CSCI 1377
Tools for Thought

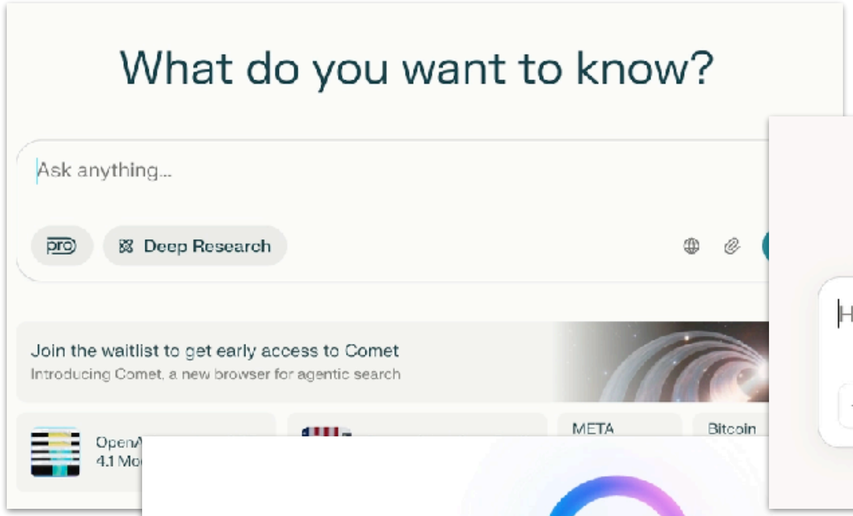
Artificial Intelligence II
Weird Futures

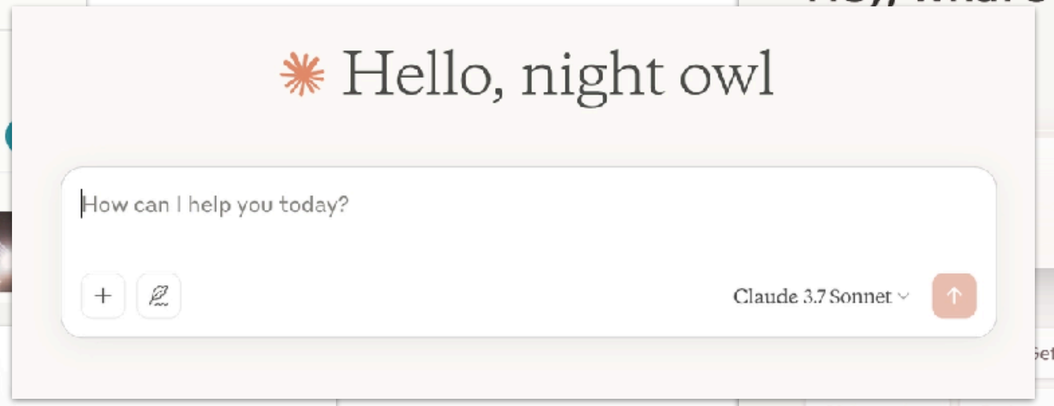
"The forced matings of minds and electrons succeed and fail with equal spectacle. Our hybrids become as brilliant as savants, and as autistic. [...] Computers bootstrap their own offspring, grow so wise and incomprehensible that their communiqués assume the hallmarks of dementia: unfocused and irrelevant to the barely-intelligent creatures left behind."

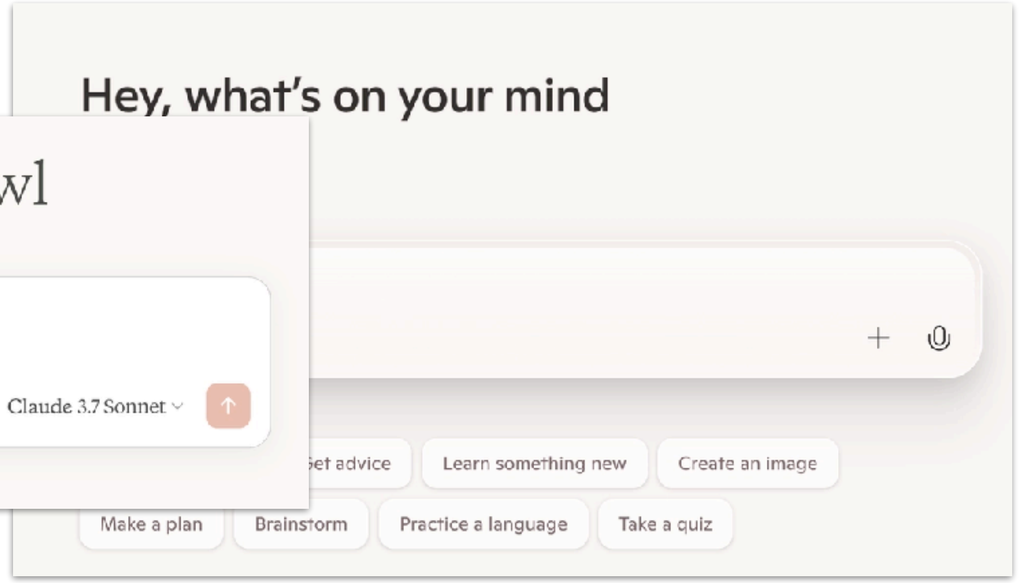
— Peter Watts, *Blindsight* (2006)

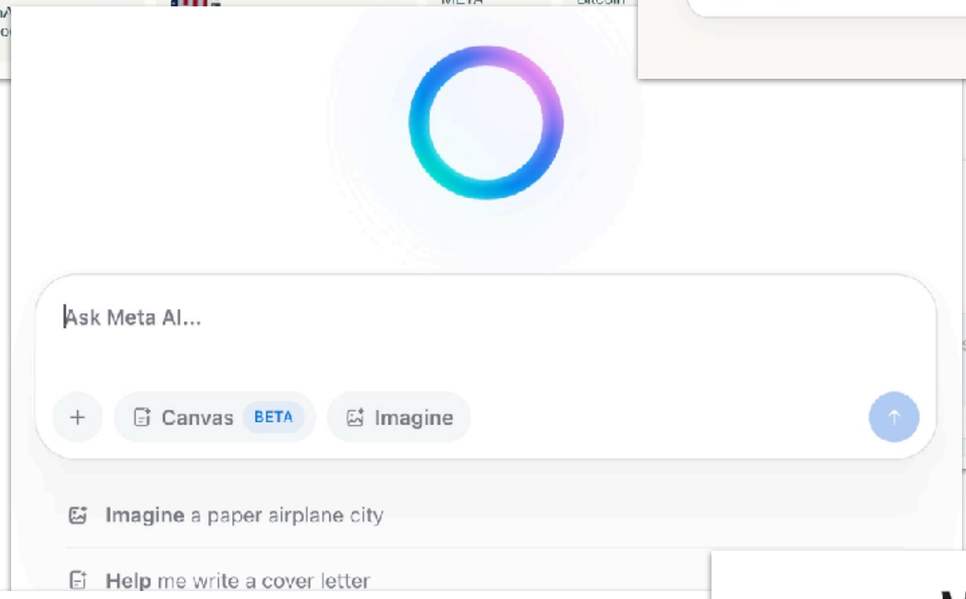
It's 1439. Johannes Gutenberg just invented the printing press. What happens next?

