curated.

COVER

What is *curated?*

A monthly newsletter for all things tech and design, with a sprinkle of whatever I like (mostly AI). I plan on sharing news, resources, tips and information I think is **interesting**, **informative**, **novel**, or **useful**.

No, will not be teaching you how to program.
No, I will not tell you which sans serif font is the new Helvetica.

The goal is to empower you with what matters. If I like a research paper, you're reading about it. If I find a cool new font, you'll find it too. If OpenAI actually gets close to AGI, you'll know here first.

THINK PIECE

AGI was achieved last week, actually.

Rajin Khan

With the onslaught of new models releasing every month *(or should I say, week)*, things we used to joke about are starting to look more and more like reality. We just haven't realized how far we've come because of how normalized everything is. Just like our everyday tech, AI too, is now being taken for granted.

What do I mean by that? Take *Devin, the AI Software Engineer* that was supposed to replace us back in 2023. It wasn't that great, and performance was mostly underwhelming. Some heaved a sigh of relief. Some went about their day *("I TOLD you it's never happening')*. Almost nobody looked at *Cursor, the VSCode fork with really good Tab completion.*

Fast forward to 2026, *Cursor* now has a valuation of *$29.3 Billion.* Most Software Engineers swear by it, using 3-4 *Agents* concurrently and opening up 50+ PRs in a day *(and then getting those PRs reviewed with AI tools like CodeRabbit or Greptile).*

It's reached a point where *Anthropic's* frontier model, *Claude Opus 4.5,* is being hailed by many as already being so good that they can solve any of their coding problems with it, be it building websites, custom software on the fly, building fancy automations, you name it.

The things we're doing right now, if revealed 3 years ago, all at once, would *100%* start a frenzy in the Software Engineering community and have people saying things like *"It's SO OVER!"* or Product Managers going: *"I don't need you Sam, I can do the website myself. Go to localhost:3000 and see for yourself"*.

Much like everything in the consumer market, selling productivity and performance drives revenue, and revenue soars when it's shipped at blazing fast speeds.

OpenAI (with ChatGPT), Anthropic (with Claude), Google (with Gemini), and XAI (with Grok), are pretty much competing at breakneck speed to push out *"The Best Model"* at an insane rate. But these models themselves don't give us the full picture. It's what's done with the models, and the tools handed to them, that push us closer to the coveted state of *AGI.*

While *Claude Opus 4.5* is a fantastic model, its usage in *Cursor,* with its own system prompt, agent behavior, tool calling *(nobody is using MCP, I'm sorry)*, is what truly makes it feel special, like your own on call Junior II Engineer.

But it's still that. A Junior II Engineer. And if you take a moment to stop and think about it, this is exactly what we were afraid of 3 years ago. But now, we're so used it, we're demanding better performance, better *SWE Bench* scores, more points on *Humanity's Last Exam.*

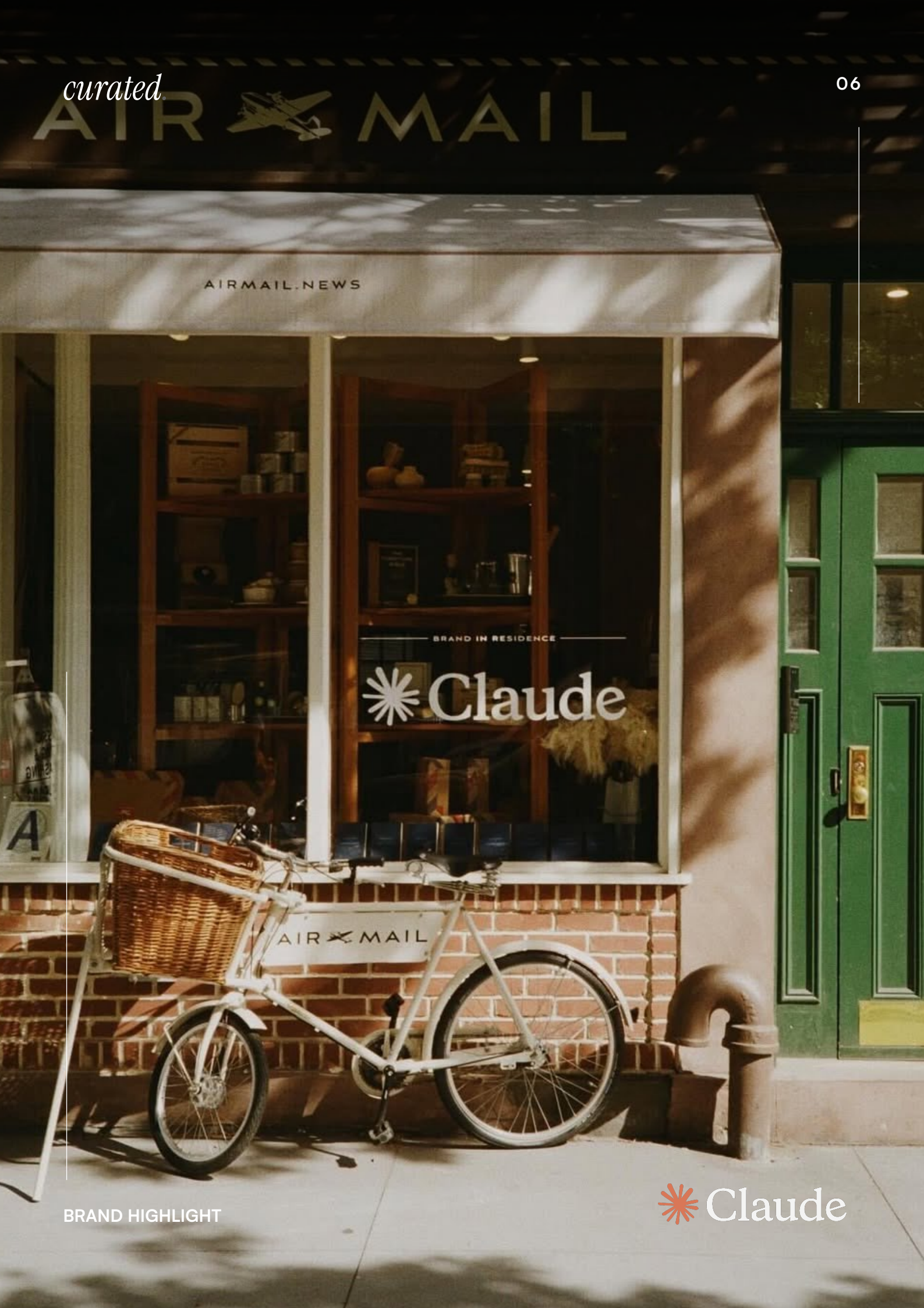
How long till it becomes as good as a Senior Engineer? It's only a matter of time, at this rate. By the time you finish reading this article, *Cursor* will have shipped 3 new features, *OpenAI* will have released *Codex 5.2 Pro Max Ultra High Fast Thinking (I'm not joking, their naming schemes are* ***really*** *this bad),* and *Google* will have dropped another *"Experiment"* that's going to kill 5 AI Startups.

