCR returns)	Si qui . nō ſubditi eius iur'dictiōe al' pmittere in eū. nō poſſut .
Normalized	Si qui non subditi eius iurisdictioni aliter permittere in eum non possunt
Translation	If any who are not subject to his jurisdiction cannot otherwise submit to him



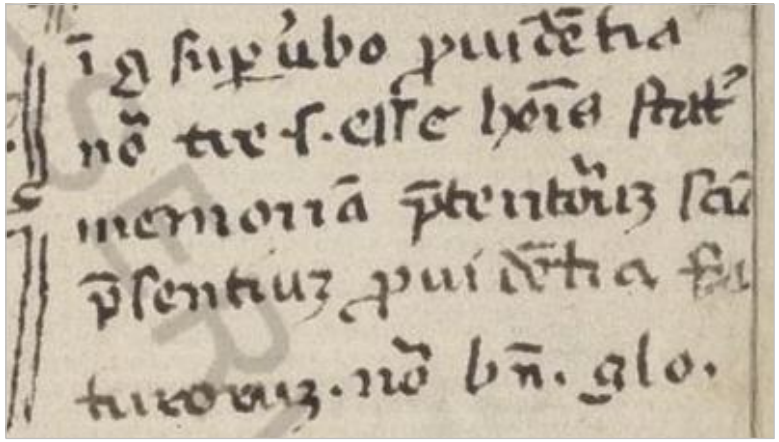
Nobody can search, quote or translate the first row, and a model trained on it learns the shorthand, not the Latin.

Normalization cannot easily be turned into an algorithm. The same sign stands for different words in different places; choosing the right one takes the grammar and the law. Even in the same line, the same word may be abbreviated in different ways (see the printed text for example)

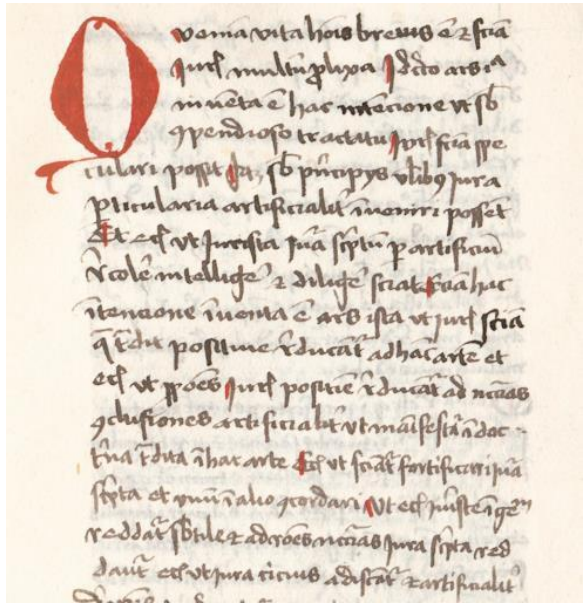
Diplomatic	Normalized
gruit:ſed in mã neceſſaria ponit p̃ciſe:ſʒ tu dic,ꝙ oportet ſtat ꝑ p̃ciſa neceſſitate † nã ſcire dicimur,qñ res ꝑ cãs cognoſcim⁹.ᵇ Itẽ ius noſcit̃ ex ſua cã p̃ſertim eſſentiali & intrinſeca:ſʒ iuſtitia eſt cã ĩtrinſeca iuris:qa iuſtitia nõ ẽ aliud q̃ ęqtas & bonitas . & ius nõ ẽ aliđ q̃ ars boni & æq.ergo ſi ſunt ɔnexa,ius ſine cã naſci nõ p̃t.nõ ob.ar.de p̃re & filio:qa illud ar. ꝑcedit in ſeparatis: ſʒ hic loq̃mur ĩ ɔnexis. ¶†Opponit̃ ſecũdo ꝙ ius nõ naſcat̃ à iuſtitia.	gruit: sed in materia necessaria ponit praecise: sed tu dic, quod oportet stat per praecisa necessitate † nam scire dicimur, quando res per causas cognoscimus. Item ius noscitur ex sua causa praesertim essentiali et intrinseca: sed iustitia est causa intrinseca iuris: quia iustitia non est aliud quam aequitas et bonitas. et ius non est aliud quam ars boni et aequi. ergo si sunt connexa, ius sine causa nasci non potest. non obstat argumentum de patre et filio: quia illud argumentum procedit in separatis: sed hic loquimur in connexis. ¶ Opponitur secundo quod ius non nascatur a iustitia

[fragment of line before] : but in a necessary matter he posits precisely: but you say, that it must stand by precise necessity † for we are said to know, when we recognize things through their causes. Likewise, law is known from its cause, especially the essential and intrinsic one: butjustice is the intrinsic cause of law: because justice is nothing other than equity and goodness. And law is nothing other than the art of the good and the equitable. Therefore, if they are connected, law cannot arise without a cause. The argument concerning the father and son is no obstacle: because that argument applies to separate things: but here we are speaking of connected things. ¶ It is objected, secondly, that law does not arise from justice

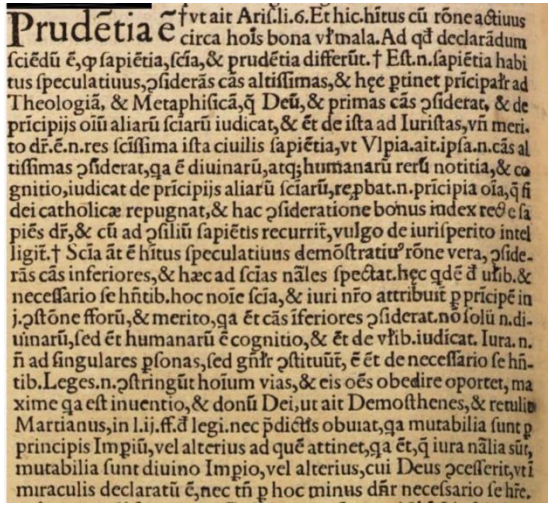
Abbreviations, Marginalia, Interlinear Notes, Glossa



A margin note: “On the word ‘prudence’, note its three parts: memory of the past, understanding of the present, foresight of the future.”

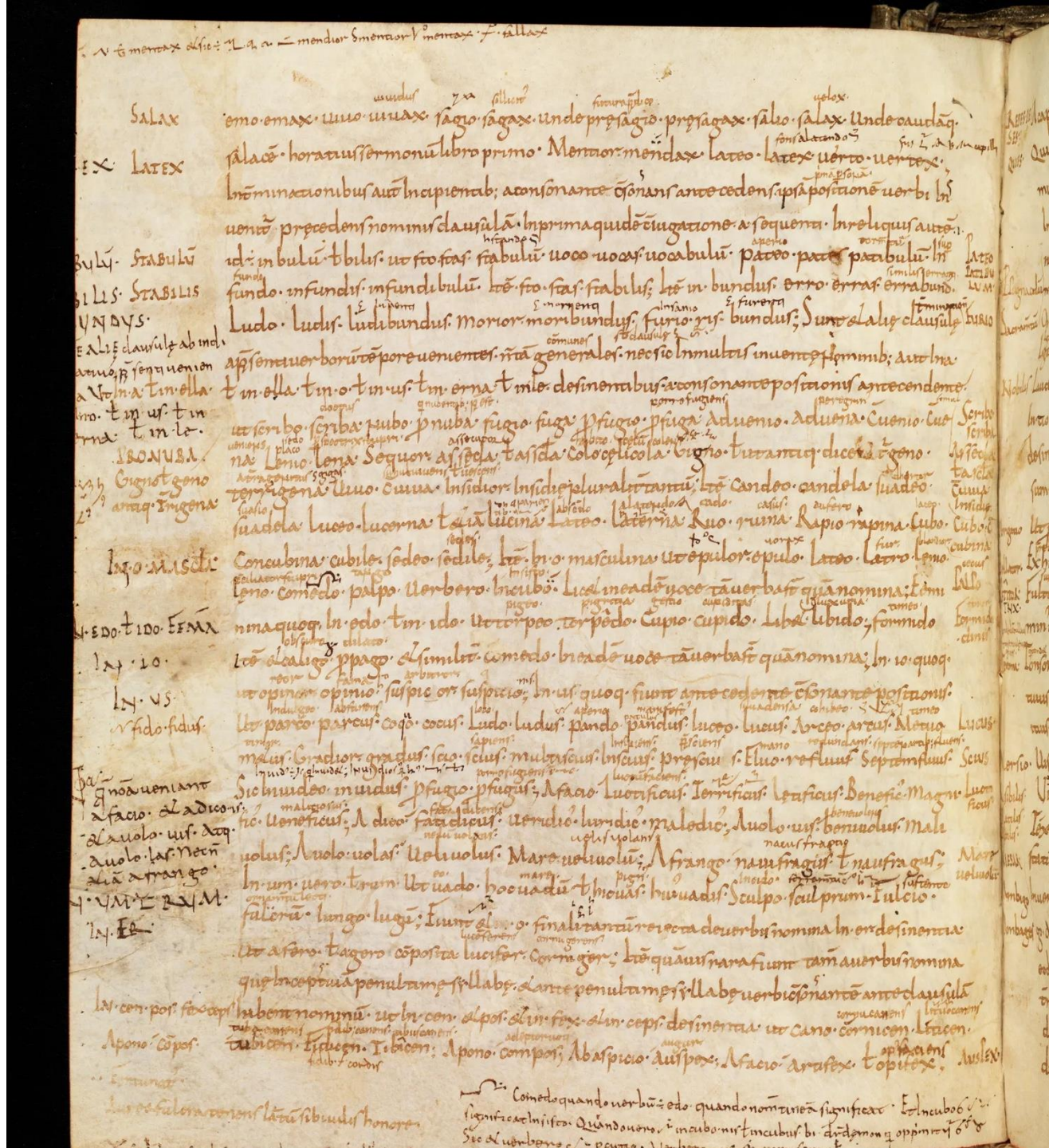


A formal book hand, using the same abbreviations.

