OpenAI vs Anthropic vs Open-Source | Token Maxing, AI Hangovers & The Coming ROI Reckoning

84 MIN · YOUTUBE · HTTPS://WWW.YOUTUBE.COM/WATCH?V=LGO_QBGV198
https://www.youtube.com/watch?v=lgo_QbgV198

SUMMARY

Matan Grinberg, CEO of Factory, discusses the evolving landscape of software development and AI, emphasizing the importance of resource allocation and the shift towards more efficient problem-solving. He believes that as AI tools become more integrated into businesses, organizations will need to rethink team structures and focus on core competencies rather than merely increasing headcount.

- *The commoditization of software development is inevitable, with every enterprise capable of building necessary tools.*
- *AI tools will lead to productivity gains, allowing teams to solve more problems efficiently.*
- *The future will see a bifurcation in engineering talent, with a focus on "load-bearing" individuals who can leverage AI effectively.*
- *Companies must prioritize core competencies and allocate resources based on what truly drives business outcomes.*
- *The rise of open-source models will challenge frontier models, as many tasks do not require cutting-edge technology.*
- *The culture of engineering will shift towards a more holistic approach, integrating sales, marketing, and product development.*
- *There is a concern about the security implications of rapidly generated code and the potential for significant incidents in the future.*
- *The market for AI infrastructure is expected to mature, with a need for model and application separation to ensure competitive pricing and efficiency.*

The world going forward there is going to be nothing that no one can build. Everyone is trying to commoditize the other. Value acrruel is a time dependent phenomenon. Meet Matan Grinberg, CEO and co-founder of Factory. Before factory, he was a physicist. He spent 12 years trying to be one of the best string theorists in the world. Now he's changing the world of software

development. He works with some of the biggest enterprises in the world. He does look like Matt Damon from Goodwill Hunting, but he is one of the best founders I've met. So many of the tasks that we're doing, we don't need the very frontier to do it. We might see a short-term contraction of usage of the very frontier models. I think it's pretty embarrassing that we don't have frontier open models in the United States. Name a legendary company that has a [__] sales or marketing team. You can't. The age of the polymath is back. We will see the best companies treat teams more and more like Seal Team 6 or like professional athletes. Ready to go.
>> Matan, it is so good to have you in the studio. You've just uh insulted my continent with the suggestion that we've only come up with bottle caps while you came up with Transformers. Um, not wildly untrue,
but this is going to be a fun show. So, thank you so much for joining me.
>> Thank you for having me, Harry. It's a pleasure to be here. Now, I was just doing a show yesterday with Rory and Jason, and Rory was basically saying, you know, the fundamental question is, will we see an increase in GDP? Um, coming from AI and the coding developments that we're seeing, and will it lead to GDP increasing above the 2% average for the last 200 years, do you think we will see meaningful productivity gains from the AI tooling that we're seeing, or is Uber's concerns validated? So I think yes absolutely we will we will see tremendous growth from these tools. I think it takes time to

permeate through um because you can tell like on an individual basis like almost like on a problem bypro basis. We can solve problems faster with these tools. Now companies generally or organize around solving problems. Um, and if you're organized around solving problems and you have some set of personnel, you might say this is the number of problems we can solve at a given time based on how many people that we have, everyone is now going to be able to solve more problems with the same number of people or solving the same number of problems with fewer people. But it takes time for the resource allocation to adjust. A lot of businesses will have to ask do we want to solve more problems now because of the increased leverage that we get or do we want to solve the same problem but now we can do it in a more efficient manner. That's I think a question that a lot of businesses will be will be uh grappling with.
>> Do you think we will have fundamentally smaller teams which ultimately suggests that number two is the option that most people take or do you think we will have actually the same size teams and we'll just go after a more expansive area?
It's really not obvious because um there are dynamics that it's hard for me to predict. But what I will say is again bringing it back to problems. All of these companies are now going to have to think okay we have all this new leverage. Do we want to solve the same problem? Do we want to increase our ambition and solve a bigger problem or do we want to solve more problems uh that might you know our users maybe have? I was watching Andre Capathy and

he was talking recently about you know the 10x engineer actually is wildly misunderstood and you won't see the 10x engineer you'll actually see a smaller number of 100x engineers and kind of the rest and this bifocation of engineering talent do you think that is the right way to look at the future of engineering talent
>> directionally yes because what is a 10x or 100x engineer I I don't necessarily agree with the language around it but like
I think like it just implies as if like 10 10x of what is it? Pure output like if if it if you when you say 10x it means like how much code they're writing. Yeah, now I can write a billion lines of code with these tools. It might be [__] lines of code though. Um so maybe the way that I like to think about it is like loadbearing individuals in an org. It's kind of like if you remove this person things fall where there in some orgs there might be people where if you remove them nothing happens and they're you know not loadbearing in that case and so you know the basically these people who have very high leverage are now being handed a tool that gives them even more leverage and so using the language of 10x or 100x yes they're levered up they can have even more impact um but uh and I think you know with that leverage language those who know how to use leverage average will be able to have even more impact and those who don't will kind of on a comparative basis be that much less valuable to a business. When we think about kind of the two different parts you said there hey you have the option of you can do

more with the same size teams or you can reduce teams and do what you already did. If I am thinking as a leader today what would be your biggest advice to me on how I should think about resource allocation for tokens internally?
>> Yes, this is a this is a a great point. um this resource allocation problem of token it's not just tokens it's like dollars it's tokens it's people this is I think going to be the thing that over the next 24 months every seuite is going to be thinking about and I think the right way to go about it is what is the core competency for our business what actually matters for the business that we are doing um and then how do we allocate resources accordingly in other words if you're a logistics company your core competency is probably not software development now you might have add a lot of software engineers as a means to an end to deliver on your logistics goals, let's say. Um, but that might not be your core competency. And so what you should be thinking about is not how do we get more engineers to make more features because that's what engineers have in the past been judged by like how many features do they ship in a quarter. Instead, it's like what are the actual output metrics that matter for our business and how do we now allocate resources whether it's dollars, whether it's tokens, whether it's headcount to uh more dramatically move the needle on that business outcome. And I think this is this is great for the world because I think part of the reason why so many organizations got so bloated is because we were in a period of time where everyone was focusing on intermediate

metrics. If you're an engineering team, we wanted to ship three features this quarter. Did you ship three features? We shipped four. What a great quarter. Like that doesn't necessarily matter for the business at all. And so now it's like finally coming back to what matters in the first place. Like what are the business metrics that we want to, you know, move the needle on. Is it customer satisfaction? Is it revenue? Is it market share? Um and you can kind of tie back every individual's work to that, whether it's marketing, sales, engineering, all of it.
Kirkland announced a $500 million spend. You're you're friends with Winston from Harvey. Uh fantastic guy. Um who obviously I'm sure has I don't know if you guys have spoken about this actually, but like that's a it's a big spend. $500 million across 5 years to internally build their own Harvey or Lora. Um
>> how did you think about that?
>> Um I mean it's fun, you know, talking about core competencies. Kirkland spending half a billion dollars to build their own AI tools. Uh my understanding is that building AI technology is not a core competency of that firm. I think I'm I'm I was surprised to see it. Now I actually think this is good for Harvey because it's nothing like trying to do something yourself to make you realize, oh [__] this is actually really difficult. This doesn't actually matter for us to have the in-house ability to build this ourselves. Let's go and have someone who is an expert in this to go and build this for us. That is my sense. My favorite is also the amount of people

that like, see, we told you how easy it was. And you're like, it's so easy. They're committing half a billion dollars. That would suggest the opposite. Um, I had Brendan on from a call the other day and he was fundamentally saying that the next 12 months would be the most value acrewing 12 months for AI infrastructure companies. We would see that the models of the products and the AI application layer companies would be most at risk. denigrated. Would you agree with that?
>> I would disagree. I'd pretty strongly disagree. Um, for a couple things. One, actually, sticking with the Kirkland thing. I think as an example, we're so used to a world where moat in software was I know how to do this and you don't and so you're going to pay me because I have the engineers who know how to build this and you simply cannot. Now, the world going forward, there is going to be nothing that no one can build. Every single piece of software anyone will in theory be able to build. Now back to the resource allocation though. Is it worth your time and your energy to go and build it or should you go to someone else who has already built it or can do it faster? To me it kind of an example of this is like suppose we had a a very busy day at work. I could probably go and pick up lunch for everyone on the team. I know how to do it. I know how to walk out the door, place an order, hold the bags, bring them in. Now just because I know how to do it, is that an efficient use of my time? probably not. I'm probably going to say, you know what, for my resource allocation, I'm going to pay someone to

go and do that for us because at factory, our core competency is not that the CEO goes and gets lunch for everyone. Um, and I think it's somewhat similar here, which is like just because you can build a lot of these things does not mean you should. And in fact, often times you want to be really ruthless about what are the few things that you and your team own and do end to end. And then if it's not relevant to your like core business and your core competencies, outsource it.
>> What would you like to do, but it's not core competency for you and so you don't do it because of focus?
>> Oh man, I I enjoy like making breakfast. I haven't done it in like 3 years. Um I like there's nothing like it's just it's it's not time efficient. It just doesn't make sense to spend my time doing that.
>> All right. Okay.
>> Um but like I I it is, you know, it's something that I do enjoy. But um but to your point on model, applications, infrastructure um I'm not sure if you've seen there's a meme of the Microsoft org chart and it shows like you know different segments and they all have guns pointed at each other. Um just to show like in Micros you know there's a there's a lot of bureaucracy and everyone's kind of fighting for who gets to do what. Um I think that image is pretty accurate to what's happening right now with the models, the application companies and the infrastructure companies where everyone is trying to commoditize the other. Everyone is trying to say oh no this one is irrelevant. This all the value is going to be here all the value is going