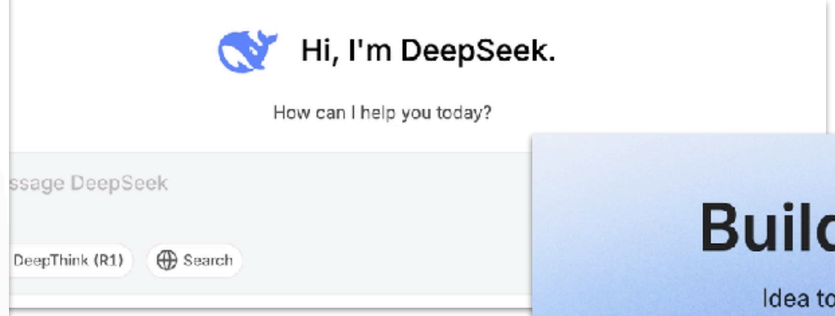
- 1500 (61y): printing presses have produced ~10 million books
- 1517 (78y): printing spreads heterodoxy, facilitating the Protestant Reformation
- 1577 (138y): printing allows the sharing of astronomical observations of a comet
- 1605 (166y): printing spreads news from the world's first newspaper
- 1618 (179y): printing spreads propaganda, facilitating the Thirty Years War
- 1820 (381y): global literacy rate estimated at 12% of world's population
- 1833 (394y): the "penny press" brings ad-subsidized news to the lower classes
- 1983 (544y): global literacy rate estimated at 70% of world's population

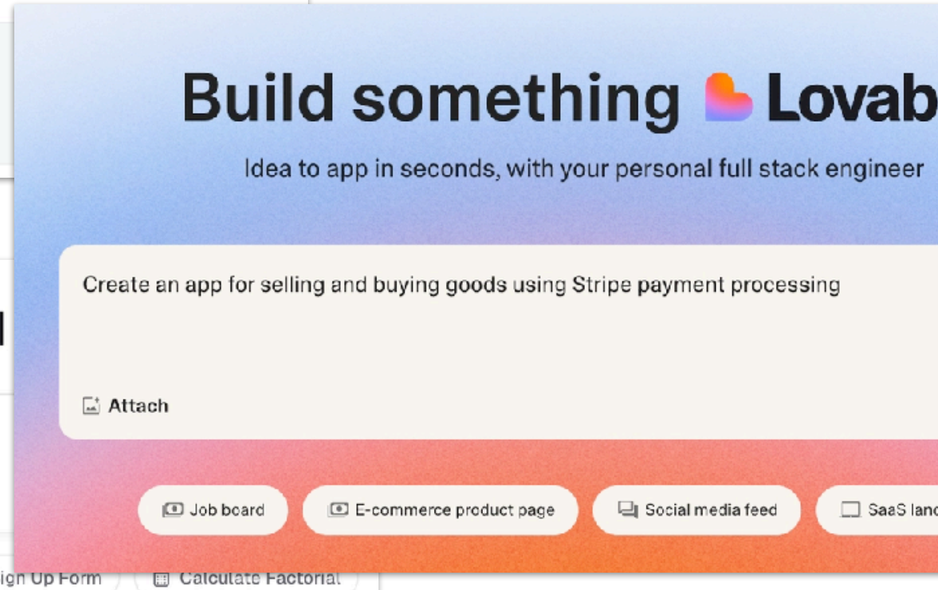


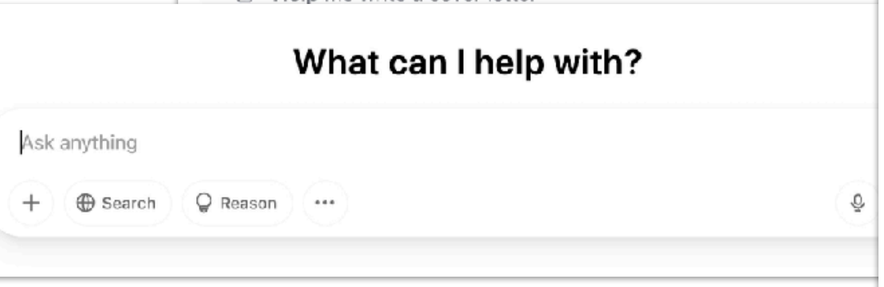


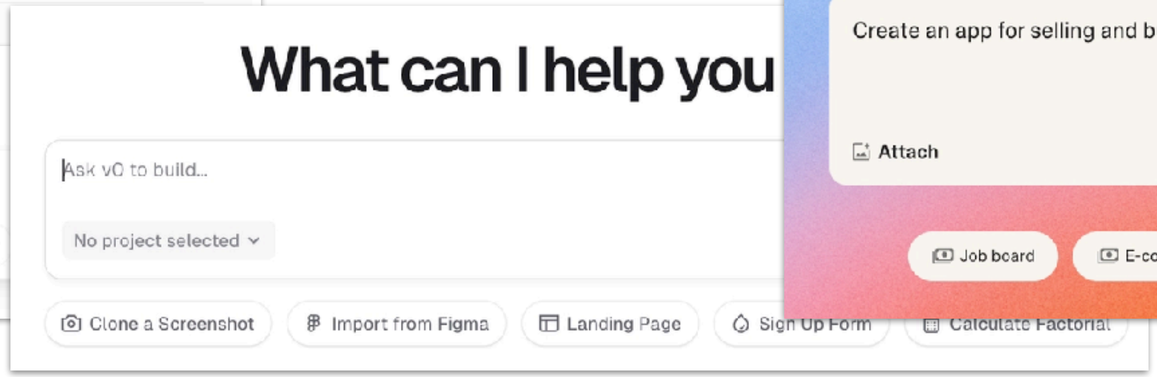












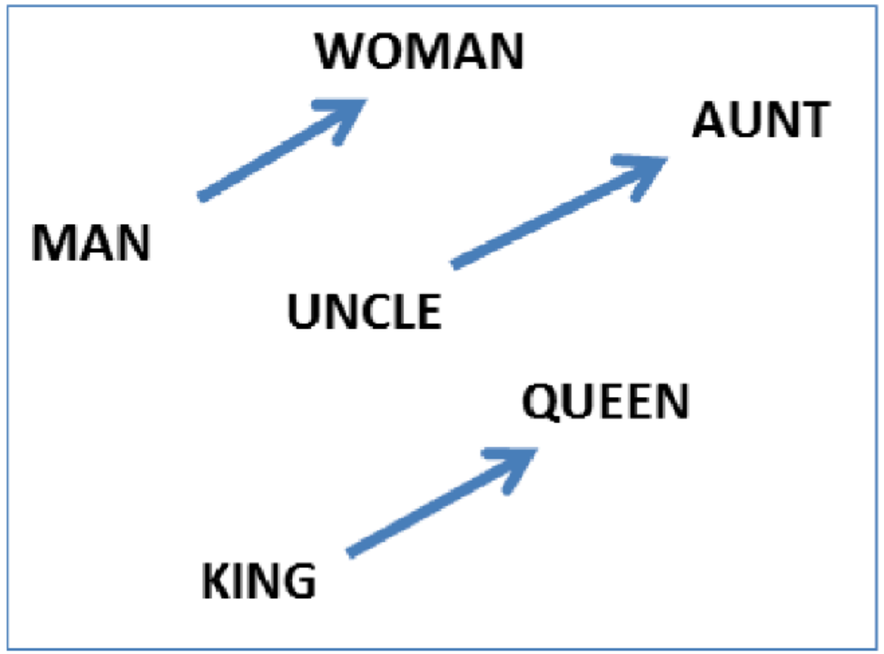
Screenshots courtesy of Jerome

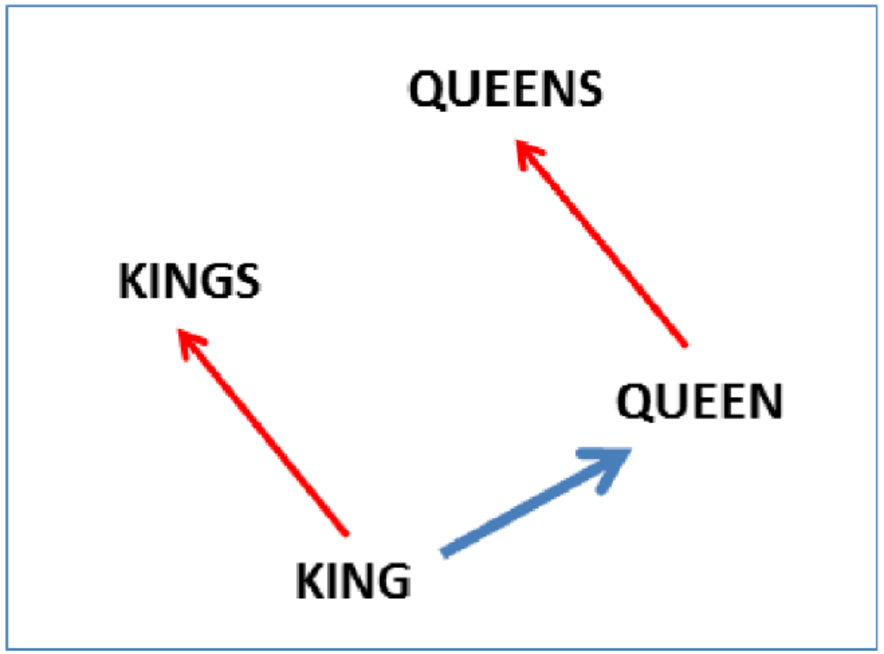
Learning from superintelligence

Vectors between embeddings can represent concepts

$$\text{forward}_\ell(\text{"King"}) = \overbrace{[0.8, 1.2, \ldots, -9.3]}^{|\ell| \text{ numbers}}$$

$$\text{forward}_\ell(\text{"King"}) - \text{forward}_\ell(\text{"Man"}) \approx \text{royalty}$$