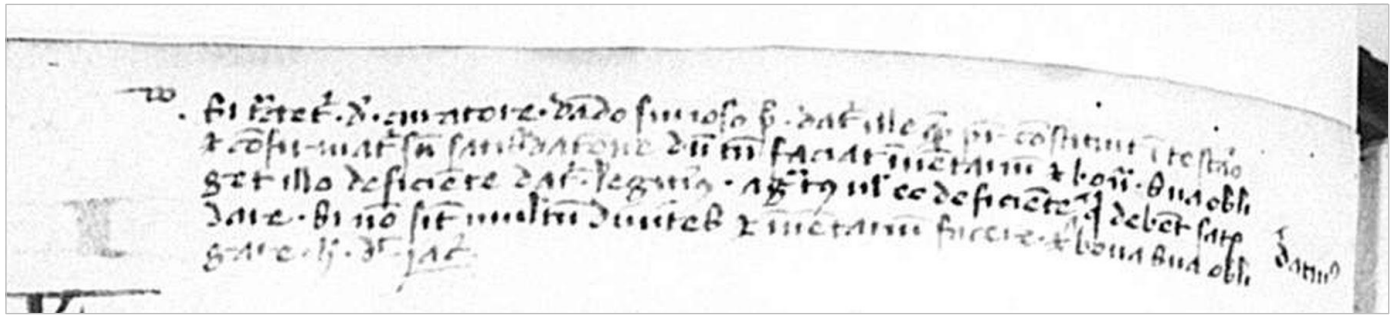


Prudētia ē ſ vt ait Arīſt. li. 6. Et hic hītus cū rōne actiuus circa hoīs bona vt mala. Ad qđ declarādum ſciēdū ē, q ſapiētia, ſcia, & prudētia differūt. † Est. n. ſapiētia habitus ſpeculatiuus, ꝙſiderās cās altiſſimas, & hęc ptinet prīcipaīr ad Theologiā, & Metaphiſicā, q̄ Deū, & primas cās ꝙſiderat, & de prīcipijs oīū aliarū ſciarū iudicat, & ēt de iſta ad Iuriſtas, vñ merito dr. ē. n. res ſcīſſima iſta ciuilis ſapiētia, vt Vlpia. ait. ipſa. n. cās altiſſimas ꝙſiderat, qa ē diuinarū, atq; humanarū rerū notitia, & cognitio, iudicat de prīcipijs aliarū ſciarū, re; pbat. n. prīcipia oīa, q̄ ſi dei catholicæ repugnat, & hac ꝙſideratione bonus iudex reō e ſapiēs dr, & cū ad ꝙſiliū ſapiētis recurrīt, vulgo de iuriſperito intelligīt. † Scia āt ē hītus ſpeculatiuus demōſtratiu rōne vera, ꝙſiderās cās inferiores, & hæc ad ſcias nāles ſpectat. hęc qđē đ uſib. & neceſſario ſe hñtib. hoc noīe ſcia, & iuri nō attribuīt p prīcipē in j. ꝓſtōne fforū, & merito, qa ēt cās iferiores ꝙſiderat. nō ſolū n. diuinarū, ſed ēt humanarū ē cognitio, & ēt de vſib. iudicat. Iura. n. ñ ad ſingulares pſonas, ſed gñr ꝓſtituūt, ē ēt de neceſſario ſe hñtib. Leges. n. ꝓſtringūt hoīum vias, & eis oēs obedire oportet, maxime qa eſt inuentio, & donū Dei, ut ait Demoſthenes, & retulit Martianus, in l. ij. ff. đ legi. nec p̄dictis obuiat, qa mutabilia ſunt p principis Impiū, vel alterius ad quē attinet, qa ēt, q̄ iura nālia ſūt, mutabilia ſunt diuino Impio, vel alterius, cui Deus ꝑceſſerit, vt miraculis declaratū ē, nec tñ p hoc minus dñr neceſsario ſe hñe.

Early print: “Prudence is, as Aristotle says in Ethics book 6, an active habit with reason concerning human goods and evils.”



Reading means Deciding: Balancing Language+Context/Visual



A five-line note in a fast legal hand. Clm 3501, f. 29v, top margin (microfilm). This is a marginal gloss by Damasius.

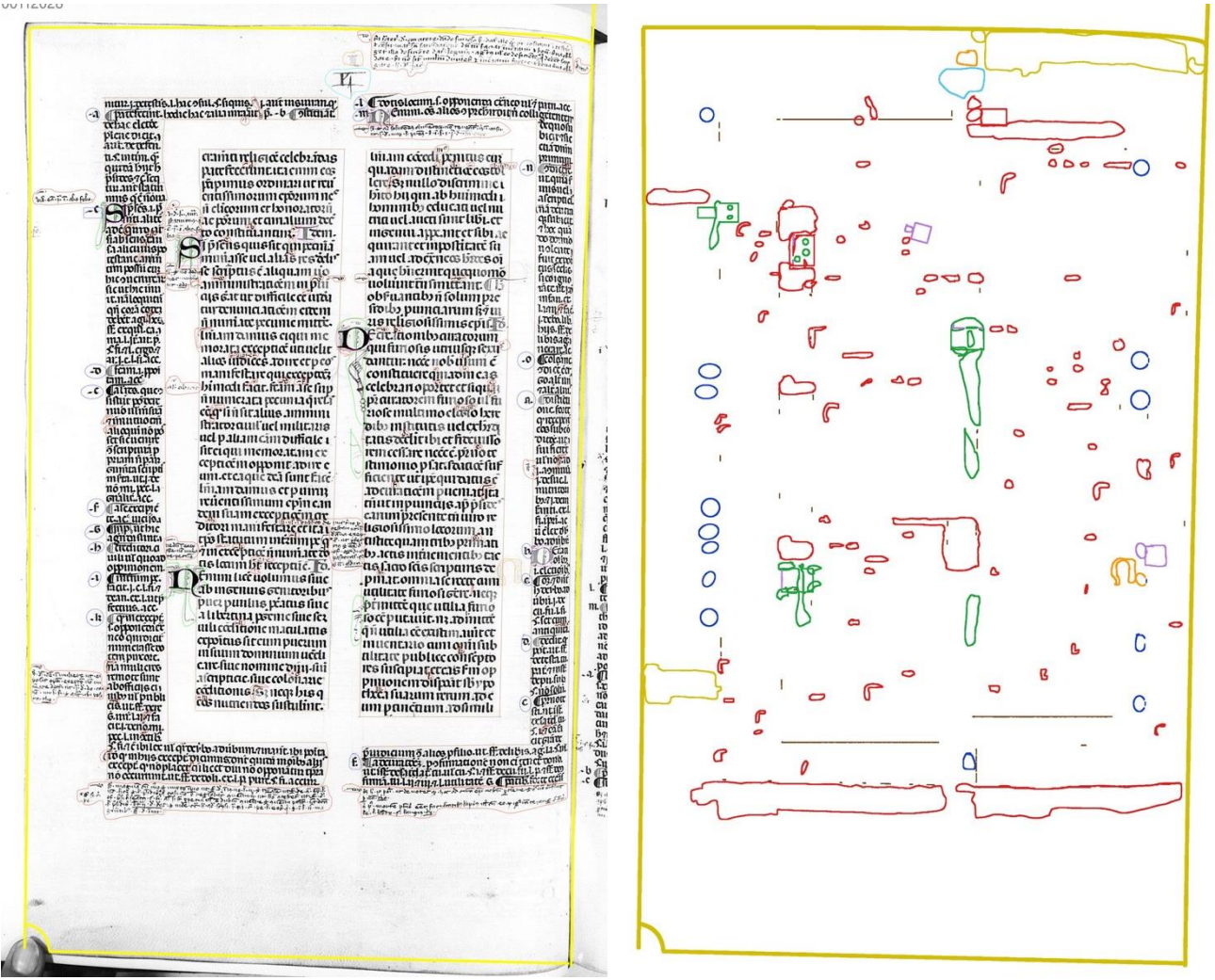
Gemini 3.1 Pro asked to simply do OCR	multū ðıuıteſ….
Gemini 3.1 Pro, open-ended (Correct)	Nota: si patet ….

A legal citation: “l. 1. § 1.” (law 1, section 1)

CHURRO	5/s.f
Claude, told only to OCR	1.l.i
Claude, in the Pipeline (correct)	l.1.§.1

Neither kind of evidence is enough alone. From shapes alone a model cannot tell u from v; from sense alone it writes fluent Latin that is not on the page. So abbreviations cannot be a clean-up step after OCR. Where ink and sense conflict, the ink wins.

The agent: the Gold Manual



One stage: every piece of ink on the page is assigned an owner before anything is read.

A written procedure an AI agent follows to produce a checked ("gold") transcription: 25 stages. Briefly:

Identify · Measure · Read · Interpret/Retreat · Emit TEI

Each stage ends in a gate: a test, written in code, that must pass before the next stage can start.

Do not …	until …
read with context	the blind reading is locked
accept a reading because readers agree	the ink has settled it, with a grammatical check
spell out abbreviations	you can say what the sentence means
call a word illegible	a reader has been told what the text is about

Output: checked transcriptions in every layer, usable as training data for CHURRO.

The “Gold Manual” Pipeline: Creating an AI agent a fixed procedure in 25 stages, each with a gate

In plain terms: The **Gold Manual** is a written procedure that an AI agent follows to produce a checked (‘gold’) transcription. It consists of one instruction file (the “gold manual” of 12,765 lines plus 141 small programs. The agent has has roughly **25 major stages** (next four slides). Every stage ends with a **gate**: a test, written in code, that must pass before the next stage may start.

Five major phases

I. Identify (Stages 0–2): what is this book, and what can this photo show?
II. Measure (3–9): the page, its ink, its areas, its lines, its scribes; one standard image per area
III. Read (10–15): marks first; a blind reading; machine readers; a reading with context; decide; prove each letter
IV. Interpret (16–19): spell out, standardize, translate, link
V. Publish (20–24): save, print, score, repeat until nothing is left, sign off

After the Stage 0, there is a basic loop:

- A: describe layout
- B: count all marks of ink on the page
- C: edit/check the page description
- D: crop pased on the ink layout
- R*: crop one image per area of ink
- M*: launch multiple machine readers for “adjudication
- E: a blind single reading by an agent
- F: a reading by an agent with context
- G: “adjudication
- V*: ranking best ”blind” readings given context
- H*: decision on wordings
- I: final output

Some rules are: : No stage may overwrite an earlier stage's record. A stage that fails its gate stops; it never guesses.
No crop until every borderline area is assigned; no decision on one kind of evidence; a problem solved by sense is written back into the letter-shape chart, the abbreviation list and the layout rules.