to be there. The reality is value acral is a time dependent uh phenomenon. So like it's not like there is one person whose steady state gets all of the value. That's not how it works. It's maybe for this next year, this person is who has the pricing power, who gets the value. This next period of time, these people get it. We are all, you know, whether overtly or not, and maybe I'm saying the quiet part out loud, like everyone is trying to commoditize the people that are not them. So, for example, we're model agnostic. We want to give our customers the best pricing, the best performance, the best speed for whatever task they want to do in their software development. Um, and we want to make sure that OpenAI, Enthropic, Google, Microsoft are all under pressure to make sure they give the best models for as cheap as quick as they can and don't feel like they can just, you know, charge whatever they want. Now similarly they you know the model companies want to make it such that the applications are all trivially easy to build and really the the product is the model and then the infer companies have their own spin on this but the reality is like everyone's trying to commoditize the one that's not them and so you know from uh from Merkor's perspective you know it's very much in their interest that models that have access to proprietary data get differentiated value and capture a huge amount of value because that validates their business model. It's a it's a big push and pull of who can who can get the leverage.

>> What is the belief that would invalidate

yours?
>> The bare case against factory is if one model provider gets significantly better than all of the others. Um so basically I think a key thing for us is that all the models are going to be roughly as good as each other. They'll be good at one is a little bit better at review, one is a little bit better at testing, one's better at Python, this and that. It all kind of fluctuates every week. Like even already people have a have a hard time keeping track what model is number one, what's the latest thing that came out. If one model ends up going way above all the others, that's a case where it's like, okay, we want to put, you know, companies might want to just completely go in with them. Um, but then that's a monopoly for the entire economy to be worried about. Is the rate of model development sustainable? And what I mean by that is like, you know, I was with the founder of Nebus the other day and he was talking about it on, oh, every few weeks we see new models. And I said, you're you're wrong. every few days, especially when we look at Chinese open source, it's like three or four a week.
>> Is that rate of model development
>> a feature of the time that we're in or is it an ongoing characteristic or trait of this environment? I think eventually we'll stop seeing them as model releases and they'll feel more continuous
>> like just how before it was you know GPT2 then GPT3 then GPT3.5 then 4 4.1 but like and then you get more and more desk and like you know 4.523 like eventually they're just going to not

announce it and it's just like hey look here's our model that's continuously getting better um because already people have fatigue like engineers at the enterprises that we work with can't keep up with every single model that comes out nor should they and I That's the the whole you know the case for the application layer whether it's us or like a Harvey or whoever else is we're going to figure out what model is best for what use case where the trade-off is between cost um quality speed and we'll just deliver that to you based on the task that you have because it's hard to focus on what matters for your business and then also keep track of all these models that keep coming up. A big question that people have is around the rise of open source and whether everyone is concerned by the amount they're spending on tokens just being so much larger than they thought. Hey, we spend our annual budget and it's May [__] Um
>> maybe we should move to open source.
>> And we're seeing more and more great companies use frontier models, see where they can get to, and then move to open source to get as close to that as possible.
>> Y
>> how do you feel about that being a considerate threat to maming the market for frontier models? I think it's a really important counterbalance because it show it basically allows you to make the tradeoffs of what tasks do you want to put what level of intelligence on. Um and I think it's a really important counterbalance because a lot of enterprises will realize so many of the tasks that we're doing we don't need the

very frontier to do it like and we can do it much faster much cheaper with these open models. Um, and again, it's part of the resource allocation. And to do good resource allocation, you want to be able to be anywhere in that cost, quality, speed trade-off.
>> I I love the gifts on Twitter or memes on Twitter when it's like, you know, me naming a file and it's like the massive cigar with the blowtorrch.
Yeah, I love that.
>> No, cuz it's such overkill. But also there's a funny dynamic that emerges uh which is there's kind of an ego thing where oh no no the work that I'm doing only a frontier model could handle. Oh this mere open model can't deal with the work that I'm dealing with. And this is like even admittedly when I first started switching over I'd be like I don't think an open model could handle this. And it's like no it probably can. And it's kind of a funny thing to to mentally deal with of deciding you know manually or then having the router do it for you.
>> My question is enterprises like security. They like reliability. They like ease.
>> Yes.
>> And when you have Frontier models which are packaged perfectly, priced clearly,
>> and it's secure,
>> they not just go for that, the easy option over trying to be smart and intelligent routing to different open models. So, a couple things. Uh, one, it's easy when there's only one of them, but again, as we said, a new model comes out every week, and if you have to go through the full enterprise process to

get every new model in, it's not very easy. Um, two is it's also really expensive and if you're seeing your costs go up like crazy and not having an ROI case, it doesn't make as much sense. And I think something that's interesting is there's kind of like three phases that we're seeing happen in these enterprises. Um, so phase one, this was a couple months ago, was board yells at CEO, "Hey, Mr. CEO, what's your AI strategy?" CEO is like, "Shit, I don't know." Uh, CTO, "Hey, what's our AI strategy? Let's make sure we adopt AI. And so then phase two was kind of AI at all costs, token maxing, part of your performance reviews. We're going to measure how much you guys use AI. Everyone, you have to adopt. That was phase two, right? Get as as many people to adopt as possible. Phase two happened a lot faster than people might have expected. And so now we're ending phase three. Like phase two was kind of like the the debauchery, the long night, you know, taking shots, having a great time, using all the AI. Phase three is the hangover where you go and look at the bill and it's like, "Oh my god, we are spending so much. I have no idea what the ROI is. Does this like is this helping our business?" Um, that's where a lot of these companies are at now. And I think this is why routing is so important cuz they're realizing and this is a true story. One of the CIOs I was speaking with realized we've been spending hundreds of thousands of dollars per month on people asking Opus 4.8 questions like, "Hey, how's it going?" like what what are my macros from the food I ate today? Like what's

the weather like? And it's like guys like we don't we don't need the the frontier of human intelligence to be doing this stuff for us. Let alone it's not even workrelated in some cases. But
>> will we see a contraction then given the hangover period being realized?
>> Um I think there are we might see a short-term contraction of usage of the very frontier models. But I think it's healthy. Um, I think it's healthier health healthier to do that than to like be blind to it and then have a real sudden kind of change there.
>> Uber announced last night, I think it was, or yesterday, that they were having like a $1,500 budget
>> per individual.
>> How do you respond or think about that?
>> We I've literally seen this with dozens of our customers where initially, and this is a lesson on our post sales team, where initially we came in, we were like, "Oh, by the way, we have these user limits, but here these are the models. Go crazy." This was before we had routing and it happened a couple times with customers where the usage would go crazy. They hadn't spent the time to actually determine what parts of the codebase do we want to dedicate these tokens to versus not. And then they were like, "Oh my god, we're spending so much. This is crazy. We need to put in token limits." And at first when we weren't like the first time this happened, we were like, "Oh my god, their usage went down. What's going on?" But spending time with them, we realized, wait, we need to make sure with every customer, we are having a very clear conversation with them of,

you know, it looks like you guys are spending a lot of tokens on some of these things. Have you thought about consciously, yes, we want to do this. Sometimes we'll proactively set in those user limits just so it's better to be aware as you're going up as opposed to just going crazy and then kind of realizing. And so what's happened with Uber publicly has happened privately with a lot of customers of ours. And yeah, there's a little bit of shock where it's like, okay, wait, let's put in these user limits. But then you come into a question of, well, wait, this team is really important. They should have a different user limit than that team. And we're just getting towards this world where you have very nuanced resource allocation throughout your or to me the biggest question that I ask myself and I think we need to ask ourselves as an ecosystem today is if Mark Ben off says that he spends 300 million on anthropic, okay, for his devs, that is 3.8% of salaries. Okay, great. What will that number be in 3 years time? Because if it's still 3.8, [__] If it's 20, [__] again. But [__] positive.
>> Um,
>> and if it's Brandon at Mccau who says that he's spending more on tokens than he is on headcount, [__] again, but even more positive.
>> What do you think that percent of dev salary is in 3 years? I think it's actually a more nuanced question than we might think. I actually think it can be as low as 0% for some individuals and it can be as high as like thousands, tens of thousands of percent for some