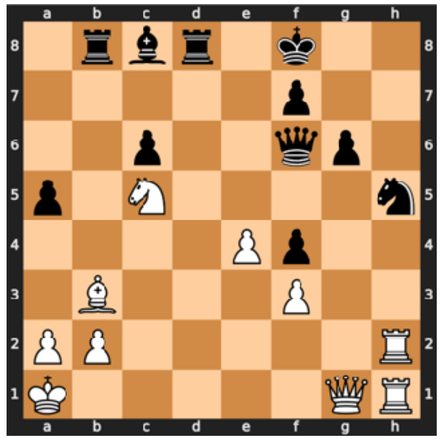


Mikolov et al. "Linguistic Regularities in Continuous Space Word Representations" 2013

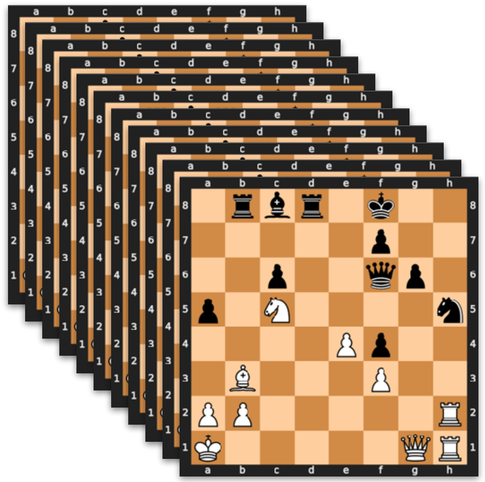
AlphaZero (AZ) is a chess-playing neural net trained by DeepMind

AZ can embed chess positions

$$\text{forward}_\ell(\quad) = \mathbf{z}_\ell$$

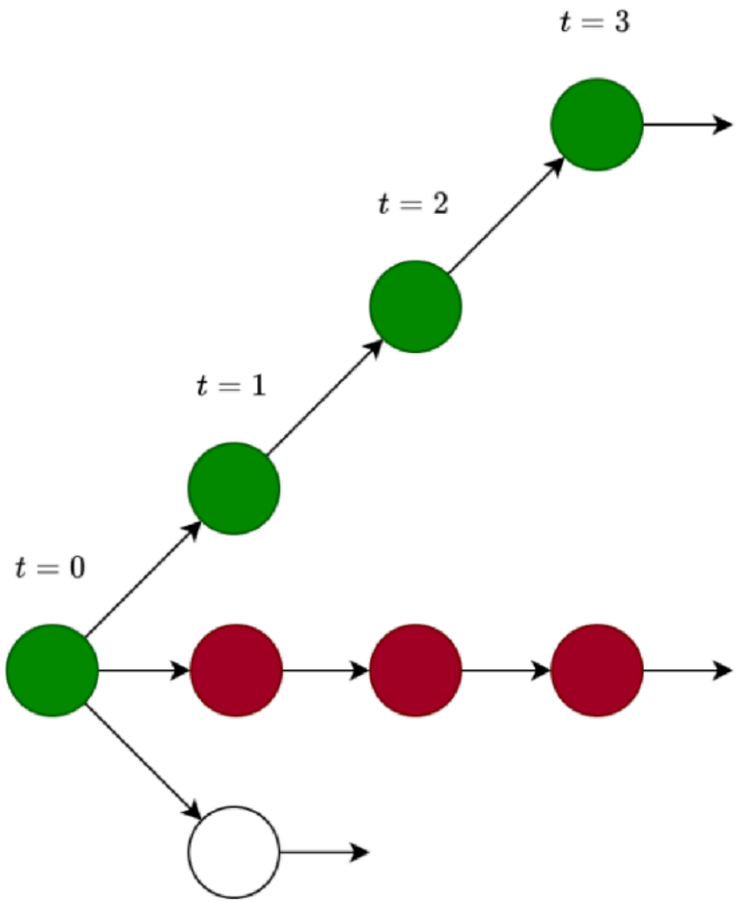


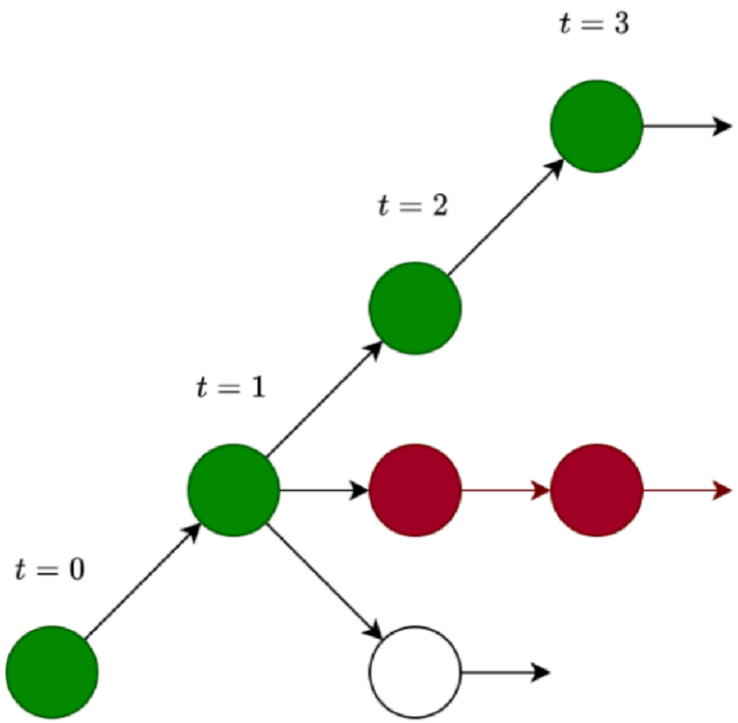
AZ has a database of self-play

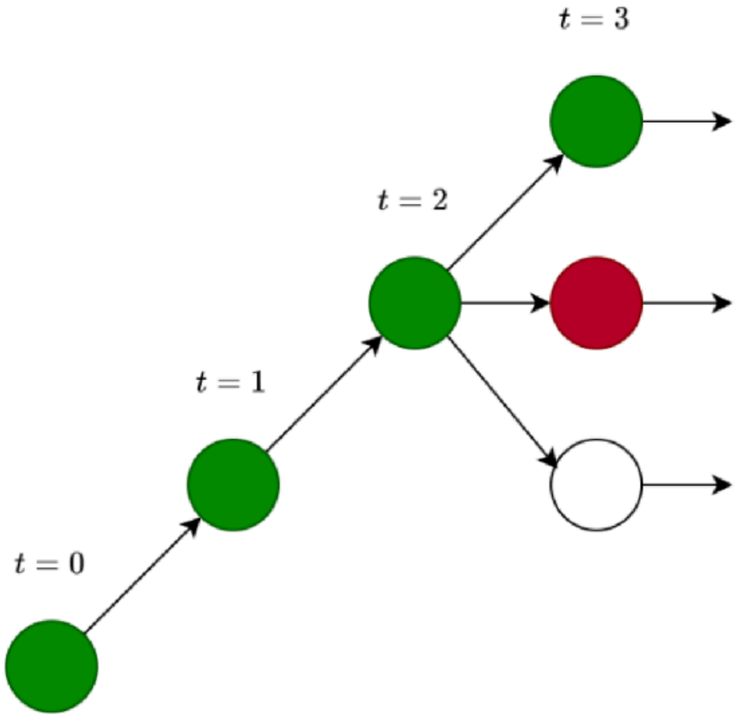


Schut et al. “Bridging the human–AI knowledge gap through concept discovery and transfer in AlphaZero” 2025

