The Two Harvard Dropouts Who raised $800M to take on NVIDIA

92 MIN · YOUTUBE · HTTPS://WWW.YOUTUBE.COM/WATCH?V=BAGWRGPWW1O
https://www.youtube.com/watch?v=BagWrgPww1o

SUMMARY

The discussion centers around the ambitious journey of a semiconductor startup focused on building advanced AI chips for inference, emphasizing the challenges faced by young founders in a traditionally conservative industry. The founders reflect on their early struggles to gain credibility, secure funding, and develop innovative technologies that could outperform existing solutions, all while navigating skepticism from investors and industry experts.

- *Inference is projected to become the largest market globally, with the most successful companies producing the highest number of tokens.*
- *Early skepticism from investors and industry veterans highlighted the challenges of young founders in a field dominated by experienced professionals.*
- *The startup's unique approach involves building a complete inference solution, including chips, racks, and power delivery systems, rather than just focusing on chip design.*
- *Key innovations include low-voltage inference technology and cluster-scale memory, allowing for higher throughput and efficiency in AI workloads.*
- *The founders faced significant hurdles in raising capital, ultimately needing $100 million to scale their operations, which they achieved through persistence and strategic networking.*
- *The company emphasizes the importance of vertical integration and rapid iteration in product development to maintain a competitive edge.*
- *Future advancements in AI models are expected to leverage the increased efficiency and capabilities of new hardware, enabling unprecedented levels of concurrency and productivity.*
- *The founders envision a future where AI agents significantly outnumber human workers, fundamentally changing productivity metrics and economic structures.*

We know inference is going to be the biggest market in the world. Whoever produces the most tokens is going to be the most valuable company in the world. >> We we had people quit. >> Yeah. >> That people literally were like, "This problem is unsolvable." And best of luck, guys. >> You kind of have to be sick in the head to join our company. You're going to convince your family to move to San Jose for the semiconductor company run by two what 24 year olds now going against the biggest companies in the world with a design that they're saying is not going to be like 10% better, but it's going to be 10 10x better. we need $100 million to do this. If we do this, we think this could be one of the most important companies of all time.

>> You're sitting in that moment, you think, holy crap, we can't afford this. And I began looking at like, huh, how hard is it to go back to Harvard? All right, gentlemen. It's been three years or so, Gavin, since you and I last did this, which is nuts. And at the time, I was just wildly intrigued by your story and what you were going to

build. I didn't know a lot about chips at the time. I was considering investing in the company. And so, I was calling everyone I could conceive of that could give me an opinion or something. And at the time, basically, the consensus was these kinds of companies are not built by young people. that the semis world the best companies are founded by 40 50-year-old people that have had a whole career's worth of experience, have learned all the problems, have done have shipped multiple chips. Two 21-year-olds are like not going to do this. Like it's just not going to work. And it was it was indicative of a of a theme which was nobody believes in us. That's obviously changed a lot now. You know, just walk the halls and talk to the people that have chosen to come work here. But in the early days, it felt like this was this was something that you had to face down. What was that like facing that down where like a set of incumbents and an industry worth of people and investors and everyone else sort of didn't believe in you and what did that anneal in you to build the company the way that that you are like what was the impact of that?

>> I think there's a certain level of naivity required to think that you could build a chip better than every other AI chip ever built and build a company to do it way faster than ever has been done. and we have the naivity. There's many times where we would say like why isn't this possible and you know really push on it and it turns out that like everybody's answers are extremely siloed to a set of constraints that aren't true anymore and the reality is the entire semiconductors and data center industry is built on buffer and what I mean by that is every part of the stack from the EDA tools to the power modules to the circuit boards to the chip design and standard cells everything is built to be general purpose for everything not just in the data center but IO IoT on the edge and so forth. and when you have a specific use case use case you're really trying to design for, you can change the constraints a lot. And I'll give you just a very simple example that we're not the only one who does, which is one of the things you care a lot about is the clock speed of your chip. It's proportional to the throughput of your system. You know, when you are doing signoff for different timing, you you know what clock speed you're actually going to be able to run on when you tape out your chip. there's this concept called corners, which is, you know, what temperatures are you going to be able to run at this clock speed. You know, the default configurations for a lot of these EDA tools assume that you're going to be running your chips in freezing temperatures. Now, I don't know about you, but I've never seen an AI data center with ice in it. So, you know, we can feel pretty confident that our chips don't need to run at full speed at 0 degrees CC. In fact, like they're never really going to be running below 80° C anyway. And just by knowing that that's a constraint that doesn't matter, we can make a ton of changes throughout the entire system. That's like a very simple one, but there's many more that you get 20% here, 50% there, 2x here, and these compound to a system that can be radically better for inference.

>> Well, I think you found two kinds of people. There are some folks who went purely on heristics. Okay, young founders. They claim they can go beat the biggest company in the world out performance. It cannot happen. And there is nothing you could go say to me that would make me change my mind. But there's also people out there who are of course skeptical that are willing to go ahead and say, "I'll spend the time. I'll do the work." And is actually possible. Like for example, one of our earliest earliest supporters was this guy Mark Ross. And Mark was a very prestigious semiconductor expert. Used to be CTO at Cypress Semi that sold for $9 billion. And when we met him, we were just a couple of guys in a dorm room.

And we came to him and say, "Hey, we want to go build hardware for inference. We think we can be much

faster than Nvidia." And Mark's like, "No, you can't. It will not work." But he want to go convince me if you write a white paper. You should go ahead and build a functional simulation and show me. And so after a lot of very long nights, went back to Mark and said, "Hey, here's a simulation. What do you think?" And he was like, "Huh, this works." But to go do a company like this, you'll need a large amount of capital. You need at least $3 million even to get get started. End up went ahead and raised five, raised a lot more after that. And then he was again surprised but got more involved and then he became an adviser and a halftime adviser and eventually a full-time CTO as he saw more and more of the development progress. And I think in general this has filtered really heavily. folks who want to go ahead and be right regardless, folks who will want to be very truth seeeking and say, "Sure, I'm skeptical, but I will go ahead and work through the numbers myself. And if I can go figure out why this is possible, well, let's go build it."

>> The specifics that you've made bets on, the way that you've built this system are immensely interesting to me. And because so many people are trying to do this now, build new chips that will do a better job of serving inference at massive scale, the world is interested in the research approaches, the different architecture approaches that people are taking to building a new AI chip. And I'd love you to just start by describing what this thing is, what it does, but maybe more interestingly and more importantly, the process that you went through to decide what bets to take, what technologies to invent, and compare and contrast those with what you've seen the rest of the marketplace try to do.

>> Yeah, I think that you go start with the product. We're not just building a chip. We're building a full inference solution. And that means a rack. That means the chip. That means the power delivery into the chip. That means the board in which it sits. That means the interconnect by which the chips talk to each other. That means the production for this mass volume of racks. Really the production is the product. We think about how we get our advantage. There are two key parts of running inference.

There's prefill and there's decode. Now we have two key tags for both of these things. Prefill is reading in a huge volume of text and decode is then using that data to generate output tokens. When you go out and run prefill, yeah, your key job is not to go predict tokens. You already know the text. Your job is to go ahead and get the model's memory. >> Yeah. >> What we call is KV cache into the right state.

>> Then you can go ahead and run decode with that same KV cache. So we will often go do is we call PD disagregation. Prefilled decode disagrees. You'll then transfer those model memories, those KDV caches over to the decode cluster and then go ahead and use that cluster to go generate the next tokens. >> So it's sort of like loading the gun and then firing it. Like if I think about it in super simple terms. >> Yeah, you got it. It's getting the model to remember the right things and then using those things to go do tasks.

>> Generally people think about this market a bit lazily or they say are you prefill chip? Are you decode chip? If you're decode chip, are you an HPM chip? Are you an SRAMM chip? Are you 3D RAM chip? are you using optics using copper? When we started this, we just wanted to understand why extremely smart people were working on these different directions. We seriously looked at architectures like having bunch of DDR memory and like a shared memory pool and looking at advanced packaging to basically break out of the shoreline. We

looked at things like, you know, is there ways to put memory dies on top of compute dies. In doing so, we realized that there's no free lunch. You know, everything has a trade-off, right?