individuals.
>> And what's the dependence there?
>> It depends on what the unique skills of those individuals are. And I'm saying individuals and not devs in particular because I think the way we even organize roles is going to be very different where I'm not sure like dev as a word makes sense. Traditionally, it's like custodians of code, right? the people who do anything relating to code are engineers or developers. I think everyone is going to be loosely interacting with code in your org, whether they're sales or marketing. Um, but I think the difference is there are going to be certain people where, again, we're coming back to resource allocation. They get more leverage by using more tokens. And then there are going to be certain people where actually they don't really need tokens at all. And that's not what how they deliver value to the business. Like for example, you know, maybe our best salesperson, the way we use them best is not by having them use tokens, but by going and meeting people face to face, right? That's an obvious example because they don't write code in the first place. But I think similarly, maybe there are some engineers who they actually do their best work by spending time with users, spending time with their customers, maybe doing some data analysis that's not very token uh expensive. But then there going to be others who are delegating to dozens of droids in parallel working on a ton of different crazy features and refactors and migrations. Um, but I don't think it's going to be a consistent number

across the board. In fact, I would argue that if your org has a standard number where it's like we want every engineer to be at this percent of their salary and token use, you're probably uh painting with way too wide a brush.
>> If I were to say give me an average number, what will that average be? Like what will the median be?
>> I would say order of magnitude. It'll probably be comparable to salary.
>> Comparable to salary.
>> Yeah. Like on the same order of magnitude
>> within a three-ear timeline.
>> Yeah.
>> What percent of tasks today be using frontier models could be done with open-source models?
>> Probably like 80 to 90%. It's typically the planning that really needs the frontier models.
>> But is is that not like the most I'm I'm sorry. I'm I'm really, you know, um dim. That's nonsense.
>> Uh but but but if it's 80 to 90%. Does that not just present the biggest bare case ever against code or cruel code cuz you're just taking away 80 to 90% of that time.
>> Well, it depends cuz those that 20 10 to 20% could be the most important tokens. It's maybe like right like 10 to 20% of the tokens, but those are really really important because it's kind of decision-making tokens perhaps.
>> Sure. Um, but it's it's very similar to how we structure human orgs, which is like often time often times leadership makes very key decisions that determine the fate of the company

>> and they don't spend the most hours. Like if you look at the human hours of a company, most human hours are not spent on making the decisions. They're they're on gathering data or implementing things, but then there's a select few hours where it's like here is where we're going to, you know, make this irreversible decision on the strategy. Um, and those people that make those decisions are also typically paid a lot.
>> Yeah. But the assumption there would be then that you'd have to increase that spend for that 10% even higher
>> and that's what's happening already. It's like the frontier models are sometimes they're getting more expensive or you're using the ultra high reasoning or you know this this type of thing. Um, and so it's like okay if these if this planning thing is the very key thing we'll spend on it but it doesn't necessarily mean that most of your tokens are going there. It's just for certain key steps maybe you want to spend a lot um and it's worth that allocating the the budget there um but then once it comes to okay we have the plan now let's implement the open models are typically really good
>> when we think about um what you're willing to spend on I asked Brandon how much it costs to hire great AI researchers
>> and he was like tens of millions of dollars
>> y
>> have you found the same and is it impossible to hire great AI researchers in competition with anthropic and open AI. We are a very uh opinionated organization and so uh it's very self

like the people who like the opinionated stances that we take are willing to not necessarily go and try to you know maximize the dollars that they can get out of in the market. Um that said it is still you know pretty competitive.
>> What do you think is the strongest opinion that you have that most people disagree with? I would say the opinion that we have that I think in the space that we are in is the most controversial um is uh the way that we treat what product is at factory. Um, I think there's some very commonly held beliefs at the labs or at some of our competitors who are also doing kind of software development. And this is honestly growing up in the Bay Area, there's a very common Silicon Valley fallacy, which is there's like research is like the pinnacle. And then there's engineers who implement the research. You know, they're not quite there, but you know, they're still great. And then there's sales and marketing and all that dirty stuff. Oh, if only we could build a better product and it would sell itself and we wouldn't need to deal with, you know, sales and marketing. Um, and it's just completely delusional. Like the product at factory is the entire journey from the very first time they hear our name till their 10th renewal after a decade of being a happy customer. Now the software is a big part of that journey, but so too is the the marketing that we do and the people that we have running that. Same with the sales process. The people that present themselves in discovery calls or in demos or in solution engineering. Like that entire thing is the product.

Everyone is first class. It's not like we have engineers who are like wholly at the office and you're not allowed to speak to them unless you're an It's like no no no we have engineers and sales people sitting next to each other. There are no like engineer corner sales corner and any of that stuff. It's like everyone is completely intermixed. Um when sales people close a deal, engineers say we closed a deal. When engineers ship a feature, sales people say we shipped a feature. It is entirely one team, entirely cohesive. Everyone, there's no like first class or second class. And this is shockingly controversial. Um, in particular in the Bay Area and in particular in coding or in AI, it's messed up. It's so messed up. And I think that the reality is, uh, it will come to haunt some of these companies one day because I think right now where there's a gold rush and everyone's like desperate to sign, you know, and get more tokens from these people, it's easy. They're kind of like, in my mind, it's kind of like they're astronauts in space where there's no gravity. Your muscles will atrophy. Gravity will come back. And if you don't have a good sales and marketing team cuz you don't give it respect, the second gravity returns, all of your muscles will be atrophied and you won't be able to compete. And I would say this, name a legendary company that has a [__] sales or marketing team. You can't.
>> No, but I can name companies that have [__] products but great sales and marketing teams. That's that's the ironic thing on the flip side. And in fact, it seems like

>> many more legend
>> like
>> I'm not going to name them because I'm going to jumble.
>> I'm sure we're thinking of some of the same ones.
>> I mean I mean yes, if Chad Pets were here, he would say them. Um
>> Yeah, that's right.
>> Does what it takes to be a great engineer change when you essentially become prompter and manager of agents versus creator and doer of tasks?
>> Yes, it very seriously changes. And this is actually why we're selecting for very intentionally like this culture that we just mentioned is really important because the best engineers are going to be the ones that don't see sales and marketing as dirty work but as again an important part of the product because as an engineer you're no longer you know just your job is ship feature. It's no you are owning full endto-end outcomes of here's the way the customer is behaving. Here's how maybe we can change that behavior that makes them uh you know uh a better user long term. It makes them more agent native. They get more out of our product. We can then follow them through that journey. Enable the salespeople so that they know how to talk about it or they know how to demo it. This like is this is like a full stack engineer that goes way beyond just engineering but into sales into marketing into enablement and all that. Um and those are the parts of engineering that really really matter. Those are the parts that have made engineers typically good founders is when they have that. And the parts of

engineering that become less important are funny enough the things that the Silicon Valley has really bragged about a lot which is like competition winning or like Olympiad type are you like as fast as possible at coding or do you memorize all the different nuances of these different languages those are the parts that don't matter like if you memorized some coding lang or some you know syntax of a coding language that someone else didn't it doesn't matter people like not the credentialism but I think they misunderstand VC mindset right now and as a VC I'm happy to share how you feel how we feel which is just like fundamentally there is intense uncertainty around what anthropic and open AI will do and who they will kill at the application layer and so in a world where we desperately seek certainty we look for validators and the validators of someone being a math olympiad or you name it whatever that is that validation
in the wake of not having other certainty that serves as a good crutch
>> yes but it's It's a crutch. It's a crutch. It's helpful. It's a good indicator. Like generally, you can't win competitions if you're dumb, right? Like it's pretty it's pretty rare. Um, however, for these types of engineers that we're looking for, it's like that's cool, but that's kind of irrelevant. Like, what have you built? How have you taken ownership and agency of things end to end? And this is not like we have people on our team that have won Olympiads and I think they're great and it's fantastic and a lot of my friends have, but there's also a certain like

especially there's some high schools that like really focus on like you must do the math Olympic like you must do this the Amy to then go to the IMO and like this is the path to success where actually that's kind of anti- signal cuz there it's like you're not owning your fate or choosing your agency. You're kind of going through the funnel. But then there are people on our team who are like from the middle of nowhere where no one else in their high school ever did this stuff. And they kind of took the agency of like I think this is really fun. I'm really competitive. I want to compete at this. And they kind of go and do those competitions on their own. Those are when the signal is still positive for this kind of engineer of the future.
>> I don't think I've told you this, but do you know who you always remind me of? Who?
>> Matt Damon and Goodwill Hunter.
>> Oh, that's so I mean I'll take that to the bank.
>> Have you Have you not been told that before?
>> Let me say that one more time. You look identical.
>> That's very We're going to put up like an image here and you're going to do a side by side.
>> All right, we have a clip. This is the clip. This is the This is the starter to the show.
>> And they're going to be like, "Uh-huh. I totally get it. That makes absolute sense." Can I ask you when we go back to actually like what makes devs great and how we think about structuring the team? What role does not exist today that you

think will be incredibly common in the next few years?
>> I think so. It's starting to exist more and more, but I think it's kind of this like GM or general manager like role for someone who used to be an engineer where basically you own end to end an outcome that is not just a shipped feature but like a business outcome. So even at factory we have this now where there are people who um will own the marketing copy if they're going to be releasing something. They'll own the outcomes in the product metrics. They'll own enabling the salespeople. Um, so it's way beyond what a typical engineer does and it kind of feels like again owning more of a business outcome, more entrepreneurial, higher agency, just like spreading their reach.
>> I think that's just like every function. It's I, you know, believe it or not, I sometimes post on different social media platforms. Um, and I just said like my biggest advice to any students here would just be just be full stack in whatever you do. If you're doing marketing, create the copy. Uh, make sure that it's ready to post. Post it at the right time. Amplify it. You have to be in every element from start to finish. Not just, oh, I just do the copy and then I hand it over to designers to create the visuals and then they hand it over to a social team.
>> Is that not just the same for every function? We're expecting everyone to be full stack in every function. The age of the polymath is back. Like I growing up I was so like I was obsessed with math and physics and I was so jealous that in like you know hundreds of years ago