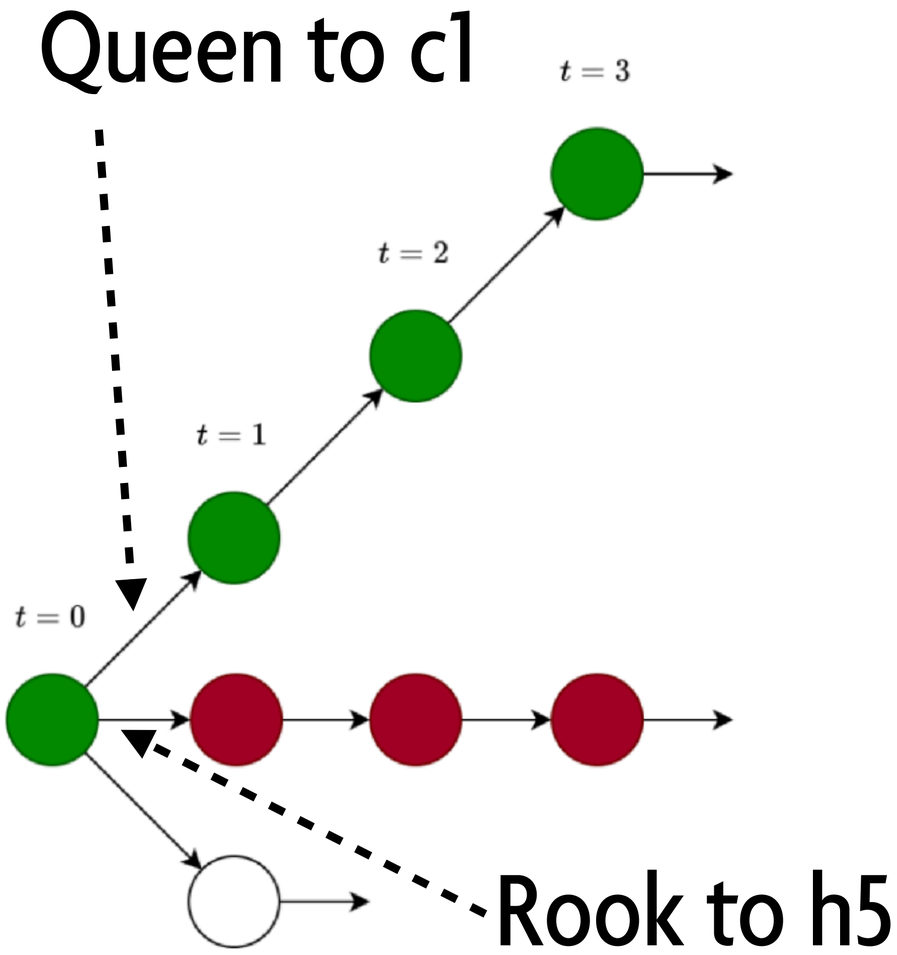
AZ evaluates a sequence of board positions to decide the optimal move

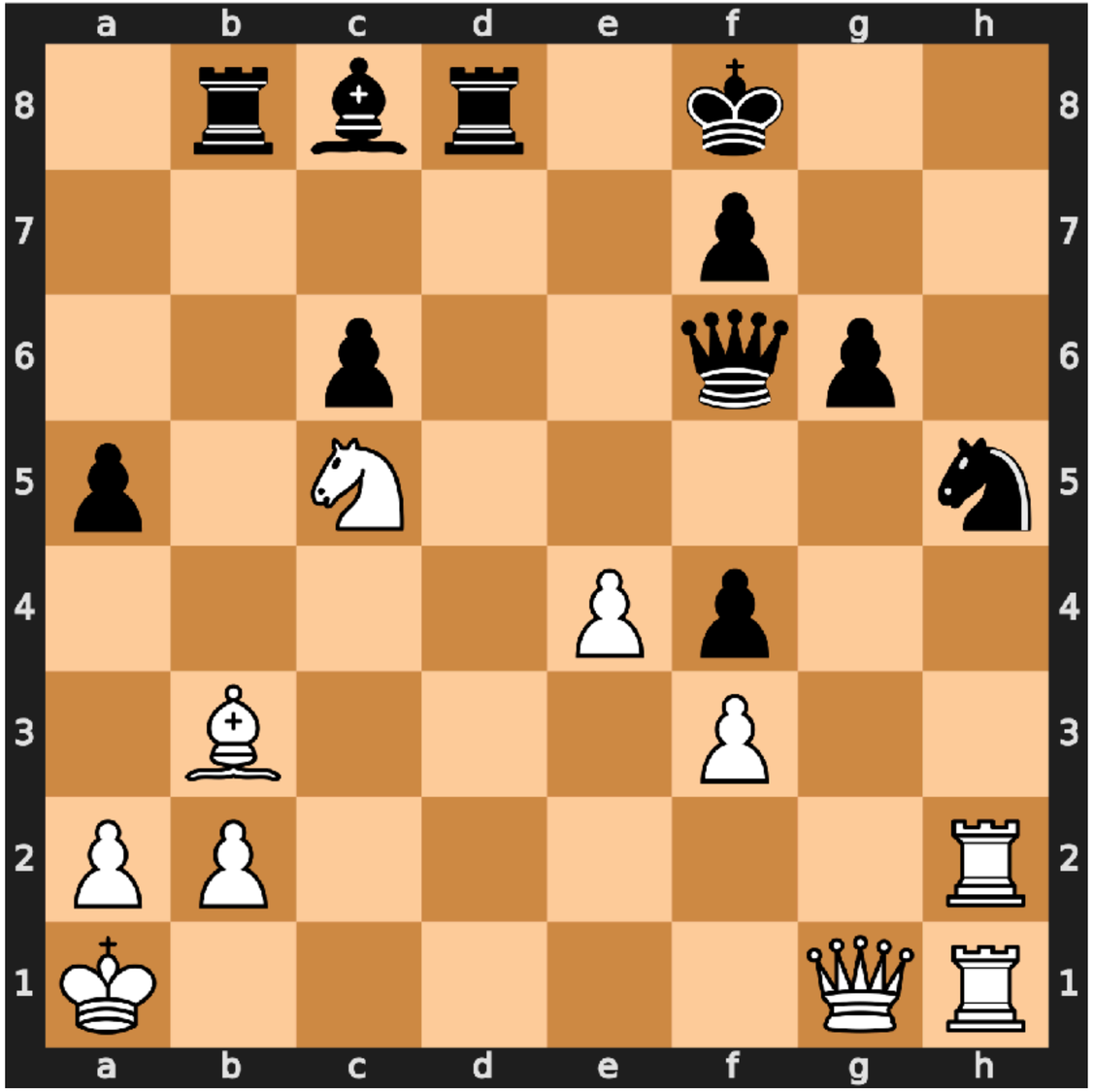






Concepts should explain why AZ makes one move and not the other





Concept discovery process:

1. Discover which vectors are meaningful to decisions
2. Determine whether the concepts can be taught
3. Estimate whether the concept is shared or AI-specific
4. Show boards w/ teachable & novel concepts to humans

Concepts are vectors differentially active in preferred board positions

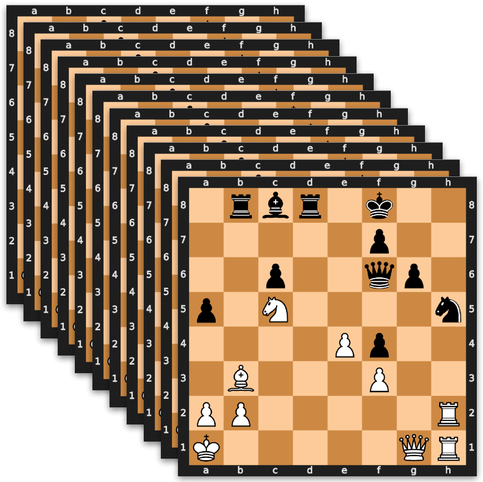
Queen to c1 = $\mathbf{z}^+$

Rook to h5 = $\mathbf{z}^-$

Goal: find the smallest vector $\mathbf{v}$ such that

$$\mathbf{v} \cdot \mathbf{z}^+ \geq \mathbf{v} \cdot \mathbf{z}^-$$

(and for all steps to a max depth T)



For every board position in the database, compute its $\mathbf{z}^+$ and $\mathbf{z}^-$, and find the corresponding concept $\mathbf{v}$

