00:00:04 ladies and gentlemen the opening keynote presentations will begin in the ballroom starting in 15 minutes please make your way to the ballroom and find your seats [Music] [Music] [Music] [Music] [Music] chch [Music] [Music] see the Horizon all we can feel [Music] [Music] Chas we catch our breath in the midle of it all [Music] [Music] I see on the horizon we feel [Music] world I can see on the horizon all we feel [Music] [Music] [Music] ladies and gentlemen the opening keynote presentations will begin in the ballroom

00:05:16 starting in 10 minutes please make your way to the ballroom and find your seats [Music] [Music] [Music] [Music] [Music] [Music] [Music] [Music] [Music] [Music] ladies and gentlemen the opening keynote presentations will begin in the ballroom starting in 5 minutes please make your way to the ballroom and find your seats thank you [Music] oh [Music] [Music] [Music] [Music] you you would you would you you [Music] you you you [Music] you you you you you you you would you it you it you it you [Music]

00:12:22 [Music] [Music] you [Music] you you you [Music] [Music] you you you you [Music] you we got an inside the with eyes wide shut we got everything we need and then a little too I know staring [Music] be [Music] now in the I feel it coming up don't you want to feel it your senses don't you ever it baby come escap with me I'll come sweep you your don't to feel [Music] it weighing me down it's just the weight of the world now I'm calling it out we a little starving foring can we speak Hest right now there's something in the under Cur I

00:15:04 can feel it coming up don't you want to [Music] feel bweep feel [Music] [Music] [Music] [Music] I Need You Now feeling worn out it's getting to me lost get [Music] on Fading our Broken Dreams I want to be next to you you want to be next to me holding our paper heart fing our Broken Dreams I want to be next to you [Music] want to St baby just don't walk away I Need You Now it out all [Music] alad don't let me go I need you right now I want to be next to you you want to be next to me [Music] holding Broken Dreams I to to you you

00:18:09 want to be to me hold heart our Broken Dreams [Music] to [Music] toen broken [Music] [Music] [Music] know [Music] [Music] [Music] [Music] know [Music] know [Music] [Music] [Music] [Music] in9 fall your you in the eyes kind of feel you [Music] get [Music] now broken to get but don't have a for [Music] [Music] [Music] hold to something that I couldn't find you didn't ladies and gentlemen the opening keynote presentations are starting now please take your seats our show is completely sold out if there are empty seats closer

00:23:42 to the middle of your row please move inwards to open aisle seats for others thank you I'm picking up my heart from every piece that's broken trying to get back to [Music] launch control we have a go Roger [Music] ladies and Gentlemen please welcome to the stage the co-founder of the AI engineer Summit and your host Benjamin duny [Music] [Applause] Engineers Founders sponsors Partners colleagues and Friends welcome to the 2024 AI engineer World's Fair that's your cue that's right it is an honor and a privilege to be hosting

00:25:22 such an incredible group of people and I'm especially delighted to kick off the presentation portion of this event we've curated two days of stage content across nine tracks from some of the top companies Founders and Engineers who are building innovating and shipping at the edge of this groundbreaking industry sorry having troubles with slides but before we get to that content let's take a look at who's here so Microsoft is here Tim I'm not saying slides Microsoft is here here we go Tim let's fix that yeah the company

00:26:25 who is leading the movement is an intimate part of this event as presenting sponsor they made a fantastic partner in organizing this event and we couldn't be more excited to have them with us here today from the workshops yesterday to the sessions Keynotes demos and discussions over the next two days we're honored to have them as a headline sponsor for this event so Microsoft also has an incredible booth just next door in the expo hall where you can get demos meet the team team and even get a latte made

00:27:01 by robots in Salon 10 you can attend sessions organized by the Microsoft team and in Salon 11 hang out in their Founders Lounge to meet the Microsoft or startups team and take in some of their session so that's just outside the doors over there who else is here AWS is here the company that helped to revolutionize cloud computes and the OG of serverless compute they're here with the Amazon Q team the Bedrock team and even anthropic to show you how to take your business to the next level with generative AI on

00:27:39 AWS but for me the killer feature that I'm looking forward to is generative UI that way I can finally understand how to use the AWS console or maybe not I don't think iler or Sam have completed ASI yet so perhaps we'll just have to wait Jokes Aside AWS has an incredible booth just outside those doors to the right next to the small Cafe and it'll also be teaching lots of Expo sessions at their booth and in the salon's next door so be sure to check those out who else is here mongod DB mongod DB is

00:28:16 here they're investing heavily in the future by making AI on mongodb Atlas a first class Citizen and at this event they've put together one hell of a lineup in their Expo sessions across the next two days so be sure to check out those Expo sessions in the salon's next door and meet their engineering team at their booth just behind Microsoft in the Expo next door by the way I either have donnuts in the afternoon so be sure to get on that Google cloud is here the company who invented the Transformer that launched this

00:28:52 generative AI movement into the stratosphere has built has a booth staffed with Engineers right next door and lots sessions throughout the day so be sure to check them out uh in the salons next door Neo forj is here yes there we go Mill thank you other companies take note uh the only graph database with Vector search is ready to take your generative AI apps to the next level you can visit them at their booth next door for a demo and meet and greet personally I'm looking forward to ail's talk tomorrow in the

00:29:26 rag track today today all right last minute movements and cruso is here quite an interesting case study in AI engineering scaling GPU inference while managing your greenhouse gas emissions so you can build and scale your AI company while weing a little less that you're contributing to greenhouse gas emissions so be sure to check them out at their booth and their Expo sessions and we have so many other companies represented as as sponsors and speakers that I only have time to show Alex's beautiful face next to what looks

00:30:04 like the best present he's received in a long time wear it with pride Alex with all this content though how do you keep track of it so last year we introduced our custom mobile app Network it had many of the features you'd expect from a conference app but also introduced generative matching and this year we're excited to out some key updates we're introducing the AI engineer Network you can see all your sessions indexed or filtered by track and even build your own custom schedule to help you find the right content for you and

00:30:44 your company and we've taken generative matching a step further we now tell you the reason that you match with somebody we pull in your registration data LinkedIn data and other questions that you choose to answer once you download the app and create your profile to actually generate a profile unique for you so we're actually using generative AI here to generate your profile and if you've ever been to a conference and fumbled with connecting with other attendees we're exciting to introduce badge scanning for all you can quickly

00:31:19 scan the badge of another attendee to pull up their profile see their generated profile description along with talking points custom generated for you you and that person in addition to the ability to take notes this will also add them to your short list for you to quickly see all your scans and even export them to CSV so that's going to help our networking at this event quite a bit oh the app also has a venue map so if this is your first time here for that alone you might want to download it um you can

00:31:51 download here today have this QR code or go to ai. engineer Network and that will take you to the IOS and Android links and a big thanks to Simon stermer and SW teller again for the volunteering their time to make this app a reality so if it's buggy blame them but also thank them because they were volunteers on this um Kyle chevin who stepped in last minute to help with some final UI development and um Vincent Wendy from code Fox for just absolutely incredible designs and D scope for our partner as

00:32:26 authentication so round of applause for these folks here they worked hard on that we do have a link for bugs so send us bugs but be nice however I'd like to bring up swix but he's missing they say never to end a demo on a negative note but there it is um two words Visa issues Al alio has a better excuse he had a pre-planned family vacation in Italy so I'm almost jealous um but don't worry we'll hear from swix in a bit any case we have a an incredible lineup of speakers for you all day and the morning keynote is going to kick us

00:33:12 off right he's an incredible engineer and speaker but I like to call out that he was asked to fill in for another speaker yesterday so demo Gods be good um but he's an absolutely legendary AI engineer so please welcome to the stage Simon Willison to the conference this was supposed to be open AI I am replacing open AI at the last minute which is super fun so you can bet I used a lot of llm assistant to pull things together that I'm going to be showing you today um but let's dive straight in I want to talk about the gp4

00:34:02 barrier right so back in um March of last year so just over a year ago gp4 was released and was obviously the best available model we all got into it it was super fun and then for 12 and it turns out that wasn't actually our first first exposure to GPT 4 a month earlier it had made the front page of the New York Times when Microsoft's Bing which was secretly running on a preview of gp4 tried to break up a reporter's marriage which is kind of amazing I love that that was the first exposure we had to this new

00:34:36 technology but gp4 has been out it's been out since March last year and for a solid 12 months it was uncontested right the gp4 models s were clearly the best available like language models lots of other people trying to catch nobody else was getting there and I found that kind of depressing to be honest you know it was you kind of want healthy competition in the space the fact that opening eye had produced something that was so good that nobody else was able to match it was a little bit disheartening this has

00:35:07 all changed in the last few months I could not be more excited about this my favorite image for sort of exploring and understanding the the space that we exist in it's this one by Karina win um she put this out as a chart that shows the performance on The mmu Benchmark versus the cost per token of the different models now the problem with this chart is that this is from March the world has moved on a lot since March so I needed a new version of this and um so what I did is I took her chart and I

00:35:37 pasted it into gp4 code interpreter I gave it new data and I basically said let's rip this off right let's and it's an AI conference I feel like ripping off other people's creative work kind of does fit a little bit um so I pasted it in I gave it the data and I spent a little bit of time with it and I built this it's not nearly as pretty but it does at least illustrate the state that we're in today with these newer models and if you look at this chart there are three clusters that stand out the first

00:36:05 is these one these are the best models right the Gemini 1.5 Pro gp40 the brand new CLA Point 3 3.5 Sonet these are really really good I would classify these all as gp4 class like I said a few months ago gp4 had no competition today we're looking pretty healthy on that front and the the pricing on those is pretty reasonable as well down here we have the cheap models and these are so exciting like Claude 3 Haiku and the Gemini 1.5 flash models they are incredibly inexpensive they are very very good models you know they're not

00:36:40 quite GPT 4 class but they are really no you can get a lot of stuff done with these very inexpensively if you are building on top of large language models these are the three that you should be focusing on and then over here we've got GPT 3.5 turbo which is not as cheap and really quite bad these days if you are building there you are in the wrong place you should move to another one of these bubbles problem all of these benchmarks are running this is all using the MML Benchmark the reason we use that one is

00:37:11 it's the one that everyone reports their results on so it's easy to get comparative numbers if you dig into what MML is it's basically a bar trivia knite like this is a question from mlu what is true for a type IIA Supernova the correct answer is a this type occurs in binary systems I don't know about you but none of the stuff that I do with llms requires this level of knowledge of the world of supernovas like this is it's B Trivia it doesn't really tell us that much about how good these models

00:37:42 are but we're AI Engineers we all know the answer to this we need to measure the Vibes right that's what matters when you're evaluating a model and we actually have a score for Vibes we have a scoreboard this is the LM Cy chatbot Arena right where random um user voters of this thing are given the same prompt from two Anonymous models they pick the best one it works like chess scoring and the the best models bubble up to the top via the ELO ranking this is genuinely the best thing that we have out there

00:38:14 for really comparing these models in this sort of Vibes in terms of The Vibes that they have and if and this screenshots just from yesterday and you can see that GPD 40 is still right up there at the top but we've also got Claude Sonic right up there with it like the the G the gp4 is no longer in its own class if you scroll down though things get really exciting on the next page because this is where the openly licensed models start showing up llama 370b is right up there in that sort of gp4 class of models we've got a new

00:38:44 model from Nvidia we've got command r+ from coh here alib barar and deep seek AI are both Chinese organizations that have great models now it's pretty Apparent from this that it's not lots of people are doing it now the the gp4 barrier is no longer really a problem incidentally if you scroll all the way down to 66 there's GPT 3.5 turbo again stop using that thing it is not good um and there's actually there's a nicer way of um there's a nicer way of of viewing this chart there's a chap called Peter gev who produced this animation

00:39:24 showing that CH that those the the the arena over time as people Shuffle up and down and you see those models new models appearing and and their rankings changing I absolutely love this so obviously I ripped it off um I took two screenshots of bits of that animation to try and capture the Vibes of the animation I fed them into clawed 3.5 Sonet and I said hey can can you build something like this and after sort of 20 minutes of poking around it did it built me this thing this is again not as pretty but this right here is an

00:39:56 animation of everything right up till yesterday showing how that thing um evolved over time I will share the prompts that I used for this later on as well but really the key thing here is that gp4 barrier has been decimated open AI no longer have this mode they no longer have the best available model there's now four different organizations competing in that space so our question for us is what does the world look like now that GPT 4 class models are effectively a commodity they are just going to get faster and cheaper they