people like Da Vinci or Oiler or Newton could be polymaths and it was because their fields were relatively shallow. So like chemistry wasn't that built out, mathematics wasn't that built out, physics wasn't that built out in Da Vinci's case like art and engineering and um sculpture. Um, and so you could get to the frontier of these disciplines within in multiple disciplines within your lifetime. And then growing up in, you know, the early 2000s and 2010s preai fields were so deep in my case, theoretical physics and strength. It was so deep that you could spend literally 50 years catching up on all of the literature and academia that's existed before you contribute anything new. And so it was like this was infuriating to me because it was so frustrating. With AI, we're now completely the opposite. These tools can get you up to speed to the frontier. Obviously, with a lot of uncertainty about certain details, you won't have the depth of other people, but it'll get you to the frontier way faster than ever before. And so now, if you're someone that's good at thinking around constraints, thinking about systems, holding uncertainty in your head, and being okay with that, like know knowing there are unknowns and knowing that you can still push the frontier forward despite that, you can be a polymath. who can push forward and create innovations on how to do developer marketing while at the same time pushing forward the frontier of like token caching for software development agents at the same time as like you know being an incredible um solution engineer like these are things

that you can now do all at once and so this is something that's very top of mind for me and in our hiring process we want to find the people that can be those polymaths um the era is totally back this polymaths are back I've had a lot of people say on the that agent operations will be like the leading uh function that doesn't exist today that will be very common in 3 to 5 years. Do you agree with that?
>> What is the definition of agent operations?
>> Agent operations is the creation of agents and the maintenance of them. So to be able to go into different functions and say ah social media I'm going to create agents that allow you to create distribute share posts. ah marketing and design. I'm going to create agents that allow you to create visuals, share them amongst each other, edit them, collaborate on them.
>> I think to some degree everyone should be able to do that on their own, but I could imagine a world where there's kind of someone whose job it is is to like find places that aren't as efficient and similar to operations now, like in organizations, but now it's just agentified. So, they're using agents to make the organization more efficient wherever possible. But I think in general, if you have people that in certain functions that aren't proactively doing that, it's probably a bad sign.
>> What do we do today that we'll look back on and go, "Oh my god, I can't believe we did that."
>> I mean, for an engineering team, like writing release notes. That's crazy that

people used to spend hours of time writing release notes or like writing documentation.
>> So, not everyone knows what release notes is. What is
>> So, it's like, you know, uh basically cataloging the changes that you've made in in the last whatever month or so. um you know and sending it out to either internal or external to your users and generally like uh this and like documentation like Stripe has a really great reputation. They had incredible documentation like so many APIs had horrid documentation. Stripe was like the pinnacle. They were so good at this. Spent a lot of time doing it. 5 years from now it's going to be like oh my god I cannot imagine cannot believe that these people that get paid so much money spent hours of their time doing this. I think that's something that, you know, we definitely won't do.
>> Does that reduce the impact of Stripe's great documentation if everyone is equalized?
>> Uh, yes. But I think Stripe has plenty of places that they can differentiate and I think it's a better world where everyone has documentation as good as Stripes.
>> Totally agree with that.
>> How does the product review and especially like code review process changed in the next few years? Yeah, I think what's cool about um this agentnative software development is review has been a big problem because basically uh you know first phase of rolling out AI coding tools was oh my god look how much code we can generate you know it's incredible I'm generating

a ton of stuff and then phase phase [__]
>> yeah phase two was some poor staff engineer who has to review hundreds of these slop PRs that are like don't adhere to your standards are like completely misformatted and all this stuff. Um, but what's great about having uh this kind of like full endto-end software factory as it were is it's now very clear the ROI of investing in things that make your agents more kind of ready for production. So, examples of this are um making sure your agents have access to up-to-date documentation, making sure agents can spin up a remote machine so that they're not just generating the code, but they can actually run it and see what the outputs are and iterate based on that to make sure that it's actually good. Um, setting up things like CI/CD or good llinters or good pre-commit hooks. These are all things that uh the best organizations at like developer experience would invest a lot of resources in. But they would do it because it makes it easier for engineers to work, easier for them to onboard. But the the impact of doing that well is just like onetoone kind of correlated to how many engineers you have. With agents though, the impact of that is now like 10x or 100x depending on how many agents you're using because the better your devx, the better your agent ends up adhering to your standards, which means there's less time that that poor staff engineer has to go through reviewing your PR, which means you're kind of faster throughput in your software development. When agents are the buyers and you're selling to agents, how does

the world change and does the value of great API increase? I think um that value is increasing especially because the thing that makes it easier for agents tends to be the same as the things that make it easier for humans. Um at some point in theory that could change where you know if you're actually training models or training agents to be as efficient as possible communicating to each other. Um but then the downside there is it's not as human readable. But if you think about agent to agent, agent to agent doesn't give a [__] about UI or design, but it does fundamentally care about data structures, um potential integrations, uh documentation, do you know what I mean?
>> Yeah. Yeah. Yeah. So, I think one thing that if you don't have careful um standards in place, it can get bloated pretty quickly. But I think the best organizations who are the most agent native actually put in a lot of guidance on like here's like the UI side of things and how things need to be um being very aggressive about like pruning anything that's unnecessary. making sure there's not like you know bloated like I don't know comments in all of your code that's like kind of gratuitous or um there are ways around it but that's kind of where the human's job changes a little bit where their job goes from I mean part of our name our name is factory um part of why it's called factory is because the future of software development is where these organizations instead of having engineers that build the software they're going to have engineers that build the factories that build their

software visually whenever I say this I always think of Tesla's factories. I don't know if you've ever seen videos of the inside of Tesla's factory. It's all these like robotic arms going and you have the assembly line going through. Um, and there might not be as many humans in that assembly line, but you know damn well that humans uh designed this process to optimize the throughput to, you know, produce more Teslas in this case. Um, and so in this new world of software development, humans, human human engineers are not going to be involved as much in like writing the actual code, but they're the ones that are going to be involved in how do we make sure it's not just creating all this bloat or it's technically getting the job done and passing tests, but doing it in a way that is really dramatically increasing debt. Um, so there's they're kind of like building the scaffolding around this factory that produces their software. Do you worry about labor displacement when we move from working in the factory to working on the factory?
>> Short-term, yes. long-term, no. Short-term, yes. Because it's just a shock to the system where, you know, there are all these big layoffs that are happening that are pretty aggressive and, you know, these are thousands, tens of thousands of people that had a job that no longer do. And so, I think that does worry me. Um, long-term though, I am very not worried because the reality is there is a huge number of problems in the world, ridiculous number of problems in the world. And a large percent of them can be solved or can be helped with

software. And very few of those problems that can be solved with software are we currently solving with software. And so if we are going to be flooding the job market with tons of engineers, that means that we can now allocate them on the broader economy to solve more of these problems in the world. And if we have more engineers who are going and solving more problems in the world, that is a net good. what problem is not currently being solved with software that will be enabled by this new technology because everyone's like climate change and I'm like great you how many people I've found doing climate change technology well
>> none
>> yeah well and maybe part of that is because all like the Googles have been hiring all these engineers so distributing great engineering talent to more problems I think is going to be a good thing the economy has to match though and properly incentivize them and that's something that I think will take a little bit of time which is like the intermediate period but like so many health problems s like so much of pharmaceutical research can be advanced with better engineering and like the thing that really upsets me with some of the people who are talking about you know pausing AI development or any of this like oh it's you know it's a bad thing and it's going to you know harm society. Dementia is kind of a go-to example where everyone understands how big of a deal that is. That is something that can be solved with better AI and better software. Like it's a matter of time like we will solve it and we can

solve it. And by saying you want to slow down AI, that's saying like these people who have relationships with loved ones who have dementia. You're like, "No, no, no, sorry. You guys, you got to maintain that relationship for a little bit longer. We're scared. We don't know about AI." I think it's like it's pretty it's pretty harmful and it's pretty selfish to say that it's something that to me it doesn't make sense. It doesn't make sense.
>> Do you agree with government intervention?
>> Uh in what capacity? In free markets when you think about like the allocation of resources there are times when it is suboptimal from a human morality societal standpoint in a lot of cases to see uh engineers at anthropic working on optimizing claw code when they could be working on optimizing healthare systems or optimizing more critical or missionritical things immediately
>> governments can intervene offer subsidies
>> offer economic incentives do you agree with that or do you believe in Adam this invisible hand.
>> Um, I think it's a it's it's certainly useful in some cases. Like I don't think anyone would argue that the government should never intervene ever in the economy because there are some things especially as it relates to like military uses or safety or things like like weapons like you're definitely going to need, you know, some involvement there. Um, I think there's some incentivization that can be helpful just because uh there's there might be some problems for a society that maybe