You know, 3D RAM, you have a thermal issue, you have a supply chain issue, you have to figure out hybrid bonding, you have to figure out the flops. So, now you're a decode chip. So we went through everything both on the prefill and the decode side. In doing so, we realized there's a few design spaces that nobody had seriously tried to explore because they were never done in AI chips. And we asked ourselves, what are the actual metrics that are going to matter the most? On the prefill side, the thing that matters is flops and flops density. And people talk about flops often as a headline number, but in reality, you should care about the flops you're getting when you're running real workloads. there's this concept called MFU or model flops utilization which is you know for every peak flop advertised how many cents on the dollar are you actually getting and on GPUs you often get somewhere between 20 and 50% depending on the workload and actually you can provably not run at 100% because you have a thermal issue where as you increase the flop utilization you have more transistors going on and off you draw more power and the chip will self-regulate and actually lower its clock speed to make sure it doesn't overheat as we looked at inference And we said if we want way more flops because we want to run at way higher throughputs, we fundamentally need to solve the thermal problem before we even think about adding flops to the chip. If I just add more flops to a GPU today or another AI chip, I'm not actually going to get more performance because it's just going to thermal throttle. Fundamentally, the essence of that is this concept of dinard scaling, which is voltage is quadratically proportional to power. So if I 2x my voltage, my power goes up by 4x. If I cut my voltage in half, my power goes down by a quarter. So we asked ourselves, how could we run voltages lower than GPUs? And we talked to a lot of people about this. We flew out to to Silicon Valley after dropping out and basically asked, you know, dozens of people in semiconductors at all these different chip companies how they did it. And the answer we got was like, you can't you can't run at voltages lower than GPUs. And this was very dissatisfying because there was many different industries of chips that run at voltages lower than GPUs. I mean, Bitcoin miners run at under a quarter of the voltage of GPUs. So this is obviously physically possible. The question is, are there issues with GPU architectures that make it unable to run at these voltages? And when we looked at the problem for a long time, we were able to create a new mechanism of running at much lower voltages, a new type of power delivery that we call low voltage inference. And we think all AI chips in the future are going to be low voltage chips. They're going to have to cram way more flops in the same silicon area and without thermal throttling run at way lower voltages. That's pretty simple. For decode, it is all a memory game.

more memory bound, you can load the model faster, load the KV cache faster, and serve more tokens per second per user. We think people ask the wrong question here. People often ask how much memory bandwidth is on your chip. You should be asking how much memory bandwidth is on your full scaleup cluster. What we are able to do is add way, way more bandwidth and a much lower latency from chip to chip to our interconnects. It allows us to be able to go serve models at this much higher speed because you can go use the SRAMM and the HPM from the full scaleup cluster as a single pool. And that's our second key technical bet. What we call cluster scale memory. And on GPUs today, the cluster memory bandwidth is often very badly utilized because the time to go hop from one GPU to another is extremely long. For example, on blackwell chips, it can be about 4,000 nonds to go point to point. And that means that if you go ahead and go to an 8xtp setup, you will get way way less than an 8x improvement in your tokens per second per user.

And what we did is built our own totally custom interconnect stack. We took everything above the second layer of Ethernet, built it full custom, and we can go out and do far better latencies and bandwidths this way too. We can go ahead and cut this by more than a factor of 5x. And that allows us to then use the memory of other chips much more effectively. As you scale the world size, your time per token goes down proportionally. It's not that surprising given all these architectures were built before chat GPT. So if we're trying to build a chip for the modern workloads, it's going to look very different. The way we, you know, organize our flops, the way we do our voltage domains, the way we do our power planes are going to look super different. The way we do the packaging is going to look super different. The way we do the board design is going to look different.

and then on the decode side, the way we connect everything is going to look very different. We're now bringing forward our first generation of this low voltage inference technology, which is running at under half the voltage of any other AI chip. >> RAMP is the only platform built to make your finance team leaner, faster, and better, saving businesses 5% annually on average, so you can stay focused on growth. Ramp customers grew revenue 3.2 times faster than the average American business. Visa, Perscell, Cursor, Stripe, Notion, 11lab, Shopify, and 70,000 other businesses all run on Ramp.

Mine does, too, and so should yours. Learn more at ramp.com/invest. OpenAI, Cursor, Anthropic, Perplexity, and Verscell all have something in common. They all use WorkOS. And here's why. To achieve enterprise adoption at scale, you have to deliver on core capabilities like SSO, skim, arbback, and audit logs. That's where work OS comes in. Instead of spending months building these missionritical capabilities yourself, you can just use work OS APIs to gain all of them on day zero. That's why so many of the top AI teams you hear about already run on work OS. Work OS is the fastest way to become enterprise ready and stay focused on what matters most, your product. Visit works.com to get started.

Felix by Rogo is a personal finance agent that turns a single prompt into finished client ready work using your firm's own templates, context, and standards. Send Felix an email like, "Take these comments and turn them for me." Or, "Udate my tracker with the context of these emails." Or, "Run the ability to pay math on this buyer." And Felix sends back finished PowerPoint decks, Excel models, and sourced research. Felix works the way your team already does, delivering work quickly and accurately around the clock. Learn more at rogo.ai/felix.

If you zoom all the way out, why is this so important? like what why is the delivery of much higher throughput, much lower cost per token better better tokens per watt like all of these metrics that the the universe is going to start talking about more and more. Everyone knows everyone's the the supply side of the equation is is a big problem right now. Why is this in a bigger picture looking out a decade like the bottleneck in the technology world?

Well, I think it comes down to productivity where we are at this extremely interesting moment in the history of civilization where there's real artificial intelligence, not like sci-fi stuff, but like these models can solve problems that most humans can't. And like it's going to create new scientific discoveries. It's going to create instant access to medical care, instant access to education, and now it's just about how many people can use this at the same time, how many products can serve this at the same time. and also the speed of of doing different tasks. So you

know when you think about wall clock time if we can take an agent that you know can run at a certain model quality and could take a year to solve a certain task using inference time compute if you have way faster decode speed you can compress that into a month. So the amount of scientific innovation and the amount of actual proliferation of technology will happen much faster. And then the second part is concurrency where today you just like it's just not possible for a billion people to use these models concurrently.

Ultimately, some people are going to get downgraded. Some people's models are going to be slower. Some people just won't be able to access the hardware. A few years from now, there's going to be giant models serving billions of users. We're very much in the early innings of AI today where the paid plans, there's only a few million users in the world using paid plans of AI models. So, we're at 1 1,000th of the global population actually using this stuff. So, if you want to serve a giant scale, a lot of things change and one of them is the number of chips that communicate together where, you know, people usually think about this in the context of training. you know, you have these giant training clusters. You have Colossus with over 100,000 GPUs that are allorked together. And you know, the inference side, you know, today people usually think about it as an 8 chip cluster or maybe just, you know, NVL72 as the scaleup domain, but very quickly this is going to become thousands of chips and tens of thousands of chips. And the way to get the most performance there, the time between sending data from one chip to another, that primitive matters way more than is getting credit right now.

So when we think about optimizing memory bandwidth for the system, you have to think about how fast these chips can communicate together because if they can only communicate really quickly with themselves and very slowly with other chips, you're not going to actually be able to serve giant models at 10,000 20,000 tokens per second. So we need multiple orders of magnitude of infrastructure built out throughout the entire stack from the power, you know, from from the wafer to the watt, you know, from the the transistor to the token, to actually bring this stuff to the world. And I think that you look at most other goods like the iPhone for example, they've gotten into this economies of scale where as a result more money does not really buy a better iPhone that if you're a billionaire or if you're just the average American, you buy the same phone. And tokens aren't like that yet. We are still in the very early days where a general purpose system, relatively small one, is kind of handcrafting these tokens like they made screws back in like the Renaissance. And I wanted to live in a world where you have the same economies of scale for token making that you do for making say iPhones or cars or anything else.

>> I think that is one of the huge unlocks that allows a huge group of people to go use the best quality models. I think that economies of scale have all made capitalism very I don't know fair. I think that allows you to go ahead and have the same product in many many different hands and you're able to go then serve way more users on a single scale up cluster allows you to get closer to that point for token serving too.

>> Yeah. And also just certain products aren't usable if they're slow. >> Yeah. >> So if you want to serve coding models you know and you want people to actually use them like there's a certain number of tokens per second you need to hit. So the question is while maintaining that per token speed how many users can I serve at the same time? and you can basically decide I'm going to shut off a bunch of the world from using this stuff or everyone's going to get a worse experience. So fundamentally you need to find ways to push out the curve and

that's why there's such a pressure for new hardware. I'd like to take some time to step back and hear both of your stories for how you came to this idea and this company and then kind of walk through what it's been like to build it because I think in so doing we'll understand the system that you've built for for the company itself that will then be able to power subsequent generations of of products like this one for this crazy inference future that that we're staring down. Rob, maybe starting with you just take it how however far back you want, but what what what I'm curious about and your personal story was actually the very first thing I ever heard from either one of you was your personal story many years ago now, which really kind of blew me away.

I'm most interested in your motivation ultimately for being here doing this thing. >> It starts back actually in high school for me. I've been very unlucky and lucky at different points in life. this was one of the tougher times. You know, at the end of my sophomore year of high school, I got injured at a martial arts tournament. the next day it couldn't walk for some reason. and they thought it was something wrong with like my SI joint or something. I went through physical therapy. I did different types of scans. They couldn't figure it out.

and eventually they found this big bump on my back in an MRI and you know told me it was a tumor. Stage four bone cancer was told I had under 30% chance of survival. it was like a 2-year crazy chemotherapy surgery learning to walk again experience. and when you go through something like that, it you know really changes the Overton window of human experience and makes you appreciate like what actually matters. and you also ask yourself like what are you going to do if you have the chance to live? Like if you actually want to get through something like that, you need to be like hoping for something. and I always knew I wanted to do something very impactful if I had the chance to to to get through it. it took me a couple years to figure out what that was going to be.

And at the same time as I got to college and met a bunch of other people building cool tech I got extremely excited by AI models especially once GPT3 came out and I was like wow this is the first model that can kind of speak English and these things are going to get really smart. What happened was when GPT4 came out there was GPT4V which was the first model with image uploading. So I went through my camera roll and I found a picture of my back with this bump on it before I was diagnosed and I said you know hey chat GPT you know pretend you're an expert doctor. A patient comes in and they says they have this, you know, bump on their back. What could it be?

>> And it immediately says like this could be a tumor. You should get an MRI immediately. Like go to the doctor. And I just kind of sat there still. It was like >> this took six months. >> Yeah, that took me 6 months. And you know, yesterday this feature wasn't there. Today it's here. I go to show my parents and I got this like notification being like, you know, you're all out of image credits today. Like you need you need to get a pro plan. And I was like, holy crap. like this is going to change everything and you know like we clearly don't have the infrastructure to serve it and there's very few things you can work on that can actually bring this technology at scale to the world faster. I mean there's like plenty of people that are super smart working on models. you know the the fabs seem maybe unreachable to work on but it seemed like the hardware was all designed before chat GPT. So like every GPU, every TPU, every AI chip that was serving these models were just like fundamentally built before this and are retrofit to serve these modern models.