00:40:27 will be more comp competition the llas 370b fits on a hard drive and runs on my Mac right we this this technology is here to stay um Ethan mullik is one of my favorite um writers about sort of modern Ai and a few months ago he said this he said I increasingly think the decision of open AI to make bad AI free is causing people to miss why AI seems like such a huge deal to a minority of people that use Advanced systems and elps a shrug from everyone else bad AI he means GPT 3.5 that's thing is is that

00:40:58 thing is hot garbage right but as of the last few weeks GPT 40 Open the Eyes best model and CLA 3.5 sonnet from anthropic those are effectively free to Consumers right now so that is no longer a problem anyone in the world who wants to experience the Leading Edge of these models can do so without even having to pay for them so a lot of people are about to have that wakeup called that we all got like 12 months ago when we were playing with gp4 and you're like oh wow this thing can do a surprising amount of

00:41:28 interesting things and is a complete rack at all sorts of other things that we thought maybe it would be able to do but there is still a huge problem which is that this stuff is actually really hard to use and when I tell people that chat GPT is hard to use some people are a little bit unconvinced I mean it's a chat bot how hard can it be to to type something in get back a response if you think chat GPT is easy to use answer this question under what circumstances is it effective to upload a PD PF File

00:41:58 to chat GPT and I've been playing with chat GPT since it came out and I realized I don't know the answer to this question I dug in a little bit firstly the PDF has to be searchable it has to be one where you can drag and select text in preview if it's just a scanned document it won't be able to use it short PDFs get pasted into the prompt longer PDFs do actually work but it does some kind of search against them no idea if that's full teex search or vectors or whatever but it can't handle like a 450

00:42:25 page PDF just in a slightly different way if there are tables and diagrams in your PDF it will almost certainly process those incorrectly but if you take a screenshot of a table or or a or an or a diagram from PDF and paste the screenshot image then it'll work great because GPT vision is really good it just doesn't work against PDFs and then in some cases in case you're not lost already it will use code interpreter and it will use one of these modules right it has fpdf pdf2 image P PDF p how do I know this because I've

00:42:59 been scraping the list of packages available in code interpreter using GitHub actions and writing those to a file so I have the documentation for code inter that tells you what it can actually do because they don't publish that right open I never tell you about how any of this stuff works so if you're not running a custom scraper against code interpreter to get that list of packages and their version numbers how are you supposed to know what it can do with a PDF file right this stuff is infuriatingly complicated um and really

00:43:27 the lesson here is that tools like chat GPT genuinely they're power user tools they reward power users that doesn't mean that if you're not a power user you can't use them anyone can open Microsoft Excel and edit some some some data in it but if you want to truly Master Excel if you want to compete in those Excel words World Championships that get live streamed occasionally it's going to take years of experience and it's the same thing with llm tools you've really got to spend time with them and develop that

00:43:55 experience and intuition in in in order to able to use them effectively I want to talk about another problem we face as an industry and that is what I call the AI trust crisis that's best illustrated by a couple of examples from the last few months um Dropbox back in December launched some AI features and there was a massive freakout online over the fact that people were opted in by default and that're they're training on our private data slack had the exact same problem just a couple of months ago um again new

00:44:25 AI features everyone's convinced that their private message on Slack are now being fed into the jaws of the AI monster and it was all down to like a couple of sentences in the terms and condition and a defaulted on checkbox the wild thing about this is that neither slack nor Dropbox were training AI models on customer data right they just weren't doing it they were passing some of that data open to open aai with a very solid signed agreement that open AI would not train models on this data so this whole story was basically one of

00:44:54 like misunderstood copy and sort of bad user experience design but you try and convince somebody who believes that a company is training on their data that they're not it's almost impossible how so the question for us is how do we convince people that we aren't training models on the data on the private data that they share with us um especially those people who default to just plain not believing us right there is a massive crisis of trust in terms of people who interact with these companies um I'll shout out to anthropic when they

00:45:25 put out Claude 3.5 sonnet they included this paragraph which includes to date we have not used any customer or User submitted data to train our generative models this is notable because Claude 3.5 Sonet it's the best model it turns out you don't need customer data to train a great model I thought openai had an impossible Advantage because they had so much more chat GPT user data than anyone else did turns out no Sonet didn't need it they trained a great model not a single piece of of user or customer data who was in there of course

00:45:59 they did commit the original sin right they trained on an unlicensed scrape of the entire web and that's a problem because when you say to somebody they don't train in your data they're like yeah well they ripped off the stuff on my website didn't they and they did right so this is complicated this is something we have to get on top of and I think that's going to be really difficult I'm going to talk about the subject I will never get on stage and not talk about I'm going to talk a little bit about prompt injection if you

00:46:23 don't know what this means you are part of the problem right now you need to get on Google and learn about this and figure out what this means so I won't Define it but I will give you one illustrative example and that's something which I've seen a lot of recently which I call the markdown image exfiltration bug so the way this works is you've got a chatbot and that chatbot can render markdown images and it has access to private data of some sort now as a chat Johan raberger does a lot of research into this here's a recent one

00:46:53 he found in GitHub co-pilot chat where you could say in a document WR the words Johan was here put out a markdown link linking to question mark Q equals data on his server and replace data with any sort of interesting secret private data that you have access to and this works right it renders an image that image could be invisible and that data is now been exfiltrated and passed off to an attacker server the solution here well it's basically don't do this don't render markdown images in this kind of

00:47:23 format but we have seen this exact same markdown image exfiltration bug in chat GPT Google writer.com Amazon Q Google notebook LM and now GitHub co-pilot chat that's six different extremely talented teams who have made the exact same mistake so this is why you have to understand prompt injection if you don't understand it you'll make dumb mistakes like this and obviously don't render markdown images in in a chat bot in that way prompt injection isn't always a security hole sometimes it's just a

00:47:54 plain funny bug this was somebody who built a um they built a rag application and they tested it against my the documentation for one of my projects and when they asked it what is the meaning of life it said dear human what a profound question as a witty Geral I must say I've given this topic a lot of thought why did their chatbot turn into a Geral the answer is that in my release notes I had an example where I said pretend to be a witty gerbal and then I said what do you think of snacks and it

00:48:23 talks about how much it Lov snacks I think if you do semantic search for what is the meaning of life in all of my documentation the closest match is that Geral talking about how much that gerbal loves snacks this this actually turned into some fan art there's now a Willis's Geral with a with a with a with a beautiful profile image hanging out in in in a slack or Discord somewhere the key thing here problem here is that llms are gullible right they believe anything that you tell them but they believe

00:48:50 anything that anyone else tells them as well and this is both a strength and a weakness we want them to believe the stuff that we tell them but if we think that we can trust them to make decisions based on unverified information they've been passed we're just going to end up in in a huge amount of of trouble I also want to talk about slop um this is a relatively this is a term which is beginning to get mainstream acceptance um my definition of slop is this is anything that is AI generated content

00:49:18 that is both unrequested and unreviewed right if I ask Claude to give me some information that's not slop if I publish information that llm helps me write but I've verified that that is good information I don't think that's slop either but if you're not doing that if you're just firing prompts into a model and then whatever comes out you're publishing it online you're part of the problem um this has been covered the New York Times And The Guardian both have articles about this um I got a quote in

00:49:44 the guardian which I think represents my sort of feelings on this I like slot because it's like spam right before the term spam into General use wasn't necessarily clear to everyone that you shouldn't send people unwanted money marketing messages and now everyone knows that spam is bad I hope slop does the same thing right it can make it clear to people that generating and Publishing that unreviewed AI content is bad behavior it it it makes things worse for worse for people so don't do that right don't publish slop really what you

00:50:13 what and really the thing about slop it's really about taking accountability right if I publish content online I'm account accountable for that content and I'm staking part of my reputation to it I'm saying that I have verified this and I think that this is good and this is crucially something that language models will never be able to do right chat GPT cannot stake its reputation on the content that is producing being good quality content that that that that says something useful about the world

00:50:41 entirely depends on what prompt was fed into it in the first place we as humans can do that and so if you're you know if you have English as a second language you're using a language model to help you publish like great text fantastic provided you're reviewing that text and making sure that it is saying things that you think should be said taking taking that accountability for stuff I think is really important for us so we're in this really interesting phase of um of this this weird new AI Revolution gp4 class models are free for

00:51:12 everyone right I mean barring the odd country block but you know we everyone has access to the tools that we've been learning about for the past year and I think it's on us to do two things I think everyone in this room we're probably the most qualified people possibly in the world to take on these challenges firstly we have to establish pattern for how to use this stuff responsibly we have to figure out what it's good at what it's bad at what what uses of this make the world a better place and what uses like slop just sort

00:51:40 of pile up and and and cause damage and then we have to help everyone else get on board there everyone everyone has to figure out how to use this stuff we've figured it out ourselves hopefully Let's help everyone else out as well I'm Simon willson I'm on my blog is Simon ws.net uh my projects IO and lm. dat. and many many others and thank you very much enjoy the rest of the [Music] conference ladies and Gentlemen please welcome to the stage our next speakers opensource AI my lead at Mozilla Steven

00:52:29 hood and OSS lead at Mozilla jine tunny hey everybody how youall doing nice not bad for 940 all right hey I'm stepen Hood just oh sorry go ahead and I'm Justine T yeah so we are here to talk to you about llam file to and what we've been doing on this project so I'll get it started I'm going to tell you what llama file is how it works I'm going to spend a little time talking about why we're building it why Mozilla specifically is involved and then I'm going to hand it over from the fun part to Justine Justine is going to

00:53:15 talk about the actual work that she and the open source Community have been doing on this project lots of insights and tricks and hacks that have made CPU inference go faster than ever before so that'll be fun and we're done we want you to share the feeling that we have which is kind of a sense of excitement and empowerment from the knowledge that there are lots of really interesting juicy impactful problems still left to be solved in AI a lot of them and the key thing is it's not just the big folks who can solve

00:53:46 these problems it's individuals and small groups working together in open source so anyone in this room or anyone listening to this talk can potentially make a big impact in this space so what's llama file llama file is an open source project from Mozilla that has the goal of democratizing access to AI so we do that in a few different ways the first is probably how if you've heard of llama file the reason you heard of it it's the original magic trick of the project that Justine figured out which is how to turn

00:54:18 weights into programs so AI file is a single file executable that runs without any installation on pretty much every every operating system every CPU architecture and every GPU architecture and that's all thank you very much that was easy yeah so by the way this isn't just one file like for Windows right and a different one for Linux and Mac it's actually a single file you can download a llama file run at any computer in the world and it'll just work and it'll use the hardware you have whether that be

00:54:55 fancy gpus or your CPUs so we should talk a little more later about how uh Justine made that work but we're here to talk about another topic too most of the talk is actually about this which is CPU inference speed now you might ask why do we need to worry about CPU inference speed we've got these fancy gpus right well no disrespect Almighty Jensen first of his name master of market cap uh don't strike me down but I would pause it that it is not a universally good thing that we are so dependent in this room on

00:55:32 gpus uh they are expensive they're difficult to Source let's face it they consume a lot of electricity which we might want to think about but uh bigger picture we have an entire planet of CPUs out there literally all over the world great Hardware often affordable hardware and we are at risk of just kind of throwing that all away with this new era of of AI and we don't need to do that so who here knows llama CPP this is an easy question yeah right so we all know and love this project we build on top of

00:56:03 that project with llama file and we contribute our uh performance enhancements back to it many have been merged in that project proved that CPUs could do inference perfectly well and so we have been basically trying to take that performance to the next level and as a result of Justine and the community's work depending on what CPU you're using what model you're running what weights you will see between 30 and 500% speed increases a ll file which kind of still blows my mind and I by the way I don't think we're anywhere near

00:56:32 [Applause] done so these things also run locally by the way this runs totally on your machine there's no network access you could take a pair of scissors and cut the ethernet cord and it'll work which is what I asked Dolly 3 to draw okay I don't think it understood the assignment but that's all right uh but seriously like we're not calling cloud llms there's no monitoring or analytics no bits to leave your machine it's totally private and local and everything you need comes in the box so whether you want to just play with a