We can't keep up, and we're getting used to it. AI is progressing FAST, and we're being super chill about it, for some reason. *ChatGPT 5.2, Claude Opus 4.5, Gemini 3.0, Nano Banana Pro,* all came out in a span of *30 days.*

Believe me when I say this, *AGI* is *not* going to be an *aha* moment on some glorious day where you'll see announcements on TV or grand declarations from any company.

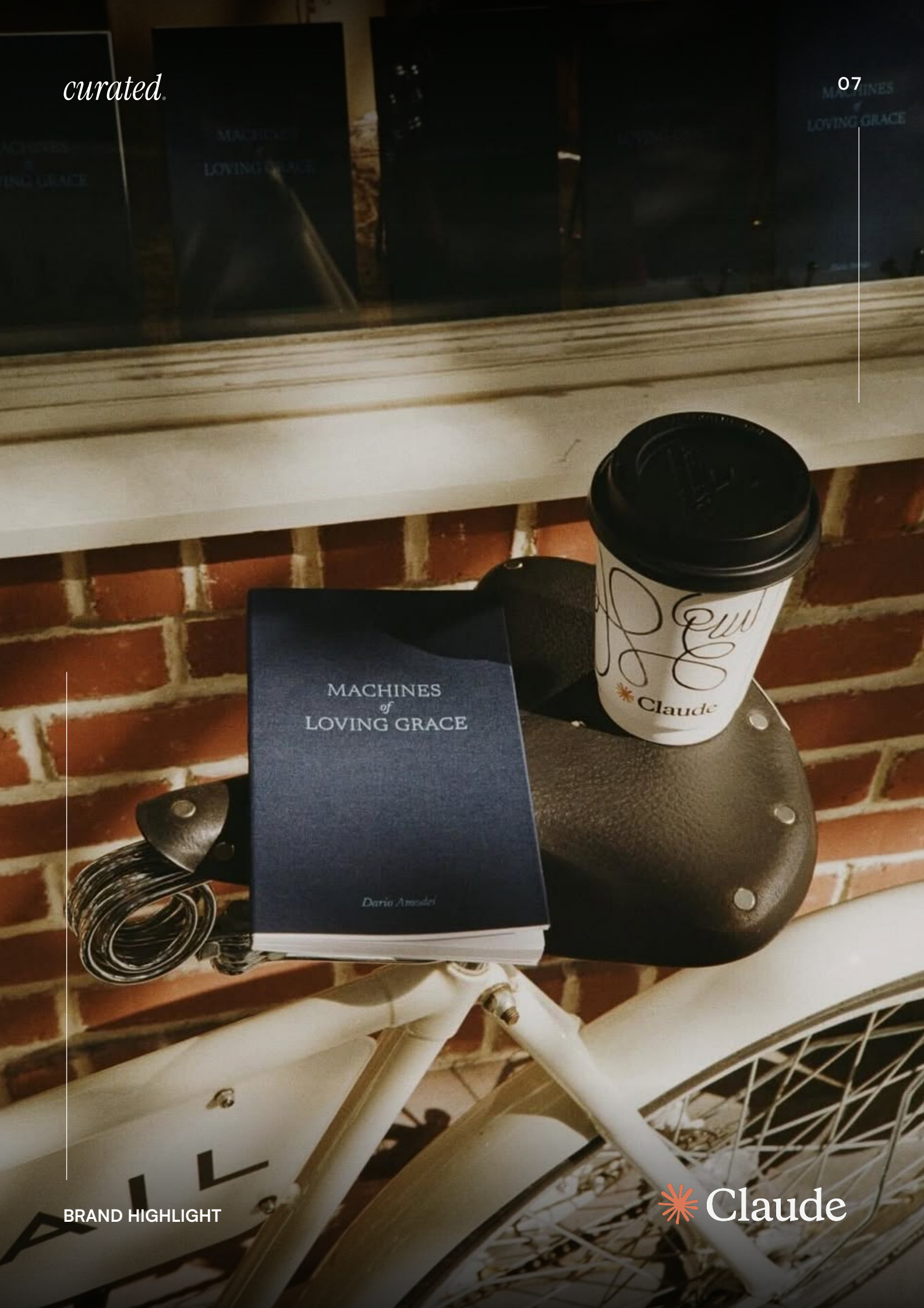
AGI is going to be a small feature which you'll read about casually over your morning coffee.