capitalism doesn't see the immediate feedback loop of and so you might want to juice the incentives a little bit to get an outcome that you're looking for. But I think generally I'm pretty reluctant. I think you need to have a very good case for why you need to do that. Even like the example of climate change um you know talking about that one um it's obviously a very sensitive subject or a very important subject for a lot of people and you know you could make the case that the faster we develop AI the sooner we solve climate change because AI you know can help us solve a ton of these problems but to develop AI faster you might need to consume fossil fuels and emit them and you know that emits CO2 into the atmosphere and so the question is like you know short term it might be slightly worse but it ends up getting us to solve the problem way sooner instead of dragging it out over 50 years or 100 years. And so there's some of these cases where the natural kind of free market will incentivize it the right way and there's some cases where it won't. But I think you need to be very very careful about the cases where you do want the government to say, "Hey, we want to step in here."
>> Do you think we are in an AI infrastructure bubble?
>> Maybe there's like some short-term blips, but like long-term absolutely not. Like not even close. I think there might be there might be similar corrections to like this thing at Uber where oh we were going a little haywire. We weren't allocating it appropriately and it's like okay let's lower consumption a little bit but like on the

net absolutely not.
>> What bottleneck do we have today that will be completely solved within a few years?
>> I think the biggest bottleneck by far working with all these organizations is the human side of it. It's just like behavior change like especially if
>> what you're saying there is like is selling into large enterprises and how they do change management.
>> Yeah. or even on an individual level like if you're an engineer who's been an engineer for 30 years it's hard to change those pattern like you're stuck in your ways to a certain degree. Um but there's also a funny thing where some of these engineers who have been engineers for a very long time or who have been engineering managers they might be more reluctant to use these tools but sometimes they're better because they know how to delegate. They know how they know how to deal with some of the like junior engineers where if you tell them the wrong thing they're off in the cave doing the wrong thing for seven days they come back with something completely useless. And then on the other end of the spectrum, there are people earlier in career who don't have as much of a standardized workflow that they're used to. So they're more eager to adopt these new workflows, but they don't know how to manage people. They don't know how to delegate as well. So there's kind of an interesting balance there. When you look at now, you sell to some of the largest enterprises in the world in some cases. Um, what do you know now about selling to large large enterprise that you wish you could tell young Matan 2 years ago?

>> So this is the first job I've ever had, which I think is always a funny thing to say. Um because prior to this I was a theoretical physicist. Literally never never like coffee shop any of that. Literally never have had a job. Like never have been paid to do anything aside from physics until this which is it's a whole separate thing. But
>> I'm Matt Damon.
But I will say the thing that has been the craziest learning and this is obvious to anyone who's in sales or like Chad and Chris, you know, to them it's obvious to me the thing that was the most kind of visceral
altering thing was um meeting people face to face makes such a big difference if you're trying to sell them something. But also, you should never try to sell something. You should always try to understand their problems and see if the solution that you might have can actually help them solve that problem. That's another Like if you go in a conversation trying to sell something especially to engineers like don't waste your time. if you go and trying to have genuine curiosity about and it it's really easy because these organizations do their engineering so differently and I find it fascinating how like all of these different banks you know consulting firms pharmaceutical companies they have the most different ways of building software and it's really interesting to go talk to them and to understand it and the best way of talking about it with them is face to face and people love talking about their problems and they love talking about all of the bureaucratic nightmares that they

have to deal with and then By understanding all of that, you can actually get a sense, you know, is our software a good fit for them. Will it help solve their problems? Um, it's also just so fun to then like meet up with them a year later and be like, I remember when you had to deal with that [__] and now you don't have to. And that's just such a rewarding feeling of, you know, making their lives better in that way.

>> In terms of like being there in person and the sales process, you got Sequoa very, very early on. Sequoa obviously one of the best and most prominent investors. Can you just tell me the story of how you got Sequoa having never had a job and only being paid to do physics?

>> Yeah. So, I was obsessed with physics basically since I was 12 because um I was a bad student. Uh and my geometry teacher told me that I had to retake geometry in high school. And like I never tried in school, but I always prided myself on being good at math. And when she told me that I was like, "Are you kidding me? She thinks I need to retake geometry? Like I'll show her." And so my first order on Amazon ever was textbooks for algebra 2, trigonometry, pre-calc, calculus 1, 2, and three, uh, differential equations, um, and maybe a linear algebra textbook. So I bought those textbooks, and then the summer between middle school and high school, I studied all of those, like did all the problems in all of them. Um, and then in high school took exams to place out of all of those classes. Um, and then I asked my dad what the hardest math was.

He said string theory, which is technically physics, not math, but I was like, okay, I'm going to be a string theorist. That was literally all I cared about for basically the next 12 years of my life. All I cared about was math and physics. Um, ended up going to uh to Princeton because they had a great physics professor I wanted to work with. Um, he's uh this famous professor named Juan Maldisa and I was like the first undergrad to work with him and write a paper with him. Then I ended up coming to Berkeley to do my PhD and you know work with a great adviser there. And then only at Berkeley I realize like it kind of all comes crashing like holy [__] I've just been doing this because it's hardened because someone said I couldn't do it. Like what the hell do I do with the rest of my life? Like this is crazy. Like everything came crashing.
>> Of course that crashing down moment. And why did it take so [__] long?
>> 12 years.
>> 12 years. You're slow.
>> I have I have tunnel vision. When I get obsessed with a problem, it is all I think.
>> I was in law school for 2 weeks. It was a quick quick realization.
>> See, some people are faster. You know, I wasn't as I wasn't quite as quick. Honestly, part of it was being a uh as part of a grad student at Berkeley, you have to teach classes. Um, and I was teaching a class to like whatever 18-year-olds who didn't give a [__] about physics. And I was like, "Oh my god, this would literally be the rest of my life." Like sitting and doing lectures and doing these classes on my

professor. I think I had a one out of five. I was like horrible. Like I didn't Yeah, it wasn't it wasn't a good fit. Um, but it was kind of this existential crisis like what what do I do? And so um you know kind of realized it was probably going to be either quant finance which is what a lot of math and physics people do um big tech or startups. Um, I ended up doing the quant finance interviews like every good physicist does. And you know, almost took it, almost went to New York to do it. And then last second, I had a a adviser that I spoke to who was like, you know what, stay at Berkeley forbid, don't do it. You're always going to be good at math. You could always go and do quant finance. Stay at Berkeley. Explore, learn some stuff, whatever. So, I was like, okay, fine. You know, I'll I'll do that. You know, so ended up taking my first CS classes at Berkeley. I learned like to code for physics for like simulations and all this stuff, but never in a formal class. And I'm very competitive. And I found that in these classes, I was doing better than some of the CS students, which was very competitively satisfying. I was like, "Oh, okay. I'm going to take more of these." And then it wasn't until I took a seminar um in what was called program synthesis at the time. Now we call it code generation. And it just completely nerd sniped me because the idea here is not machine learning for video or audio or images, but it's code with the explicit purpose of creating itself. And there's something just so fundamental about that. And a decade of physics like physicists and mathematicians, they're

never interested in the case of like n=
3 or like n= 4, four dimension. It's
always like what is the n-dimensional
solution? What is the arbitrary, the
fundamental, you know, solution to
things? And there was something so
fundamental about this idea of like code
generating itself and it just got me
obsessed. So I stayed at Berkeley and
for the next year that was kind of what
I spent my time on. My adviser was very
chill and just allowed me to just you
know take AI courses. Um and eventually
I realized that the way to actually
solve this problem was not in academia
um but in the industry um and to
properly solve it in the industry you'd
have to start a company. Um but I knew
nothing about starting companies. So
because again all I cared about was math
and physics. Didn't know anything about
this. So, what uh does someone who wants
to learn about starting companies do?
Well, they order on Amazon Peter Teal's
0 to1 and they look up on YouTube how to
start a company. And so,
and so, uh, you know, read 0ero to1,
incredible book. I know it's so cliche,
but like to someone who didn't know,
growing up in the Bay Area, shockingly,
I just like did not care about any of
that. And reading this, it was like so
concise, beautifully written, all that,
you know, loved that. And then, you
know, was watching these videos, a lot
of them like Y Combinator, you know,
videos and all this stuff. And then I
stumbled upon this uh I think it was
like a Stanford VC club podcast um with
this guy whose name I recognized because
at Princeton when I wrote that paper
with Juan Maldena, I had cited one of

his papers. So, it was a a theoretical physicist. Like I remember this guy's name, but he's on this podcast talking about how he sold a company for a billion dollars and was a investor at this place called Sequoia. And he also in this video seemed like pretty sociable and normal, which I don't know if you've interacted with.
>> I'm not sure, dude.
>> Theoretical, you know, compared to theoretical physicists though, like he can maintain eye contact, you know, he was somewhat normal. Very rare. That's such a low bar.
>> He can he can hold himself in a social setting.
>> Yeah. And so I was like, okay, who is this guy? You know, I got to talk to him. So, I ended up writing him an email being like, "Hey, I'm Maton. I also used to be a physicist." I wrote a paper with Juan. Didn't say the last name cuz it's like if you know, you know. Um would love to get your advice. And you know, he responded that day and invited me down to Sand Hill and it was supposed to be a 30-minute meeting. Um but we end up going on this walk and it ends up being a three-hour walk. Uh and on this walk, it turns out we had very similar reasons for getting interested in physics, very similar reasons for leaving physics. And at the end of it, he was basically like, you know, it was great to meet you, Maton. You absolutely need to drop out of your PhD and you should either join Twitter right now because Elon just took over and it's hardcore like for your resume if you like voluntarily go there or you should start a company. And I was like, okay, thank you so much. I