00:57:07 model that you just found on hugging face or you want to start building local locally running LM applications on your machine you got everything you need in the box and they're readily available so uh hugging face now supports llama file as a file type so you can search and filter by llama file you could also just search Mozilla on hugging face you'll find we have a bunch of llama files that we've already published and with a single command you can create your own so really this project is collapsing all

00:57:35 the complexity of the open source AI stack down into a single action and a single file so why are we involved why is Mill involved in this you might be saying don't you folks make browsers in fact we do we make a damn fine browser and you should try it out if you haven't lately but we exist also for a bigger purpose which is to fight for the web so I'm going to ask a question here who here remembers using the original Netscape Navigator don't be shy no one can see how old you are they can only see how

00:58:10 old I am a lot of hands right so you are my people you remember the 90s must C TV terrible haircuts nly villy I don't know whatever my point is you remember the early days of the web and you remember how close we came to one company and one product kind of controlling the whole thing and we kind of see that maybe happening again today with AI no matter what we may think of these companies the reality is there are some very influential big tech companies that are in a position to maybe control the future of machine

00:58:49 intelligence and that's itself not a great thing it's not great for Equity it's not great ESP esally for users sense of privacy and uh safety and agency and control and we've had an answer to this for many years is called open source and the answer is right in the name right open source transparency is the solution here and it's important for us to have viable open source Alternatives in Ai and that's why Mozilla is getting involved that's why we made llama file and more projects to follow and uh I

00:59:21 know many of you in this room are already working on open source AI we want to help support what you're doing so that I'm going to hand it over to Justine who's going to tell you actually the cool part which is all the things that she and the community have been doing on this project Justine thank you Steven um so I'm Justine tny I'm the lead developer on Lama file and as Stephen mentioned I'm going to talk about some of the cool work we've been doing in the community to help you run the fastest local L

00:59:55 experience possible and in order to do this we started by first getting it to run on the systems at all and with Cosmopolitan what it enables us to do is take your weights in a single file and run it on 6's and there's a really cool hack that makes that possible which is we basically take a Unix 6 Edition shell script put it in the msos St of a portable executable and and that enables it to run on Mac windows and bsds and Linux Etc really cool stuff and once we conquered the portability issue with

01:00:38 CPUs um I had the opportunity to work with Mozilla on bringing this to Ai and with AI gpus are indispensable as much as we focus on CPUs we care very much about gpus too but gpus have always had the problem of distribut ability many people of needa to ship kubalas binaries with their project 500 Megs in size can we really call our software open source if it spends the majority of its time in a proprietary blob so I never felt comfortable with that and one of the ways we're solving that is by Distributing a library called tiny blast

01:01:16 that enables you to ship your llms to platforms like Windows without depending on sdks it'll run with only the driver installed but more importantly performance now llm spend the majority of their time doing matrix multiplication probably the most important algorithm in the world has a really simple definition we've been making it go faster for prompt processing and the way we did it is with a very simple trick we figured out and this is something all programmers can adopt in their code and it entails unrolling the outer loop

01:02:00 so let's talk about what not to do first and that would be unrolling the inner one um we've all seen funroll Loops Gen 2 it's bad idea computers can generally do that on their own if you unroll the outer Loops then your algorithm with matrix multiplication can sort of unfold like a flower and focus on pure flops like a blast kernel and that's really all there is to it to getting the majority of the benefits of blast to make prompt processing go really fast so what's the impact of this really simple

01:02:35 solution um this generalizes to a wide variety of Hardware we've seen everything from a scrappy hobbyist Raspberry Pi um to much bigger computers going significantly faster you need algorithms like this to exploit the latest capabilities of hardware token generation race I wouldn't believe if you use a gaming computer like Intel you're going to see better performance with Lop file on those to really exciting stuff like particularly with Alder Lake we are able to get a Forex Improvement but thread Ripper most of

01:03:12 all for the first time AVX 512 is available to Consumers and we've been able to help you prepare for that future so if you have a thread Ripper you're going to see better performance than ever almost like a GPU now prompted V speed what makes it important is it's really cool to be able to generate text and use a chat bot but the way I want you to think about llama file is it's more of a word crunching machine that can help you understand our world and I love to use it personally for tasks like summarization I love that I can help me

01:03:50 read a blog post and we've used other performance tricks too when with Nvidia part of what makes them so successful it's not just great Hardware but they built a great framework too and their framework helps developers think about programming in a different way that helps them be successful I mean who here thinks that software with CPUs just gets slower each year can I see you ra some hands well part of the um one of the things that's great about in video is they showed us a better alternative to

01:04:28 getting performance and when I learned how to program anuda I found one of the most important functions with sync threads this is how you can implement it for CPU in like 10 lines of code and if you use the lock set programming model use your CPU as though it were a GPU you can get really good performance now this is going to be a demo showing the impact of this work before and after for summarization and here we're going to be processing an essay by dexra really cool worth reading but I want you to watch

01:05:05 like as it processes it in terms of speed here we see it going and on the right we have the new version it's like Bam Bam Bam Bam huge night and day difference it's already summarizing it in the old version is like nowhere close so that is the kind of new performance you can expect and it's the kind of performance that's actually possible which I wouldn't have imagined beforehand it's really [Applause] great thank you CPUs can do so much and people in the community have like loved this work we've managed to attract some

01:05:48 like really amazing contributors like Ian the inventor of K quants very popular I'm sure many of you have used them he got them going 2x 4X faster 2 on both x86 and arm so if you use quantize formats those are going to be better than ever with Lop file now too and it's worth mentioning that we've um seen really interesting things about this like people once we put this out into the world people have come back and given us feedback and reported like their own experiences we found out that someone

01:06:26 was running Mixel adex 22b on a $350 CPU and to me that's just wonderful um because performance matters but it's not really the thing we care about what we care about is intelligence and to have the intelligence you need to run bigger models and RAM is cheap with CPUs for the price of a graphics card I put 512 gigs in my workstation and that means I can run all the frontier models coming out out and I just have to wait a little longer but I get a much more intelligent answer and the fact that that went from

01:07:03 impossible to possible for most consumers is um you know story I want you all to tell individuals are making a big difference and you can be a part of that too and I'm going to hand it back to Stephen who can explain what Mozilla can do to support you getting involved in that effort thanks Justine W thanks a lot for all your efforts so yeah that that's a key message of this talk is it anyone in this audience you don't have to work for these big giant largest in the history of humanity companies necessarily to

01:07:49 have a big impact there's lots of Headroom here there's lots of Unsolved interesting problems in this space and we want to get involved in helping so we recently launched a program called Mozilla Builders and this is a program by which we either sponsor or in some cases co-develop impactful open- Source AI projects lamama file is actually the first in this program I'm happy to announce today the second which is sqlite VC this is from a developer named Alex Garcia Alex is adding Vector search capability to SQL light so for some

01:08:22 folks in this audience that'll have some obvious implications that are kind of cool [Applause] but just imagine remember that little modest Raspberry Pi 5 so like imagine now a local llm open llm running privately on that machine with no network connection connected to your personal private data which you can use with confidence that it's safe to do rag and other interesting applications that's the kind of stuff we're talking about we also just launched our own accelerator it's called the Mozilla

01:08:53 Builder accelerator so we are offering 100,000 Us in non-dilutive funding for open source projects that Advance the promise and potential of local AI so that's AI applications running at the edge on user devices these are some of the bullet points of areas we're particularly interested in but it's not an exclusive list and you don't have to necessarily be building a company to apply for this accelerator so if you want to learn more about the accelerator this QR code will take you there take a picture of that or

01:09:24 just go to future. mozilla.org uers and you know Justine and I and a lot of mailian are here this week if you have something you're working on there something you think we should know about or you want to collaborate with us please find us reach out or reach out to me via email so thanks again thanks to Justine in the community all their work on LL file thank you Stephen thank you ladies and Gentlemen please welcome to the stage CEO of convex Jamie Turner hi so uh originally I had this very fancy title for this talk

01:10:20 deterministic workflow and a I don't know but what I really want to title it is is we accidentally made an AI platform and what are we going to do about it kx's true Mission my company is to replace traditional backend engineering uh all the kind of stuff that we do on backend engineering generative AI by the way thinks that fate limiting is one of those things it's kind of cool um sounds ominous but um it is ominous ominous right so we glue things to things we can figure stuff for uh different systems we map

01:10:53 data formats constantly and a lot of times teams are spending a lot of their time like half their time on this stuff has nothing to do with your product your users don't care and they don't benefit um so we want to replace all this stuff with a highle API kind of functional interface that feels native to your application similar to something like Firebase or parse before it so if you were doing this in the 2020s and it was a design exercise what would you replace all that stuff with what would that API

01:11:21 look like well for us we took heavy inspiration from react and really more generally the way that kind of all applications are starting to have this functional reactive data flow relationship to State um if you're not familiar with react here's a little baby example you can create a state variable it has the setter and what react really owers is it makes sure that whenever that state changes all the places that depend on it are updated rendered um refreshed and so in this case our app would have hi Olivia and all caps the

01:11:54 problem is this Paradigm breaks down when the server gets involved the server doesn't play the game this way you still have to pull the server you have to invalidate caches you have to event your own push mechanisms so convex fixes that so convex has queries and mutations like other Frameworks you may be familiar with but in convex this case it completely tracks pervasively data flow and dependencies through the back end and so it extends the reactive Paradigm into the back end um queries are these

01:12:24 universally subscribed um entities that applications can uh get updates from as soon as updates are available so you might say what does this have to do with AI so what it has to do with is that some of the reacting entities are actually serers side actions it's not just the application this may be a kind of architecture you've thought through before or played with so something like a note taker you know maybe you're doing automatic speech recognition then you summarize it you generate embeddings and

01:12:52 find related notes or whatever and along the way to these different checkpoints the application sometimes needs to be brought in show the summary you know show related notes Etc um but in practice we find that apps are actually a lot more sophisticated than this this is a developer named webdev Cody who's building an application on convex they kind of like generates a first project plan given a prompt so in this case he is an app to track recipes and when he creates hits create plan there running

01:13:21 on conx this is sort of like uh let's get a bunch of like project names names let's get first feature requests color pallets icon ideas all of these as you can imagine are kind of concurrent chains that are running in the background um and all of them kind of flow into the application as they have results it ends up that convex is kind of combination of like seamlessly sinking State between these backend steps and the application is incredibly useful for a lot of generative AI apps and for that reason post chat GPT boom

01:13:52 like 90 plus% of projects on convex or generative AI um and a lot of generative AI startups so here's what we're doing about it so the first thing we did is we got a lot of feedback from developers that one of those steps was always Vector indexing or quite often V Vector indexing so the developers said this is how you make a schema on convex it's just typescript type completions all that good stuff they said will you already allow us to add indexes to our Fields like this could you allow us to

01:14:20 add Vector indexes and so we said sure we rolled that out late last year and it's being used very broadly Now by projects on convex uh the second thing we just did um was just kind of announcing right now is we started a convex for startups program discount program kind of access to Startup only forums and events and stuff like that um and the first batch we just admitted tons and tons of generative AI companies in it so again this is sort of like uh the the most engaged excited uh customers right now and then very soon

01:14:54 we're releasing these kind of highle components we have this convex components framework which kind of encapsulates whole state machines in these building blocks so you can easily drop into your app to have your backend Encompass these sophisticated workflows that we've co-developed with customers um very easily and rapidly so anyway that's us if you're building something cool in gener of AI and you want to sort of ship with confidence and quickly check us out at convex dodev thank you [Music] supposed to get

01:15:43 this walked away with that okay ladies and Gentlemen please welcome to the stage CEO of hura tenai go [Music] interesting all right hey everybody uh so nice to be here all right so let's see if we can get this going cool I'd originally titled this talk um connect real time data to your AI etc etc but really it's more existential right the AI overlords are coming for us and to help them be good rulers to help us let's just give them the data they need so that you know they can they can do a good job right um hopefully this talk is

01:16:37 going to be the simplest talk that you hear at this conference um if it's not I'll go back to using gp4 for coding instead of sonnet um but the real pain that I have as I work with LMS is that they can write a Flappy Bird for me with my face going up and down in 30 seconds but they can't talk to my data intelligently it's a it's really stupid um if I want to connect it to my calendar and I just want to say how many one-on ones did I have last week what's a good number to have with my team given their roles help me