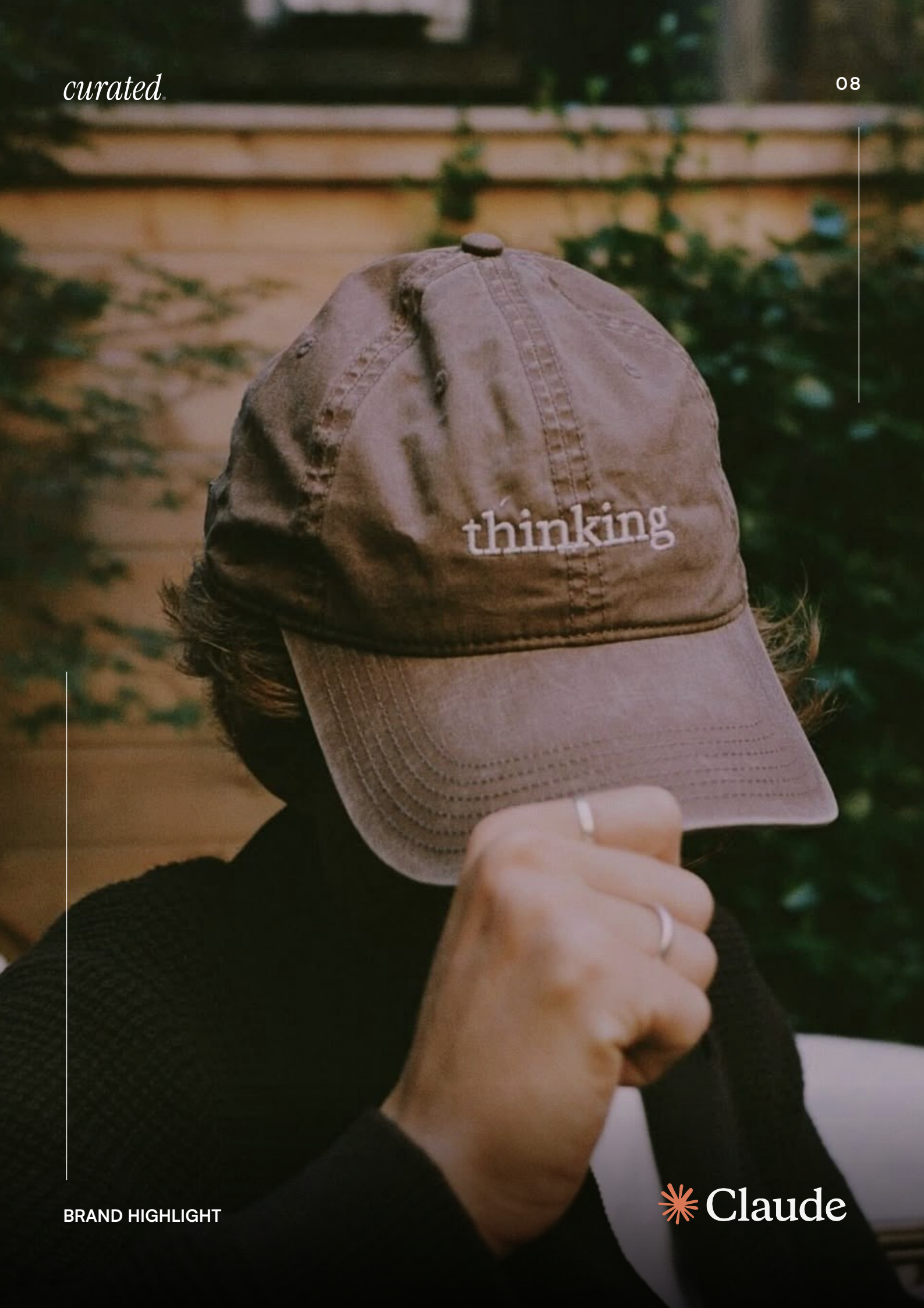
BRAND HIGHLIGHT



BRAND HIGHLIGHT

Claude


MACHINES
of
LOVING GRACE
Dario Amodei
Claude



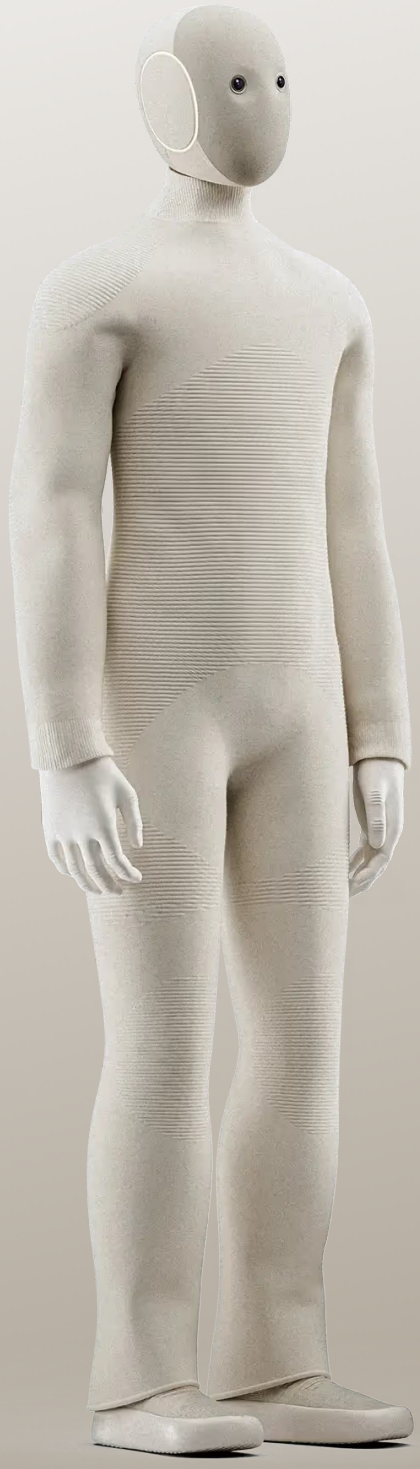
BRAND HIGHLIGHT

Claude

TECH

Is Neo the chosen one?

Rajin Khan



If you're talking about the one from *The Matrix, yeah, absolutely.* The humanoid *"AI"* robot from *1X Technologies?* It's complicated.

Neo is trying to be a lot of things. The claims are bold. It's a helping hand, it can automate chores, it helps you reclaim time, and it *"Grows With You".*

Consider me sold, right? We're living in *Detroit: Become Human*, already! I can sit by and ask it to fold laundry, do the dishes, clean my house, while I focus on what actually matters *(doomscrolling).*

Well, not really. Putting aside the *$20,000 (yes, twenty, sweet, thousand dollars)* price tag, while it does technically deliver what it states, the means to achieve it takes away a lot of the awe.

Reading so far, with the *"AI"* label thrown in too, how did you think the robot operated? Automatically, right? Well, kinda. Again, it's complicated.

You see, *Neo* divides its operation in two modes, *Fully Autonomous*, or *Expert Mode.*

Under *Fully Autonomous,* you have completely independent actions that the robot can perform, without any human intervention.

These include tasks like basic navigation and movement (walking around your house, moving to a person when called, self charging), simple household tasks (putting away dishes, *basic* cleaning and tidying, *basic* chore execution if it knows how), and voice interactions (it's loaded with an LLM).

While that sounds like a lot, and very fancy, note the usage of the word *basic*. It can do simple cleaning like sweeping the floor, putting away trash, but anything more complicated, like perhaps dusting your bookshelf, it'll have to learn that via *Expert Mode.*

Now, what is *Expert Mode?* I'm glad you asked. *Expert Mode* is fancy marketing speak for Human operators. Yes, a literal human operator will connect to your robot, watch and listen with all the cameras and microphones attached, and carry out a specific task from their control centre remotely *(they are fully transparent about it though).*