There's going to be an entire new wave of architectures that came out and like what a more exciting thing to work on than bringing this to everybody. Very different angle at the same time. I I was running a startup incubator called ProR which has incubated a bunch of different companies. Some of the earliest ones being cursor and anyphere which merged and mer and etched went through it a handful of others. Yeah. At the time as these models were getting smarter, it was yeah 2022 I was realizing all of these companies are spending all the money they raise on compute and I had had this realization as I was working on some of my own stuff that like oh my god all the products I want to build are going to cost tens of millions of dollars a year in inference.

Hey, this is not going to be tenable. Like the cost structure of every software company, the COGS is not going to be like zero anymore for an incremental user. It's going to be like quite high and it's going to be a function of inference and then the opex of every business is going to also be inference as people use more and more coding agents. So fundamentally it seems like inference is going to be really important and it feels like we are on a like decade march for inference to become you know the biggest market in the world. so when you think about that 10 years from now there's going to be these giant projects where everything in that data center fundamentally hasn't been designed today like we should go pick something and work on it. And you know that's that's kind of how it got started.

>> Gavin I I'm really excited for you to go back about as far early in high school maybe even earlier and tell the the your favorite hash marks on the timeline that ultimately led to your ambition to to drop out of Harvard and and start this company. My first job ever was at a company called Exnord where I did a colonel's development. I was 17. >> Yeah. >> And a 17-year-old can't sign illegally binding contracts. So rather than go ahead and do a traditional CIA, they went ahead and sat me down and said, Gavin, don't share this information. And Extern was one of the only companies that saw, hey, maybe this is a good trade. And got to work building kernels. And I've done this at another number of other companies since. Extern got bought by Apple for 200 million. Did the same thing at Octo got bought by Nvidia for hundreds of millions of dollars. But when you do this sort of colonel's work, what you realize is that the math is relatively easy.

>> Okay. >> But to get high-speed decode, the thing that matters is data movement. Almost all the work that you do is optimizing how do you move data around a single chip or across multiple chips. And that's why we went ahead and built this cluster scale memory tech. we bring that interconnect time way way lower. You can go do way more movement and as a result get a much faster time to generate each subsequent token and build these crazy things Rob's talking about for doing a year's worth of work in a month or more than that in the future.

>> Can you talk about the competitive drive that's evident in some of the high school competitions that you participated in and won? >> we did a couple. for example I was very active in the FTC robotics and I was lucky to have a very talented partner Sanford that for a long time we were part of a traditional school team where it was about 20 guys all working together as often typical in first tech challenge and the goal is to go out and build a robot that scores the most points and a bunch of other things too. and first they put a lot of emphasis around collaborating with other teams around trying to do really good documentation around trying to go ahead and get others inspired to go do the same thing. Sanford and I decided rather than go ahead and do it this way we're going to win and we did nothing else besides build that scored the most points >> as a twoerson team >> as a

twoerson team >> rather than a 20 person team >> that we were much much smaller than almost every other team in the competition. We figured that if we were going to go specialize, if we were gonna go out and do this win the damn games really well, we wouldn't need to go ahead and advance based on the quality of our documentation or of our outreach, >> we were just gonna go win.

>> And so we did. We have branched off, built this twoerson team, built a robot, and decided we were going to go ahead and redesign it every 3 months. And we did. And we actually had the world record for the highest score during this competition at one point. we were rated by OPR third in the world for software development and it was a damn good machine. >> What from that episode can I translate as an analogy onto how you built Etch the company?

>> We think about how you want to go ahead and do a full rack scale product like this. There's a couple key ideas. One of them is like velocity velocity velocity that you win by shipping. You're not going to go out and win by having the best outreach or the best communications. We you win by having the best product. >> Chosen to have no communications. >> Exactly like your I'm just realizing how exactly like the robotics.

>> There are many ways you can go win in business, but we'd rather go focus on just building the best product. >> And similarly, we think we can go do it with a lot fewer folks that if you're willing to go ahead and just focus on product, product, product product and paralyze relentlessly, you don't need 20,000 people like the big companies have, you can do the best product in the world with far fewer people. You know, there's a saying of the best part is no part. I think for us, it's also the best vendor is no vendor. As much as possible, we want to vertically integrate the entire product. Both because we get more performance, but we can move way faster. So, you know, everything from the chips to the boards to the cold plates to the interconnects to even the production. We want to do all of it as in-house as possible. We're actually I think we're the only startup right now that's building its own rack as well as its own chips. and we did it all at the same time. A couple years ago is the last time we were public.

at that point we just started building our rack team and we brought over Brian Lerer who built all of Nvidia's HGX and DGX systems which is like 80% of the revenue and we said we're going to build the rack at the same time. We actually went through multiple iterations of the rack before the chips even came back. before the chips came back we made thermal chips that had the exact same hot spots as we expected our chips to have. So we could build the cold plates.

We could overpressurize them and blow them up. We haven't had a single leak since our chips came back with the cold plates cuz we already validated them. Yeah. We have a a factory in Taiwan. We have a few dozen people out there. We built a clone of a bunch of the test stations in our office. we have a 2 megawatt data center on this floor. and we did 24/7 development cycles. You know, people are doing day shifts and night shifts to actually get the hardware up and running as quickly as possible. you know, it's that extreme vertical integration and extreme parallelization of the schedule that lets you get products to market way faster. M if you think about the building of the early team and what it required as two young guys building this company, there's lots of very talented young entrepreneurs out there maybe for the first time of this scope or magnitude in a long time all of

whom probably could benefit from the lessons that you've learned. Getting very sophisticated, talented people to come join you even after careers at the you know the other great companies. if you were teaching this as a class like here's how to get you know elite talent when you're young and inexperienced and naive like what what would be the syllabus?

>> We have a pretty biodal talent philosophy. It starts with what we call the legends which is when we're trying to solve an incredibly hard technical problem and generally do something that hasn't been done before. We need to find the very best person in the world and often the number one guy in the world versus the number 10 guy versus the number 100 guy. Huge difference and whether it's actually possible to solve the problem. We created this system we call project-based recruiting where we map out all of the hardest technical problems across all industries that anyone has ever had to solve. We look at temporality. So who are the people who did the zero to one? Who is in charge quote unquote who actually did the work?

we talk to as many people as possible. And then we just track it. And you'd be surprised by, you know, the amount of people who say yes after the first conversation is pretty low, but the amount of people who say yes after the 20th conversation is surprisingly high. You really got to keep at them. That like when you hear no from somebody who really is the best in the world, then that really means, hey, you should go ahead and come back when you have a few more milestones proven out.

>> Yeah. >> And I think it's one of the most convincing things to see is, hey, we make bold claims. And when you go ahead and hit those again and again and again, that is really belief inspiring. When we decided we wanted to build a rack and not just a chip, we were looking at this and we're saying, you know, how many products have actually shipped at scale for a rack scale system that actually have the power density that we're trying to solve? And you know, we just kind of said, if we were going to wave a magic wand, what would the best possible person in the world look like? and be like, well, if we could find somebody who like started at NVIDIA and built the entire rack team through all their different generations, learned all this different stuff, but is still scrappy, still understands the startup culture, but has seen scale, like that would be the best possible person. So, we mapped all of the different teams that related to all of the different rack scale products at video and we found three people that we thought, you know, could fit the bill. and we talked to all of them and two of them have just retired and one of them was planning to do one more generation for Nvidia and and then retire. his name is Brian.

and over time we convinced him to join. Brian started the HGX and DGX team at NVIDIA. you know, which was, you know, a majority of Nvidia's revenue, you know, tens of billions of dollars a quarter. and the other two guys ended up investing, by the way. But when you have somebody like that, they just know what good looks like because they they've seen it. And like there's so many times where we'd talk to Brian and he'd just point to us and be like that's a billion dollar like a billion dollar lesson I learned. Billion dollar lesson I learned like you know that just saves us cycles. And you pair someone like Brian with somebody like Sanford.

>> Do you have a name for them? So Brian's a legend. What's what's Sanford? >> Yeah we say chips on shoulders put chips in data centers. >> Yeah. So you know Sanford and Gavin in in in high school were world

robotics champions. and Stanford was finishing his senior year of college and we called him up a couple years ago and we said hey can you come you know check out what we're doing we need some help on the platform side and he comes for a week and we say can you build the cold plate this week and if you asked like any thermal engineer like you know anything like that they they would think you're just like totally naive right I mean these things take months to do and like to be clear they do but you can make real progress in a week if you if you put your mind to it and you think it's possible and like he built a contraption in a week that like actually derisked like a pretty key power question we had. and you put those two together and they've done incredible things and like one is not one is not possible without the other because you need the extremely driven people that just keep asking why and don't know where the bodies are buried to like take tons of aggressive risks and then you need the people who've seen scale and still have the startup scrappy mentality to help them along the way. It's really the It's really the legends plus the raw, you know, some naivee raw first principles type talent, but it's the com. It's not just that you have both in the company. It's that they're working together.