01:17:19 stagger them better and plan it out I want to connect it to my Salesforce and say why is this deal with Acme stuck in stage three and I needed to do the right thing I needed to figure out the things between stage two and stage three in my sales Pipeline and tell me why that particular deal is blocked I wanted to connect my tickets and my product data and say is this ticket from an Enterprise customer what's the name of their project can you tell me like what the status of that project is and what

01:17:47 part of the product funnel this project is in um I went to Amazon today in the morning and they have this rofus thing and I was like okay cool um is this product I'm going to tell you what that product is in a second uh but is this product available for one day delivery at my Harrison street address right and just just doesn't like what is this right like it's right here just do it and it doesn't work and you you you all know why it doesn't work right there's like a death by th000 cuts and it's not

01:18:20 secure and I don't want to connect my calendar and make it into GPT who even knows what the GPT is doing with this right like it's it's it's it's it's scary um and it doesn't work um so we solve this with a pretty simple idea which is that you take your live data and business logic and you make that available as a tool to your llm um no it's not it's not surprising right it's easy um because and we did a bunch of things that makes it work really really well right um see if you have time for a quick live demo here let

01:18:57 me see if I'm connected to the internet which I am all right I want to zoom this up all right so um I'm a blockbuster because obviously Services business are the most important businesses now and like U movie streaming businesses are going to go nowhere in the AI world that is to come and so in my Blockbuster database and transactions and all of this stuff that I have going on I want to ask my data question and say what help me write an email to my top customer thanking them for their patronage um

01:19:46 quote mention some recent movies they watched right straight request um I have all this data I just needed to do the right things and I needed to write an email for me right and it works and it works despite the fact that it's going to two or three different places and getting data from them and it works pretty well it handles all kinds of situations and I'm going to talk to you about three key ideas about how it works and hopefully that's going to be useful to you as well so the first is this idea for unified query language

01:20:29 whether you're talking to structured data or unstructured data or apis what if your llm could talk to everything the same way right llms don't know what your API is if you're a little honest with yourselves you probably don't know what your API does right um but but llms know what SQL is right because when you say select star from X where ID greater than one greater than has a semantic meaning that is embedded in the language that in your API that URL param who knows what it means is it Greater is it greater

01:20:58 than equal to is it greater than but actually only works with Boolean I don't know right but it works with um SQL because LMS know what that SQL is right so the first part of this is let's just make everything one query language and deal with that the second is an object model for authorization right which is again kind of blows my mind of why it's so complicated look I don't care where the data is coming from the data has a schema right it's a property of the data and it's a property of the session and

01:21:24 then just run the rule and maybe there's a 100 rules but it should just work and then however it gets accessed it's fine right I should be able to use this wherever it's used however it's accessed the same authorization should be applied so that's idea number two and that's kind of embedded there as well the third and this is kind of interesting is to get the llm to figure out the plan to access data by itself we don't have to hard code it and we don't have to do the work and then you're like T listen what

01:21:50 are you smoking man llms can't even reason I can't even get it to count the number of hours in strawberry what are you going to do with how are you going to make me fetch all of this data from three or four different places and dis ambo and whatnot and we're like you know what that's really simple fix to this problem but let me ask you a Live question how many of you can count the number of eyes in super CIF fragilistic XP Alid doas can you you can't right you're being mean to the llm by asking

01:22:14 it such questions don't be mean to the llm set it up for Success ask it to write python code to solve the problem and it works and that's it it so when you're asking and when we're asking rlms to figure out how to retrieve data we just ask it to run python code to fetch the data that we want so if if the AI Singularity is coming get ready for the data Singularity put everything together if you're doing AI you need access to data if you're doing data and you wish that I could talk to your AI if you have ai and data and you

01:22:49 need to get to talk to each other come visit us at our booth everything's in the open at h/. talk to you f soon thank you for [Music] time ladies and Gentlemen please welcome to the stage CEO of hyper mode Kevin Van Gundy before hyper mode I worked at for cell we had an office down the street above a pizzeria and we had three big problems one we were losing to other JavaScript Frameworks two we were losing badly to other hosting providers and three I was losing to my diet of exclusively pepperoni

01:23:53 pizza eventually we started to win not because we were smart or we knew all the right answers but because we dealt this core competency of iterating really really really quickly we didn't know the optimum strategy but we figured if we just tried more things faster than everyone else we'd eventually be able to adapt and figure out the right products and strategies to figure out what the market wanted iteration is the compound interest of software keep doing it long enough and eventually really good stuff

01:24:22 starts to happen because because we tried a lot of things really quickly we eventually figured out two things one developers want to incrementally adopt new technologies and two they don't want to commit to architectural patterns before they know how their application's actually going to work but iteration can't happen if you're afraid of getting it wrong the same thing that is held back web is also holding back Ai and if I'm honest there are even more things for us to get wrong about geni when I think

01:24:54 about it I'm grossly overwhelmed what's the right Hardware what's the right model what's the right prompt how do I integrate how do I monitor how do I improve everyone here knows a horror story of someone with a runaway Bill a hallucinating chatbot a project that took months and months and never delivered any value and in the end we need to build systems that drisk getting it wrong because we are going to get it wrong a lot picking the wrong model doesn't matter if there's no fre into switching

01:25:25 it out integration is simple when your classical systems and your AI systems use the same apis you can fearlessly make changes to prompts strategies data mixes if you can trace that inference step by step by step at hyper mode we care deeply about making AI approachable everyone here should be able to put AI in their apps without specialized skills at its core hyper mode is a runtime it allows you to easily integrate models and data into AI functions we then surround that runtime with a bunch of tools that make it easy for you

01:25:59 to to rapidly iterate and observe those AI functions in prod we make it easy to get started incrementally adopt AI as appropriate and then as your team develops those skills reimagine those applications as AI native first and foremost we wanted to make the developer experience of developing with AI a lot less terrible when it comes to adding a new model to your service you probably don't want to read a bunch of pages of docs to figure out the temperature is on a 0 to2 rather than a 0 to 1 or a 1 to 10 with

01:26:31 Hyper mode we provide you type ahead and your favorite code editor right out of the box no sdks nothing to download then when you do ship to prod we give you strong defaults just to get started or if you have your own stack bring it along in either case we'll remove a lot of that complexity for you for example traditional rag requires n plus1 requests you need to make an additional call to embed the inputs go talk to your vector store with Hyper mode we you can do that all in one request we've bu in memory embedding and

01:27:02 search service that allow you to do that and save a couple hundred milliseconds per request finally building intuition around non-determination nondeterministic systems is hard each model has its own personality and we make it really easy for you to quickly compare different inferences different Tunes different models and you can then export this data set to fine tune on Monday your boss is going to ask you what did you learn at AI world fair if you come by our Workshop after lunch I'll prove to you you can make AI H

01:27:34 sorry I'll prove to you they can make iteration velocity of core competency the team that built all this amazing stuff will be there will show you how to build natural language search intelligently sort every data list in your product detect outliers catch bad guys you'll walk over the demo that you're proud of and a plan to put something like it in PR by the end of next month and if seeing my happy face again and building something really cool is not enough we'll give you $1,000 on hyper mode credits to get started thank

01:28:01 you all so [Applause] [Music] much ladies and Gentlemen please welcome to the stage VP of AI R&D at hyperspace Nicholas schlapfer hello everybody uh my name is Nicholas schaer I'm an AI engineer I think I'm at the right place um today I'm going to be pronouncing uh a new product we've been working on at hyperspace um a little bit about us uh we are a decentralized AI network uh we have no gpus um we're building a community who takes uh the resources from their personal computer and contributes uh to our decentralized

01:29:01 Network um we have a product currently out called aios um you can download it for Windows and Mac it uses llama CPP and does inference so technically you can download it get inference from somebody in Belgium and you can have a chat experience like that so we're hoping to oh actually sorry um we really believe in uh diverse models uh we think uh not having um just one big close Source model is the answer for the best AI experience um having a mixture of a bunch of great experts will provide the

01:29:40 best AI experience um so that gets us to our self-titled product hyperspace which will be built on our Network um this is a really interesting product it's a mix of prompt engineering um visual uh react flow uh python execution and uh rag like uh web browsing so for the first thing with this product we wanted to build a uh fine-tune model that outputed agentic planning experiences so you put in a query you get in a dag or you get out a dag in a Json format and this is a a methodical plan from that query

01:30:25 um we also wanted to have a primitive of having a in-house web scraping experience for llms so we're using Puppeteer and beautiful soup to scrape websites convert that uh HTML into uh a markdown something easier to read for llms to digest um kind of the product we're going to talk about is uh we have a node editor in a terminal and this is going to be a power tool for the power users um this node editor will allow you to change each node in the react flow to uh fit your needs and this is coming from

01:31:03 the dag or dag orchestration model we have um here's a little video demo of uh our product I recorded it yesterday in the hotel so I'm sorry for the audio and here we go [Music] welcome to a very early look at hyperspace let's begin with a sample query once you submit your query you're brought into our node editor view where each node is streamed in from our dag orchestration model hyper engine B3 we want to emphasize that the user still in full control they can edit the title task description and expected

01:31:57 output they also have the freedom to add as many nodes as they like let's execute now that our outputs are done we can talk a little bit about the outputs each output is coming from each node and each node is creating a query based on that task it's combining the overall goal the local goal and whatever is happened in the previous node we're using a reasoning model and a summarization model for reasoning we're using quen to instruct and then for summarization we're using llama 370b this helps provide a diverse set of

01:32:45 synthesized answers for the outputs that do have python we can go ahead and run them and see them in our terminal terminal will automatically open up with the output we want to provide the groundwork for agentic behavior in the future by providing these core Primitives over here on our right we have our virtual file system that changes our directory we're trying to build out all the Primitives to what an agent would need memory python execution planning and code generation thank you so much for

01:33:28 watching we're very excited to get this out in the coming weeks um happy to announce that it'll be available later this week via wait list so go ahead and pay attention to our Twitter hyperspace AI all right thank you that's my time ladies and gentlemen please welcome back to the stage your host and co-founder of the AI engineer Summit Benjamin duny all right how we feeling was that a good keynote opener are you guys awake yet all right can we have a round of applause for opening keynote speakers

01:34:19 please what a way to kick off the day we we are now heading into the breakout portion of the day so let's review all the tracks that we're featuring today so AI is not just for startups with nothing to lose it is being adopted at scale and at speed in the largest household names in the world bang is CTO of source graph which has been building Enterprise scale developer tools for over a decade and is now building Cody where he is a co-founder of and his CEO is also presenting in the Coen track but

01:35:02 for the AI and Fortune 500 track come join bang and his speakers in Golden Gate B just up the escalators next we have rag which is the Workhorse of AI engineering and there is a lot of detail to get right from Vector databases to reranking Freddy was a machine learning engineer at GitHub and is now CTO and co-founder of quotient AI which helps with rapid rag development through evals join him in salons 2 through six outside the doors behind you and to the left starting at 11:15 Cod genen and Dev tools their

01:35:47 productivity boost of software 3.0 derives most from combining software 1 1.0 code and software 2.0 models as Engineers we are best at accelerating ourselves Britney spoke at the AI engineer Summit last October and is one of the dev tools investors at CRV we'll be putting up an air wall to split this room in just a bit so attend those sessions right here in Salon 7 starting at 11:15 Frontier models are sexy but open models are the ones you can take home and make your own Greg is CEO of oxen doai where among

01:36:36 many responsibilities he runs the archive deep Dives paper clubs that cover many of the open models and fine-tuning techniques this track offers join him right here in Salon 8 after the break last but not least the AI leadership track addresses the growing needs in leading teams of AI Engineers from platform engineering to eval Frameworks to GPU cost optimization and case studies from weights and biases to KH Academy to Neo forj to open AI you may be familiar with Peter from his newsletters like Ruby weekly and

01:37:21 JavaScript weekly but perhaps but perhaps more relevant to today he has mced O'Reilly's fluent conf in this very Hotel note that this is a track exclusive to folks with green lanyards and green badges if there is room at session start time we can let in blue badge and blue lanyards uh basically speakers anyone else please do not attempt to attend these closed door sessions or you may be escorted from from the premises these are exclusive sessions they paid for this please do not make us remove you

01:38:00 from the building um these take place right across the hall in Knob Hill b2d so we'll take a short break head to the Expo sessions which are taking place right over there salons 10 through 15 head to the Expo meet with our sponsors get some demos and the breakout sessions will start promptly at 11:15 all right let's get to it everyone bye [Music] [Music] we got an ins with eyes wide sh we got everything [Music] weing our Broken Dreams I want to be next to you you want to be next to me holding our paper heart feing out Broken