Outside of the absolutely *ridiculous* concern for privacy, what makes *Neo* a less of a sell is just how many things need to be done in *Expert Mode*. Folding the laundry? *Expert Mode.* Cleaning the dishes? *Expert Mode.* Clean your Bathroom? *Human Operated, Expert Mode, baby!*

It needs to learn, with time, different chores via *Expert Mode*, and when those chores will become *Fully Autonomous*, we still don't know.

So what's the value proposition here? Spending *$20,000* on a robot loaded with more cameras and microphones than you can count, always getting a 24/7 live stream of your house, only for it to be operated by another human remotely?

While I may sound appalled, they're doing something no other company has, and that's *taking the first step.* This is the kind of progress that needs to happen, and the people that use this now will shape the future.

That being said, do I want to pay *$20,000* to do A/B testing for a visionary product? Yeah, no thanks.

SOFTWARE HIGHLIGHT

tailscale

Networking, made stupidly simple.

Tailscale is a *Zero Trust* networking platform that makes secure connectivity ridiculously easy. Think of it as "your own personal internet", a *mesh VPN* that connects all your devices securely without the typical VPN headaches.

What's a *mesh VPN?* Think of it like this: you install and connect *Tailscale* on your home PC, and do the same on another computer, or laptop, on a *different network*, even in another part of the world. *Tailscale* will make it appear as if you're on the *same* local private network (they call it *"Tailnet"*).

If that one sentence doesn't immediately sell this to you, then it's probably not for you. If it does, I've got even better news. It's fully free (for up to *100 personal devices*), and dead simple to set up too.

Literally, just install and click *connect.* It takes less than 2 minutes to set up, works anywhere (Linux, MacOS, Windows, iOS, Android), and is configured to be *Zero Trust* by default.

This means every connection is *fine-grained,* uses *Wire-Guard* encryption, and provides *identity based* access *(not network based)*.