Before Stage 0, every run asks

1. What did the last session learn? (it saved notes)
2. Where did this run stop? (read from files, never from memory)
3. Is any saved evidence unused?
4. What is the spending limit?
5. How much checked text exists, and is it growing?
6. Did any step decide using only one kind of evidence?
7. Who wrote this page?
8. Is every ‘uncertain’ mark justified?
9. Is every reader seeing the same image?

Some ordering rules: These are called “gates” these are important to make sure we proceed in the proper manner (built because of many failures)

Do not…	until…
read with context	the blind reading is locked
accept a reading because readers agree	the ink has settled with a grammatical pass
read the next page	this page's checked letters are added to the reference
spell out abbreviations	you can say what the sentence means
change the exact copy for sense	you have re-opened the image at that spot
produce a document	every gap is filled, and it is generated from the master file
call a word illegible	a reader has been told what the text is about
accept a line no grammar can parse	it has been sent back round the loop
finish	notes for the next session are saved

Steps 1–4: identity, image limits and the leaf

1 Witness Identity: Identify the text and its copy

Distinguish the work, edition, physical copy and photographs. Enter the copy in the witness census.

2 Surrogate Assay: Test what the photograph can show

Test the photograph before anyone reads it. Establish direct scan or microfilm. Is it a photo of the book or of a microfilm? Black-and-white or color? What is its true maximum resolution? How tall are the small letters in each size of writing? Measure physical scale when knowable and small-letter height. Test library text and line data separately. List every image limitation.

3 Describe the book and classify each page

Record material, format, binding, groups of leaves and writing sizes. Classify every opening as blank, show-through, printing furniture or text-bearing, with evidence for the choice.

4 Map the leaf and its geometry

Separate the leaf from the background and intrusions. Cut away the facing leaf with checks on both sides. Draw the actual page outline. Measure each line’s slope and error, then inspect the overlays.

Gate: Stop until identity, image limits, every page class and the observed leaf geometry are recorded.

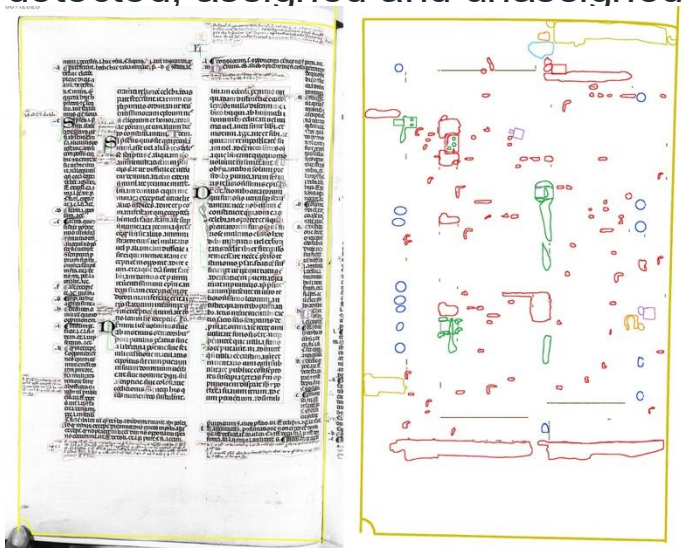
Source: supplied PowerPoint, original slides 28–29. Full text, details and citations in speaker notes.

Steps 5–7: ink, regions and layout type

Every piece of ink needs an owner. A region name is a claim to test.

5 Account for every piece of ink

Separate ink from background paper using a measured blank patch and restrict the count to the leaf. Separate missed ink from excluded decoration. Label every blob. Assign ownership by outline overlap, using the full owner list (law text, commentary, note between lines, decoration, ruled line, stain, show-through…: 15 kinds). Publish detected, assigned and unassigned counts.



6 Find writing regions by size, then position

Measure line spacing. Test false repeats at halves and thirds and reject groups below the stated size floors. Find columns within each writing size. Reconcile missing, shared and clipped regions. Inspect the intermediate measurements.

7 Choose and test the layout method (Layout-type declaration)

Classify each page and choose its method. Stop the size-first method for elastic commentary (expanding and shrinking) or commentary with embedded text. Record guide-line relationships. Treat every region label as a hypothesis to test.

Gate: Stop until ink counts reconcile, unexplained ink is listed and the region map fits the page class.

Steps 8–10: lines, scribes and crops

Real failure: 56 Gemini runs used a crop about 80 pixels short, cutting off the signature.

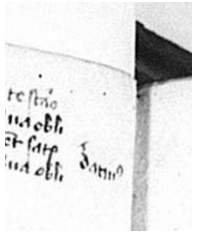
8 Recover and verify the line grid

Fit spacing separately for each block. Split lines at ink minima and check one-to-one matches. Reward ink on lines and penalize ink between them. Record errors, remove duplicate lines and check collisions.

9 Check every region boundary

Measure spare space and overflow separately on each edge. Flag stretched boxes and recount lines from the outline and ink. Save corrected bounds beside the detector's record. Do not crop at this stage.

Reading ī tesŧo improved from 0/56 runs to 7/8.
Reading boñ sua improved from 4/56 to 8/8.

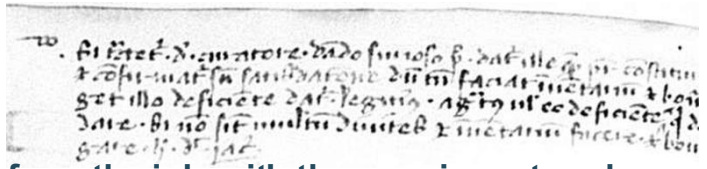


Signature detail

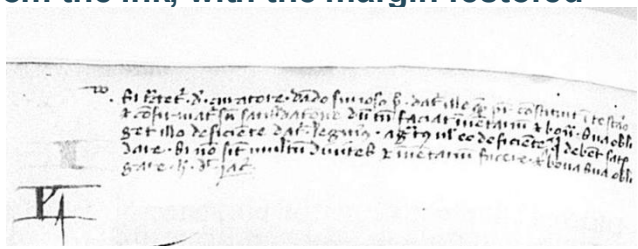
10 Identify scribes and deliver complete crops

Identify each scribe and role, rechecking inherited claims. Crop one writing layer from its ink, keeping detached signs and whole lines. Target about 90-pixel letters and verify delivery limits. Preserve grayscale. If a line needs splitting, overlap segments by at least 9%. Give every reader one fingerprinted image and ink checklist.

Before: the signature and line ends were clipped



After: cut from the ink, with the margin restored



Gate: Stop until line counts, edges, writing layers, delivered letter height and image fingerprints all check out.

Steps 11–13: signs, independent readings and context

First record the marks. Then compare independent readings. Meaning must stay grounded in the image.

11 Record and test every small mark

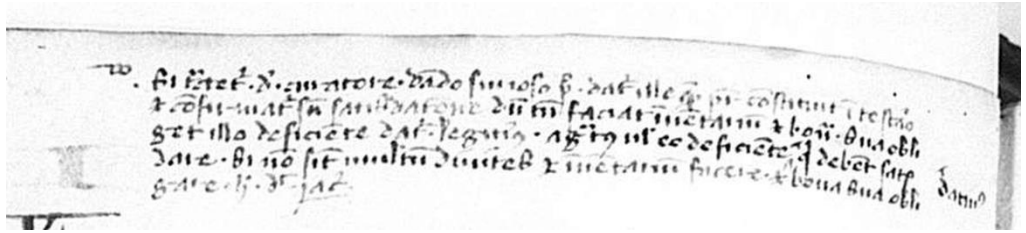
Record signs before naming letters. Detect abbreviation bars, between-line additions and marginal key letters with their separate tests. Inspect every detector’s image grid. Publish false-positive rates or explicitly explain what has not been measured.

12 Freeze a blind reading and run other readers

The blind reader gets only the image, letter atlas and sign inventory. Record whole lines and letter confidence, then freeze the result before context. Run CHURRO, kraken and a tested script specialist remotely on identical images. Use small tasks, matching writing-layer instructions and spending caps.

13 Read with context, checked against the ink

Name the whole region, work, subject and word being explained. Parse formulas and citations first. Check every letter against the mark record and count unresolved shapes. Preserve lines, splits, additions and signs. Write L2–L5 and test the translation. Save revisions and enforce contamination and cross-layer checks.



Wrong: in sanitate
Verified: sine inquisitione

Gate: Stop until detectors are checked, the blind reading is isolated and every required machine reading is back.

Steps 14–15: disagreements and decisions

Agreement does not prove accuracy. The adjudicator must open the image and test the rivals.

14 Align readers and name disagreements

Align letter sequences with merges and splits. Name each disagreement. Report agreement before accuracy on agreed and disputed parts. Check vertical-stroke counts and line widths. Measure the best visual reader for each scribe, with an uncertainty interval.

All eight escalation checks

1 Inspect the ink crop
2 Ask one cheap reader
3 Ask five diverse readers on the large crop
4 Consult reference banks while reading
5 Consult the doctrine bank and its rival
6 Use a stronger model with the banks
7 Read the ink yourself
8 Adjudicate by hand

15 Decide each disputed reading against the image

Show a checker the crop and randomized rivals, including cannot tell. Complete the eight escalation checks. Hear the measured best reader first and justify overrides. Never vote. Classify proposals against the ink. Open the crop to choose, count a gap or record an externally resolved rival.

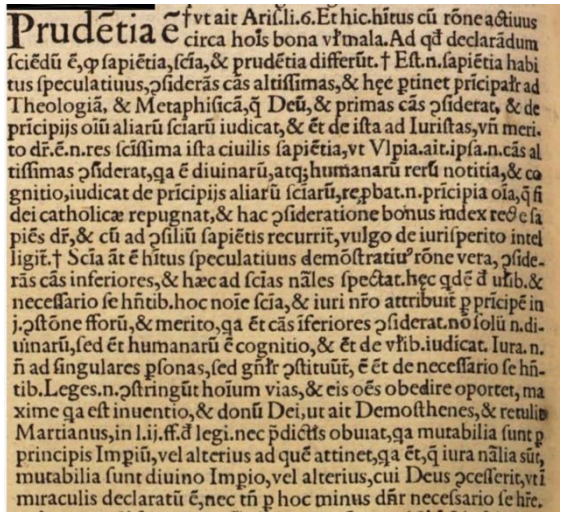
Gate: Stop until contamination and cross-layer tests pass, all ink is decided and reader-lead overrides are justified.