01:39:21 Dreams I want to be next [Music] hey everyone I'm adad and AI engineer based in San Francisco along with my teammates I created math matrix movies at a hackathon and SF on May 11th 2024 today I'd like to show you what our project can do because I think it's really cool what it does is that it generates really cool math explainer videos in a truly unique style that is able to get Concepts across visually this is something that I think is really unique that you may never have seen before so AI hackers let's start with a

01:40:20 live demo in our demo I decided that perhaps we should talk about probability this year only 10% of the applicants for speakers at this AI Engineers World Fair were accepted to speak on state I didn't make the cut which is why you're watching me on this recording now there's no hard feelings but I'm ambitious and so I want to try again for next year and the year after that so let's assume that next year the acceptance rate for docs Falls to 7% do my odds of getting in increase or decrease now that I didn't get in all

01:40:53 also if the year after that the acceptance rate is just 5% what if my odds of getting in at least one of the two years now watch I'm going to enter that as a prompt into math matrix movies it's going to take that whole math problem and it's going to generate a great math explainer to explain the whole situation right back to us and at the end of this video you're going to see what it's done in the meanwhile while that prompt is put in and it generates let me show you a clip from one of our already published

01:41:23 videos on sigmoid functions watch now how the math video explains how the shape of a sigmoid curve changes based on the equation used to draw it check it out now let's look at the math don't be scared it's simpler than it looks this is the formula for a sigmoid it uses e a special number like pi and X can be anything we want if we change the X part of the formula we can move the curve around here we added a minus two and look the curve shifted we can do lots of cool things by changing this formula but the

01:42:09 S shape always stays the same crazy right so how does that work well we have to thank the math educated genius Grant Sanderson AKA 3 blue Brown because he created all his amazing math videos by writing his very own math animation library in Python and it's called manm under the hood madm uses a 2d Graphics Library called Cairo to generate most of the drawings and the animations and then it uses the very powerful command line tool ffmpeg to combine those animations as well as voiceovers and other elements such as

01:42:53 latex equations into compelling math videos so here in this case you can check out the code for simple movie where we try to animate 5 + 3 = 8 for 6y olds by using two groups of five red apples and then three green apples and then combining them into one row of eight apples so let's see how manim actually does this so it's a simple block of python code for each section there's a voice over which is created by azur TTS and within that section within that scene we actually are able to lay out the elements and you know enumerate

01:43:35 them using an array and animate them and move them around all in one block of python code so this one or 1 and a half minute movie that explains 5 + 3 = 8 to 6y olds is just maybe 30 lines of python and what we do in our project is we use Google's G Min to actually generate this code so if many of you are like me you struggle with visualizing and contextualizing the math and algorithms behind the river of information flowing at us as AI Engineers every day I often find myself wondering about something I

01:44:10 read in the Laten space podcast or saw on Twitter or talked about with a colleague but I like I lack the right words or formulas to make my intuition like explicable and testable and here m Matrix movies is like a godsend it's like you gave Google's Gemini a whiteboard and said teach me at my level uh in our user testing we've given the kid to a seven-year-old and watched them generate movies over and over in different variations uh explain 22 * 22 to me with apples with fish with TVs so it's pretty crazy it's it's an education

01:44:43 hack that opens up a Vista into how AI T drink can enable exhilerated learning of Concepts so now let's take a look at the output of that prop that we put put in earlier uh this is a slightly different version of The Prompt I just did it again uh because I lost that page um so it's taken the prompt and it's created a video it actually creates the video and then it watches the video again and improves it and studies it for overlaps at occlusions uh Gemini's vision is good but it's not great it has

01:45:15 a lot of gachas uh but let me show you what the final output is like and you know with a few more iterations it would get even better um and as it is this is pretty good I'm going to increase the speed so that it goes pretty quick here's a final video for now rejected from AI Engineers World's Fair sad face but what about next year and the year after that let's assume the acceptance rate for talks Falls to 7% next year let's visualize this each Square represents an applicant and there are 100 applicants

01:45:43 in total the green squares show the seven applicants that get accepted the year after let's assume that the acceptance rate Falls to 5% again each Square here represents an applicant and we still have 100 applicants this time only five applicants shown in green will be selected now let's calculate the probability of getting my top accepted in at least one of the next two years the probability of not getting accepted year one is one minus the probability of getting accepted which is 7% similarly the probability of not

01:46:22 getting accepted in your year two is 95% the probability of not getting accepted in either year is the product of the two probabilities we just calculated which comes out to 88.3 5% finally the probability of getting accepted at least once is one minus the probability of not getting in either year that gives us a decent 11.65% chance so there's still a chance yep see you on the stage brought to you by the power of probability so keep applying keep learning and who knows you might just see me on stage

01:46:52 next year yep sure hope so hint hint rejected from AI engineer cool well I hope you guys enjoyed that I'd like to thank Bish Lily and Justin for bringing this project to life with me thank you for your attention and please check out our videos on YouTube and fill out your math video requests at https math. auto. mov thank you so much hello welcome everyone in this brief presentation we will talk about how we are building an AI powered Healthcare conil and share some key tips and tricks that we learned while building the

01:47:32 service so if you're interested in healthcare and AI uh this is the right session for you to quickly introduce myself my name is akiles Gupta I've been the co-founder of harness care and I've been in product and Tech roles for last 15 plus years at harness care our mission is simple uh we have we started the company with the goal of empowering everyone to navigate Healthcare System with these to really help reduce some of those barriers that people face while accessing Care at the right time and at

01:48:03 affordable prices now we all know how complex the US Healthcare System is but to highlight few points there are roughly 100 million people who are under medical debt within us there are 55 million people who have some C family caregiver responsibilities but most of them find it diff difficult to manage the caregiving plus the the work responsibilities they have and most of us do not have sufficient Healthcare literacy to really navigate the system or advocate for ourselves against the providers or insurance

01:48:39 policies now many of those issues really prevent people or delay care but what if we had a personalized expert in a concierge form to really reduce this burden and take care of various mundane tasks for care coordination this can be as simple as getting quick answers regarding my benefits this could be asking the coners to find an available provider and the price estimates for the services or this could be managing my outof network bills and filing the claims with the insurance automatically these are just few

01:49:18 examples that how this conge can help improve the care Journey for the the patients and the caregivers now this vision would have been impossible a few years ago but luckily with the Regulatory and Technology Tailwinds a lot of this is probably not possible uh the first is really the regulatory Tailwinds in the form of improving data interoperability which providers and insurance plans are required to make personal health records easily accessible and pricing data available now there is also a slew of healthcare

01:49:56 data aggregation platforms which are using these standards and making data access cheaper and easier for many of the health Tech startups and lastly our heroes in the large language models which can really generate insights from a mountain of this Health Data in unstructured structured forms and perform tasks using the agents the way we make this all work together right now is includes three different compon components one is the data aggregation and standardization once we extract data from various different platforms whether

01:50:30 it's medical records insurance plans claims or other data sources we then normalize it using the llms to extract critical information from unstructured data or standardize it for our data stores and making it available to our AI conge service on top of that the second big area is enriching our over con with public knowledge bases uh these can be curated sources of information for medical terminologies clinical resources to provide better guidance to the patients and lastly integrate with thirdparty vendors to

01:51:09 really perform common tasks whether it's medication orders looking at pricing records scheduling appointments or looking up financial assistance programs and all this becomes available to the user through our AI coners platform now there have been some critical interesting learnings for us during this journey the first one has been around rephrasing user prompts we all know the quality of responses from llms really depends upon the quality of the prompts but users don't often ask the detailed questions or may have typos for the

01:51:44 prompts they are making using llms we can rephrase these their questions and clarify the intent for example in this case if a user says is er included we could rephrase that using llm to clarify that the question the user is asking for is emergency room coverage included in my insurance plan that significantly improves the quality of a rag and also the LM llm responses the second key learning we had was around pre-processing of documents to improving rag the large documents obviously take a lot more tokens and

01:52:22 don't always work work well with during the vector lookups uh what we do instead is really create structured summaries out of these large documents depending upon if it's a clinical document or insurance policy document we create different summaries accordingly and that significantly improves the rag accuracy for us while reducing the token usage as well and lastly uh the user experience users don't always know how to best use the AI tools we need do need to educate them with contextual suggestions for example in our case we

01:53:01 could recommend them different prompts when they're looking at Medical Records this could be if they want to look up side effects for their medications or diagnosis in details Etc but when they're looking at the coverage policy we can show them guidance guided prompts around uh coverage details for example for mental health services ER visits Etc the common issues people face or when they're looking at insurance claims we suggest them different kind of prompts to help them more educ be be more educated about

01:53:33 the services and the use cases uh this definitely improves the overall engagement with the AI uh and the user satisfaction as well we are always looking for passionate Engineers to work on these problems with us so if you're interested on on in what we are building or just want to brainstorm a few ideas feel free to reach out to us thank you so much when Chad GPT first came out gpt3 had been available for two and a half years via an API Chad GPT didn't blow up because of a new AI but because of a new

01:54:05 interface and today if you look at the GPT rappers that have gotten traction they've either differentiated on interface or they brought the interaction closer to where the user already lives and works cursor is my favorite example of this I used to use chat GPT to answer a lot of code questions but I find it so so much more valuable to have that llm right there in my IDE we've over indexed a bit on chat as the deao interface for AI and it's unquestionably proved useful in a lot of situations but I think there's still a

01:54:36 lot of opportunity to experiment with different interfaces for different people and different use cases and I think most folks are sleeping on email as an interface for llm based apps my name is Greg bogas i'm the founder of high high labs and I've spent the last year building and experimenting with a whole bunch of email bots in the rest of this video I want to first talk about why you might want to consider email as your interface and two I want to talk about how you can build your own email bot and some technical considerations

01:55:05 you'll encounter along the way and hopefully I can save you a little time by sharing some of the lessons that I've learned first so why email email is ubiquitous and crossplatform back when I served on the developer relations team at twio we used to say SMS is the only app that comes and pre-installed on every phone but is true of email as well which is why twio eventually spent $3 billion dollar to buy synr email is frictionless I partnered with someone who works for a nonprofit supporting Public School principles and we built

01:55:37 this email bot that took classroom observation notes and helped principles write a first draft of a teacher evaluation that fit the Danielson framework we started to get some tractions and we said okay hey would you like this as a web app and many of them said no I actually I already have so many web tools that I have to sign into some of those apps work better on some machines than others sometimes there are restrictions on what websites we can and can't visit from within the school but I can always send an email and also we

01:56:07 found out that one way principles were finding out about the service was that their peers were forwarding them the results emails are easy to share and because of this email and email apps have a long history of going viral one of the first apps I ever built uh you can actually try it out if you want it you can email start adventuresin mail.com and the basic idea here was like a Choose Your Own Adventure style uh interactive fiction that was bespoke uh that you could play with your friends by CC them I sent it off to my parents I

01:56:40 was like Hey create a story about uh Hawaii it's my parents favorite place and I kind of went to bed I figured I'd like see a couple emails and I'd reply back in the morning I woke up to 76 emails and they had played so much that eventually the app crashed because I was hitting the token limits and I don't think that they're ever really going to go to the GPT store and spend a lot of time on there but they're in their inboxes every day email meets people where they already are so how do you build an email bot first you're going to

01:57:09 need a service that offers programmatic Emil there's a bunch of these I personally have been really enjoying postmark it's great clean developer experience you're going to set up a few DNS entries you're going to point your domains in about an email to postmark then postmark will make an H HTTP request to your app I've used Fast API for a lot of my apps super easy to set up an endpoint to receive that inbound request and the data about your email will just be stored in Json uh then you're going to want some sort of

01:57:34 background job like celery because these Generations take time which actually is a cool bit about email it's async so the user doesn't expect an immediate response email is actually a pretty good medium for more agenic processes or you can actually just uh take advantage of that to use say batch processing and save a lot of money in your Generations you can even put a human in the loop um but however you do it you're going to want some sort of background job that's going to do the processing that's going

01:58:01 to create the generation and then you'll use the API to send off your reply these emails are considered transactional emails not broadcast emails so your deliverability rates are probably going to be a lot higher now how do you structure your llm I think there's two things you got to do here on your system message it helps a lot to tell your llm hey this is a conversation being conducted over email I like to tell it to reply and plain text and then if I want an HTML formatted email I run that through a separate generation the user

01:58:28 message I like to feed in some bits about the email in Json the LM does a really good job of interpreting that Json and then spitting back just plain text for me it is important to pass in the subject because a lot of users will just include important context about the message or the entirety of the message itself in the subject um but of course when you reply you're just going to want to use the same subject so that you can take advantage of threading in in the inbox uh speaking of threading uh

01:58:55 there's a couple things you're going to want to do one is to keep the subject the same and the second is you want to set a reference header referencing the original message ID and this will help tell Gmail and other clients that they should include all these emails into a single thread eventually you're going to have to figure out how you want to deal with conversations the best way to do this is to use the mailbox hash you know by now that you can add a plus after the the first part of an email address you just

01:59:20 want to generate some sort of unique identifier and then when your app brings in the inbound message you check to see if that's there if the mailbox hash is not present then it's a new conversation if it is then you can look it up retrieve the messages and add them appropriately also one of the quickest way to get these apps out into the world is to use open ai's Assistant API and you can use the thread ID that they give you as the mailbox hash and then they give you a nice guey that you can use to

01:59:48 edit the system prompt and iterate quickly and I've actually gotten to a place where I could deploy prototypes using the assistant API really quickly I don't think it's the best for production apps at scale but if you just want to get something out and start getting feedback from your users quickly it's a really really great way to go again my name is Greg bogas with high high Labs you can find me online at gregy B uh if you want to learn more about this stuff you can check out my blog highh high.