Tailscale is perfect for home labs, remote server access, CI/CD pipelines (integrates with *Docker, Kubernetes*, and cloud services), and sharing services between devices without exposing them publicly.

So if you wanna forget about the horrors of subnetting and port forwarding, *Tailscale* is worth checking out. It's one of those tools that changes how you think about networking.

PREDICTIVE CONCEPT DECODERS: TRAINING SCALABLE END-TO-END INTERPRETABILITY ASSISTANTS

Vincent Huang[*], **Dami Choi, Daniel D. Johnson, Sarah Schwettmann, Jacob Steinhardt**
Transluce

ABSTRACT

Interpreting the internal activations of neural networks can produce more faithful explanations of their behavior, but is difficult due to the complex structure of activation space. Existing approaches to scalable interpretability use hand-designed agents that make and test hypotheses about how internal activations relate to external behavior. We propose to instead turn this task into an end-to-end training objective, by training interpretability assistants to accurately predict model behavior from activations through a communication bottleneck. Specifically, an encoder compresses activations to a sparse list of concepts, and a decoder reads this list and answers a natural language question about the model. We show how to pretrain this assistant on large unstructured data, then finetune it to answer questions. The resulting architecture, which we call a *Predictive Concept Decoder*, enjoys favorable scaling properties: the auto-interp score of the bottleneck concepts improves with data, as does the performance on downstream applications. Specifically, PCDs can detect jailbreaks, secret hints, and implanted latent concepts, and are able to accurately surface latent user attributes.

1 INTRODUCTION

Interpretability seeks to explain the internal computations of neural networks, for instance through circuits (Olah et al., 2020; Elhage et al., 2021; Wang et al., 2023), probes (Alain & Bengio, 2017; Hewitt & Manning, 2019; Belinkov, 2022), or concept dictionaries (Bricken et al., 2023; Cunningham et al., 2024; Templeton et al., 2024). Since neural networks have complex structure and open-ended behaviors, explaining them by hand is not scalable, which has led researchers to develop automated interpretability techniques to explain neural activations in natural language (Hernandez et al., 2022; Bills et al., 2023), and *interpretability agents* to propose and test hypotheses about how activations relate to external behaviors (Schwettmann et al., 2023; Rott Shaham et al., 2024).

A downside of hand-designed agents is that they are bottlenecked by the capabilities of off-the-shelf models, which are not specialized for interpretability. Yet the core task—predicting model behavior from activations—provides a natural training signal, since predictions can be verified against actual model behavior. This suggests turning behavior prediction into an *end-to-end training objective*, by directly training assistants to make accurate predictions.

Concretely, we train an encoder-decoder architecture with a *communication bottleneck* (Koh et al., 2020). The encoder reads the subject model's activations $\mathbf{a}$ and outputs a sparse list of active concepts; the decoder reads this list along with a question q about model behavior and must produce the correct answer. Because the encoder does not see q, it must produce a general-purpose "explanation" of $\mathbf{a}$ that is useful for many different questions. The sparsity of this explanation then aids human interpretability.

We instantiate these ideas through an architecture we call the **Predictive Concept Decoder** (PCD). Concepts are encoded by a linear layer followed by a top-k sparsity bottleneck, and the concepts are then re-embedded and fed to a LM decoder. We jointly train the encoder and decoder on FineWeb (Penedo et al., 2024), using next-token prediction to provide scalable supervision for understanding the subject model's activations without requiring labeled data. We then finetune the decoder on

[*]Correspondence to vincent@transluce.org. See transluce.org/pcd for more information and to interact with the trained model.

Talking with Black Boxes

Rajin Khan

We treat most *AI models*, or *Neural Networks*, like "*Black Boxes*". It usually means we don't know what's going on inside it. We build it, give it an input, and get our desired output by adjusting different parameters.

But we have Chain-Of-Thought Reasoning, I hear you say. Well, *COT Reasoning* is still a part of the model's output, if we're being technical. The model chooses what to say about its reasoning. This means it can can *lie*, *be wrong*, or *not know* why it did something.

Now, I can't be the only one that thinks the enigma behind this *Black Box* model is too intriguing to not explore. And I really am not. A bunch of cool people from *Transluce* just published a paper that explores novel *Predictive Concept Decoders*.

What they're essentially doing is building systems that can read and explain what models are thinking, but *not* with reasoning. They're actually reading a model's *internal activations* (the numbers that fire through *Neural Networks*), and turning them into something we can actually understand.

Let's get technical: it's an E*ncoder-Decoder* system with a *communication bottleneck*. The *Encoder* reads the subject model's *activations*, and then *compresses* them into a sparse list of *concepts* (think of it like summarizing a 1000-page book into 10 bullet points).

Then, a *Decoder* reads *only* those *concepts*, along with a *question* about the *model's behavior*, and has to *predict* what the model will do. The catch? The *Encoder never sees* the question, so it has to create *general-purpose explanations* that work for *any query*.

So it's just a translator that speaks *Neural Network* and converts it to English, literally. This is what we're calling our *Predictive Concept Decoder*.

And these *PCDs* learned to do this automatically, and get better with training. Isn't that crazy?

The results are pretty impressive, too. *PCDs* can detect *jailbreaks* (sneaky prompts designed to bypass safety measures) *even* when the model itself can't verbalize that it's being manipulated. They can surface *latent user attributes* that models are using but won't admit to.