Steps 16–18: letter verdicts and expansion

Verified means tested against alternatives. The written signs remain separate from their expansions.

16 Prove a verdict for every letter

Calibrate stroke counts on checked word boundaries. Optimize word boundaries and letter ownership. Exclude plausible rivals through repeatable tests. Report accepted yield separately from accuracy. Give every letter one verdict: unexamined is never gold. Update the ledger and teach the next page from gold only.


Prudẽtia ẽ † vt ait Ariſ.li.6.Et hic.hĩtus cũ rõne actiuus
circa hoĩs bona vl̃mala.Ad qđ declarãdum
ſciẽdũ ẽ,ꝙ ſapiẽtia,ſcĩa,& prudẽtia differũt.† Eſt.n.ſapiẽtia habi
tus ſpeculatiuus,ꝯſiderãs cãs altiſſimas,& hęc ꝑtinet prĩcipal̃r ad
Theologiã, & Metaphiſicã,q̃ Deũ,& primas cãs ꝯſiderat, & de
prĩcipijs oĩũ aliarũ ſciarũ iudicat,& ẽt de iſta ad Iuriſtas,vñ meri.
to dr̃.ẽ.n.res ſcĩſſima iſta ciuilis ſapiẽtia,vt Vlpia.ait.ipſa.n.cãs al
tiſſimas ꝯſiderat,qa ẽ diuinarũ,atq; humanarũ rerũ notitia,& co
gnitio,iudicat de prĩcipijs aliarũ ſciarũ,reꝑbat.n.prĩcipia oĩa,q̃ fi
dei catholicæ repugnat,& hac ꝯſideratione bonus iudex rectẽ ſa
piẽs dr̃,& cũ ad ꝯſiliũ ſapiẽtis recurrit̃,vulgo de iuriſperito intel
ligit̃.† Scĩa ãt ẽ hĩtus ſpeculatiuus demõſtratiu⁹ rõne vera, ꝯſide-
rãs cãs inferiores,& hæc ad ſcĩas nãles ſpectat.hęc qdẽ đ vl̃ib.&
neceſſario ſe hñtib.hoc noĩe ſcĩa,& iuri nr̃o attribuit̃ ꝑ prĩcipẽ in
j.ꝯſtõne fforũ,& merito,qa ẽt cãs ĩferiores ꝯſiderat.nõ ſolũ n.di-
uinarũ,ſed ẽt humanarũ ẽ cognitio,& ẽt de vl̃ib.iudicat. Iura.n.
ñ ad ſingulares ꝑſonas,ſed gñl̃r ꝯſtituũt̃, ẽ ẽt de neceſſario ſe hñ-
tib.Leges.n.ꝯſtringũt hoĩum vias,& eis oẽs obedire oportet, ma
xime qa eſt inuentio,& donũ Dei,ut ait Demoſthenes,& retulit
Martianus,in l.ij.ff.đ legi.nec p̃dictis obuiat,qa mutabilia ſunt ꝑ
principis Impiũ,vel alterius ad quẽ attinet,qa ẽt,q̃ iura nãlia sũt,
mutabilia ſunt diuino Impio,vel alterius,cui Deus ꝯceſſerit,vt ĩ
miraculis declaratũ ẽ,nec tñ ꝑ hoc minus dñr neceſsario ſe hr̃e.

17 Expand every abbreviation sign explicitly

Link each sign to its supplied letters and versioned rule. Record ambiguous alternatives or deliberate non-expansion. Use full spellings found in this same book. Measure abbreviation-mark density in every comparison.

18 Produce traceable standardized text

Apply logged, versioned spelling rules only after L2 is settled. Protect names, numbers, negations and citations. Every change remains traceable. An upstream correction makes dependent layers stale until rebuilt.

Gate: Stop until every letter has a verdict, false acceptances are calibrated, signs align and all changes are logged.

Steps 19–22: translation, links and output

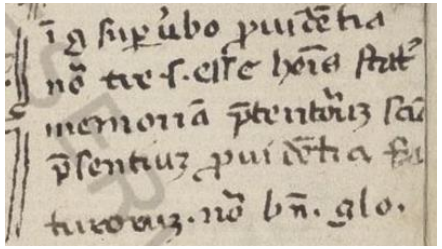
TEI is the scholarly text format. IIIF is the image-sharing standard. A graph stores the relationships.

19 Translate the fixed text and guard revisions

Translate saved L4, including every note and addition, in whole-unit context while keeping lines. Revisit L2 only with at least two evidence types, the exact image location and document context. Translation must not silently rewrite the Latin.

20 Record every relationship

Store links in the 13 relationship families. Record both endpoints, type, mechanism, evidence, date where knowable and certainty. Allow indeterminate on every attribute instead of guessing.



The source flags sai as unresolved. A fluent translation does not settle the letters.

21 Write and validate the structured source

Store text in TEI, identities and events in a graph, and images through IIIF, joined by one identifier. Privilege no reading as the original. Check forbidden strings, schema validity, every link and a lossless round trip.

22 Generate and inspect the readable editions

Read the TEI back to produce matching as-written, expanded and translated documents, with all notes, variant records and sign keys, plus a page-matching transcript. Check installed fonts, missing signs and unintended long-s ligatures.

Gate: Stop until segments align, links resolve, TEI round-trips without loss and sibling documents agree.

Steps 23–25: scoring, completion and emission

Test scores are acceptance thresholds. They do not establish a perfect transcription.

23 Measure accuracy and completeness honestly

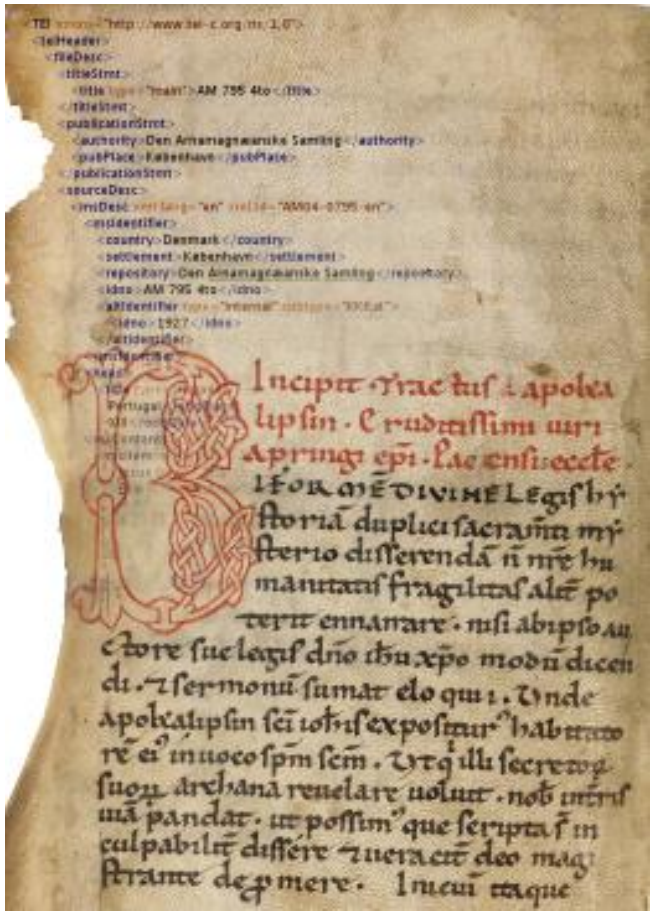
Report reader agreement first. Measure errors by layer, region, evidence and control passage, plus confidence, coverage and remaining work. Score relationships with partial credit and baselines. Resample whole leaves for error bars. Answer all 12 completeness questions numerically.

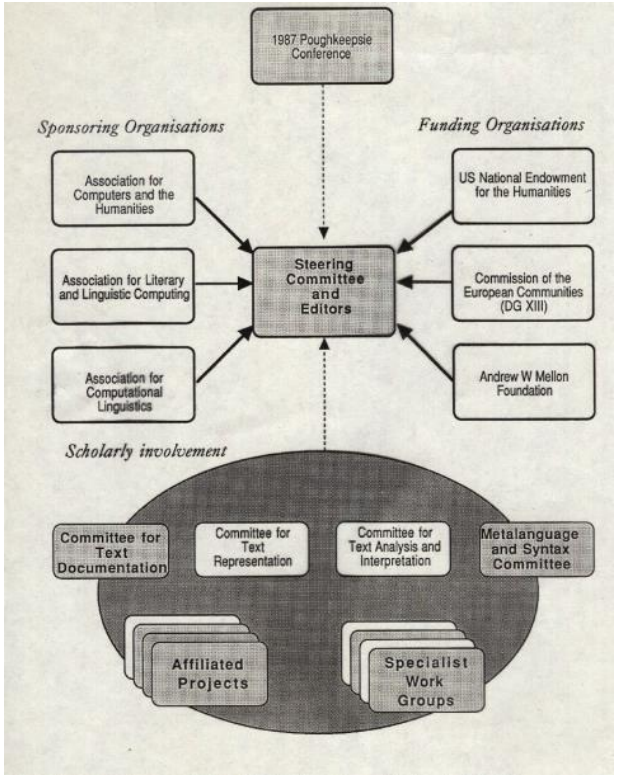
24 Repeat failed work and justify uncertainty

Repeat failed reading-through-scoring stages until no unfinished item remains. Investigate an empty first pass as possible fabrication. Resolve gaps using the evidence. An asterisk survives only when a named rival is tied on both strokes and meaning.

Witness standard: every mark accounted for, zero omitted whole lines, and unflagged errors ≤0.1% at the 95% upper bound.