>> That's right. >> If I think about that funnel, anything else more interesting to say about how much better you've gotten at recruiting and like why why those metrics keep getting better. One of the shocking things is like being such a contrarian bet kind of self- selects, >> right? that like you're the kind of person who I think is some opportunistic going to go join whatever the hawk company is rather than go ahead and do deep digence you will not come work here >> right >> and it's one of the things I worry about as we announce more and more of the product and its specs we may lose some of this if we're not very careful >> you kind of have to be sick in the head to join our company you think about it on paper it's like you a person who you know is probably a very accomplished engineer making a good amount of money it's liquid it's predictable somewhere else you're going to convince your family to move to San Jose and live in this apartment on this housing program for the semiconductor company run by two what 24 year olds now that's pre-product that is going against the biggest companies in the world and the most supply constrained environment ever created with a design that they're saying is not going to be like 10% better but it's going to be 10 10x better. Something must be wrong with you to do that. And you know people are just wired differently here that they like really want to not prove people wrong who don't believe but prove people right who do believe and like they just take it personally and you know that's really fun to find those people and like you know frankly just the nature of the companies you know makes it very easy to whittle out the people who aren't like that. One of the very first things you and I talked about, Rob, was I I started asking about Sohoo, which is the name of of the first product here. And you said we could talk about that in in great detail, but the thing you should know is that what we're really focused on is building a machine that can at scale produce these things and generations of them as efficiently at the highest possible quality level.

So we want to build like the company or the machine that is the company is the thing that totally >> will produce this thing and then subsequent things. So I'd like to talk about a few principles or cornerstones of the company. One of them you've talked you've alluded to some of them. You've said velocity, you've said vertical integration. You know have become more popular topics. parallelization is something maybe that we should talk about. But I'm especially interested in your guys' willingness to take huge risk to go faster. Maybe tell your favorite story about why this is the philosophy, what it's allowed you to do that maybe other companies haven't done.

>> There's a number of stories here. but one of my favorites is there was a time where we were getting close to taping out the chip. We realized, wait a minute, one of our vendors is way, way behind schedule. And we had two very bad options. One option is to keep the current vendor and push our timelines out by on the order of a year. >> Another option was to switch vendors, start over, and also push timelines out by a year. And neither of these was a good option.

So we had to go look for option number three and what that was was we we figured out they're all in Bangalore team actually going and doing the work. We went out and shipped a dozen of our top engineers across the world to Bangalore for 6 months. I was there as well. I lived in Bangalore for four and a half months personally and every morning we'd go ahead and walk across the crazy busy Bangalore streets into the office. we'd be the first ones in. We'd go out and built a wide variety of tools, both things like, hey, auditing a huge amount of the code that was going in, building a bunch of tools as well to make this go even faster, making sure we're making the right design decisions on the spot right there. No 12-hour back and forth. We could go ahead and decide immediately, and then at 1:00 a.m., we'd go walk back through the now empty Bangalore streets and do it all again the next day. We still had the a bunch of the team in the US. We ran these 12-h hour on each side, you know, handoffs where we had 24-hour development cycle where at 8 a.m. and 8:00 p.m. every day, we'd all get on the Zoom, we'd share all the data, we'd say, "When I wake up, like, this must be done. Like, we must get this chip out."

and it was extremely intense. At the same time, we saw other chips, you know, at the same stage as us with that same vendor that ended up taking years that still aren't out today, that still haven't even taped out today. and it's that level of extreme urgency that's required to bring products to market. >> What is the key to doing this? Well, this has become a a trope because of Elon mostly that like his special skill and others that seek to emulate him would try to do this too is figure out like what the binding constraint is and just like flood the zone personally on that thing which is kind of like going to Bangalore or something. It seems like this is this is a central tenant of the business and of any business that's going to do this kind of vertical integration. What's the key to doing that? Well, like again, what have you learned about that specific act? For me, I think there are two key tricks to this. The first one is that you can't build a chip alone. It's got to be a team problem. And your most important job is to go get great people to go with you and great people to go ahead and be inspired and excited to go ahead and do crazy things like this. It is a huge ask to go say, "Hey guys, uproot your lives for six months or in one case 12 months." That we had sent one guy out well ahead. It sucks, but we're lucky to have team members who are in it for the right reasons. But I think the second big thing, too, is being able to make decisions very fast. That one of the worst items is when there's a factory or there's a vendor who is waiting for you to go ahead and make some call and is then just stalled.

>> And this happens all the time, even for very small things. So send folks, delegate a big amount of responsibility to them and say make a reasonable call. It's okay if you're wrong every now and then, but I would much much rather be right most of the time and give an answer immediately than wait every time for the perfect response. Speed wins. >> What about spending money to go faster? There's this learn by doing thing which has become so interesting and as the world has gone away from software and towards more hardware again in in the world of technology that we've outsourced so much of the learn by doing totally by shipping stuff overseas and effectively just being the idea guys here in the US. Seems like that obviously is reversing and

you've adopted this way of learning by do like you want to be in that iteration learning loop.

>> Absolutely. >> And and part of that is willingness to spend and take risk with dollars. Yeah. >> Can you talk about that a little bit? there I think there's a great quote of like the biggest risk is not taking risk very similar here which is like every day there's over a billion dollars of revenue in this category and a lot of it's inference so every day we don't ship we're just leaving tons of opportunity on the table so your willingness to spend money should be extremely high if you can get a very clear ROI out of it so we have this concept that we call pre-fetching which is when you're waiting for one thing to get done when you know you're going to do other things once you have it is there ways that you and parallelize the entire schedule. So for example, like we know our chip is we know our chip is going to come back on a certain date. We want it to be that everything possible that could be done without the chip is done before the chip lands. And this costs a lot of money, right? This means that like we want to build our entire software stack beforehand. That means we want to actually like we shipped racks to customer data centers without our chips in them with all the networking, all the CPUs, all the storage all set up so we could bring all that data center software up before the chips came back.

It meant that we took over 700 FPGAs and put the entire full reticle chip on an FPGA cluster and ran a dozen different models with our full inference stack on them before the chips came back. It means that we built a thermal chip to mock the thermal profile of our chip and built cold plates based on that before the chip came back. It means we had the entire production line ready. It means we did many revs of the circuit board.

It means the entire product was ready to go before the chips came back. And this is what it gives you. There was another very famous AI chip company that took 10 months to go from getting their silicon back to having them running inference in Iraq. And this was like publicly announced to their investors and was it was it was a really big deal. we were able to do it in 40 days. And it's because by the time the chip came back, everything was boring. The software was already written, the rack was already there, the production line was already set up. We were just go go go get everything together. You know, you don't always catch everything. You make some tweaks on the fly and then off you go.

>> Although in that particular case too, that was a big part of it. But also like the shift I think made a big difference too. >> Oh, totally. >> That like we went out and literally had a day shift and a night shift. There were team members who would go in, come in at 10:00 a.m. and leave at around midnight. Yeah. >> Which would come in at midnight and leave at 10 a.m. running around the clock to get to that 40 days.

>> Yeah. I mean, over half the company lives next to the office. so it makes it easier to do that type of thing. >> You pay them to do that, right? Pay them extra. Do you still do that? >> Yeah. the the invisible hand does wonders. >> I mean, hey, it works for me, too. >> We're both there. >> I'd love to take one big step back and talk a bit about just the broader ecosystem here. The amount of shortages on the supply side, the exposure of risks in the global system and the supply chain around this stuff has become like everyday Wall Street Journal front page news. like the stocks that people are watching and investing in and excited about. You know, if you think about the memory stocks, these were, you know, boring commodity like nothing burgers five years ago and now they're the center of global attention. If you just assess because you've been building in it, the global connected

supply chain that's required to make stuff like this possible. Just riff on it. Like what scares you? What's working well? What needs to change? What do you hope you change by virtue of how you build this thing? like just what's your assessment of of this story right now?

>> I think that one of the most undervalued pieces of the supply chain story is almost none of these things are you buy them and you don't talk to the vendor again. You have to go collaborate. That is the most important part being successful I think in ships with TSMC or with memory vendors. You need that partnership. I think that for TSMC in particular, people don't understand why it is so valuable. You look at the tech and the tech is the best in the world.

But for me, the real value is all in the service. The TSMC customer service is way way better than I have seen at any other company in any other industry. It's the kind of thing where if you say, "Hey, we you can improve your yield by making this change. You can go make them a recommendation and then we'll go run an experiment on their own dime in our case to see if they could actually get the higher yield." And when we found that we were right and the experiment worked, they moved over the rest of the line. And that kind of thing just doesn't happen in most most industries.

If I go to like the steel works plant, say, "Hey, I want you to change the composition of the steel." And they'll say, "Screw you. Not TSMC. It is why they are the number one and why they're they're going to win." >> One of the things that matters a ton is power availability and time to power. And the problem is the more power you want, the more shortage there is. It's actually very similar to chip clusters, which is like why is Colossus charging $12 an hour for black holes? It's because they're the only place you can buy 20,000 of them at once, right? Like why is the like 500 megawatt data center so hard to find? It's it's the exact same reason. You know, one of the things we need to think about is like how do we get way more juice out of each megawatt?

and like people are looking throughout the entire stack, whether it's just improving the PUE, but also entirely new hardware to get the most tokens per megawatt to solve this problem. but fundamentally like building new buildings is hard. it's much easier to go from a 100 megawatt to a gigawatt than from a gigawatt to 10 and 10 to 100 and you know we are pushing the limits of of what's possible on these timelines. So there's a lot of people trying to scale in their data centers as much as trying to scale them out.