appreciate you taking the time. Like, I'm going to, you know, I'm going to go think about it. But in the meantime, I like had already known about factory. It was just I didn't want to ruin the meeting with like a pitch. Um,
>> you didn't want to transactionalize this.
>> Yeah. cuz it was so like it was incred like we had the exact same reasons for getting interested
like didn't want to didn't want to dirty it with
>> No, it's like kind of like an LP where at the end you're like I don't want to ask for a track.
>> Yeah. Exactly. Exactly. And and then the crazy thing the next day I go to a hackathon in San Francisco and see across the room this guy who also went to Princeton who I like recognized but I didn't like know super well. End up talking to him. He's also interested in this problem. We like we joke that it was like intellectual love at first sight. This is my co-founder, Eno. Um, and basically that day forward, we spend like every day talking non-stop. I had some shitty demo that I built. Eno is thousandx better of an engineer than I ever will be. Um, and so he and I for the next like 72 hours like put together this better demo. And then I call up this investor and I'm like, "Hey, I have something cool I want to show you." So we hop on a call and I show him this demo and I'm like, "What do you think?" He's like, "Eh, it's okay." I'm like, "Are you [__] kidding me? This is going to change the world. what are you talking about? He's like, "Okay, well, would you work on it full-time?" And I

was like, "Yeah, absolutely." He's like, "Okay, drop out of your PhD and send me a screenshot." And keep in mind, like, my parents immigrated from the Soviet Union to the United States with basically nothing. The the fact the fact that I was doing a PhD to them was like their pride and joy. Like, it was the thing that they were the most proud of. Um,
there was so much momentum. So much momentum. I, you know, I he answered my email. We got along well. I met the co-founder the next day and I was like, "You know what? [__] it." Dropped out, sent him a screenshot and he was like, "All right, you have a meeting with the Sequoia Partnership tomorrow morning. Be ready to present."
>> You You've never presented to a Venture Partnership before.
>> No.
>> So, so what happens?
>> I made a shitty deck.
>> We We go to the Sequoia HQ, put some slides together.
>> Keep in mind, I didn't even know who the hell they were. Like, I didn't It was just like, "Oh, these random people like, "Okay, whatever. Yeah, I'll go talk to them." I wish it was recorded. I wish it was recorded because I'm sure I came across as so arrogant. How did it go? I thought it went fine. They asked some questions. I think I was pretty uh again I didn't know anything about VC land or startup land or any of that stuff. So like retrospectively, I know like Alfred and Pat and Rolloff, they were all like in there. They're all asking questions and I was probably dismissing some of oh yeah, we'd solve

that easily. We do this, we do that. Um keep in mind this was in uh April of 2023. So, this was like way before anyone was thinking about agents, way before people were even using co-pilot. We were talking about fully autonomous software development agents. Um, and uh, it was kind of a blur. Uh, and uh, you know, the next day Sean calls me and he's like, "Hey, we want to give you a check."
>> How big was the check?
>> A million dollars.
>> A million dollars.
>> And you know what he gives me [__] for? You know what the terms?
>> Five post.
>> Five post.
I mean,
>> what? I'm not being rude. Why did they bother doing a partnership meeting like in the nicest way? That's like a coffee. Like, I know it's a dick comment, but like when you managed like seven different time, it was a different time. Early 2023 was a different time.
>> 20% post.
>> Yeah.
>> Just for for you know, listeners, I mean, on on last funding round, that'd be like a $300 million position, not including dilution. And it was one of those things a lot of my a lot of people I spoke to were like, "You should go shop that around. You can get better terms because it's Sequoa." And it's just like when you have a connection like that, there's a certain thing to me where like obviously you want to maximize, you know, the the uh position for the business, but like no one else would have believed in me except him. No

one else would have understood like I literally had never had a job before. No, like no other partner I would have met. Retrospectively, it's like oh yeah, what no one else would have done it. And it's one of those things where like trust and loyalty and like belief to me that matters so much more than like the price tag you get or whatever. I want to make sure that the people that I have in my corner because we're building a legendary company. It's not just going to be 10 years. This is like a lifetime. Would you tell founders to take a discount for Sequoa? So generally yes. I mean they're the best firm in particular if there's like a special connection with you and the partner or there's a special reason why them in particular. But I think what really matters is you want to have people that are there for you when the days are tough and when it's not obvious. Because when you're a hot company raising a hot round, everyone's your best friend. Every it is their job to make you feel special and they are really good at it. Well, it's the best way someone's tried to we just some people I don't even know. I don't I don't want to name names. There's this one investor in particular who's like still in the game but more of the old guard. I'll say that much. Um, and I remember beforehand people were like people told me like, "By the way, he's really good at making you feel good about yourself." And I was like, "Yeah, whatever. I'm I can deal with that. That's fine." And then I remember leaving the meeting being like, "I'm the [__] man. Like I am like this is my destiny. I'm going to build a legendary

company. Like I got this." And then like 30 minutes after when it wore off, I was like, "Oh my god, he got me. Like he did it. Like he did the thing. He made me feel special." And like a lot of investors when a company is hot are going to do that and they're really good at it. That's why they're great investors. Um I think for me what's really important as we've built out our board in particular is people who have like deep conviction when it's not obvious. Like that's what really really matters because when a company's hot everyone's going to be excited. Um it matters when it's not and there are going to be tough times. How how do they behave then? How did you get Ivanka Trump as an ambassador?
>> Through so one of the best hires that I've ever made at Factory was uh this woman Franchesca. And so the way that Francesca and I met um was at a random conference I was seated next to her and Alex Paul who's one half of the Chain Smokers. And obviously people know them as the Chain Smokers. They're also incredibly good investors. Incredibly good investors which sometimes people are surprised by. Um, and you know, we got along quite well. And um, and weirdly enough, Francesca and I also grew up in the same hometown, which is a whole and had a ton of that was another kind of weird coincidence, but um, like just in the process of like them wanting to put a check in and the way she did diligence and just the way that she kind of carried herself was so clear like she was a killer and they wanted some allocation. I was like, "No, no, no, sorry." Like, you know, it's going to be

this. And she was [__] relentless.
Like came to our office like was like,
"Hey, like we need to get to this much.
How can we do it? I'm going to make
these. If I do this and this and this,
the business value that we provide to
you is going to make it worth this more
so than giving that allocation to
someone." She was like kind of hounding,
you know, we were having a conversation.
I was like, "Look, Francesca, like if
you want more ownership of factory, you
could just join us." And he was like
kind of as a joke. I was like, "Oh, you
could just join us if you want more.
Like this is the highest we can do." But
then we kind of both were like, "Oh,
interesting." you know, we talked about
it a bit more and then realized, wait,
this is an incredibly strong fit. And so
we ended up bringing Franchesca on
board. Alex was uh, you know, it was
kind of tough because she was incredible
and they were very close. You know, he's
since been happy because she's helped us
deliver a lot of, you know, returns for
them and um, you know, we're we're the
biggest fans of theirs and, you know,
kind of we still have a very deep
relationship. Um and she was very close
with their firm uh affinity um from her
investing days. Then you know we were
introduced we got along really well um
and so then that was how the connection
was made there.
>> Does Ivanka Trump provide value? People
will look at him and be like h branding
just a name whatever. And I don't mean
that disparagingly at all. I think
people often think that with kind of
famous celebrity names.
>> Does she actually provide value?

>> Yes, she is. First of all, she's one of the kindest and smartest people that I've met. And like there are people that you meet that, you know, are famous that are kind of like a let down or like, oh, they're different than I expect. She is genuinely so kind, so intelligent, and like people just in throughout tech, throughout the world really love her and for good reason. And she has an incredible network. She's so generous with her time. like there is like kind of dirty work, investor help that she helps out with that some other investors who are more known as investors do not do. And so
she and the firm more broadly really earned that right um on the cap table.
>> That's really good to hear. I I hate the statement. I I I'm not sure if an was quite your hero. Uh but like people say never meet your heroes, they always disappoint. And I think that's just total [__] Yeah, I remember meeting Doug Leone, who was one of my heroes, did not [__] disappoint. Like I laugh being more like, "God, he should have been even more of like a a poster boy for me cuz he was so great."
>> So I I I totally agree with you that that's very funny. I would love just your thoughts on some market composition that I'm struggling with, which is like when you look at cognition, you look at claw code, you look at codeex, you look at curs now with grock,
>> how does this market evolve and mature? Is this an AWS is your GCP? Is this an Uber lift? What is the mature state of this market?
>> Yeah. So, I think what is necessary for the best outcome for the consumers is

going to be models that are separate from the applications. You as a consumer do not want to um use applications that are provided for you by the same people that are giving you the model because the incentives are misaligned. We get optim
>> the incentives are misaligned. Why? just
>> because if let's say uh the example of coding like if I'm a model provider and I'm working with a large enterprise and I'm giving you a coding tool I want you to use as many tokens as possible because I'm an API business and I get more money the more tokens you use um and I don't have a huge incentive to be more token efficient um other than like you know yeah I want to give a good product experience but not strong incentive um versus if you have model providers and you have an application layer that allows that enterprise to decide between the different providers. If you're a model provider, you better damn well be the best or the cheapest or the fastest or else you'll never get tokens through to you. So, it puts the best incentives on the model providers. There's that independent um agent in our case there. Um and then that gives the best prices to the enterprise. It also gives them the best in terms of like if one model is really good at this language or that language, it allows them to kind of adjust between them. Um and in the world where there you're like vendor locked in then you can slowly get like laziness and slower shipping and uh and as a you know consumer you end up getting a worse experience. Okay so it's not good for the consumer if the model is tied to the application.