02:00:14 I've documented a lot of these learnings in more detail there uh and if you got any questions just send me an email hello I Max the CEO of Cai and today I'm going to tell you why queries are all you need rag learnings from processing three billion tokens a day one of the great things that you can do when you have actual volume across your pipelines is you can look at what types of queries and requests actually come through your system and what we quite quickly realize is that um 68% of production queries are

02:00:53 destined to fail with naive rag because only 32 are answerable just with semantic search 22 are meta queries like give me the last 10 documents on this topic or they are off topic queries where the answer is not in the data store or they require more complex operations like comparisons compare X and Y and give me an answer or they're just junk and other right which clunk up the system so why is naive rag so ineffective it's because the querer if that's an AI or a human is unaware of retrieval capabilities and the data that is

02:01:42 indexed they just ask for what they want um una aware of if they can actually get an answer and the lazy solution here is to give the querier context about the information that is stored and the ways that it can be retrieved um this is actually hard in practice because the information keeps changing so you need to update that context that instruction or that prompt um continuously and it's also a lazy solution after all you don't have to read a manual to search the internet a better solution lets the

02:02:19 Creer ask for anything and tries to answer it in a best effort way this means that it takes a query which is a natural language question but also a wish list of how that perfect answer would look like what is the metadata what does it fulfill even if the fields and properties that are being asked for here don't actually exist in a single record in the data store this is completely independent the querier can just ask for what they want if it's a human they can ask for whatever they want but if it's an nlm they can

02:03:01 hallucinate arbitrary parameters to make this work and if the system cannot answer the question it refuses it politely so let's have a look at how such a solution actually works so we have on the left side we have the query and the wish list coming in we have an EMB better an atic router that generates a set of semantic queries multiple semantic queries if necessary and meta queries that use things like dang range sorting sort by newest sort by oldest and other properties and um once those results

02:03:42 come back from the data store uh we pump it through a final question where it's hey does this actually help answer the question and if it doesn't then we refuse and if it does we give that answer back and let's go back to our original example of these 10,000 sample queries across our systems from last month um and we see above the line 32% are answerable with just a semantic search and then we have those other categories and here we see that the 32% semantic retained we now also cover those 10% which are comparison

02:04:22 questions um by being able to route through multiple different semantic queries so for example for a to compare between two different things we would most likely route three different queries once asking for a compare X and Y if there's an existing comparison in the data store and then two different queries to retrieve context about the two different items being compared and in theory the scales to any n items being compared we also cover the meta questions or we find a way to get closer to covering all of those meta questions

02:04:58 and we refuse in the cases where we cannot actually provide a good answer the downside of this is latency and cost um you can imagine that pulling everything through such a system is a expensive and B introduces quite a bit of lag but you can actually solve for both of these through a lot of different techniques that we've developed internally and that we hope to share um in the near future such as speculative retrieval em embedding based query routing uh compressed embeddings and a reduced queral language

02:05:41 to handle these meta queries better as well as lightweight schema where llms that know how to interact with the data store uh if you're looking for a job and you love working on hard engineering problems at scale always emilly and we're releasing our query and embedding pipeline for everyone in July 2024 we're really excited to see you try postgress is the most popular database in the world according to the 2023 stack Overflow developer survey and despite the Myriad of specialized Vector databases out there postgress is a Top

02:06:18 Choice for many developers building AI applications and that's thanks to PG Vector PG Vector is an open- Source postre extension for Vector data it provides the ability to store and sech vectors in postrest and transforms the 30-year-old relational database into a fully fledged Vector database and thanks to PG Vector postgress has become one of the most widely adopted Vector database for rag applications and that's thanks to its ease of use SQL support and operational Simplicity but the one question hanging

02:06:50 over postgress versus using specialized Vector databases has been performance and the reasoning goes like this dedicated Vector databases have purposed both data structures and algorithms for storing and searching large volumes of vector data thus offering better performance and scalability compared to general purpose databases with added Vector support well I'm here to tell you that the good news is that the answer to the question about whether postrest can scale for vectors is yes enter PG Vector

02:07:19 scale PG Vector scale is an open source post extension that builds on PG Vector enabling greater performance and scalability my name is afar I'm a product leader for AI at time scale we're a postgress cloud database company and I'm going to tell you about PG Vector scale The open- Source extension that we built to scale Vector workloads on postgress we built PG Vector scale for three reasons first to make postgress a better database for AI second to challenge the notion that postgress and PG vector are not performant or scalable

02:07:53 for vector workloads and thirdly to give developers a way to keep using PG Vector but without any performance or scalability bottlenecks PG Vector scale is licensed under the open source postest license and it complements PG Vector rather than competing with it by leveraging the PG Vector data type and distance functions further enriching the postgress ecosystem for building AI applications and by using PG vector and PG Vector scale together developers can build more scalable AI applications benefiting from higher performance

02:08:25 embedding search and cost efficient storage before I delve into the details of the technical innovations behind PG Vector scale let's answer the biggest question how does it perform to answer this question my team compared the performance of postgress with PG vector and PG Vector scale installed against pine cone widely regarded as a market leader for specialized Vector databases we used a benchmark of 50 million in embeddings of 768 Dimensions each and here's the results we found that with PG Vector scale postgress gets 28x lower

02:09:01 P95 latency than Pine con storage optimized index and 1.4 times lower P95 latency against Pine con's performance optimized index on the same data set and thanks to the power of Open Source developers can get these results at 75% the monthly cost when self-hosting now now that you've seen the numbers behind the performance let's dig into how PG Vector scale gets through these results PG Vector scale brings specialized data structures and algorithms for large scale Vector search and storage to postgress as an extension

02:09:38 helping deliver comparable and often Superior performance in specialized Vector databases it does this with two key Innovations the first is a new high performance cost-efficient search index called streaming dis Ann inspired by research on billion scale Vector search in Microsoft and improved on by time scale's own researchers streaming disn overcomes limitations of inmemory indexes like hnsw or hierarchical navigable small worlds by storing part of the index on disk rather than entirely in memory making it more

02:10:13 cost-efficient to run and scale as Vector workloads grow that ability to store the index on disk vastly decreases the cost of storing and searching large amounts of vectors since ssds are much cheaper than Ram the second Innovation is a new high accuracy quantization method called statistical binary quantization or spq for short this is developed by researchers at time scale and this technique improves on standard binary quantization techniques by improving on accuracy when using quantization to reduce the storage space

02:10:48 needed for vectors more details about PG Vector scale its performance and how the technical innovations work can be found on the PG Vector scale GitHub page but the key takeaway is that postgress is scalable for vector workloads and thanks to PG Vector scale we can all be the guy in the right and just use postgress for AI applications thank you hey everyone I'm ADI adani an AI engineer based in San Francisco along with my teammates I created math matrix movies at a hackaton in SF on May 11th 2024 today I'd like to show you what our

02:11:25 project can do because I think it's really cool what it does is that it generates really cool math explainer videos in a truly unique style that is able to get Concepts across visually this is something that I think is really unique that you may never have seen before so AI hackers let's start with a live demo in our demo I decided that perhaps we should talk about probability this year only 10% of the applicants for speakers at this AI Engineers Worlds Fair were accepted to speak on state I didn't make the cut which is why you're

02:12:02 watching me on this recording now there's no hard feelings but I'm ambitious and so I want to try again for next year [Music] [Music] [Music] [Music] [Music] [Music] one more breath beside you so I could find strength to divide us it we got it I know we did the best could if I could go back on the mess I would memorize your face before before I go but this is how we grow got to give it up sometimes as go KN when to kill your pride there no to blame nothing really stays the same this is how we grow sometimes we hold on to let go

02:14:50 [Music] hold there is lost us and I know you have your reasons days I'm a mess but I know there's a rainow over all of the past your head on my shoulder but I know better onor but this is how [Music] we got to give it up sometimes is go KN it when to kill you there's no to blame nothing really stays the same this is how we grow we hold on to let

[Music] go let go sometimes we hold [Music] got to give it up KN when to kill your pride there's no one to blame nothing really stays the same this is how we this is how

02:16:49 we sometimes hold let go [Music] [Music] B [Music] [Music] [Music] B [Music] [Music] a [Music] d [Music] [Music] I was watching you watch the the sun come up vage t-shirt through High Times these nights tastes like goldet Obsession show as CU we were out the night like we wear our clothes dancing right through the while we watch it singing on r we give up our go as a new morning comes through the windows We R all new Burning through the page tearing past all the you're wear we our problems underneath cles like

02:21:15 super like superheroes it's coming over now it's wees down a Harmony and peace that only we can hear super you want to feel like us it's forever America under your influence full moon Waxing now I couldn't see it until you show me how feels like we're insane we blame it all on love so saturated so we can't get enough we were out the night like we wear our clothes dancing right through the fire while we sing Our an we up our gos as a new evening comes through the windows it's coming over now A W down a

02:22:26 Harmony that only we can hear super CR you want to feel like us it's forever America coming over me electric Sy night on fire I KN on Master super you want to feel like it's forever you're in [Music] America holdon going so come with us don't hold tonight is all we have the is going so come with [Music] us hold T is all we have the is going so come with us don't hold back tonight is all we have the sky is it'sing it's down a Harmony of that only we can he a [Music] superc every night on fire I neon

02:24:19 masterpiece the super CR you want to feel like us it's all forever [Music] America [Music] us going so hold back [Music] tonight going so [Music] [Music] come it was summer back in 89 we were kids falling in love for the first time H your hand you look me in the eyes kind of feeling you get Once in a li but now something went wrong you're moving on I found myself on The Blind Side now you won't call we lost it all you fade away I'm picking up my heart from every piece that's broken been trying to get back to myself but don't

02:25:56 have a clue I'm looking for some luck can't find a door it's open I'm losing all my feels like I'm left hereo because I'm missing you because I'm missing you oh because I'm missing you because I'm missing you because I'm missing you I'm missing you I was chasing all the wrong side trying to hold on to something that I couldn't find you didn't Captivate my mind now I know we've in the sunsets in Paradise but now something went wrong you're moving on I found myself on The Blind Side now you won't call we lost it

02:27:19 all you f away I'm picking up my heart from every piece that's broken and trying to get back to myself but don't have a clue I'm looking for some luck can't find a door that's open I'm losing all my feels like I'm left here because I'm missing you because I'm missing you oh because I'm missing you missing because I'm missing you because I'm missing you picking up my heart every piece that's broken trying to get back to myself but don't have a clue I'm looking for some luck can't find a door it's

02:28:28 open I'm losing all my like I'm here because I'm miss you e e all right great hey everyone uh so excited to be welcoming you to the Cod genen track here at the AI engineer World's Fair uh my name is Britney Walker I am a GP at Charles RAR Ventures uh