They can catch when a model is using a *secret hint* you gave it, even when the model *claims* it figured everything out on its own (as outlined in a lot of recent *Anthropic* papers).

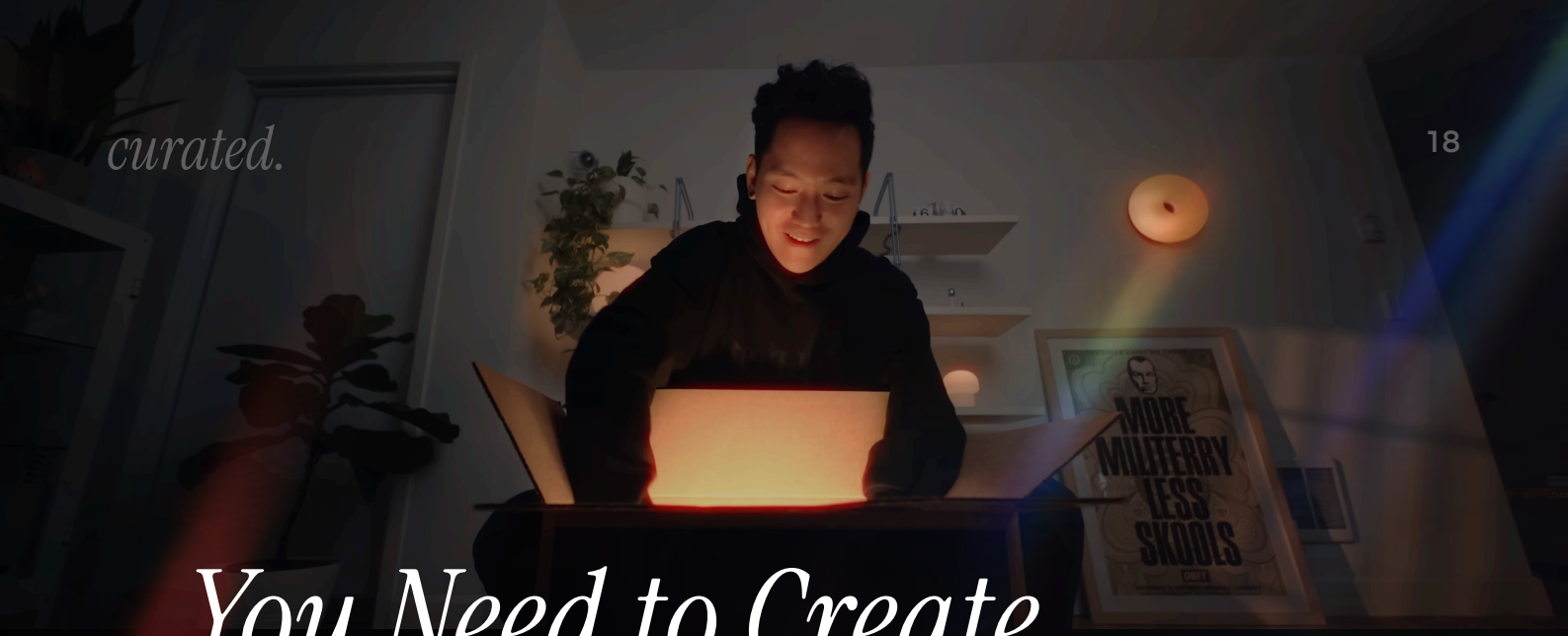
Don't get me wrong, this doesn't eliminate the *Black Box*. It just makes it a really good *translator*. The model, by all means, is still a *Black Box*. We don't fully understand what's happening inside. But now we've got a *system* that can read the tea leaves and tell us what the model is *actually thinking*, in terms we can actually understand.

The paper also interestingly shows that the quality of these explanations *improves* with *more* training data *(more data = better interpretability)*. That's a scaling property we haven't really seen before in this field. While most other interpretability work hits a wall, this one seems to keep getting better.

So is this the end of the *Black Box* problem? Not quite. But it's the first time we're treating interpretability as something we can learn, rather than something we have to engineer. And if there's one thing we've learned from the last few years of AI progress, it's that things we can learn tend to scale really, *really* well.

The Black Box isn't illuminated yet, but it's starting to talk.

ENTERTAINMENT



You Need to Create.

Rajin Khan

There's this moment in a *Scott Yu-Jan* video where he's explaining *Dieter Rams'* tenth principle of good design while 3D printing a key holder. *A key holder.* And suddenly you realize: I've been living with badly designed shit my entire life.

Scott's a *real* product designer (ex-*Google*, *ex-Amazon*) that doesn't gatekeep any of his expertise. He looks at everyday interactions, household objects, activities we take for granted, as a problem to solve with design.

He'll apply professional product design thinking to a key holder. He'll build an *iPhone dock* based on a 1976 alarm clock and make you care about design history you didn't know existed. He'll build a portable Mac Mini that makes you question why laptops even exist.

It's like getting design education disguised as entertainment, and that's *so* much more valuable than a random tutorial or another *hey, look what I made* video.

He won't just teach you how to think like a designer, he'll show you *why* you should.

But what got me hooked, or why I love his channel, isn't the projects themselves (although they're incredibly good). It was actually this core idea that he keeps coming back to: *Make Your Own Things.* It's so refreshing.