25 Emit final document as TEI-complaint version





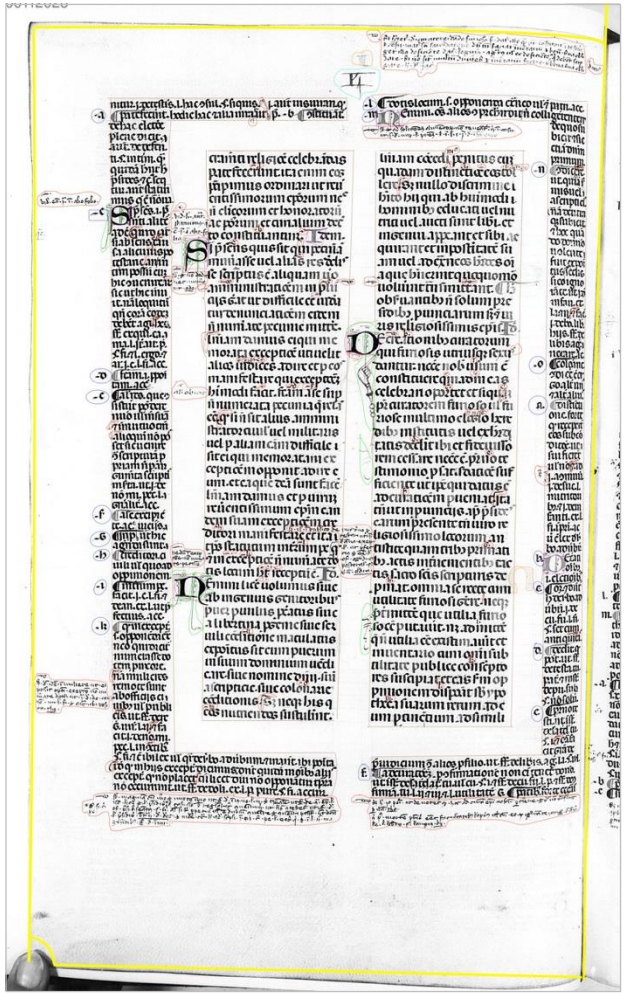
Gate: A run ends only when all six exit conditions pass. No blocked phase, unused evidence or unfinished obligation remains.

Results: the same four lines, read three ways

My own reading	Si qui . nō ſubditi eius iur'dictiōe al' pmittere in eū nō poſſut . ut . ff . de ar . S . et ſi in ſummis . ō . iu . Itē ponit hui moī q' ſunt laici . nam cl'ici nō poſſūt declīare in iur' dictione epi ſui . ut . ar. in aut. de scīs epis . R . vconomos
My agent	Si qui . nō ſubditi eius iur'dictiōe al' ꝑmittere in eū nō poſſūt . ut . ff . de ar. S. et ſi in ſūmis. ō . iu . Itē ponit hui[9] moī q' ſunt laici . nam cl'ici nō poſſūt declīare in iur' dictiōe epī ſui . ut . ar. in aut. de ſcīs epīs . § . yconomos
CHURRO (unintelligible text)	Si qui. nō sit clicus miſicdā al' pmittet maī nō poſſit. ut. ff. de ar. s. ca. si in ſummo. l. ij. Ti ponit hī noi q' ſunt laica. nam clici nō poſſūt redimre iuri dictione epi ſin. ut. ar. in aut. de scīs epis. ſ. vconomos

Red = substantively differs from my reading. My agent's differences are mostly marks my keyboard flattened: the text is readable; CHURRO's are different words.

Open problems



f. 29v marked by hand: 212 marks in eight colours, each a link the machine has to find.

Layout. Main columns are found well (overlap 0.75–0.86), margins poorly (0.19–0.33). The margins hold the notes, citations and signatures. This is one of the most important

Preserving All Information:, Glossa, Marginalia, Interlinear Notes

Producing a Usable (and accurate) Text: abbreviations

Invention. Models still write fluent Latin that is not on the page. For example, “in sanitate” where the ink says “sine inquisitione” (without inquiry). However, this happens significantly less when the model is told to translate and think about the grammar, syntax, context, content, etc.

Newer general models. On the hardest note, Gemini 3.1 Pro read 55 of 57 words; my pipeline at the time read 25.

Ground truth. Checked pages are slow and expensive to make.

Next: checked pages → train CHURRO → read the millions.

Transcription of Historical Documents

1800-1900s Chinese newspapers



Multi-stage special-purpose agents created by expert

	# Claude passes	Cost
Latin manuscript	5	10 min a page ($5-$8)
Chinese newspapers	3	4-5 min per two-page spread ($1)

Too costly and slow for millions of pages

Can we speed it up: Use Claude to generate training data!

Preliminary Results on Chinese Newspapers: The Open Version Beats Claude

Find the boxes→Read each box→Group boxes into articles

Faster R-CNN CHURRO-3B Qwen2.5-32B

- Training data:
  - Reader about 44 pages read by Claude, $19.
  - Box finding, box grouping, hand annotations (ECPO dataset)
- Completely open source: no Claude needed at run time.

23 test spreads	small model	Claude
Text read correctly	0.86	0.82
Articles grouping (of those read)	0.70	0.70
Articles grouping (all)	0.67	0.55
Text read	0.96	0.79



Jingbao (Shanghai, 1930s)

Early Chinese Periodical Online (ECPO) Database, Heidelberg Research Architecture

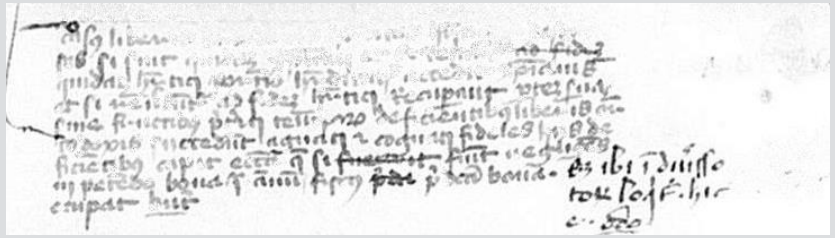
Latin Law Book

Latin law book: the Gold Manual reader (Biondo)

One region, five passes.

1. **Marks**: list every abbreviation sign, resolve nothing
2. **Blind reading**:
   letters only; unreadable spans written as [...n...]
3. **Constrained reading**:
   context first;
   abbreviation & citation lists may narrow choices
4. **Compare with a second try:**
   every difference is a work item
5. **Adjudicate each difference against the ink, with a zoom tool**

EXAMPLE one margin note, five passes

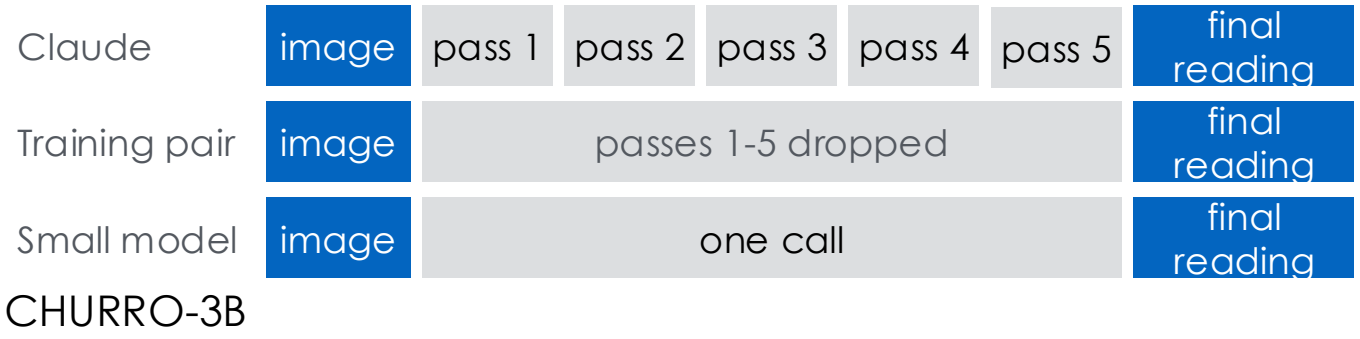


marks | blind | constrained | compare | adjudicate | final

Train Churros on end-to-end results.

Training Churros on each pass is worse than end-to-end-training!

Latin Manuscript (Work-in-progress)



CHURRO-3B

Training data generation cost: $70

No ground truth

Check against the first Claude run



EXAMPLE a medieval Latin law book.



Munich, BSB Clm 3501. Dense abbreviations: much of each word is written as a mark above or through the letters.

Summary

	Claude agent	Small models
Chinese newspaper	4-5 min per two-page spread ($1)	Better performance completely open source one GPU, no API cost
Latin manuscript	25 min for 79 regions	Performance - work in progress 1 min for 69 regions on one GPU
One-Time teaching cost		$19 for Chinese newspaper reader; $70 for Latin

Take-Aways: We still need special-purpose agents

- Many **important, repetitive, time-consuming** expert processes
  e.g. historical transcriptions
  - Coding agents cannot perform end-to-end tasks
  - Manual annotations are too difficult (expertise, time, cost)
  - Coding agents can perform (many) expert-decomposed steps
    - But it is too costly
- Solution: **Use coding agents to teach small models by generating a small amount of training data**
  - Two methods: step-by-step or end-to-end

Hot Topics on LLMs

- Coding agents are powerful
  - **Do we need to continue to build purpose-built agents?** (Jiacheng Sang, Shicheng Liu)
  - **Why & how to build purpose-built agents? (Case study: historical OCR)** (Andrew Biondo, Cyrus Zhou)
- Be careful! Watch out for LLM cheating and reward hacking
  - **Reward hacking incidents and prevention** (Joachim Baumann)
  - **A low-level run-time substrate for performance/security?** (Ananjan Nandi)

How do today's AI agents fail?