>> Yeah. I mean one of the one of the interesting things about a system like this is what it replaces. Yeah. >> So if if I think about a rack like this versus I don't know a set of black wells or something or Reubens or or whatever is coming next. How should I conceptualize that? Like because not it's not just watts, it's also physical space to your to your point that people all of a sudden cerebras talked about this in their in their recent earnings call that literally this is a problem that there's literally no space to put the systems. yeah, how how should I conceptualize what this represents or replaces in terms of other units of compute? Here's how customers think about stuff deploying models generally which is you know when I'm building a data center or I'm building a cluster it's not like in the abstract of like oh I like these chips and like you know this is the power footprint and so forth. It's like I have a real production workload I'm trying to serve and you know for my product to be useful there's a certain speed I need to serve it at and for certain products it's really fast and certain products it's really slow whatever

the speed is this is my speed the question is in a given amount of power how many users can I serve while guaranteeing that speed >> so another way to put it is IS iso what's called interactivity what is my throughput we are just finishing kind of the early innings of of the AI infrastructure boom where people really just cared about speed you know GPUs were not able to reach a lot of the speeds of other types of chips like all these SRAM chips you know thousands of tokens per second and that enabled tons of new use cases that got people very excited. There's an entirely new wave of AI chips you know us being one of them that are all going to be able to hit these speeds. The question then is if you're hitting these speeds, what is the number of users you can serve at the same time? And by proxy, if I have a 100 megawatt data center, how many, you know, software agents can I run at the same time? So when people are doing that evaluation, our hardware is going to generally be able to get you an order of magnitude more concurrency at a given level of interactivity. So that directly translates into, you know, tokens per watt, tokens per dollar, all the things people care about when they're actually serving, you know, these giant mixture of expert models at scale. M there's these now famous interactivity curves, right? So you can pl not many people publish them. but you could see a Blackwell curve. You could see an AMD curve which is a little bit worse than Blackwells and it's still an $800 billion company. So if you think about what then the impacts are of shifting that curve not just a little bit further out but but much further out. Yeah.

>> What are the things that most excite you about what this will enable? I mean, I want to go out and solve some of the hardest problems and I want to go solve these in much less time. I mean, there were things growing up that I was not sure I'd be able to live to see. For example, the unit disc conjecture, one of the things we talked about in college. >> Yeah. >> And I was not sure I'd see that proven in my life.

>> And this was done by an AI model and it was done over a long period of time. But if you're able to then run the same model 10 times faster, you can go shrink the time to go have these breakthroughs. And there's a huge number of other problems in math like this as well that I worry it will take 100 a thousand years to go prove a thing like this. You can either have a much smarter model or a model of the same intelligence running much faster. You can then shrink that and I can see it. And it's so cool seeing these breakthroughs get made. I am so so excited to see much more of this happen. I think too often people think about tasks and applications and stuff in these very short time horizons.

So, you know, doing a chat and it's like 50% faster is nice, but it's not like gamechanging. As these agents go longer and longer time horizon and the models get more and more capable, you're going to see gigantic bodies of work that would take months of compute. And we think about this in wall clock time. Like if you talk to a pre-training researcher at a lab, they'll tell you that wall clock time often is one of the most important things that matters. And what wall clock time means is the time from, you know, starting your run to finishing it to actually get data back. If if you can shrink this time, you know, from a six-month run to, you know, a two-month experiment, you're going to be able to do many more iterations and people will make changes on the model architectures to actually improve the wall clock time.

Very similar here in terms of how we think about the use cases, which is, you know, the exciting part about super low latency decode is wall clock time on long horizon tasks becomes much shorter. So a year-long compute build would now take a month and that month-long compute build will now take 3 days and that

3-day compute build will now take 7 hours and so forth and so forth. so that's the thing that I think is really hard to internalize because the models are just getting capable capable enough to do this stuff. I thought it was really cool months ago when cursor published that they had a bunch of coding agents build an entire browser from scratch in a week. Totally nuts.

And that that will soon happen in under an hour. and there's going to be many of those types of things that are going to happen with these massive parallel agents all working on a given task. >> What are the ultimate limitations of these systems? Is this just like a physics question like like how many times faster cheaper can we get theoretically? Like how do you think >> there's a lot there's a lot of room at the bottom as they say that if you think about chip and chip latencies on an NVIDIA product you're looking at 4,000 nconds to go from one chip to another.

Mhm. >> We'll be able to do much better than that. >> What's the mathematical limit >> is speed of light. >> You can do it in just a handful like 2 three nconds >> and they're 4,000 >> 4,000 today. There is a lot of room at the bottom. I think there are things like power efficiency that sure we're able to go and save huge amount by bringing the voltage down by so much. But you could go lower. You could go much much lower. It's very challenging.

But when I think about 20 30 years in the future then I think it's inevitable and also for economies of scale for cluster scale up for a long time eight chips was the biggest scale up domain and it even had 72 bringing it to well 72 but you can be way way bigger you look at like a fab for example you have a $40 billion single monolithic building with only a handful of lines running through it you could have the same kind of thing for some futuristic mega cluster $40 billion hundred billion as a giant mega token factory serving one or a handful of models for a massive number of users to get that same economies of scale thing. Same model, massive number of people.

>> You you mentioned kernels engineering and that being your first job that has emerged as a thing that nobody had ever heard of in their lives to now something that you hear about all the time. the importance of it to ek more performance out of the you know the bare raw metal when will that just be something that AI doesn't entirely as well are humans still the best kernel engineers are they doing it with the assistance of AI systems like how far down will humans still be in the loop of designing these things like when will that go away >> today it's all very hybrid and the best kernels are still written by human AI collaborations well so any AI model is built up of these fundamental primitives like mat moles like convolutions like chipto-chip operations collectives and making these overlap and making these really fast matters enormously and it's the kernel designer's job to go ahead and figure out where can I overlap how do I allocate memory how do I verify that if there's some issue like a retransmit needed it doesn't stall the whole pipeline >> these things are very challenging but they can go make your performance be say 3 4% better per optimization and you can do so And we thought about our software stack. We wanted to go skate where the puck is going to be. And three years ago there were kind of two ways you could build software. One of them was to invest heavily in graph compilers. These things are not very performant but they work out of the box. They don't require a human to go come in and tweak all the kernels. But we went the opposite direction. We are kernels first programming. And that means that yeah it for a long

time did not work out of the box. But if you were a colonel's expert, you could get incredibly incredibly high performance. And the thing about this is that now as the coding models get better and better, they're doing more and more of the kernel generation task. And when the models keep getting smarter, it'll eventually do all of it. They will become superhuman. So we're going to build for where the world is going. And even today, we think about our profiling tools or a debugging stack. We think about it from how will the model use these tools more than we think about how will humans use these tools.

>> That's we sometimes run experiments internally and we had codeex actually get GPOSS running from scratch just based off of our docs completely by itself. >> Wow. >> And it did it I think overnight. we think about game selection a lot and what we mean by that is making sure we're investing our energy in the right bets because regardless of what you choose to work on it will take tremendous effort and you know one of the things that we started with was you know the decision explicitly not to build an arbitrary graph compiler not to support arbitrary pietorch not to support arbitrary CUDA not to support arbitrary onyx graphs but instead we envisioned a world where there was going to be under a hundred models that actually mattered and they were all going to very similar from the underlying mathematical perspective and that we were going to build primitives using physics that were going to accelerate these as much as humanly possible and we were going to allow the most sophisticated customers to have direct access to the hardware and do whatever they want and that has saved us a tremendous amount of time not having to build a compiler and that has allowed us to actually get much more performance and funnily enough when we started a lot of people dismissed this idea and the only people that took us seriously were in high frequency trading because they all they all hate compilers too they all write their own kernels and we've had dozens of people from high frequency trading join the team because they saw this philosophy too.

>> What are the limits to vertical integration? Like where do you how do you know where to draw the line? And I'm I'm starting with this question to talk a bit about the broader market. The circumstances of the broader market are really interesting to me where the vast majority of chips of AI chips get bought by a very small set of customers. Yeah. >> Many of those customers are themselves trying to design their own AI chips.

OpenAI announced Jalapeno. It seems like this very funny circumstance where like the most valuable thing in the world all kind of flows through a couple chip makers, a couple chip buyers. They all seem to be kind of doing thinking about doing each other's job. And then you've got the circumstance where like okay then these things go in a data center and you've got neoclouds and inference providers and this other part of the stack. you've got model builders and providers. Like I can imagine a world where because you have the best hardware you design models and you build data centers you know like you leak outside of your current vertical. So like how do you think about where to draw the lines for the business?

>> We have a saying that production is the product. >> Ultimately what matters here is we know inference is going to be the biggest market in the world. Whoever produces the most tokens is going to be the most valuable company in the world. So all the decisions we make is how do we get the most token capacity online as possible and part of that is building a really good product that's has way more throughput that can run at way better latencies and so forth so we can you know per chip we make get way more tokens online. Another part of it is

like not doing parts of the stack unless we absolutely have to to get to giant scale. So there are parts that we decided to do because it was absolutely required to get to scale like building the rack instead of just building the chips. like doing a CM model instead of doing a JDM model. But you know there are parts of it that are kind of noise to us right now. Like we're not going and building our own data centers today.

That doesn't actually help us get more capacity online. You know in general our customers are actually making power and moving their clusters around to get our chips online because they're such high throughput. if there was a world where other things were a constraint we would totally go and integrate with them. But the reality is we're just purely focused on getting as many tokens online as possible. I think it just comes down to economies of scale. Again, that at certain parts of the stack there are huge economies of scale and others there aren't. For example, on designing models, huge economies of scale there.