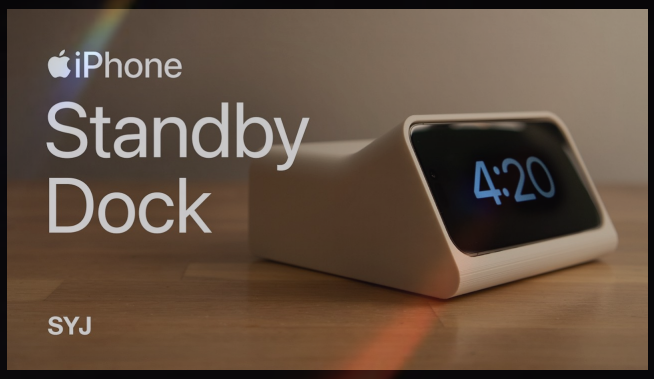
The world is full of boring, mass-produced solutions. We've all bought that generic key holder from *Amazon*, that uninspired desk organizer, that one-size-fits-all solution that technically works but feels... *empty.*

Scott's whole thing is asking: why settle for that?

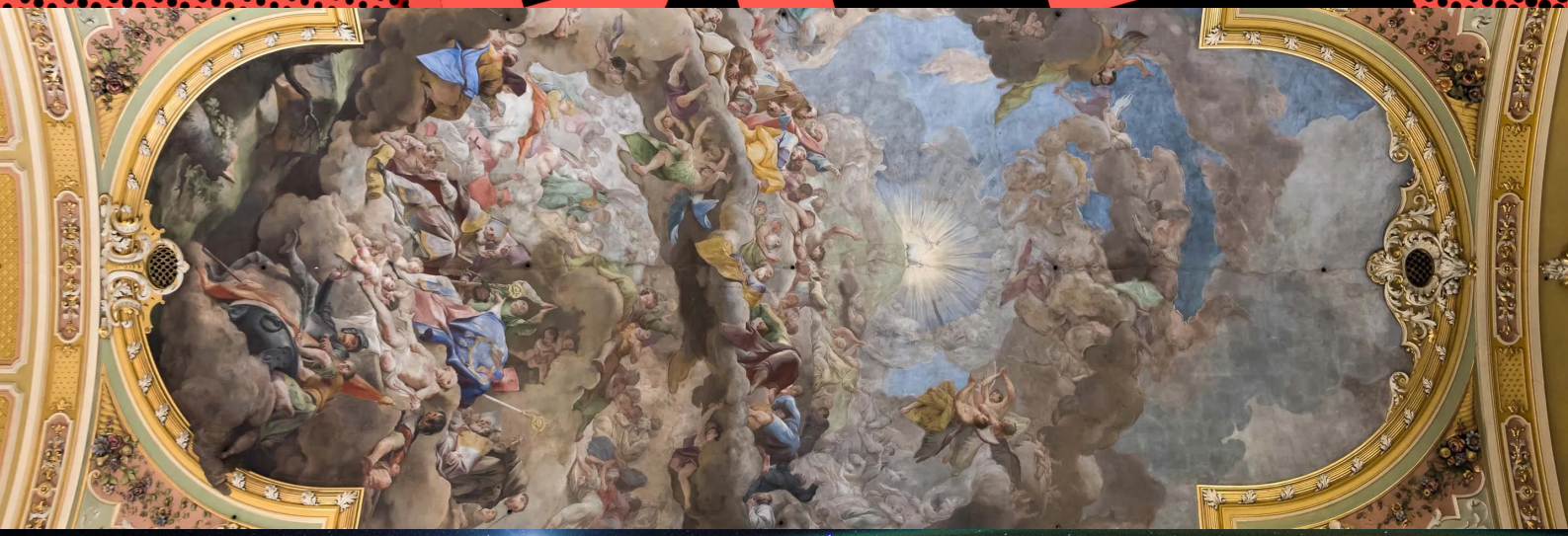
His philosophy is basically this: *Everything around you was designed by someone. And that someone wasn't necessarily smarter than you. They just had the tools and the opportunity.*