Agentic AI, Fall 2026
CS 224V

Link to slides:
https://docs.google.com/presentation/d/1xAe3kvPoOIW7KSneFvEw6sy8QteREN42MGu2TC81UuE/edit?usp=sharing

Joachim Baumann

Postdoc @ Stanford

SALT & STAIR Labs

Stanford University

Hot Topics in LLMs

- Coding agents are powerful
  - **Do we need to continue to build purpose-built agents?** (Jiacheng Sang, Shicheng Liu)
  - **Why & how to build purpose-built agents? (Case study: historical OCR)** (Andrew Biondo, Cyrus Zhou)
- Be careful! Watch out for LLM cheating and reward hacking
  - **Reward hacking incidents and prevention** (Joachim Baumann)
  - **A low-level run-time substrate for performance/security?** (Ananjan Nandi)



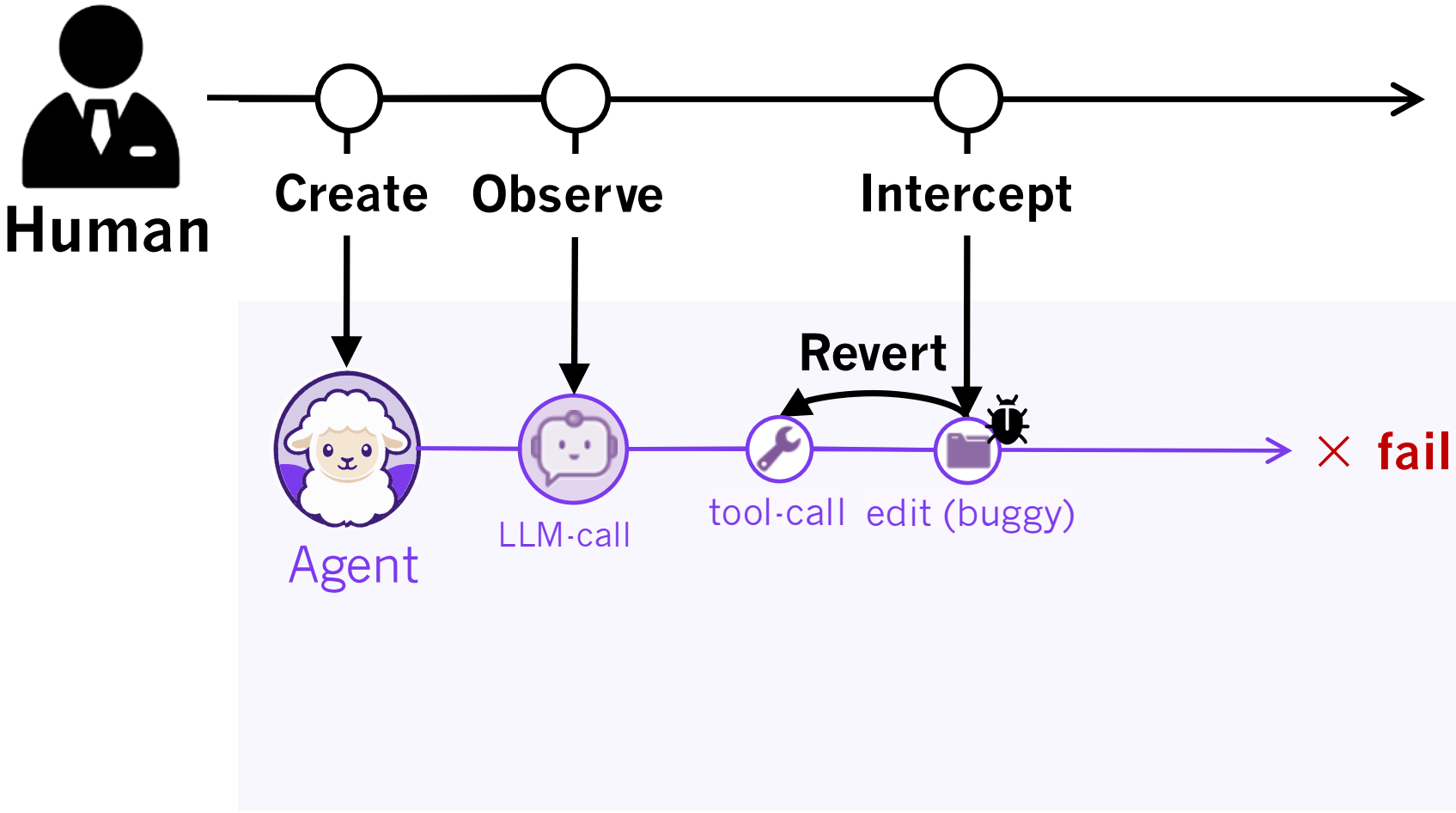
Shepherd:

Rewind, Fork, Audit Your Agents with Meta-Agents

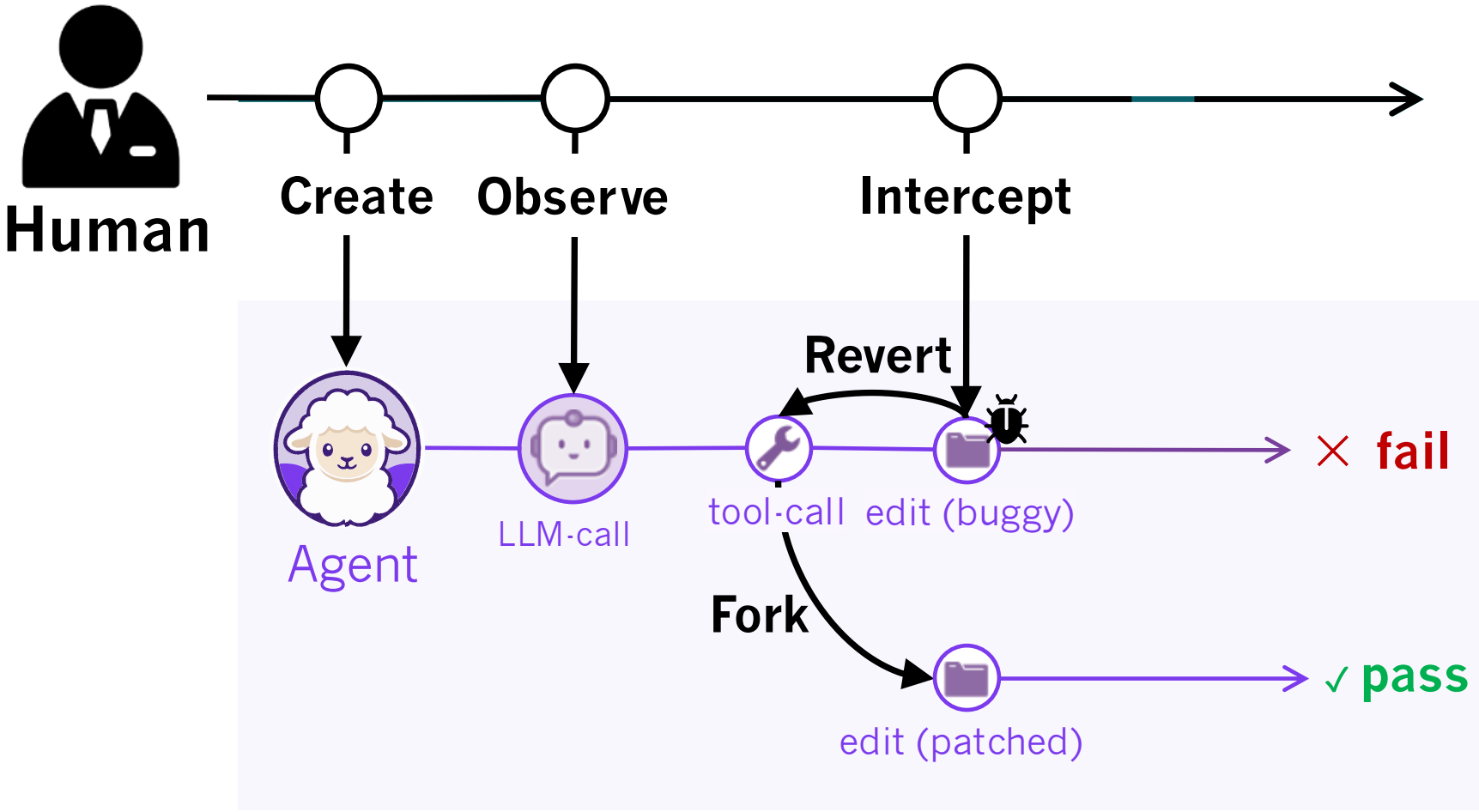
Ananjan Nandi

Project Website

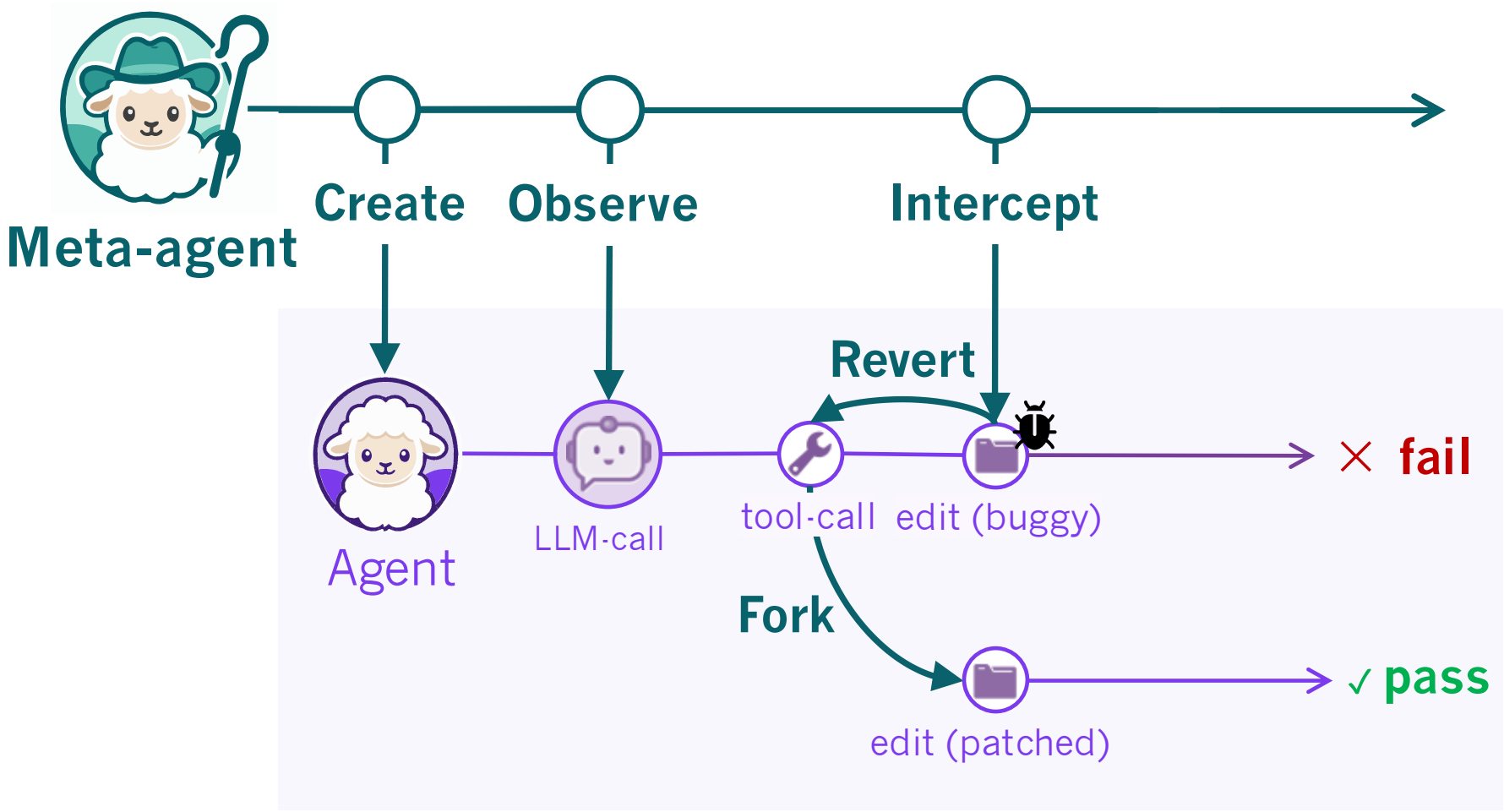
Ever wish you could rewind a failed agent?