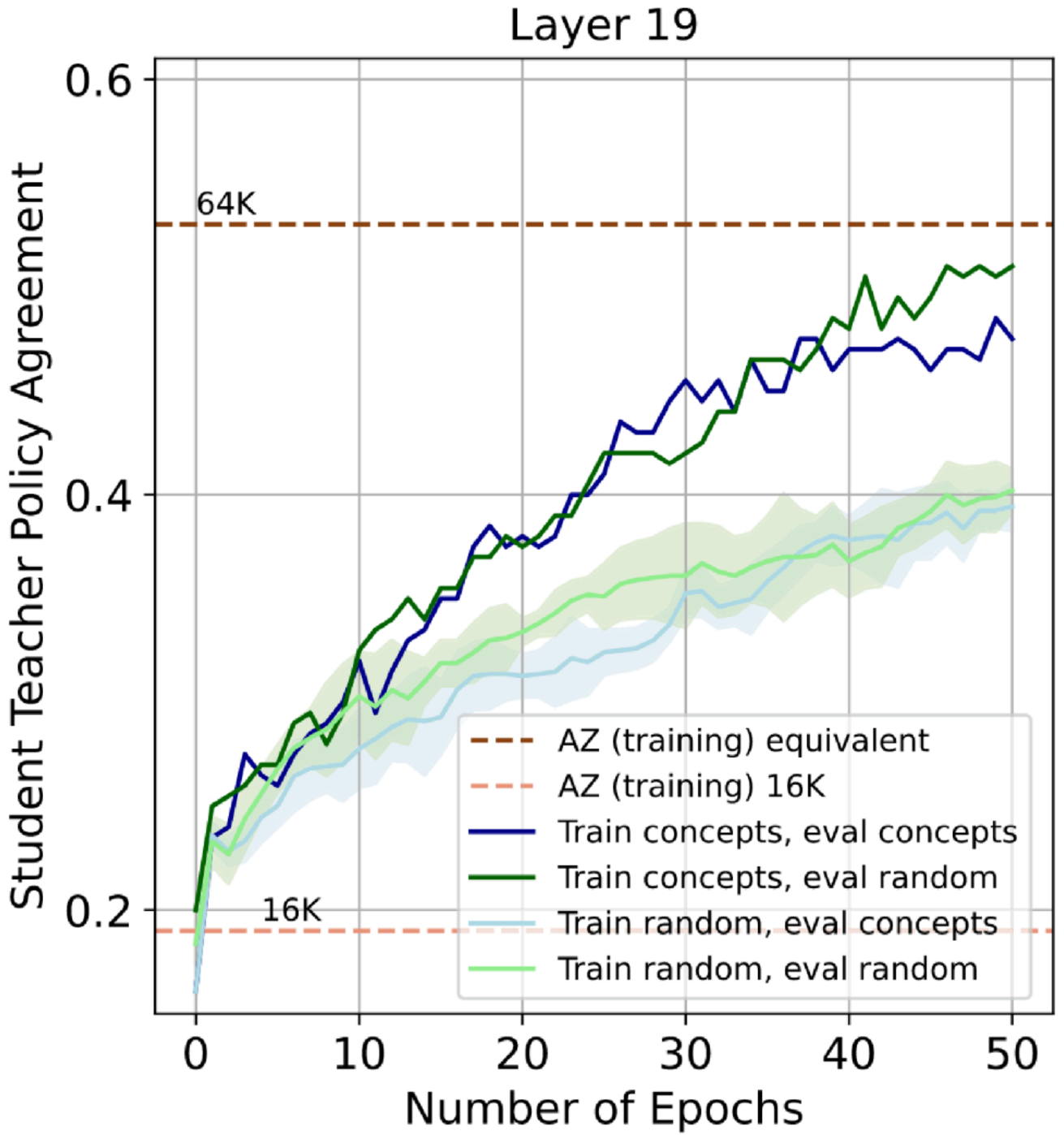
Evaluate teachability by training an early checkpoint of AZ on the concept

1. Find every board position in the database satisfying $\mathbf{v} \cdot \mathbf{z}^{+} \geq \mathbf{v} \cdot \mathbf{z}^{-}$

2. Find a checkpoint of AZ such that it disagrees w/ the final AZ for at least 80% of the board positions

3. Train the checkpoint for 50 epochs on the board positions against the final AZ's answers

The model learns better training on concept-related positions than random positions



Evaluate novelty by finding concepts that are not well reconstructed by the human concept basis

1. Build a matrix of embeddings for every position played by AI or humans

$$\mathbf{Z}_{\text{AI}} = \begin{bmatrix} \cdots & \mathbf{z}_{\text{AI}}^{1} & \cdots \\ & \vdots & \\ \cdots & \mathbf{z}_{\text{AI}}^{n} & \cdots \end{bmatrix} \qquad \mathbf{Z}_{\text{HUM}} = \begin{bmatrix} \cdots & \mathbf{z}_{\text{HUM}}^{1} & \cdots \\ & \vdots & \\ \cdots & \mathbf{z}_{\text{HUM}}^{m} & \cdots \end{bmatrix}$$

2. By SVD, compute the orthonormal row basis $\mathbf{U}_{\text{AI}}$ and $\mathbf{U}_{\text{HUM}}$

3. $\text{novelty}(\mathbf{v}) = \text{reconstruction-error}(\mathbf{v}, \mathbf{U}_{\text{HUM}}) - \text{reconstruction-error}(\mathbf{v}, \mathbf{U}_{\text{AI}})$

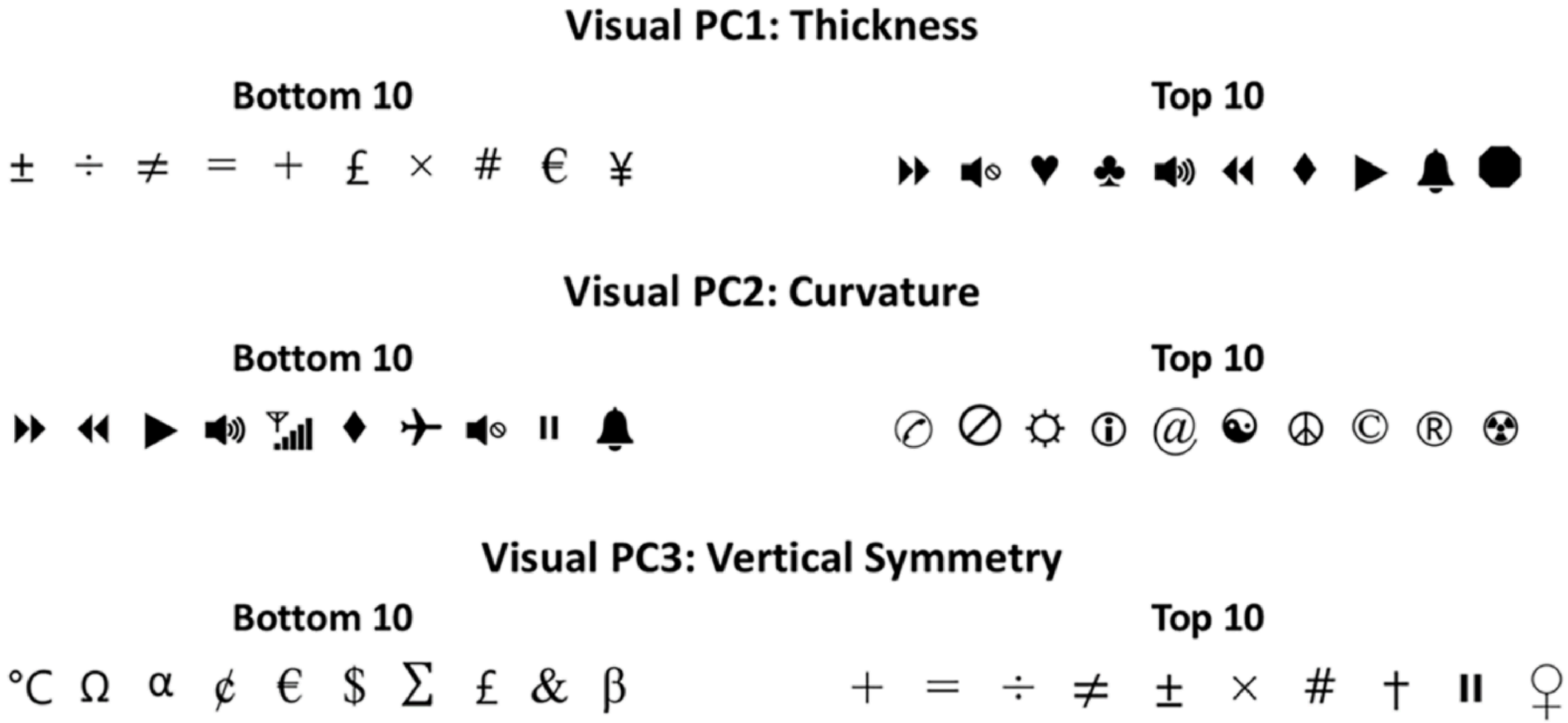
World-class chess players improved play on concept-related boards through training

Study: 4 grandmasters shown concept-related positions, evaluated before and after for agreement w/ AZ on concept-related puzzles