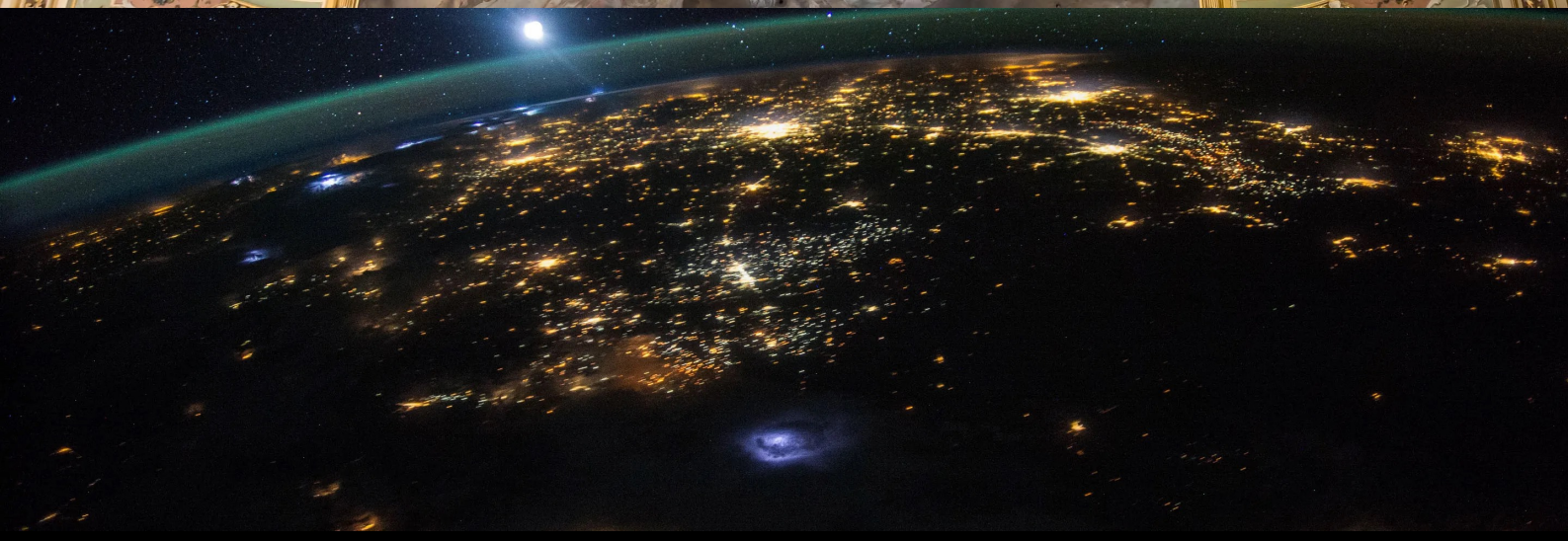
But now? With 3D printing becoming *actually* accessible (prices less than a gaming console), the playing field has been leveled. You can make things that are *better*, more *personal*, and actually *delightful* to interact with. That key holder you use every single day? It *doesn't* have to be boring. That drawer organizer that you open hundreds of times? It deserves more consideration than whatever generic plastic thing you found on *Amazon*.

They’re not just objects, they’re *interactions*. And interactions can be *fun*, *satisfying*, even *beautiful*. You’ve been settling for boring. Once you start thinking this way, you can’t really unthink it.



It starts small. You're opening that drawer for the hundredth time and thinking, *I could fix this.* You're at a friend's house noticing their *terrible* cable management. Then you're lying awake at 2AM sketching a better *soap dispenser* on your phone. You've *stopped* being a consumer. You've become someone who sees problems *everywhere*, and that's *not* a curse, it's a *superpower*. You're infected, and there's no going back.





DESIGN RESOURCE

Open Foundry

https://open-foundry.com

Good Fonts. Oh so free.

Your Note's a Mess. Good. It should be.

Rajin Khan

I hate taking notes.

No, wait. I love taking notes. I hate that I'm terrible at it.

You know the feeling, you're in a lecture, or a meeting, or it's 2AM and you're having the best idea of your life, and you're typing frantically. And then you look at what you wrote and it's just... fragments. Half-sentences. Bullet points that don't connect.

What was I even trying to say here?

I started working on my own solution because I was tired of that feeling. But doing that taught me the problem wasn't what I thought it was. I'd try to take notes on my laptop and fall behind. I'd even jot things on my phone, but they'd come out as word vomit.

We've been conditioned to think good notes are clean from the start. Organized. Structured. Ready to read and digest.

But that's not how thinking works. Thinking is messy. Ideas come in fragments. Connections form later, sometimes days later, usually in the shower. The best thoughts start as half-formed sentences scribbled at midnight.

All of this inspired me to condense my solution into one, simple goal: *dump your messy thoughts, click a button, and watch AI turn them into something readable.*

And *voila*. *PuffNotes* was born.

What if instead of forcing yourself to write perfect notes, you could just... dump? Write whatever comes to mind, however it comes out, and let something else handle the *structure*.

That's what the beauty of *PuffNotes* is. It's not replacing your thinking. It's removing the friction between having an idea and capturing it properly.

Look, note-taking apps aren't bad. I love me some *Obsidian*. I really like *Notion's* all-in-one vibe. I think good old *Apple Notes* is elite.

But they're solving the wrong thing. They're trying to make you a better note-taker when what you actually needed is permission to be messy.

Let your tool to clean it up later. Your notes deserve better than generic solutions that don't fit how you actually think.

PuffNotes is live right now *(at https://puff-notes.vercel.app)*. Use Offline Mode for complete privacy (on a computer), or sign in with *Google* for seamless *sync* across *all devices*.

Oh, and it's open source on *GitHub.* Look it up and help me close a few PRs *(and drop feature requests too)*.

A NOTE

Thank You,

for finishing December's *curated*. Seriously, you deserve a pat on the back for reading a full newsletter through. Everyone's attention span is so fried right now, it's crazy.

Got thoughts? Let me know, I love discussing.

Disagree with something? Talk to a wall.

The structure will be similar for most issues (unless I feel otherwise). I am not sponsored, affiliated, nor associated with any of the brands and images used or mentioned throughout this issue.

I will never accept payments for any Highlights (be it Brand, Software, or Research Papers). I talk about what I find interesting, and think you guys might too, that's the whole point.

See you next month (or next week, depending on how much coffee I drink).

Rajin Khan

END.

curated.

a newsletter by Rajin Khan

find me and subscribe at rajinkhan.com