For chip fabrication, same story. But for example, if you think about building some small metal part inside of that rack, there's not that same effect. >> We think the natural boundaries are on the chip side, on the bottom, and the model layer at the top. and we'll fill the whole gap between. >> A few weeks ago, there was a guy who was running a next generation AI chip for, you know, one of one of the frontier companies. and he's trying to recruit one of our architects. and this person actually kind of did an UNO reverse card and started recruiting the person trying to recruit our our guy and and within a week we hired him.

and I was going on a walk as like we were kind of finalizing the offer. I was like, well, you're leading this super important project. Why are you deciding to join? And his answer was super interesting which was it fundamentally is not existential for my company for this product to win for Google with TPUs like their revenue comes from search. >> Google won't fail if TPUs fail. >> That's right. Meta won't fail if MTIA fails. Microsoft won't fail if Maya fails and OpenAI won't fail if Jalapeno fails. Ultimately this is our product.

It it is like completely unsurprising that the best chip in the world is built by a company that only builds that chip. It's Nvidia, right? And like for us like it is completely existential for us to get as much to token capacity online as possible. And you know it recruits a set of talent and it recruits a support from suppliers and from customers that view it with the level of intensity that we do. Look at the raw flop stats. You compare any of these chips built by the labs or by the hyperscalers. The flop density for say FB8 times FB8 is lower than the black B300.

>> Y >> and that makes sense because they don't have to go take the risk. they just have to go and build a similar enough product and not pay the Nvidia tax. >> As I think about you guys building the solution, you're the process of doing so is is solving a sequence of really hard challenges. What has been the single episode that was the hardest to overcome? >> When we were designing the chip, we built this massive massive FPGA cluster to go out and verify the full chip worked as is. And FPGAAS are digital entities. You can go test digital logic but not analog logic. And it turns out that when the chip came back, we began to go see issues in our attention adaptation of incorrect results. And we realized, wait a minute, there's a problem where the back pressuring logic across a clock domain crossing is failing. And this is going to cause the chip to produce wrong results. And it is very very hard to solve. And we realized that there was one and only one way to solve it as we had to go line

up two clock signals on our chip to within 50 picosconds.

That is literally 50 trillionths of a second. And we had to go get these signals aligned to this super small granularity and do it on every chip two billion times a second. >> A lot of people said this was impossible. >> We we had people quit. >> Yeah. That people literally were like this problem is unsolvable. And best of luck guys. Well, when you have a problem like that, step one is okay, let's assume the problem is solvable.

How would it be solved? Well, first we realized what we have to be able to do is find a way to go ahead and move our clock phase by a picoscond, 10 picosconds. And we had an idea. What if we had these two clocks? We set them just a little bit apart from each other. And we figured out that hey, if we go out and figure out the phase, if we then go out and use the drifting mechanism to go ahead and wait for just the right amount of time to get that 50 picosconds always lined up, we can do this extremely reliably and then lock the phases exactly where they have to be. We can guarantee this never happens. People were, I think, somewhat blown away that this worked and it worked as well as it actually did. but we made it work.

>> How long did that take? This is actually about two weeks that it >> a dark two weeks. >> It was a very scary two weeks, but it was the kind of thing where when that kind of thing happens, that is the most important time to go ahead and be investing effort. That that is the hardest time to go do it when you feel like things are hopeless. But like the sooner you solve that problem, the sooner you can get back to building and scaling up production to mass volumes.

>> I think a lot of our story is like as Gavin says, assume it is possible. Like assume it is way possible like possible to have a chip with way more flops on it. Assume it is possible to have a system with way lower latency between chips. Assume it is possible to create a shared memory pool that can run at way higher bandwidth. Like how would one do it? A lot of the time when we do experiments, we will dozens of experiments and all of them will fail.

But like we only need one to work. There was multiple times. I mean Gavin I think was was leading the charge when our and our chip bring up with I think 30 different board experiments and and three of them worked and all three of them are worth their weight in gold. >> This is one of the things where people come to me and say Gavin almost none of your experiments works and I would say I only got to get lucky once.

Ridgeline is the first endtoend system of record with embedded AI for investment management firms running portfolio accounting, reconciliation, reporting, trading, and compliance on one unified platform. Firms are moving off legacy technology and onto Ridgeline because of how far ahead Ridgeline's AI features are compared to anything else in investment management software. I've been hearing from a lot of investment managers about AI and they fall roughly into two camps with some unsure of where to even start and others convinced they can build their own order management system over a weekend. The reality is that running an investment firm will always require governance controls and a single source of truth for your data and no amount of AI enthusiasm changes that requirement. Ridgeline is built on exactly that foundation, which is why I believe that firms that come out ahead in the AI era will be the ones running on Ridgeline's unified platform. If you're serious about your firm's AI strategy, Ridgeline should be part of that conversation. You can request a demo at ridgeline.ai.

So, one idea for one of these stories that I'm asking about, you know, difficult moments in the company's history is around the ability to raise capital to fund the thing. I think when you started it, you knew you'd need capital, but you did not know you'd need the quantum of capital that you've ultimately raised and and are spending to build the solution and you hadn't raised money before. These are all new things, right? And there was moments where it was really really difficult because I was I was there, I saw it. There moments where it was extremely difficult to raise the money that you did that without which the company would not exist. It would have died. And like many great stories, you know, there there are many near-death moments, but money specifically in this new world. This isn't software. You don't just need a little bit of money.

And then maybe you could tell the story about the true hardest part about raising money early on. Before you had something that you could show people and be so proud of and performance that you could show them and blow their blow their socks off, it was just you guys talking about an idea. But talk about the early difficulties raising money because it was pretty hardcore. >> We've had some intense moments. it reminds me of probably early 2024 before we raised our series A and we were at this point where we had done enough of the architecture, done enough of the design that we knew that the chip like architecture was sound. We had to go build it. There was a lot more to do.

We were ready to go into what's called the physical design stage. We needed to sign an agreement with a physical design vendor which you know will cost you at least $40 $50 million. and then we had this realization as the models were getting bigger and bigger and you're seeing these giant models come out that we were going to need to build the entire cluster, not just the chip, but we were going to need to build boards.

We're going to need to build interconnects. We need to build cold plates. We're going to need to figure out all of the networking and everything. And that this was going to cost a lot more than the $15 million we had in the bank. >> And you're like, man, that was scary. >> Yeah. >> That like you're sitting in that moment, you think, holy crap, we can't afford this. and I began looking at like, huh, how hard is it to go back to Harvard?

>> I mean, I mean, at the end of 23, we put together this like memo. I mean, we spent like a hundred hours on this cuz we're like, >> we have no idea how people are going to believe us when we ask for the amount of money we're about to ask for. and it was like 30 pages. It was like extremely technical and in-depth of all the different

things we needed to build and like all the milestones we needed to hit and how the market was going to evolve and all the new use cases and the cost per token and all this modeling. And then we went and talked to investors and every major investor in the valley passed immediately. They were just like, "Okay, two kids that just finished Harvard, haven't taped out a chip, no test chip, you know, inference like you know, who knows if this is going to be a big market. Everything's going to be training. you know the models still hallucinate. This could all you know all be a bubble. You know at the time the biggest semiconductor fundraisers for a series A was around like 40 $50 million. We were looking at this and we were just like tallying the bill. We're like we think we're going to spend $100 million in the next 12 months. Like if we really want to do this like if we want to actually get to scale and like actually get the performance we're talking about like this is going to be extremely capital intensive. How the hell are we going to pull this off? I think that one of our key ways we got started in this process, >> we thought to ourselves, what is the cheapest possible way we do this? And we decided, well, if I made almost nothing.

>> Yeah. >> And if I ate nothing but ramen, then we would go ahead and spend basically just the money for the mask of 11 tape and that would be that, right? >> And if so, we could probably do it on $30 million, an obsanely low number. And we actually went out and got a debt provider to be willing to go ahead and lend us the money we needed to go cross this barely barely ramen to a chip threshold. From there I think it was first to go catalyze a series of other hey maybe we can go do one more thing one more thing. One more thing.

Yeah. So we're at this moment where we're like if we really want to build this company because we're not going to halfass it. Like we're not going to go do a test chip and spend years on it and like let the entire AI market boom like while we could be building the product. Like if we're going to do it, we're gonna go all the way. We're going to need to find a way to get $100 million.

I mean, I remember Gavin and I were like sitting down in the office in Certino late at night just like looking at each other and we're like, could we cut 500k here? Could we cut 100K here? How long could we convince everyone not to take a salary? And we're like, holy the math is not going to close. we we we really need to solve this. And there was a period of a few weeks where you kind of just go into survival mode and you call every person that could possibly know an investor and you're like, "We need $100 million to do this. If we do this, we think this could be one of the most important companies of all time. Do you know somebody that wants to take an aggressive bet that wants to like believe in us? Like here's all the information. Like we're an open book.

Here's the team. It's great people. We've been working super hard. we've done these things in record time, but we have these, you know, hundred things to go. Like, do you want to do this? And like the snowball starts and you get a million here and 2 million here, and you're like, "Okay, we're not going to run out of money this month." You get a $5 million check, $10 million check, and you're like, "Okay, maybe I can buy those FPGAs." And, you know, the snowball snowball happened where, you know, we were very lucky that we ended up putting it all together. I mean, we had a board meeting. I show you the spreadsheet and we look at it and it's like 103 million. It's like these are all like soft commits and we all look at each other and we say we're going to take it and that was the series A and luckily I I think it it's been much easier since then and we've raised almost a half dozen rounds since then.