Grandmaster	Puzzles solved Phase 1	Puzzles solved Phase 3	↑
1	0/12	5/12	+42%
2	4/12	7/12	+25%
3	3/12	5/12	+16%
4	6/16	7/15	+6%

Hacking our brains

Thickness, curvature, vertical symmetry reduce the memorability of a symbol



Roberts et al. “Reverse engineering what makes a symbol memorable” (2026)

DALL-E 3 can generate more or less memorable symbols

Word	Memorable	Forgettable	Word	Memorable	Forgettable	Word	Memorable	Forgettable
impression			opinion			revenge		
intelligence			opportunity			risk		
journey			patience			sin		
kind			personality			society		
knowledge			pity			soul		
lack			pleasure			success		
language			possibility			talent		

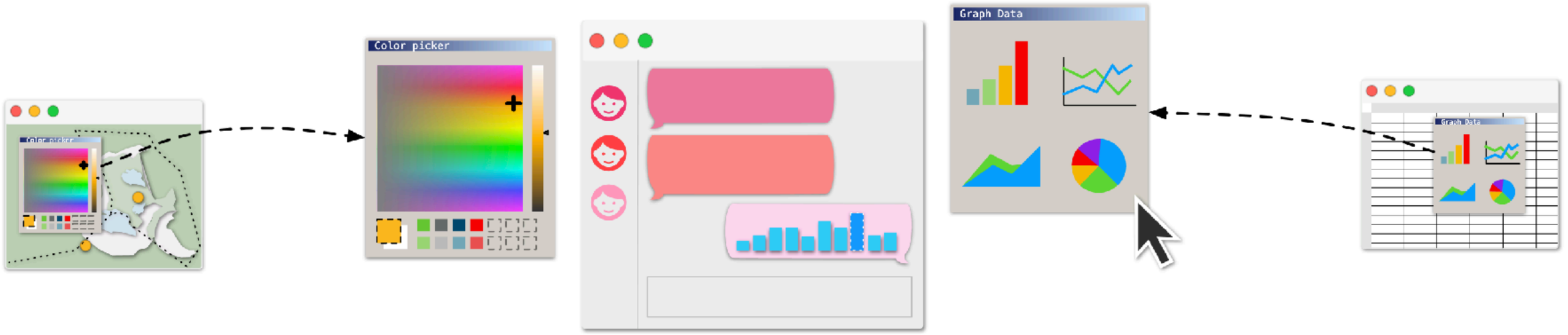
Generative interfaces, malleable systems

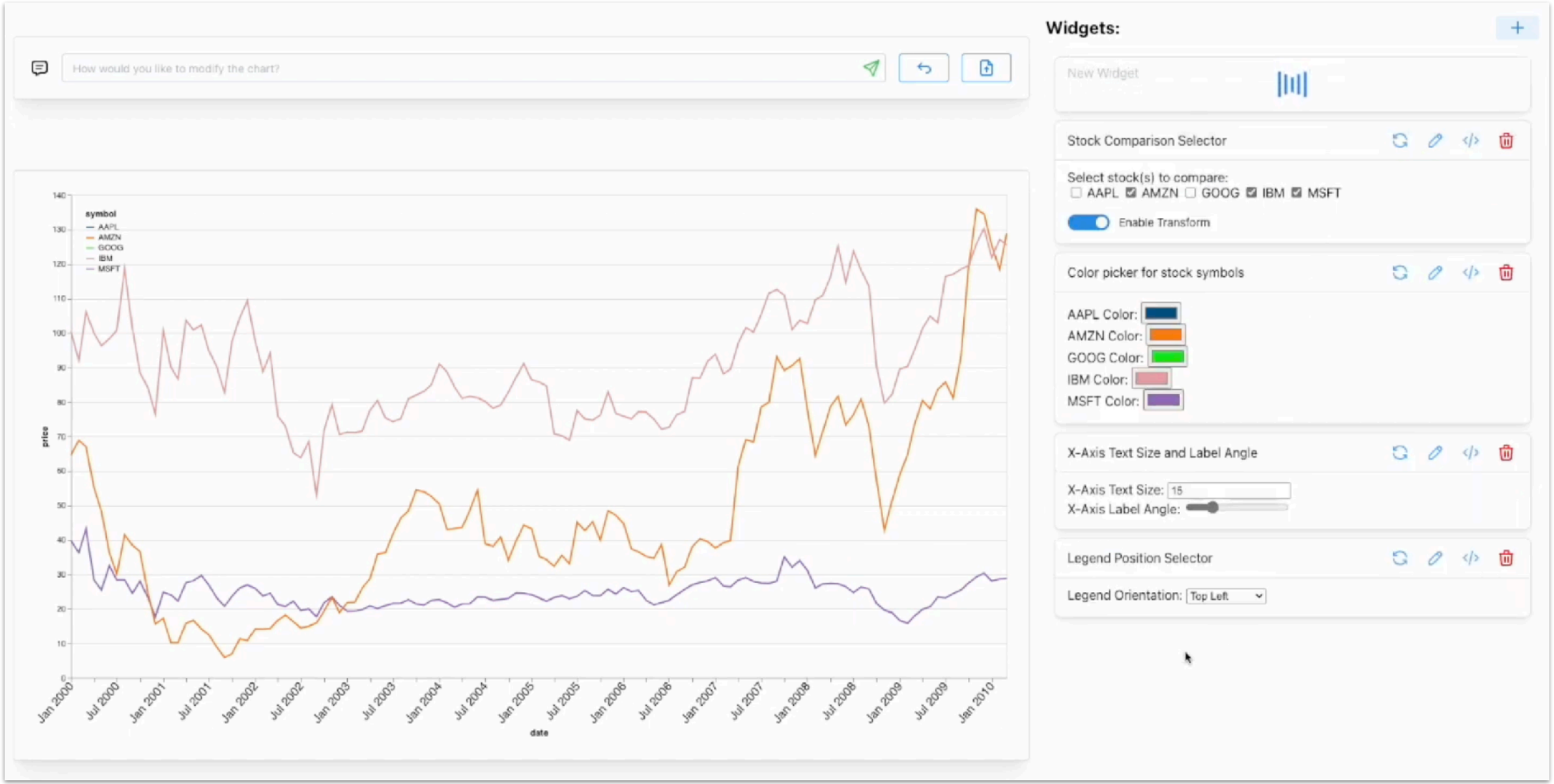
We live in a world of apps

- Spreadsheets are an app, document editing is an app, social media is an app, etc. etc.
- Apps only talk to each other through pre-baked integrations
- Every sufficiently complex desktop app implements keybindings, tabs, window management, folders, tags, ...
  - And each implementation is slightly different from the others!
- If I want a spreadsheet and a document editor together, that's a whole new app!