Okay cool
but bluntly we are seeing codeex and claw code eat a huge part of the market. What does the market look like in three years in terms of market maturation? This is going to be different from cloud. I think cloud a lot of people suffered because, you know, the cloud providers came and said, "Hey, look, sign this three-year deal. We're going to give you a big discount. We'll get everything good for you. It'll be all right. Come on in." And then they would do that and then they would jack up the prices and once you're standardized on one, it's going to take you 2 years to switch to something else. So, good luck. You're stuck with us and we're going to charge you more. Everyone has scars from that. So, now every CIO I speak to is really keenly aware of we cannot, you know, throw our lot in with just one model provider. we're going to need to be agnostic. And so, you know, you could be agnostic by saying, "Hey, every engineer, we're going to give you cloud code and codeex and Gemini CLI and all these other tools." Um, but then the problem is now you're asking your engineers to use 10 different tools. Or you can use someone like Factory where you can use one tool and you can kind of decide um kind of like in an auction on a task bytask basis which model provider do we want to use? Do we want to use an open model? Do we want to use Frontier? You know, which one of those? How can you help me understand the paradox of hey we need to be more cost efficient with replet we're going to run the same prompt on three models at the same time
>> and I don't mean that there's no

diminishment to replet that's like them providing a great product but also I haven't seen them or that use case as much in the enterprise I could see for maybe consumer use cases where you're not as cost sensitive because you're not doing things at crazy scale where it's kind of fun to see oh I wonder what Gemini does versus open eye versus anthropic um and you know for some enterprises if there are things that are like very sensitive or very secure Sure, you might want to do that, but for a lot of like if you're a nontechnical person building an internal dashboard, you probably don't need 10 different models to generate different iterations of it. Totally get that. In terms of the market maturation,
what happens to the lovable and rap market? We saw open eye kind of release a competitive product last night.
I just don't know what happens that. Can you help me understand it? It's not obvious to me. Um, and part of it is because not too many people that are close to me use those tools frequently. Like most of the people that I know either don't use like AI tools or they're like technical and using factor. Also, like I'm not going to be no none of my friends don't use factory. Like what do we come on? We wouldn't be friends. Um, uh, so I need to understand a little bit more about that user. My sense is they're probably, and we're still in the early innings, so I'm sure they're quite agile to figure out what is the exact niche that they want to occupy. Um but it's not super obvious to me what the kind of focus is because my understanding is some of them have been

pivoting towards the enterprise a little bit. Um but I think from the enterprise perspective the like I think they've been pivoting towards the enterprise in non-developer centric functions. So like hey if I'm uh lovable of the world I'm going to sell to sales teams marketing teams uh customer support teams to allow you to create amazing uh materials
>> uh with no experience developing.
>> Sure. I mean, in that case, I think that that niche does make sense a little bit more. I think the if I think it would be ill- advised if they were to try and go to the niche of non-technical people writing code for code sake because I think that is going to be run by like if you're going to need enterprise controls over who has access to what databases and what code and all that stuff that's going to be run by the engineers. That's going to be where factory goes. If it's things like, you know, if a salesperson wants to build a customized demo app or customized website for something, I could see in some cases that that having some value there.
>> Are we entering a danger zone for security? A huge amount of net new code created that may not be as secure as previous and we're seeing just the worst hacks, security leaks, and this is just the start.
>> Yes. Yeah, it's going to be crazy. Um,
>> when you say it's going to be crazy, like what does that actually mean? like the amount of code that's being generated. The uh I think code generated is growing exponentially. The security efforts aren't growing in kind and I so I think there's kind of a lag there. Um I think there probably going to be in

the next couple years some pretty big incidents that occur because of generally probably have been. I just whatever incidents that have occurred no one's going to admit or typically they'll be reluctant to admit um if it was like AI involved or not. Um, but also I think we haven't even seen the most adversarial behavior yet. Like I think people can use these tools to be quite adversarial and so I think security like the higher the stakes it's going to grow in importance and so I think the security part of the market is really important.

>> Do you think US startups should be allowed to operate so extensively on Chinese open source models?

>> Yes. Um, using an open model is fine. I think the like there are kind of two concerns. There's one concern is if you're sending your data externally to like a different nation which is one concern and I think that the concerns there about like we don't want to send our data um to China or generally I mean you should probably want to keep your data to yourself regardless or like within country regardless but I think the the separate concern is like oh the model itself like even if we host in the US is their concern with the model itself and to explain some of the concern there I think the idea that some people have is like I don't know if you've seen in like those spy movies where there's like a code word where suddenly someone starts acting like you say the right word and then they're like in robot mode where they're going to go act adversarily. I think the concern is that some of these models might secretly

have that ingrained within where you say a trigger word and then suddenly even if it's hosted in the US it's going to like send data somewhere else or it's going to start you know trying to intentionally kind of break whatever it is that you're doing. Um suppose any nation were to try and do that. Suppose they wanted to make a model that had one of these trigger words that's going to go and act adversarially.
Theoretically, you would want to do that as late as possible because if you do that in an early model and someone discovers it, they're literally never going to use your models ever again. I don't see that as a big concern. And also, if you're deploying correctly, like not as a consumer, but in the enterprise, if you're deploying correctly, data exfiltration or like kind of some of this um adversarial stuff generally you can fight against. I do think just from a you know, I'm quite patriotic. I think it's pretty embarrassing that we don't have frontier open models in the United States. Um, so I do hope to see us, you know, reclaim superiority there.
>> Europe is significantly behind, especially on the model development side. Do you think Europe is too far behind to catch up?
>> Probably on the like frontier model lab side
on the like there's so much to do on the like infra buildout and energy side of things. Um but again the thing that's very difficult uh in the different parts of the world is you have like democratic countries where things generally are slower. Suppose you say we need to do

this thing you need to get a lot of support you need to convince certain people to do things you need to pass legislation it takes a long time. Um but the benefit though is you know theoretically we get this balancing act where we don't go too crazy in any which direction. Um you know other parts of the world where it's more authoritarian is like this is the thing we're doing we are doing it we're acting now. you get to move quickly. Now, there's less kind of correction because what if you're going on the wrong course? But in cases like AI where it's pretty clear like for buildout, you need to build data centers, you need uh energy. Um and energy requires a lot of buildout as well that has a huge amount of lead time. You can act faster. In the west, things are slower. So, that's one thing that kind of goes against us. It's a little bit slower to get this stuff done, especially when there's all the politics that you have to deal with. Do you worry about the public backlash to data center development that we've seen? I think it's like 40 out of 100 data centers post approval don't actually get built out in the end. Do you think data centers will be seen as a symbol of wealth concentration and technology superiority?
>> Yes, but I think that's at least in the United States the beauty of having states uh is we get some like selection where we can have different experiments of like what's it like for a state that says no to all data centers. Well, okay, there won't be as many jobs that get created there. versus the states that do allow for data centers to be to be

created, you know, people will prosper. They're going to have great jobs. They'll, you know, see the the downstream benefits of it. Um, but it's nice. It's like we have little petri dishes to test out and see how things work. That is the beauty of the United States. Um, and I think in Europe, I mean, it's a it's tough. I think the there were some there was some good positioning that Europe had, you know, a few years ago, a few decades ago with nuclear that I think hasn't been, you know, delivered on as much as of late. But that would have been a world in which Europe would have a way to bounce back a lot in AI on the on the energy side.

>> Well, 100%. I blame the Germans.