Many of them from those investors just doubling and tripling down. That's allowed us to get to market so quickly. Like this rack would not be possible had we not have been so aggressive. >> I also think like suppliers too I think deserve a little bit of a commenation here. Yeah. that TSMC was willing to work with us back before we'd raised any of that hundred million dollars back when it was still really really scary actually went ahead and let us get some of their emulators on extremely favorable terms where we pay over many years.

>> Yeah. basically a big loan and like it takes a lot of belief from your partners to go do this but at the end of that you come out with this very strong team and all the folks who back you are not in it just out of pure financial incentive they believe >> why did TSMC believe do you think >> this is a great story even before you joined in there was a conference as event and I was one of the only young CEOs of semiconductors I think it's kind of a novelty asked me to come in there and speak. I get to the semi event and I am the only speaker there under 40 and only person there under 30. I was at the time 22 22. So I go up and I speak. There was a speaker dinner afterwards and by pure luck happen next to this very senior TSMC VP. It's a very nice dinner. There's like the former CEO of ARM there. It's very bougie.

Everyone's in a suit and I'm there with this VP and it turns out we both studied math in college and we go we get a little piece of paper and we begin talking in great detail about how do modern AI models work at the actual per tensor by tensor level and the guy just gets it and we begin talking about hey how do you run this very effectively? why is Louisville such a critical technology to make this work? And the following day, I get an email from TSC saying, "Gavin, I want to work with Etch, >> find a way to make it happen."

>> Crazy. >> They've been a great partner ever since. And >> it's amazing to think about some of the tropes and obviously like should break the fourth wall here. Like I'm a big Etch investor. I've been involved for a long time. I think the absolute world of you guys. So like I'm incredibly biased, you know, in this conversation. I'm trying to ask, you know, questions that are that are broader and interesting and could be objections to what you're doing and we we'll keep doing that. but it's so interesting to me that like when you read about investing, everyone cites this idea of like contrarian and right as the quadrant that makes all the money and it sounds really nice, but contrarian means like everyone else thinks you're stupid. And so when you go and you you get immediate nos from literally everybody, it is a fascinating quadrant to exist in before you become consensus.

>> I mean, what was it like for you? I'm super curious. >> Well, it's interesting. At the time, it was the it was the largest by a lot first check that I had written. Sice to say, I'm not a a math expert or a semiconductor expert or an AI expert really at the time. and so it was much more of a believed in the concept of this market potentially being huge. you having made very very clear bets on how the future was going to look, having positioned the company in order to attack those things in a in a in a hardcore way. and then just the two of you and and what I felt about you was the majority of the reason why we made the bet when we did in 2023 or whatever it was. but at the time it was it was the biggest. And I think the same thing you said about naivee applies to investing as does to maybe building a semiconductor startup which is like I didn't know what I didn't know. And when I called experts they were basically like this is stupid. They laid out in in very logical terms like why this wasn't going to work and why it was such a low probability bet. And I think one of the things I've learned from it is just like you kind of have to damn the base rate.

Like if you invested on base rates you should do something other than what we and I do. There's always the index fund. >> Yeah, there's always an index fund. Exactly. So, it's actually never been scary for me. I think probably most of that is because I don't know. There's a lot I don't know. And if I knew more about what you guys have done and the difficulty, I probably wouldn't have done it. I don't know what that says about like maturing as an investor. Like maybe I don't want to know, you know, a lot more and and have some of that healthy naive. I don't know.

>> Funny. I mean, I think a lot of the traditional semiconductor funds missed the entire AI chip like all the AI chip companies. and like all the coding experts missed all the coding companies and like I think it's very hard to realize that constraints have changed and like you know when you've looked at tape outs for 20 years and you've seen so many of them not work on the first try or the second try or the third try like you couldn't even run a workload you totally forget that EDA tools are way better and that FPGAAS like exist today in a way that they didn't exist before and all the types of validation you can do today just wasn't possible so I think for us a lot of our believers either like they were kind of on two sides of it.

They were just believers in the market and the team or they were building chips today and extremely technical like the high frequency trading firms where they would literally audit everything from the micro architecture and the RTL to like the board designs and like the schedule and the software stack and we would sit down with like 10 of their people who build their own chips and they're asking us such detailed questions that we're wondering are they going to build the chip? It was really on either of those sides. And if you were anywhere in the middle, like you just wouldn't understand it.

>> In the investing world, they often talk about variant perception, something that you see or believe that others don't, right? And and and that perception creates the opportunity. >> I think I've invested, I don't know, five or so times in that. And every time when you do it, stakes are getting bigger and bigger. And so it it does get a little scarier and scarier. and and because you guys have been so quiet in the marketplace, I think it's very easy to dismiss you. As the stakes get bigger and bigger, those dismissals are harder to hear.

>> And and so I I do think like betting on something that you see when what you hear from the outside world is very different like that that perception gap is is oppos. >> Exactly. The last thing I would say is the accumulated evidence of your guys's and your team's ability to solve seemingly impossible problems is one of the most interesting like things a company can have. It's like a binary like companies do this or they don't.

>> That's the thing is a big advantage of people who have been here for a long time. You get some new joiners who are scared shitless. When you see a thing like this and there are oldtimers who've been here for all of two years >> smoking cigars in the trenches. Another one. Another one. >> Yeah, there's definitely a find a way mentality. If you're here, you're here because you assume it's possible. So, like we can't be saying it's impossible.

Everything is solvable and we're just going to work at it until we figure it out. there's a f a favorite story I have this guy who's kind of a legend in in silicon validation who who joined our team and we were doing the early stages of what's called wafer sort. and you know when your chips are coming out of fab, they go out on these

wafers and you have this thing called a probe card that attaches to the wafer before you dice it with these probe pads and you send these electrical signals to basically test which chips are good and bad. So when I, you know, slice the wafer into a bunch of chips, I can package it and only package the good ones. We go through our first wafer and, you know, it's like 2 3:00 a.m. because we're doing it with like TSMC over the phone in Taiwan. We have the screen with the wafer that's all gray and each chip is gray and then as you start running the patterns the the squares are supposed to turn green or red and they all turn red. We're like like this is really bad. Like everybody's like guys take a breath. He leans back and he's like the puzzle begins. like that. You have to have the attitude of like, yes, you will go out and you will go stare into the abyss and you will go see scary things and we'll solve them.

>> When did you see the first green square? >> Within a day of that, but in the moment you're like, I have worked for years for this. I put my life on the line. I've asked my family to like, you know, stake everything on it. And then like it's red. It's extremely scary. And there's a certain type of person who just like is addicted to that feeling of just feeling the fear and solving it. And you know we are lucky to have a lot of those people here.

>> If you think about applying all of this earned knowhow from this last several years and now thinking ahead to Gen 2, Gen 3 and beyond. >> Sure. What will you be doing most differently as a result of everything that you've learned? Just like from a conceptual standpoint, like the way that you will attack designing and producing this next one based on what you learned doing it the first time. >> It took us a while to get to the primitives that we think are really what matters for scaling inference. We tried a bunch of things early on from compilers that would turn different models into FPGAAS to burning weights in silicon to splitting your HBM to KV cache and weights and all of these different things. and there was a lot of cycles of learning till we got to the point that we realized that like fundamentally if you want to run a majority of the tokens in the world. You need to do three things. You need to build a chip with the most flops in a given power budget. You need to build a chip that has the lowest latency between other chips. So the biggest scale up domain possible. And you need to produce as much of it as possible.

>> And I think probably in the first half of our journey so far, we learned the first two and that informed the design a lot. And that informs a lot about the bets we're making in the future with the low voltage inference and the cluster scale memory. But the production part I think in the past year has become extremely obvious. How much people want to deploy this stuff if you can have it available today. You know the best ability is availability. You know if I have a thousand chips today someone's going to use them. And you know we need to build a chip that's not just like way better than what's been built before but it needs to be available at many gigawatt scale. We need to be able to be building a product that is producable at gigawatts per month from the limit. As we think about that, a lot of the design decisions we're making with our nextgen, which which you've seen already, is just about simplicity, removing tons of parts, trying to assemble and disassemble a thing again and again, and learning how to make it as quick and the cycle times as possible in production, making sure it's going to be reliable, making sure it's going to be serviceable, and making sure it's going to be producable at gigantic scales. What about other problems in the ecosystem that are outside of your control such as capacity at the leading nanometer at TSMC or availability of HBM4 memory or you know some of these other things where like everyone is fighting for scarce unit of capacity or whatever. How do you face up against those realities when you're trying to produce as much as humanly possible? The people deploying the most comput in the world do

think about supply a bit zero sum which is there's only so many wafers being produced on a given nanometer node on a given fab right and there's only so much memory being produced and that's why actually for our firstg product we built it on a different supply chain than the Rubins so you know we're on 4nometer Ruben's on 3 nanometer you know we're on a different HPM than Reuben's and so forth so it actually is not a zero sum thing it's a positive something where more is more so often when we're talking to people deploying at scale.

It's not a decision between a gigawatt of a GPU and a gigawatt of us. It's 2 gawatt. and I think as much as possible thinking about supply chain early in the design decisions because if you have the most performant product and you can't produce it, you know, then you're just a podcast. That's other big thing about vertical integration too is or certain things like for the chips and for the memory, you have to go ahead and partner for most of the other stuff. Those are also very highly in demand components. And the more that you build yourself, the more stuff you can go do on top of what the world can currently build. It is not, oh, you're taking availability from somebody else.