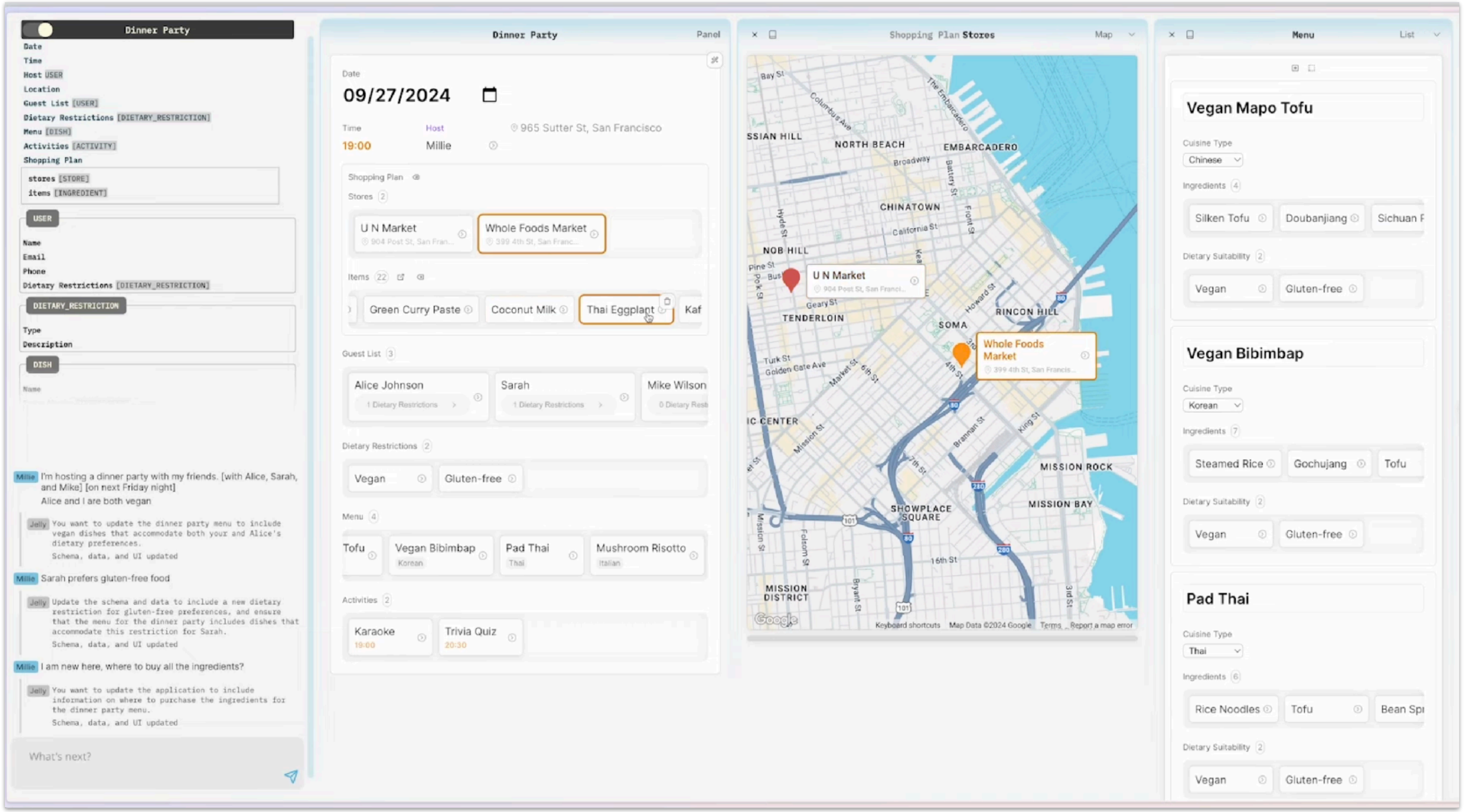
Malleable software aims to increase the power of existing adaptation behaviors by allowing users to pull apart and re-combine their interfaces at the granularity of individual UI elements, such as toolbars, widgets, menus, documents, and devices. In other words, the goal is to erase the boundaries between apps and create an end-user accessible "physics of interfaces" that dictate how different interfaces and documents can be assembled.

— Philip Tchernavskij, *Designing and Programming Malleable Software* (2019)