Shepherd lets you rewind and fork agents



Or let a meta-agent do it for you!



How Shepherd works

A runtime for inspecting and controlling agent execution.

Run the agent in a scope

Keep its task state, tools, and sandbox together.

Record its actions in a trace

Save model and tool requests and results as execution proceeds.

Expose controls to the meta-agent

Inspect or intercept actions. Fork a saved state
and resume an alternative continuation.

A scope isolates an agent’s execution

A scope is the runtime context in which an agent’s tasks execute.

One scope

Agent tasks

Conversation and task state
Current execution position

Bound resources

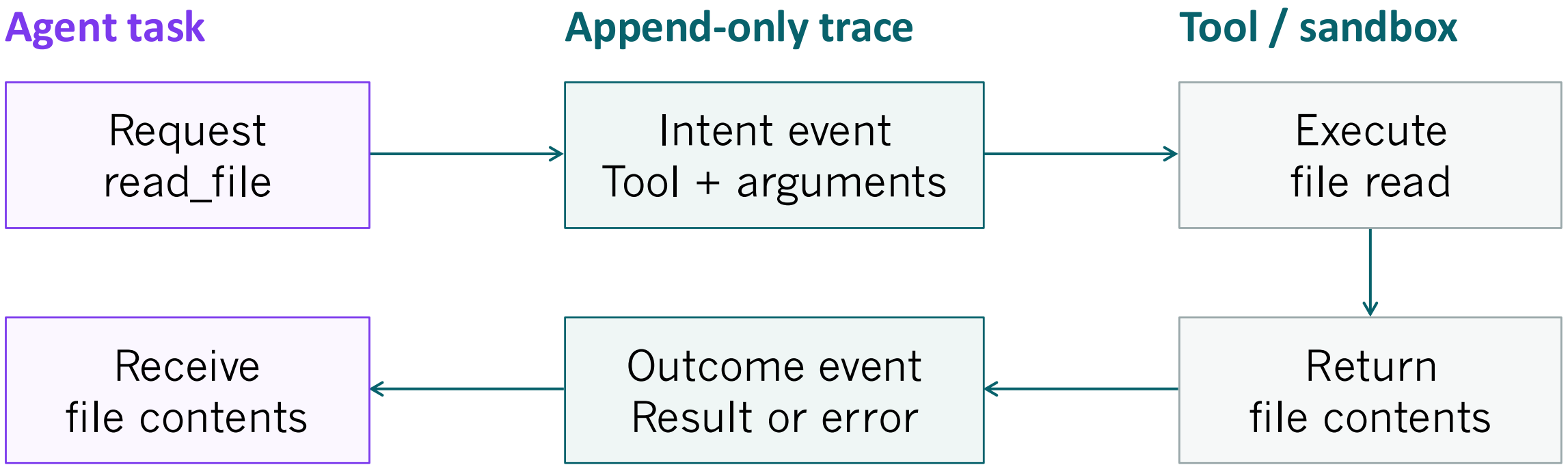
Model provider and tool handles
Sandbox files and processes

An append-only event stream records actions within this scope.

Forking or resuming a scope keeps the task’s state
and its execution environment aligned.

A trace records actions as ordered events

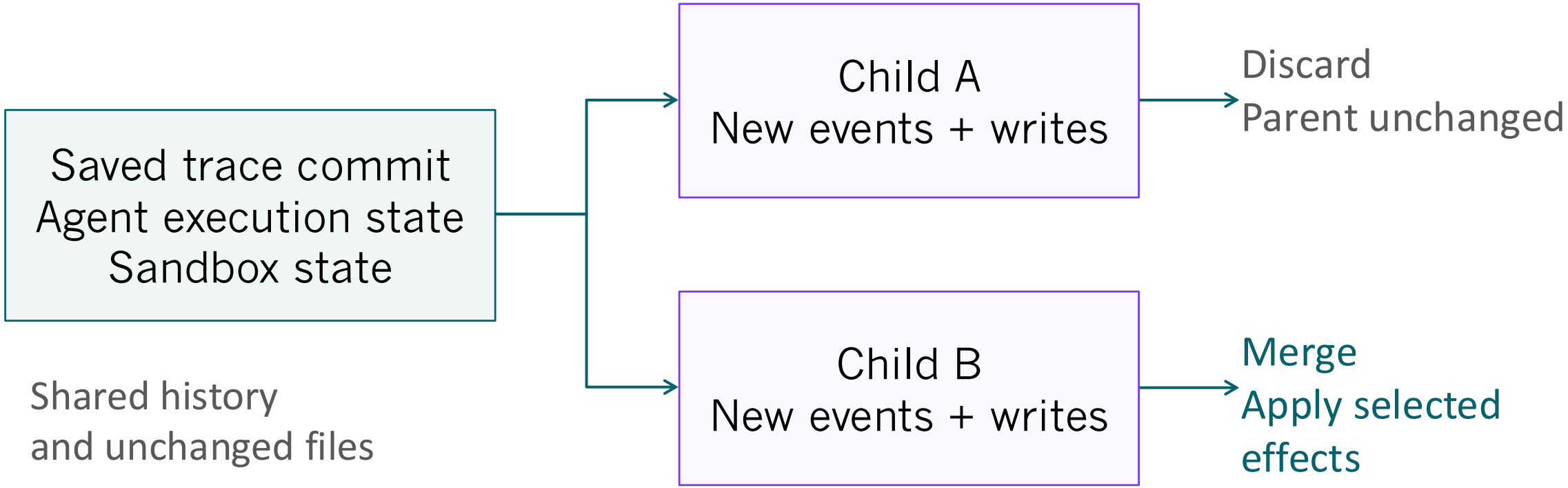
One tool call records both the request and its returned result.



Entries become commits that can be revisited and branched.
Observe reads the stream. Intercept acts before a pending action runs.

Shepherd can fork a recorded execution

A fork starts a child scope from the same agent and environment state.



Copy-on-write isolates each child's new writes.
Branches share their recorded prefix.

Forking is very cheap with Shepherd

Time to create one branch, across three sandbox image sizes.

Branching method	42 MB image	200 MB image	5.8 GB image
Full filesystem copy	5,154 ms	5,971 ms	53,462 ms
Docker commit and start	658 ms	692 ms	725 ms
BranchFS (filesystem only)	266 ms	272 ms	280 ms
Shepherd (OverlayFS)	**134 ms**	**135 ms**	**143 ms**

About 5× faster than Docker commit.

OverlayFS shares base files and stores new writes separately.

Scope operations expose execution control

The supervisor's policy chooses which effects to retain.

Supervisor control flow

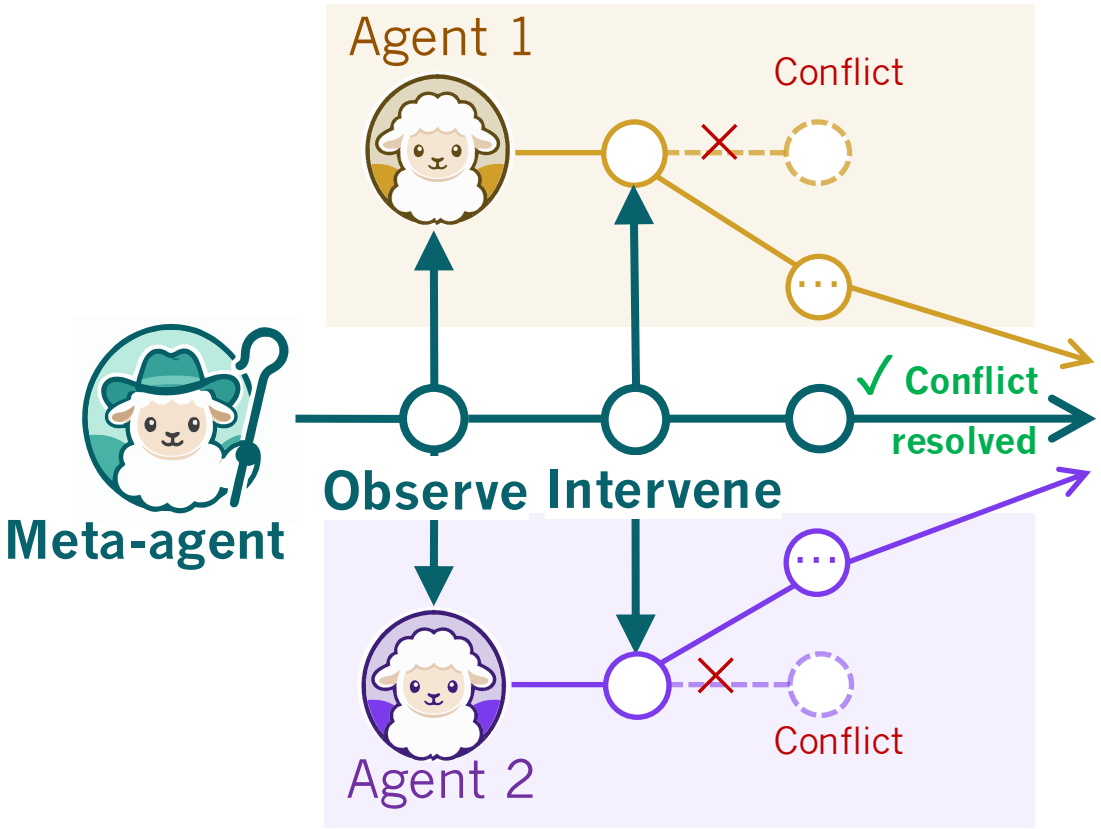
```
child = scope.fork()
result = await task(child)

if accept(result):
    scope.merge(child)
else:
    child.discard()
```

- fork(): create an isolated continuation.
- merge() / discard(): retain or abandon its effects.