You're adding way, way more. And, I think that's how you win. >> One of the things I realized we haven't talked at all about is the models themselves, which is kind of crazy the things behind all of this demand. Anything interesting that you would say about the way that you see models progressing based on what we've seen so far? I guess I'm more interested in how you as thinkers about hardware think hardware might impact where the models themselves go in the future. One of the most important ideas that we believe in is that machines don't think like people think. You look at airplanes for example that airplanes don't fly like birds fly.

That when you think about how mechanical devices have to work, it's often very different. And in much the same way for people storing data and loading memory is very cheap for neurons and doing math is relatively expensive and it is the exact opposite for chips. Generally loading data is very expensive and doing math is very cheap and as time goes on I think you will end up finding that math gets cheaper at a rate that is faster than memory gets cheaper due to this fundamental limit on any kind of DRM device. You should go ahead and think about how can I make my model use a huge huge amount of compute. What if I had for example many copies running at the same time? What if I activated a huge number of experts? What if I had gigantic experts that I can go ahead and run on multiple server acts at the same time? That is how I think you'll build models that are the next generation of intelligence and context too. There's been a lot of work on hey very efficient inference. what if I don't load the full context into memory and most of the time I think that makes a lot of sense you want to go build a super intelligence why can't it go look at a billion tokens of context why can't it spend a huge amount of compute to go ahead and read all that in a super fast I would love to go be able to talk to a machine that was able to go attend to well every book ever written and it's short-term memory and I think you're going to get to a point where you can't >> a theme in models right now is this focus on something called dynamism which is this ability to control the level of computation and memory spent at a per token or per user level when doing attention as well as this ability to dynamically in your chip on the fly send data to other chips for different models doing certain types of operations and you know the reason is fundamentally as we are scaling context length as we're scaling model size as we're scaling the amount of computation per user we're looking for ways to be more efficient so you know the first thing is Like Gavin says, you know, mixture of experts, you know, architectures where, you know, maybe we don't need every parameter being used for every token. but maybe there are things where even at a token level, we can say, well, this token needs this context from this other

token. They can share that memory, so we don't have to have overhead of of using the memory as much. maybe this token is really important, so we should spend more compute. We should have longer context on that token. So hardware that really accelerates these types of very dynamic computations extremely important. And you you can imagine current hardware that was designed before those types of architectures have lots of overheads in doing them. So you basically end up in these really bad worlds where you have inefficient hardware at doing this dynamism. so therefore you can't run it very well or you have these very blocky architectures that are kind of applying blunt force to many different tokens that all need more or less computation.

>> I have two questions about the future. we've talked a lot about what you built so far and how you built it. The first is about the new ways that people might start using these systems. the tech the raw technology, log of run times, you know, things of this nature. When inference gets gets much cheaper, faster, more accessible, there's more total supply and it's better. What are the things that you think people will use that capacity to do that are the most interesting, exciting to both of you? Yeah, there was a viral tweet by Noan Brown where he said that as these models are having longer and longer time horizons, they can do tasks that take say six months and there's often not enough time to go and evaluate them for such a long period of time because by that point you'll have a new model app you'll want to go evaluate instead.

And with tech like we built our cluster scale memory, you can go ahead and run that six-month job much faster. But there's a second piece of this too talking to him about it. He's now an angel as well where it's not just the time, it's also the number of people or agents who are working on this. If you're trying to go and evaluate can a human build a rocket, you will find that the answer is no. No one person can go build a rocket. Instead, you have to go put a team together. And I believe the same thing will be true of agents too.

If you want to go ask can agent go out and build some crazy futuristic piece of software, you'll probably need a very large team. maybe that's 10, maybe that's a million. You have to go and have this enormous amount of both cluster scale memory to go ahead and have that very short time per token and a huge amount of flops to be able to go run that whole fleet. I'm going to be a little futuristic. I firmly believe we are on a global march of inference becoming majority of global GDP. and it may may take more than 10 years, but it's it's going to happen. and right now we measure productivity as a society as GDP per capita, but really it's going to look much more like agents per megawatt or maybe agents per gigawatt by then. And while we're being futuristic, I think this is the second to last year where a majority of the workforce is going to be human. I think in 2027, you're going to see there's going to be more agents doing knowledge work than humans. And it's going to be extremely interesting to see what happens. You could imagine a world where for countries a majority of their energy ends up going into data centers doing inference and the energy efficiency of those data centers basically governs how many agents and therefore you know how big their workforce is. So you're going to see like as Gavin is saying you know right now we have you know one agent or a team of five to 10 agents working on group projects for a couple days. So you can do pretty cool stuff because they're smart but it's not going to be civilization scale. What happens when you have countries that can have literally a billion concurrent agents, like a billion people in the workforce working 247 concurrently on the same stuff. I mean, it's it's just kind of unfathomable what's going to happen.

and it's going to be the biggest proliferation of technology humanity's ever seen. >> I think as well, like when you have these huge huge amounts of demand, you get this idea of economies of scale again. >> Yep. >> Or if you think about people, I have a brain. I'm not using the whole thing all at the same time. That only a part of is going to be active. And this is the way healthy brains work. And for MOE models, it works much the same way. On MOE model, only a small fraction of the parameters is being used for any given token at any given moment. But if you have a large number of users on a piece of hardware, you can go kind of take that brain, cut it up into many different experts on many different servers, and run a huge amount of volume through it. So you'll have a bunch of different pieces of traffic. You'll have many of them using each part of the brain at any given point in time. And you'll also make the cost per thought, cost per token way, way lower.

>> So I think you're going to end up with these giant scale distributed brains which is going to look like the real the form factor of this is a big data center with a bunch of chips, a huge amount of flops, and a huge amount of scale up interconnect. >> You think we'll see a trillion dollar individual data center? >> Absolutely. It is a matter of time. It's like asking what you see a billion dollar fab or a10 billion dollar fab or hundred billion dollar fab. It is inevitable that the economies of scale don't stop at oh $40 billion is the magic number for fabs. No, the cost wafer keeps going down as you keep spending more money. And the same thing will be true of say plants that go out and make steel or plants that go out and make tokens. a very smart alien lands on on Earth and wants to know from each of you how you would frame up this opportunity that that you guys are tackling. What what do you say to them?

>> Frame it as thinking is really valuable that every company in the world runs on thinking and we are entering this really unique moment in time where you have machines that can go think almost as good and as soon as good and as soon better than the best humans can. Building these machines is going to go be a huge opportunity. But more important than that, the way in which you go ahead and run this kind of thinking is going to be very very different as demand goes higher and higher and higher and higher. There's a unique moment right now to go build a new set of solutions, a new road map for how do you run the future quadrillion parameter models for a billion people all at the same time on a gigantic scaleup cluster. We are in a new era of intelligence where the cost of producing intelligence is dramatically so much cheaper than the value of the intelligence that we are in a many year probably many decade supply shortage of these tokens. And basically any chip or any system that can produce tokens is likely to be extremely valuable. And you should find some part of the supply chain of the token. it can be everything from model training down to what we're doing in the silicon and otherwise to spend time on and push the frontier. and that the companies that are the largest are frankly going to be the companies that produce most of the global supply of tokens and own a majority of the supply chain of that token. And importantly, it's people who build systems that as they get more and more chips put together cheaper. That the way you want this to scale is not that, oh, if I want to go serve 10 times more tokens, I buy 10 times more servers. It must be some solution where I want to go serve 10 times more tokens, then I get some economies of scale benefit with my cluster scale memory tech that allows me to then not charge as much as 10 times more for those stacked tokens. What a ridiculously exciting future that you guys are building to enable. When I did this with Gavin last time, I asked him my traditional closing question. So, this time I'll ask you, what is the kindest thing that anyone's ever done for you?

>> During my cancer treatment, there was a big decision I had to make. the doctors came to me and said, "It's

time for you to decide. Do you want to get surgery or do you want to get radiation?" Here's the trade-off. If you get surgery, you're more likely to live, but you have to assume you'll never be able to walk again. if you get radiation, you'll be able to walk again, but it's not the same probability that you'll live. You may die. What do you want to do? And I was 16. and my parents, I said, you have to make this decision for yourself. and I thought a about it for a long time and decided I'm going to do the surgery. I get the surgery. One of the things you do when you get a tumor resection is they do something called a necrosis analysis where they look at all the different cells and say is a cell dead or alive because if you have a bunch of cancer cells that alive you have a problem. And they looked at it and they said you know you usually want 98 99% necrosis for us to say you're in the clear. You're below that. you should go get radiation.

and there was only a few machines in in in the world that actually could do the type of radiation I needed. one of them was in Boston. I was in a wheelchair and I needed to move to Boston for multiple months. And both of my parents decided to move out and drop everything they were doing and and live with me. and I'm eternally grateful. beautiful. Thanks, guys. Amazing conversation. Your finance team isn't losing money on big mistakes. It's leaking through a thousand tiny decisions nobody's watching. Ramp puts guardrails on spending before it happens. Real-time limits, automatic rules, zero firefighting. Try it at ramp.com/invest.

As your business grows, Vant scales with you, automating compliance and giving you a single source of truth for security and risk. Learn more at vant.com/invest. Ridgeline is redefining asset management technology as a true partner, not just a software vendor. They've helped firms 5x in scale, enabling faster growth, smarter operations, and a competitive edge. Visit ridgelineapps.com to see what they can unlock for your firm. Every investment firm is unique, and generic AI doesn't understand your process. Rogo does. It's an AI platform built specifically for Wall Street, connected to your data, understanding your process, and producing real outputs. Check them out at rogo.ai/invest.

The best AI and software companies from OpenAI to cursor to Perplexity use work OS to become enterprise ready overnight, not in months. Visit works.com to skip the unglamorous infrastructure work and focus on your product.