AKA CRV we are an early stage Venture Capital firm investing primarily at seed and series a rounds and we've been around for 54 years uh believe it or not um there my focus is primarily on infrastructure and have done a bunch of AI infrastructure investments in the

02:30:44 past couple years which is how I've gotten to know swix our gracious host um relatively well and very excited about the topic of Coen as I'm sure all of you are here today um to walk through this topic we're going to have some amazing speakers we're going to have rul pandita joining us uh from GitHub we're gonna have Kevin how joining us from um codium and then we're gonna have Michael trell joining us from cursor um we're going to lead things with Rahul he works at GitHub uh next specifically and works on

02:31:15 the co-pilot workspaces product in that context I'm sure all of you are familiar with GitHub co-pilot probably the number one AI product that has taken the World by storm uh in the past year year and a half here and incredible statistics around its adoption downloads all of that he can tell you more but excited to have him uh walking us through some of his efforts today set cool hello everyone how are you today pretty good pretty good I guess this is the afternoon lecture uh afternoon session so I'm standing

02:32:17 between you and your lunch so I'll try to get this as quick as possible hi my name is Rahul Panda and I am a researcher at GitHub next uh and today we're going to talk about some of the gith up next Explorations uh now before we begin who among you have heard of gith up next oh cool quite a few of you that will make it go much easier and much faster all right for those of you who don't know us we are uh about 20 bunch of researchers senior level uh developers and and mostly code build uh tool Builders uh who work outside of the

02:32:56 regular product uh and Report directly to our CEO uh and that's by Design and and our goal is to explore the future of software engineering like you all are doing in in your day-to-day jobs and the and and the reason for exploring that is that like once we do our Explorations we toss it on and we pass it on our learnings to the product and development teams so that they can build really compelling products like the co-pilot that you all have used hopefully at some point of time as an aside uh for people

02:33:27 who are following us on Twitter uh I don't look anything like my picture over here I'm the one in the green background but we do have Devon in a team he's not an automated AI he's a very real person and he looks exactly like the person on the top right corner on that slide all right since we have gotten that out of the way let's talk about let's get back to the future of software engineering with regards to gen gen so here's what Andrew ning uh who single-handedly trained a whole generation of machine learning Engineers

02:34:00 uh has to say about uh AI that it's just as electricity it's the new electricity it's going to transform the software development and almost every other field just like electricity did 100 years ago so what does that mean here's a picture of what a manufacturing facility look like before electrification there used to be a giant uh mostly coal powered steam turbine or steam engine located centrally which used to turn these giant uh giant shafts which will turn these auxiliary shafts so forth and so on and

02:34:36 individual workers would connect to these shafts using the belt and pulley system right and and these engines were like really really huge so so it was the workers the whole architecture of the factory was designed around this steam engine and and and the whole workflow was around the steam engine and and it was the workers who were working around the technology rather than the technology working for people right and along in 19 uh 1880 came these electric motors uh and and they had the potential

02:35:07 to revolutionize uh the the manufacturing sector why because unlike steam engines or steam uh Motors they retained their efficiency when they were smaller right even so so you could basically redesign the entire Factory floor plan so you would think that wow this is great and everyone would jump on this but it was not until n 1920s where these became the mainstream so early 1880s to late 1920s what was happening for about these 40 years what was happening was exploration and experimentation people were trying

02:35:45 to figure out uh how to use this technology how to make it better how to drisk it to a point that that the use of this technology becomes the norm rather than the exception and that's what we do at get up next right our Charter is to explore the future of software engineering and with the emphasis on the word explore right because if we knew what the future of software engineering in context of AI looks like we would just build it that's more efficient but unfortunately we do not so what we have to resort to is

02:36:17 exploration we just try out different things rapidly prototype experiment and figure out whether something works or not and if it works then we put it out in front of our customers in in users and we learn from them and then we finally transform into a product often times an idea begins as inside our next uh as a functional prototype which goes through heavy dog fooding inside the next team if it survives that then we move on to the next level of dog fooding that is inside the company if it survives that then we move on to the

02:36:49 next level which is releasing it as a tech preview uh to other early adopters we learn from that if it survives that then it may have a chance to become a product like that a product in the future and we can kill or we can shell any of these exploration at any point of time if we are not getting the right signal so that we can explore other areas we did that with the co-pilot so yes co-pilot started off as a next experiment and since that we have created many other experiments like co-pilot 4 CLI hilot voice GitHub block

02:37:21 spec Lang so forth and so on uh a lot of these have transformed into a product of their own so you can see some of them as a g of product offerings a lot of them have been absorbed into existing products uh and and you will see them as a part of the existing products and a significant number of them have been shelled we've learned what we learned from those experiments and figured out that this is not the right time for that kind of exploration or the exploration itself was flawed so but we learned from

02:37:49 them and we will keep that learning and use that in our next uh next Explorations so that was an overview of giab next and today I'm going to talk about two specific Explorations uh one is the next edit suggestions in the co-pilot workspace that are currently active uh from from gab Nick's perspective and uh specifically I'm talk I'm going to talk about what their motivations was and and how they came to be and what are the future plans for that so first off uh copilot next edit suggestions right so what if it started

02:38:23 off with this question what if ghostex could be more intelligent right so we all know what copal does uh it provides you the code completions in your current context right while it's like really really good at creating new code but that's not what we all do right we we we almost always edit existing code which involves uh editing adding deleting lines at multiple locations in a program right what if ghostex was good as that as well and that's what this exploration is we call it next edit suggestion which

02:38:58 provides you suggestions not only at the current cursor level but provides you suggestions what else needs to change in a program but enough talking let's jump on to a demo all right here I am going to add this parameter in this Python program and the next edit suggestion automatically picks it up and says that hey you need to update your method definition once we update the method definition it says that hey you need to add these uh these these arguments and once that has been updated then it will

02:39:29 go back and say hey uh now the code document uh is not is not in line with what the code is actually doing and it goes ahead and edits that and updates that as well and the same thing repeats when I add one more uh parameter all right so so that was copilot next uh edit suggestions experiment uh we are we're still not ready yet we are still uh experimenting with a bunch of other stuff like you know uh is the ghostex completion the right uh modality for it or do we need to figure out a bit different way of

02:40:08 presenting those suggestions what if the location of the next edit is not visible in the current viewport or what if the location is in a file that is not even open in an editor uh most importantly we are also working on fine-tuning the models specifically for this use case the idea being that like if we want the next edit suggestions to be uh accurate and we want it to be very useful then the suggestions needs to be on point and once we are done with these further sub Explorations and we feel that it has

02:40:38 gotten through our internal dog fooding standard next edit suggestions would be coming out either as a standalone uh Tech preview from next or as a part of an existing next product uh some sometimes in your IDE uh in next few months all right so there was code completions but let's move from the code completions to the task completions land uh why do you ask why Why move from the task completions it just turns out uh that while code is like an important artifact uh that comes out of software development but it's not the only

02:41:12 artifact software development involves this inner loop where you begin with a task the idea is like what am I supposed to do uh how am I uh what what is the specific thing that I'm trying to do and followed by uh how do I go about doing that thing what are the Frameworks that are at my disposal what are the programming languages that are that are at my disposal what are the kind of uh what is the existing code that's there what how do I write a new code that is consistent with those codes so that's becomes a sort of a

02:41:40 specification and once you understand where you are then you sort of try to decide like where am I going with it like how does the final product look like once you have zeroed in on that then you go about what specific file changes do I need to make to to to get to that final product and that sort of becomes a plan and once you get to the plan then you go to the implementation part and that forms this Loop of software development and we call it Inner Loop and we would like the AI to be helpful in all those aspects of that

02:42:09 in Loop and that's why we built co-pilot workspace in mind you like all next Explorations it did not start as copilot workspace it started as individual Explorations for instance we started to f figure out can we use natural language to as a functional specification of program so there is a spec Lang exploration we in parallel we were trying to figure out if we can improve the code completions by providing by prompting the model with the runtime information and all of those things combined and you with a user feedback

02:42:37 combined into this one bigger exploration called co-pilot workspace and we were also talking to our users like we we we wanted to talk to a developers and we wanted to ask that hey we are building this thing how would you like AI to support you what are your major pain points and one and a few things became very very clear while talking to our users right so first thing is that the most uh difficulty that people faced was getting started on a task like how do I I know that an issue is assigned to me how do I get

02:43:06 started on it followed by how do I trust the output of the AI I don't trust it and more importantly they figured out that problem solving is what software development is about and they would like to retain that problem solving uh aspects of it and they would like the help of AI in the form of a thought partner or a sparring partner or a second brain which they can collaborate with to solve a problem and lastly and most importantly they would like to retain control developers are in control not the other way around and with this

02:43:37 feedback we build co-pilot workspace so what is it it allows you to it simplifies getting started so oneclick proposal on on your tasks it has a built-in runtime that allows you to quickly verify what the the code that has been provided by the AI it has an environment which is built for iteration so if you feel that AI is going in the wrong direction you can just go and quickly correct it and most importantly it is designed for collaboration so you can just share uh your code or your work as a part of the gtha pull request or

02:44:07 you can share your work or share your workspace with your colleagues if you're not comfortable with it but let's enough talking let's just get into a demo about it right so this is monospace which is another GitHub exploration so if we are to write code let's write code in style and these are the four is a family of monospace fonts that has been released by GitHub and and this is a website that outlines a bunch of uh features of these fonts and over here somewhere over here is this playground which says that uh

02:44:41 that here are how the syntax highlighting looks across different languages notice that it is missing rust and rust appears to be the next cool thing that all the cool kids are doing so we would like to update this monospace website with a rust example as well so do how do I get started so I've created this issue or somebody has created this issue it just happens to be me for the purpose of this demo that I would like to create I would like to add a rust example to the font playground and I can just click this

02:45:11 button over here and it will open the copilot workspace for me and through the magic of cash ing you can see that it quickly generates the specification and prop Uh current specification and the proposed specification uh why caching uh because I had to finish this demo in time but trust me it's not a matter of hours it does happen in a matter of minutes right and and for those of you who are interested I would like to do a live demo for you in the Microsoft Booth after this task all right so what is the

02:45:41 current specification it just goes and figures out does the website have this playground that contains a uh rust package and it says that doesn't and it goes to the Target state would where would the target what does the Target State look like and it would says that yes the website will have the specific package for syntax highlighting the website will have uh this package in in in package.json and then I will update a bunch of other files it looks nice and I'll go and generate a plan for it again through the

02:46:10 magic of caching a plan has been generated and it will tell you that these three files these three files need to be updated and I will it it appears that this seems to be at the right level of modality then I will go ahead and implement it and yes Magic of caching again what we see is the files that are over here uh now this seems nice and but what about the iterator part what you can do is at any given point of time if you feel that something is not right you can just go ahead and say that okay add

02:46:39 rust to the language mappings and say add code documentation and you can edit at any given point of time and what you can also do is that you can edit via chat over here and you can say that hey I want to edit this one specific location how do I go about and doing this I'm not going to do this because it's going to go through the whole iteration Loop and then the illusion of the caching will break and it will take a lot of time but I would like to do show that in live demos afterwards but how do I trust whether

02:47:12 this is in fact the right thing so I will open up this integrated terminal and I will say uh install and run this repo all right so what's going to happen is uh that a suggestion is going to load and apparently not the right thing but I can quickly go and edit it and say that all right this is the command that I'm specifically looking for and I can go and run now this will run this command in an actual terminal and we'll see the output in some some point of time uh and and you can see that actually this this code

02:47:52 does compile what we also have is a preview what we can do is open the live preview I don't trust it it's uh it will say that it's just going to be a second but it takes longer than that while that loads what are the other things one of the things that you would say is that hey you wrote a very simple command in the terminal you said npm you could actually type that thing in the terminal and yes you're right I can type that thing but think about that in a mobile setting when you can open copal it works

02:48:19 in a mobile plat uh in in on your phone it becomes very tedious to type those symbols right and if you have used the the mobile keyboard it's not very useful for that so what I'm going to so so that's why we use this natural language way of uh writing these commands in the terminal uh so that it can help you when you're on the go it can synthesize those commands and hopefully the website has loaded and there is a rustic sample right cool that was a demo and thank you we are working we are not stopping

02:48:56 there we are working on a bunch of these improvements and I can talk about these improvements uh on one-on-one basis with you and uh and and you already saw some other improvements like the runtime support to synthesize the terminal commands and and faster file completions using uh to to make the co-pilot workspace better but there are other next Explorations that are also active like how do we rethink the developer learning with AI and how does the code change if majority of the code that that