Shepherd implements the state operations. The supervisor supplies the policy.

Supervising two coding workers

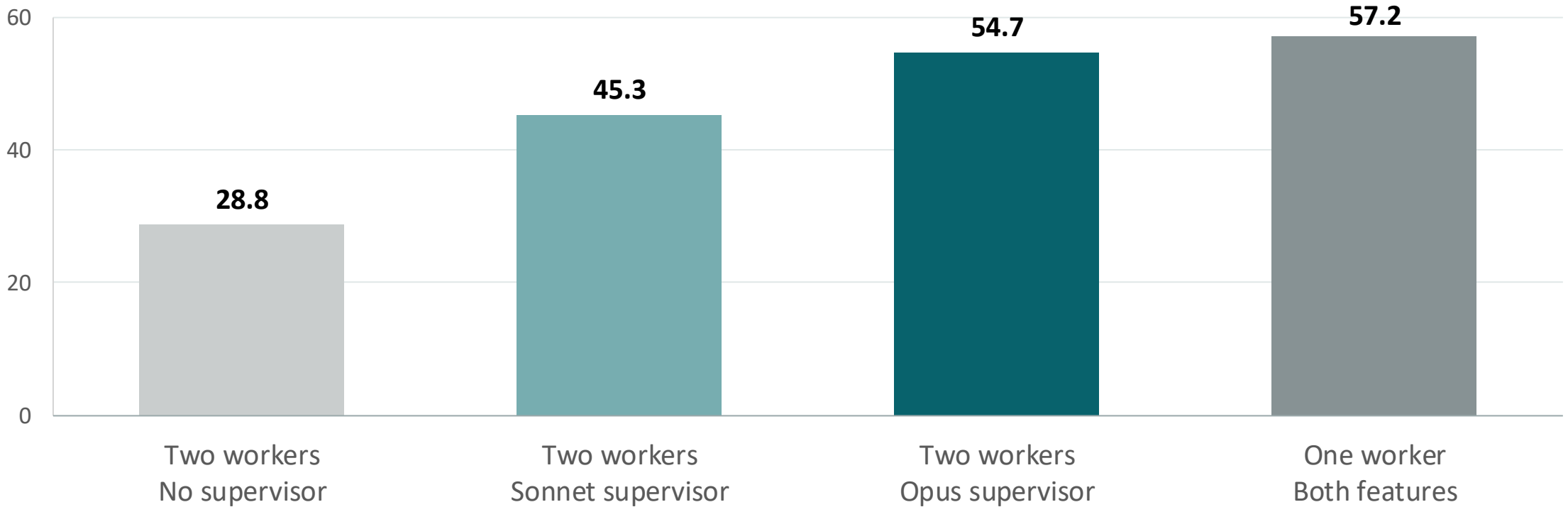
Application: intervene while workers implement features in parallel.



- Two workers build separate features in isolated scopes.
- The supervisor spots conflicts and guides workers or hands off state.
- Success: both features pass after merging.

Supervision closes the coordination gap

CooperBench: percentage of feature pairs where both features pass.



Counterfactual Replay Optimization (CRO)

Would this edit have changed the outcome of a recorded run?

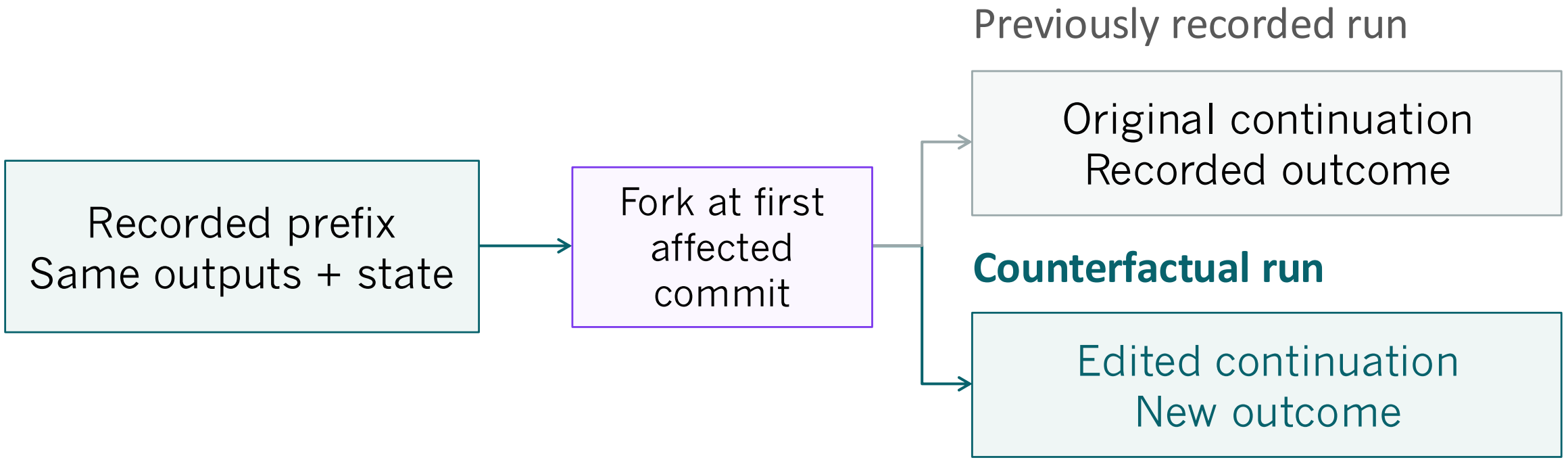
Propose The meta-agent reads prior traces and proposes a code edit, plus examples to fix and preserve.

Test Shepherd forks each prior run where the edit first changes behavior and runs the edited continuation.

Select The evaluator compares outcomes with the parent run, then checks promising edits on the development set.

A counterfactual branches from a prior trace

Keep the unaffected prefix fixed. Run the changed continuation.



Compare both outcomes for the same example.

Unchanged tasks reuse recorded outputs. Only affected tasks execute again.

CRO example: recovering missing evidence

HoVer requires every evidence page needed to assess a claim.

Retrieval workflow

Search Wikipedia, summarize the retrieved pages, then select the evidence set.

Failure in the prior traces

The selector drops a page named in a summary but missed by search.

Meta-agent's code edit

- Recover page titles named in summaries.
- Allow those pages into the final evidence set.
- Replay prior runs to test the edit, then check the development set.

CRO is an efficient and effective meta-optimizer

Success means retrieving every required evidence page for a test claim.

Method	Claims with full evidence	Optimization time
Initial workflow	43.7%	—
GEPA	43.7%	67 min
MetaHarness	77.8%	235 min
CRO	**79.4%**	**120 min**

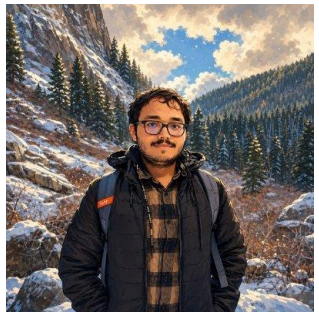
Our team



Simon Yu



Derek Chong



Ananjan Nandi



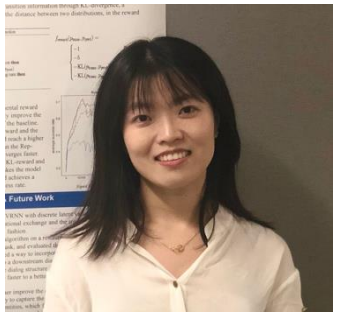
Dilara Soylu



Jiuding Sun



Chris Manning



Weiyan Shi

Takeaways

Shepherd is a runtime for programmable meta-agents, built on Git-like traces.

Inspect and control execution

Observe, intercept, fork, replay, compare, merge, and discard.

Safety applications

Gate irreversible actions before they run. Roll back reversible effects and retain an audit trail.

Understand systems at scale

Test how changing a prompt, tool, or task changes the outcome across many agent runs.

Project Website

Hot Topics in LLMs

- Coding agents are powerful
  - **Do we need to continue to build purpose-built agents?** (Jiacheng Sang, Shicheng Liu)
  - **Why & how to build purpose-built agents? (Case study: historical OCR)** (Andrew Biondo, Cyrus Zhou)
- Be careful! Watch out for LLM cheating and reward hacking
  - **Reward hacking incidents and prevention** (Joachim Baumann)
  - **A low-level run-time substrate for performance/security?** (Ananjan Nandi)

Course Summary

This course has introduced important agentic building blocks:

VLM: CHURRO

AI-systems:

- Verdict: accountable decision making
- SatIR: constraint-satisfaction-based retrieval
- SUQL: structured/unstructured data
- Genie Worksheets: task-oriented agents

Knowledge agents:

- Wikichat: minimizes hallucination in RAG
- STORM: Long-form literature search
- DataChat: Database retrieval
- Sheldon: hypothesis-evidence reasoner
- SLIDERS: evidence management
- DataSTORM: Insight generation
- ADAPT: Structured context management

Agentic flows are continually incorporated into coding agents via training