>> And that's our German audience gone. Um, dude, I want to do a quick fire around with you. So, I say a short statement, you give me your immediate thoughts. Nebus versus Coreweave. Who has a larger market cap in 5 years time and why? to me and this is this is speaking from strongly biased as an application uh person like I'll take the grab bag it doesn't matter as I actually hope for a world in which I don't even our users don't even know which one is is under the hood

>> for you I would want core weave to be bigger

>> why because Nebius I think have more ambitious plans to be full stack which will eat into some of your plans in a way that corewave don't

>> ambitions what are ambitions I don't think it makes sense for them to do Okay. Um, go businesses need to think about their core competencies and like

if people are trying to expand beyond their core competencies, Kirkland and Ellis, great. Look, have fun. It's not your core competency. I don't think it makes sense.
>> Yeah, I get you. I think you could argue that it's a lot more adjacent there. But I get you. Do we have a series of businesses like a Nebius, like a Mccor where customer concentration is like 90% of revenues?
>> Will we see more of that?
>> Yeah.
>> Yeah, probably.
>> Is that a bad thing or a good thing?
>> Um, it's bad if you're an investor in one of those companies cuz it's a little riskier, but I think you can find a steady state. It's just scary. You just know there's kind of a sort of damic above your head of like if they ever, you know, it's just risky. It's risky. a sort of dam first time it's ever been said on the show. Um, tell me, can you sell to enterprises today without an FTE model? Yes. Have a good product. This the thing about the FD thing blows my mind is like you like for us when we do FTE like the way I think about it is their goal should be acceleration. It basically if there's a customer where if we just give them our product they'll scale to like uh a million in 6 months. I'll throw in FTEEs if they're going to scale them to a million dollars worth in three months. Great. They accelerated that. If I'm sending in FTEEs as services, like I'm not Accenture here. Like I'm not trying to be like or Infosys or Cognizant or whatever. Like we are not a services company. If we need FTEES to make the product work, we

have a [__] product. Like the point of FD should be accelerate and get them consuming faster. If you're putting in FTEEs cuz that's the only way you'll get a deal done. I'm sorry my friend, you have a [__] product. What do you think of the whole grind slop uh element? We we talked a little bit before about the show with Nico at Corgi which generated a little bit of discussion online.
>> Yeah, just a little bit.
>> Just a little bit of discussion online.
>> You're ruffling feathers as always.
>> Dude, I said nothing. This is honestly it's like it's like someone comes to your party and does something well. It's like that's me.
>> Um what do you think of the grind slop? I feel like a lot of the things we've talked about actually is like something everyone needs to be wary of is intermediate metrics and grind slop comes from intermediate metrics like oh you know generally to do things you need to spend time on it so let's focus on how much time do we spend instead of like are we doing the thing right like the analogies I use is uh imagine trying to measure who won a basketball game by who sweat the most like you could sweat a ton but look at the scoreboard like are you doing what actually needs to be done or not and I For us, we want to focus on like getting the best players. I don't care if you sweat a ton or if you sweat very little. If you're scoring a lot, great. We want you on our team. Um, now generally for most people, you have to sweat if you want to, you know, get things done. Um, but I think you are doing a bad job on hiring if you need to like mandate certain crazy hours or you

need a bed in the office. It's like, dude, get a good night's sleep. Like, you don't need a bed in the office. Like, just go get a get an apartment nearby that's nice and cozy. get 8 hours of sleep. If you as an important member of your team at your company can get your job done on 2 hours of sleep, you're not doing very high leverage work.
>> But do you know I did think it was an amazing opportunity for eight sleep to do like an amazing social campaign. I would have delivered it. I would have got the founders outside being like we got you covered.
>> It's funny like that's literally like we wouldn't that be funny when we were when we were 30 we had like a 30 people. We had like a what we call a surge like a pretty aggressive like two week sprint. Um, and as part of it, I got everyone on the team eight sleeps, like fully free, whatever, $3,000 per person, like you know, the decadence of startups, right? But I think that the idea there is like
>> we are optimizing for output and
>> the people that we are bringing onto the team, it's like Seal Team Six, like the NBA Allstars, like it is worth every dollar to make them more productive and, you know, to deliver on these ambitious goals that we have. And so we can do that and you know this at least for me I think eight sleep helps with my sleep engineer like great let's do it let's they're going to be better they're going to have more of their wits about them they'll be sharper and the type of engineering work that we do is not just like grunt work how can we spend as many hours to do it we have droids for that

the work that we do is like might require like really deep thought really kind of like every ounce of brain power that you have in which case if you didn't sleep well like you're not going to make as good of a decision
>> if I gave you unlimited money. What would you spend on today that you're not spending on?
>> I think gen generally um we will see the best companies treat teams more and more like whatever seal team 6 or NBA like professional athletes. Not in the way that Google did it with like oh you get like a bounce castle and all this like weird [__] but like where like athletes it is kind of like it seems like they're getting pampered but it's kind of a burden like your diet is monitored. you get like you have to do your like hourlong massage after a game to make sure your muscles are recovered for the next game. You have to do like an ice bath and all this stuff like it seems glamorous but sometimes it's not. I think spending on that type of stuff but obviously in the more like intellectual domain I think that's what more and more companies will do. If I could spend an incremental dollar to uh make every person sleep that much better, recover that much better, be that much better at making decisions, it's probably worth it.
>> You're such an American. Do you know what I like? I like lemon cello. You know what I like? I like smoking. Do you know what I want to do? I want to sit in the sun under the intense vitamin D rays and I want to take in life with my friends. Yeah.
>> And you guys are like optimized to the

extreme. Did you see the Steven Bartlett video the other day? You might not know this guy Steven Bart. He said, "I drank three two glass of wine and it ruined three days of my life."
>> Yeah.
>> Because I didn't sleep and then the next day I ate more. I podcasted worse. I didn't go to the gym and then I slept badly again and three days ruined. Okay, honestly, I get that to be fair. The first year at factory, I would drink a whiskey every night. And my argument was, and you'd probably agree with this, was like for robustness, like if you want to be a robust human, you can't have like one drink, you know, ruins the next 5 days of your life. Like the wind blows and then you're like you're ruined, right? So, to some degree, I get it. Like, you want to have some of this stuff. Um, but I also think maybe, you know, again looking to athletes, what they do is they have inseason and out of season. And maybe it's like when you're in season, you're [__] locked in. You're not drinking. You're like optimizing all this stuff with your eight sleep. And then, you know, take a week off, go on the beach, drink some, you know, mojitos or whatever the hell people drink on the beach.
>> Well, out of season, you Charlie Sheen.
>> Yeah.
>> If you can recover after, you know, to each their own.
>> Oh god, that would be the funniest thing ever.
>> Work hard, play hard. Yeah. Uh, okay. You can invest in one company uh on IPO day. Sorry, dude. Anthropic or Open AI?
>> In my mind, the answer here is I think

they're approximately equivalent. Like to me, it doesn't really matter. The biggest reason that affects like the EV is like volatility of the company. Um, that's the only like because I think from the business perspective, they're both very well suited and like kind of well positioned there. Um,
>> so you were saying anthropic
>> probably just past is an indicator of the future and like there's just been more like random chaotic turbulent events at OpenAI. Um, but like from a business perspective to me that's like the both great choices but anthropic. Okay, we got that. Uh, good. Uh, can I ask has Dario done a miss or disservice to the ecosystem by saying we're going to take your jobs? We're going to take your jobs. We're going to take your jobs.
>> Yes, it actually this like really upsets me. On one hand, I maybe just implied anthropic there, but on the other hand, I think that has been not only like disingenuine and wrong, but it's like really hurt the psychology of a lot of people, developers, like just people in the world does AI uh a disservice, does the world a disservice cuz this is again talking about the use cases that are going to the problems that will be solved for society. This feeds fuel of like we should slow down AI, we should stop doing it. And honestly, it's for selfish reasons that they did that because if you're trying to raise unprecedented amounts of money, you know, hundreds of billions of dollars, whatever, the best way to convince people to do that is to say all of capitalism is gone. The only company

that's left will be me, so you better give us your dollars. And then suddenly when it comes to IPO, when now suddenly all the humans and the people that you might be replacing now have money that you want them to put in your IPO, then suddenly it's whoa, whoa, whoa, oh no, humans are pretty important. There are going to be jobs again. You know, we like you guys. That that pisses me off.
>> I totally agree. And what's ironic is the ones who've never said it are the ones who've never needed the money. When you look at a Zach or a Damis, they've always had a very different stance to Sam and Dario when it comes to labor displacement and jobs.
>> Yes.
>> It's really interesting. The ones who need it and the ones
>> I mean it's it's just it's a shame because like for all the philosophizing about AI and intelligence and all this stuff, it's like incentive is driving the outcome and the incentive is I want to raise a lot of money. Which legacy company do you think has most embraced AI? Well,
>> honestly, EY, the accounting firms, is one of our largest customer. I know that's you said shocking.
>> N it's [__] shocking. You win.
>> They are so agent native. It's crazy. They're one of our largest customers.
>> They just push it down the organization.
>> They're just like basically they saw what happened with the cloud. They saw scars of being like late and kind of not jumping onto it aggressively and they have some great engineering leaders there who are like look this is going to be scary some people are going to get

upset you know it's not going to be the easiest thing but we are going to make our or agent native if it's the last thing we do and they were honestly pretty early to it as well to me I think that's one of the most interesting things seeing is like they're more agent native than some like startups which is wild
brave new world brave Brave New World. Uh, final one. What have you changed your mind on most in the last 12 months?
>> What I've changed my mind on the most in the last 12 months is there was a brief period of time where I thought it might be just one or two companies that run away with being kind of the frontier and the best. What seems pretty clear to me is it's probably going to be at least four that are going to probably be approximately as good. And that is a win. Like that is the win for humanity. Like the bad case for humanity is when there's one that's really really good. Like I think there's probably going to be at least four if not many others. Um and that's something that it seems there's like kind of growing evidence of which kind of my sense is it's a hot take because I think right now people are a little bit enamored with maybe one or two. But
>> listen Matt Damon, it's been so wonderful to have you on the show. I'm going to let you go back to Robin Williams. Uh
um and the show was brought to you by Eight Sleep. Uh
kidding, dude. It's been so much fun. Thank you so much.
>> Thank you for having me.