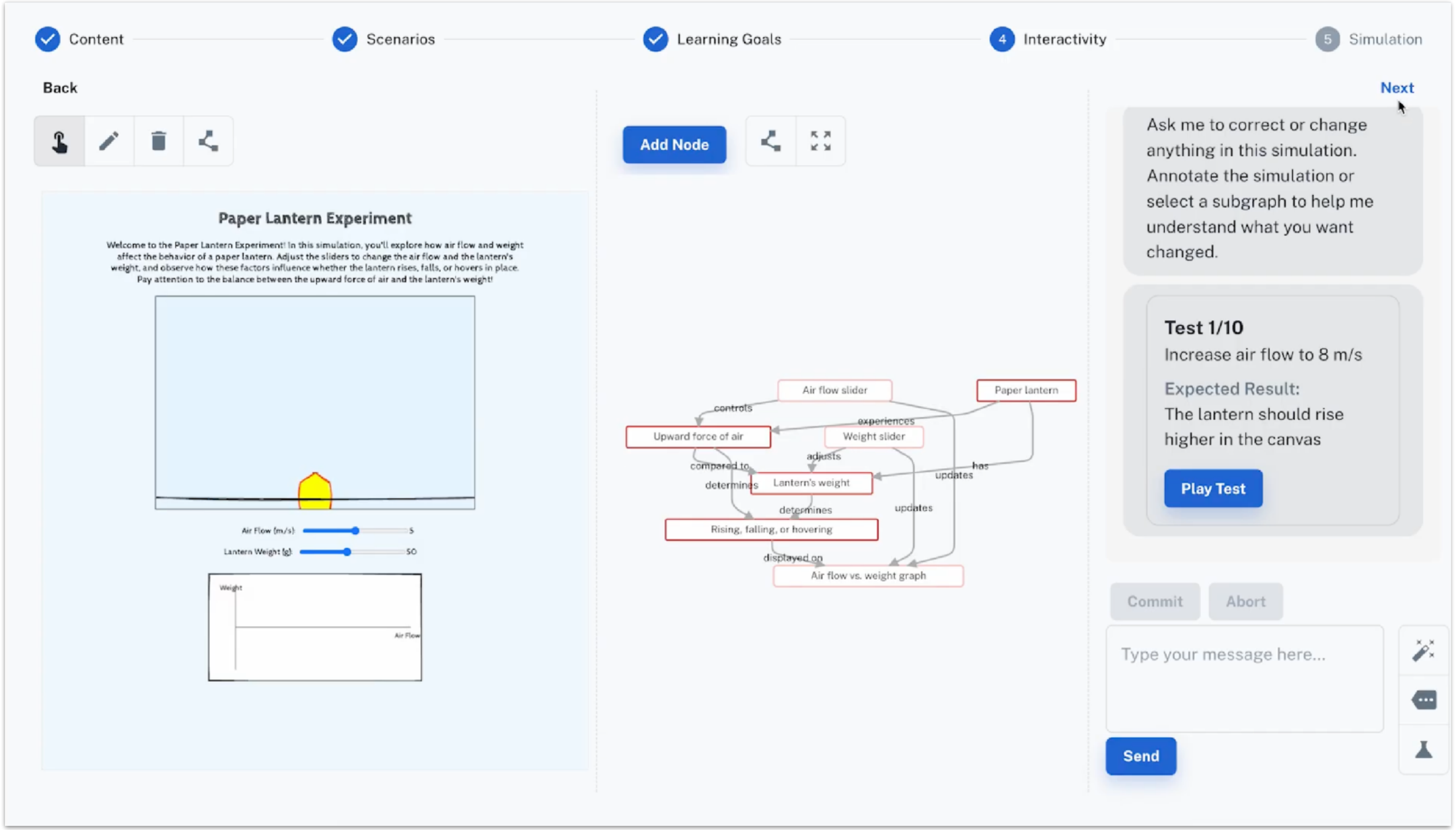




youtu.be/8Z4yKqm0AAg



youtu.be/X3cf1U1dFGQ



youtu.be/7IRl9Jgbu1s

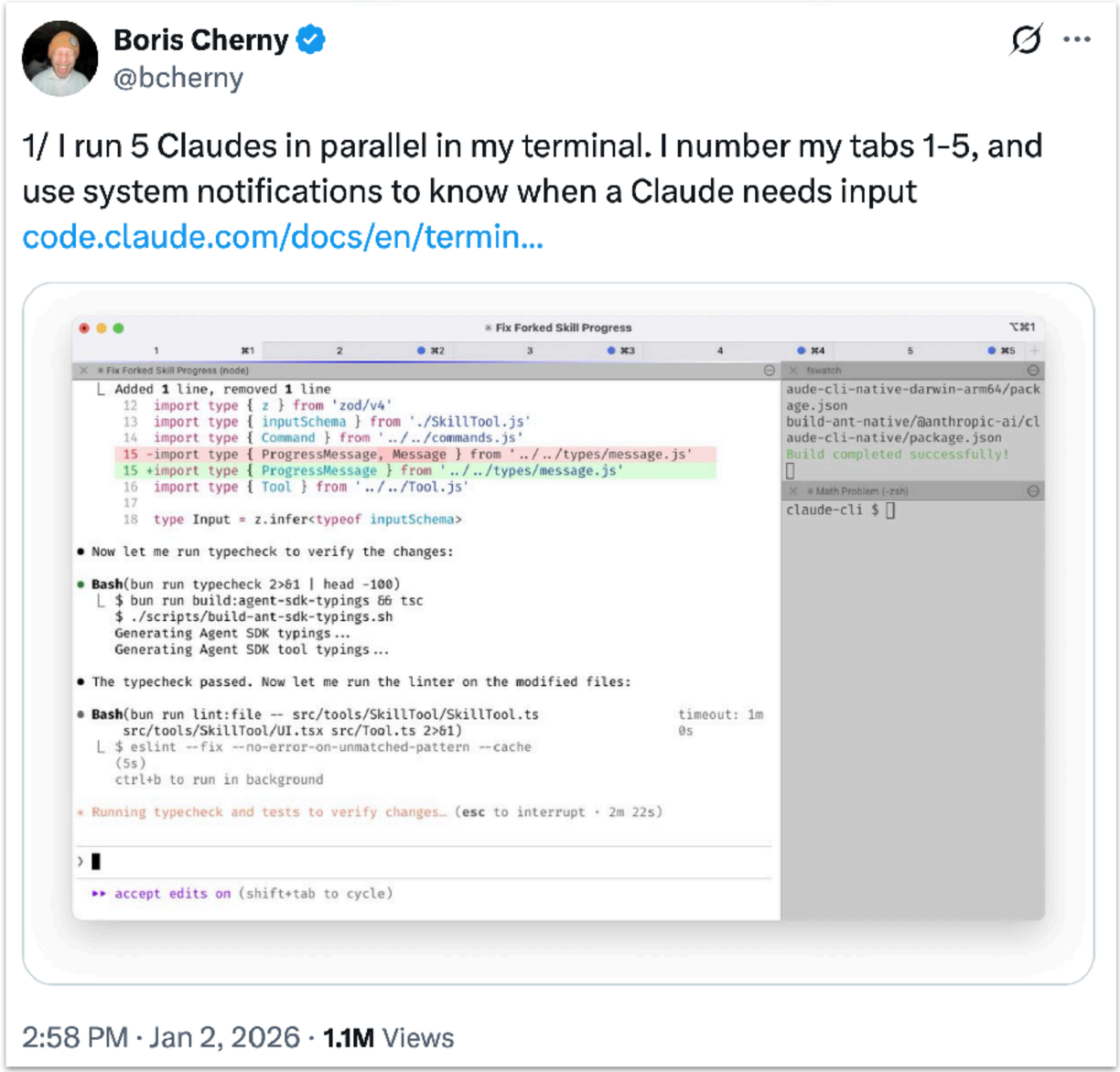
Brief discussion

- How would you want to grow or bend your day-to-day tools?
- Where do you think these demos will fall apart?
- What software architecture do we need to sustain malleability?

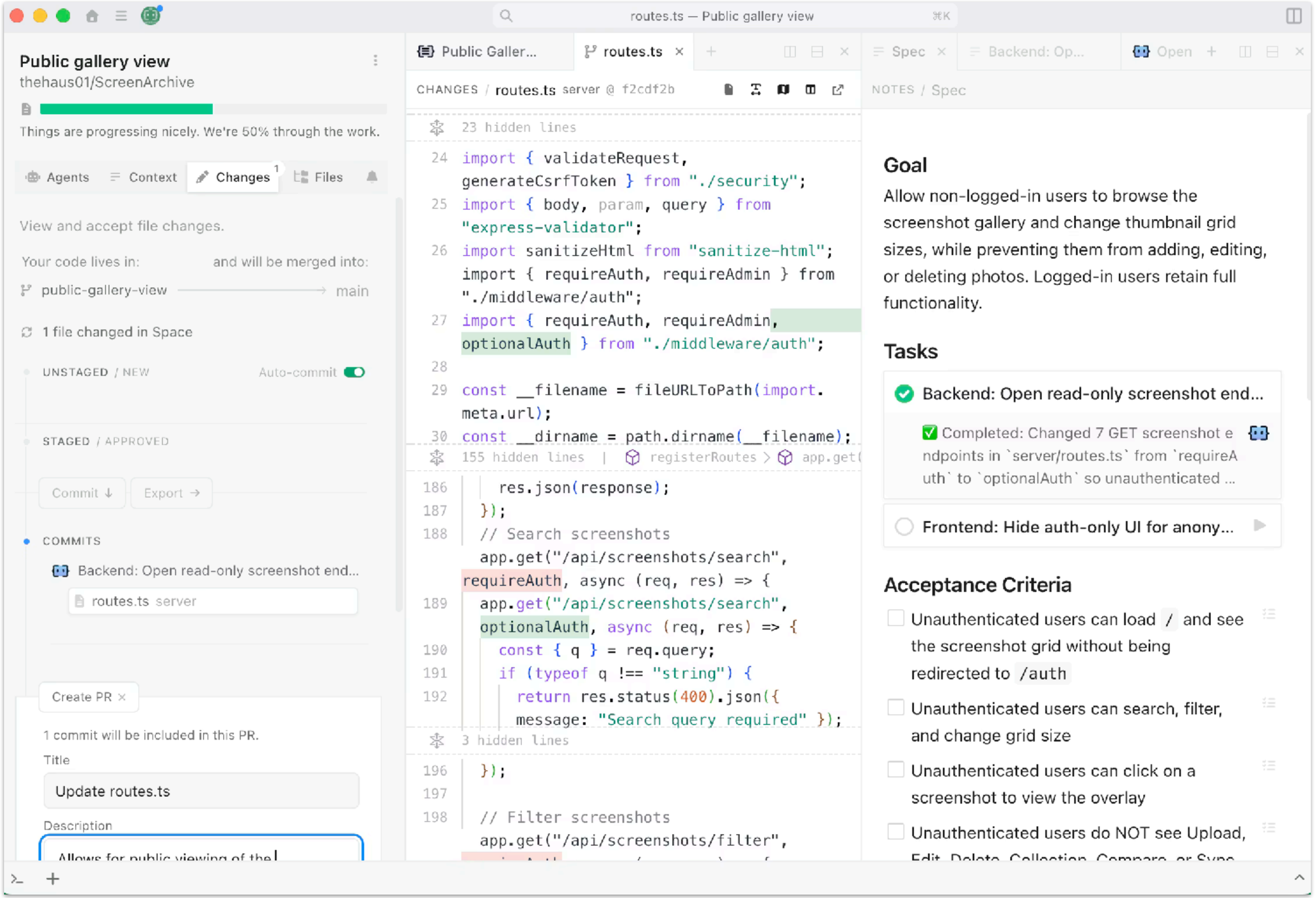
In the future,
everyone is a manager

What will the coding interface of 2030 look like?

- Last year has dramatically shifted towards agents and conversational UI for coding
  - Claude Code runs in the terminal but could easily be elsewhere. It's a React app for god's sake...
- Managing multiple agents is currently "more terminal tabs"



Intent IDE


routes.ts — Public gallery view
Public gallery view
thehaus01/ScreenArchive
Things are progressing nicely. We're 50% through the work.
Agents
Context
Changes
Files
View and accept file changes.
Your code lives in:
and will be merged into:
public-gallery-view
main
1 file changed in Space
UNSTAGED / NEW
Auto-commit
STAGED / APPROVED
Commit ↓
Export →
COMMITS
Backend: Open read-only screenshot end...
routes.ts server
Create PR ×
1 commit will be included in this PR.
Title
Update routes.ts
Description
Public Galler...
routes.ts
Spec
Backend: Op...
Open
CHANGES / routes.ts server @ f2cdf2b
NOTES / Spec
23 hidden lines
import { validateRequest, generateCsrfToken } from "./security";
import { body, param, query } from "express-validator";
import sanitizeHtml from "sanitize-html";
import { requireAuth, requireAdmin } from "./middleware/auth";
import { requireAuth, requireAdmin, optionalAuth } from "./middleware/auth";
const __filename = fileURLToPath(import.meta.url);
const __dirname = path.dirname(__filename);
155 hidden lines | registerRoutes > app.get(
res.json(response);
});
// Search screenshots
app.get("/api/screenshots/search", requireAuth, async (req, res) => {
app.get("/api/screenshots/search", optionalAuth, async (req, res) => {
const { q } = req.query;
if (typeof q !== "string") {
return res.status(400).json({ message: "Search query required" });
3 hidden lines
});
// Filter screenshots
app.get("/api/screenshots/filter",
Goal
Allow non-logged-in users to browse the screenshot gallery and change thumbnail grid sizes, while preventing them from adding, editing, or deleting photos. Logged-in users retain full functionality.
Tasks
Backend: Open read-only screenshot end...
Completed: Changed 7 GET screenshot endpoints in `server/routes.ts` from `requireAuth` to `optionalAuth` so unauthenticated ...
Frontend: Hide auth-only UI for anony...
Acceptance Criteria
Unauthenticated users can load / and see the screenshot grid without being redirected to /auth
Unauthenticated users can search, filter, and change grid size
Unauthenticated users can click on a screenshot to view the overlay
Unauthenticated users do NOT see Upload,

For next time...

https://docs.google.com/document/d/1rAzujTFcmQyVEbTSGmu3OH43BpXju4qEuYPXOPU9vRE/edit?usp=sharing