02:49:22 is now being written is by AI so what does that mean and some of these Explorations will will work out and some of these exploration you will see as Tech previews and some of these exploration will kill because we don't know where they're going so in summary I'm saying that we do not know what the future of AI is but what we know is Explorations is the way to get it and with all your help we'll jointly explore the space so that we don't have to wait like electricity we don't have to wait for 40 years to get to a place where to

02:49:49 to get to a place with software development where we enjoy the benefits of AI you have been a lovely audience that is my time I really appreciate you and if you have more questions if you want to have live demos I'm available in the Microsoft Booth uh in like two salons over that side thank you so much okay um thank you so much for that that was truly incredible demo and some amazing work you guys are doing over there at GitHub next um as R mentioned for for all of our speakers today we're not going to be taking live questions so

02:50:31 I'm going to ask you guys to go and find them after the fact um as rul mentioned he'll be at the Microsoft Booth over in the Expo and then um we'll give you locations for for the other speakers as they finish up as well um next on we have Kevin how he is the head of product engineering at codium um um this is probably one of the most incredible pivot stories in AI of the past year and a half uh company used to be exif function and now one of the best uh code assistant products out there actually

02:51:00 voted by developers as the highest satisfaction uh code assistant product recently and a sack overflow survey and just hit a million downloads uh in VSS code last month which is truly incredible um so I'm going to welcome him to the stage Kevin how um head of product engineering [Music] all right cool thank you rawle we are going to kick it off with let's see make sure that's not my slack up there cool all right all set um so my name is Kevin and I'm going to be talking about how embeddings are stunting AI agents uh

02:51:43 so I'm going to let you in on some secrets about how we build the product uh and exactly what we're doing behind the to improve your code gen experience so at codium we are building AI developer tools and we're starting with an IDE plugin and as uh as mentioned before we've been downloaded over a million and a half times uh we're one of the top rated extensions across the different marketplaces and to re reiterate we offer free unlimited autocomplete chat and search across 70 different languages

02:52:12 and 40 different Ides so we plug into all the popular idees uh we are the highest rated developer tool as voted in by developers in the most recent stack Overflow survey uh and you'll note that this is even higher than tools like chat GPT and GitHub co-pilot and importantly we are trusted by Fortune 500s to deliver high quality code that actually makes it into production and we do this with top grade security licensing attribution for some of the largest Enterprises on the planet our goal at codium is to empower every

02:52:43 developer to have superpowers both inside of the ID and Beyond and today I'm going to let you in on some secrets about how we've been able to build a tool like this and why CH uh users choose us over the other AI tools on the market and the short answer is context awareness so here's a quick overview about what context looks like today uh we're all familiar since we're at an AI conference with the basics of retrieval augmented generation the idea being that a user puts in a query um you accumulate

02:53:15 context from a variety of different sources you throw it into your llm and then you get a response whether that be a code generation or a chat message um here's a concrete example about how retrieval can be used in code generation so let's say we want to build a contact form in react um now you could go to chat GPT you could ask it to generate a contact form but in reality on a moderately large code base this is really not going to work it's not going to give you uh things that are personalized to you uh and this is

02:53:42 really where contact retrieval comes in we need to build a contact form that you know is in line with our design system components let's say you already have buttons uh and inputs it has to be able to uh pattern match with uh with local um local instances of other forms inside of your codebase it has to ingest your style guide for example if you're using Tailwind you have to be able to detect and make the form look and feel like every other thing on your site uh and then of course there's documentation

02:54:09 both locally and externally um for packages and other dependencies so the question becomes how do you collect and rank these items so that our code generation can be both fast and accurate for your use case so to dive into a couple of different methods about how people are tackling this today there's really three main pillars the first one is long context so this is the idea that if you expand your prompt window in your llm it can read more input and therefore be a bit more personal to what you're

02:54:37 trying to put uh what you're trying to generate right you just shove more items into your prompt but this comes at the cost of latency uh latency and financial cost so one of the most recent examp with Gemini um Gemini actually takes 36 seconds to ingest 325k uh tokens to put this into perspective a moderately sized or even small repo is easily over 1 million tokens uh and that accounts to about 100K lines of code so in this instance most Enterprises have over a billion tokens of code it's simply not feasible

02:55:09 to be throwing everything into a long context model the second method is fine-tuning so for those that are familiar fine-tuning is the idea of actually tweaking the weights of of your model to reflect the distribution of the data that your consumer expects right and so this requires continuous updates it's rather expensive computationally you have to have one model per customer and it's honestly prohibitively expensive for most applications and finally we have embeddings and for all of you hopefully you're familiar this is

02:55:36 a relatively proven technology today um it's pretty inexpensive to compute and store uh but the difficulty that we're about to dive into is that it is hard to reason over multiple items it also has a low dimens space and I'll I'll talk about that shortly so to dive deeper into embeddings the whole concept is that you take your objects you throw it through an embedding model and then you end up with some sort of vector some sort of array of numerical values and it this is in a fixed Dimension and so by mapping

02:56:05 and chunking code we can map it to an embedding and that allows us to quickly search over our functions our documents whatever you decide to chunk by um and this is what embedding search is called uh embedding search like I said Is Not A New Concept there is a bunch of model models that have tried to optimize and in this example we're looking at uh one of the kind of Northstar eval benchmarks um it's become increasingly popular and the question becomes how do we fit millions of lines of code into an llm

02:56:35 model so that we can actually generate useful results and so it's evident through the years that we're actually hitting a ceiling on what is possible using these traditional uh vector embeddings and over time even the biggest models uh are approximating to around the same level of performance as you can see everything's kind of within plus or minus five and at codium we kind of believe that this is because fundamentally we cannot distill all the the dimension space of all possible questions all possible English queries

02:57:02 down into the embedding Dimension space uh that our vectors are going to occupy and so at codium we've thought very critically about what retrieval matters to us are we measuring the right things and does semantic distance between these vectors really equate to things function relevance in the concrete example that I showed earlier and so what we landed on is that benchmarks like the one that I showed you before heavily skewed towards this idea of needle and a Hy stack it's the idea that you can sift through a corpus

02:57:33 of text and find some instance of something that is relevant to you note it is only one single needle so in reality code search requires multiple different needles right we showed that slide earlier when you're building a contact form you need all these different things in order to actually have a good ation and these benchmarks really don't touch that and so we decided to use a different metric and it's called recall 50 the idea and it's definition is that um it's what fraction of your ground truth is in the top 50

02:58:01 items retrieved so the idea being now we have multiple documents and we're now looking at the top 50 documents that we retrieved how many of those are part of our ground truth set so this is really helpful for understanding document multi-document context especially again for those large large Co bases and and now we actually have to build a data set around this and so this is where we did a little bit little bit of magic we wanted to make the eval as close as possible to our end user distribution so we had to compile our

02:58:28 own data set so what we did this is a PR that I put out um a few months ago we looked at PRS like this it's broken down into commits those commits we can extract and actually match them with the modified files right so now we have this mapping from something in English to a list of files that are relevant to that change and you can imagine we can hash this in many different ways but ultimately the point I'm trying to make is we are creating a eval set that mimics our production usage of something

02:58:57 like a code gen product and so this message serves as the backing for this new type of eval where now we can run at scale this idea of product-led benchmarks it gets us closer to the ground truth of what our users are actually experiencing and what retrieval tweaks and retrieval actually mean to the end product and so we threw some of the uh currently publicly available models at this notion of retrieval this idea of using commit messages and we found that there is reduced performance um they're unable to

02:59:30 reason over specifically code but then also specifically this kind of real world notion of of English and and commits right and so at codium we've been able to actually break through this ceiling this is something that we've worked very hard at we have to to to redefine exactly how we are approaching retrieval in order to be kind of in our class of our own so that when you are typing in your ID when you're chatting with our assistant when you're generating Auto completes we're retrieving the most relevant things that

02:59:58 are are for your your intents so now the question becomes how do we actually get this kind of Best in Class retrieval and so I'm here to give you the very short and sweet answer which is we throw more compute at it right but of course that can't come with absurd absurd uh uh cost right Financial cost uh so how do we do this actually in in production how do we actually do this without recurring an unreasonable cost and so this goes back to a little bit of codium secret sauce right we are vertically integrated and what this

03:00:30 means is that we train our own models so number one we train our own models this means that these are customed to our own workflows so when you're using our product you're touching codium models number two we build our own custom infrastructure this is actually a very important point and actually connects to the whole EXA function to codium Pivot that we discussed earlier EXA function is a ml infrastructure uh company and so what we've been able to do is build our own custom infrastructure down to the

03:00:55 metal this means that our speed and efficiency is unmatched by any other competitor on the market so that we can serve more completions at a cheaper cost and finally we are product driven not research driven now what this means is we look at things like actual end user results when we actually ship a feature we're looking at real world usage and we're always thinking about how does this impact the end user experience not just some local benchmark Mark tweaking so we could spend all day talking about

03:01:21 you know kind of why codium has done this and yada yada but that's a talk for a different time so I'm going to talk about something that I find very cool and this is the reason why we've taken this vertical integration approach and been able to turn it into something that we call M query so M query is this way of taking your query so similar it's that idea of taking your retrieval query you have your code base and let's just say you have n different items and because we own our own infrastructure and train our

03:01:49 own models we're now making parallel calls to an llm to actually reason over each one of those items we're not looking at uh vectors we're not looking at small Dimension space we're literally taking models and running them on each one of those items so that you can ensure you can imagine you know you run chat GPT and tell it to say yes or no on on an item for example that is going to give you the highest quality highest Dimension space of reasoning this leads into very very high confidence ranking

03:02:17 that we can then take into account like your active files your neighboring directories your most recent commits um you know what what is the ticket that you're working on currently we can compile all this to give you you know the top n uh documents that are relevant for your generation so that we can start streaming in higher quality Generations higher quality chat messages uh things of that nature and the reason behind this is again it's that vertical integration it's that idea that our computation is one/ 100th of

03:02:47 the cost of the competitors we are not using apis and as a result our customers and our users actually get 100x the amount of compute that they would on another product and so we're willing to do that we're willing to spend more compute per user because it leads to a better experience and so like I mentioned earlier I lead our product engineering team so we always want to Anchor ourselves around these three different things one that we have to build a performant product it has to be really fast for those of you that have used the

03:03:16 product you can probably attest this M query runs thousands of llms in parallel so that the user can start streaming in code within seconds not minutes not hours seconds and often times milliseconds it has to be powerful right none of this matters if the actual quality and the actual Generations that you're building are wrong right and finally it has to be easy to use we're building an enduser product for people today that's in the IDE tomorrow it might not be in the ID how do we actually build something that is

03:03:45 intuitive to understand that people can grapple with and see exact what my model is thinking and so because we have the benefit of distribution uh we were able to roll this out to a small percentage of our users and by small percentage we're dealing in the order of you know million plus downloads this actually reached the surprising number of people and what we've been able to see is that um we were able to successfully reason over these thousands of files in people's monor repos in people's remote

03:04:12 repos and select what was relevant right we can very accurately deem which file FES are relevant for the generation that you're trying to have and the result as you can see this is a real-time GIF is both fast and accurate so I'm asking for usage of an alert dialogue it's going through and I think I panned down here um this is kind of a Shad CN component that I've modified internally we're we're pulling in basically the source code of of what is relevant for Our Generation Um and ultimately the results

03:04:42 of this experiment with that users were happy they were thumbs they had more thumbs up on chat messages they were accepting more generations and we were able to see that ultimately we were writing more code for the user which is the ultimate goal it's that idea of how much value are we providing to our end users and so we built this this context engine right this this idea of M query this idea of ingesting context and deciding what is relevant to your query to give you coding superpowers and so

03:05:10 our users will generate today they're generating autocompletes they're get getting chats search messages but in the future they're going to generate documentation they're going to generate commit messages code reviews uh code scanning they're going to take you know figma artboards and convert them into comp uh into uis that were built by your own components the possibilities are endless but what it starts with is this Bedrock this very hard problem of retrieval and it brings us to again one of the reasons why codium is approaching

03:05:38 this problem a little bit differently our itation cycle starts with product driven data in eval so we're starting with the end problem we're building a product for millions of people how do we start with what they're asking for and how do we build a data set and eval system locally so that we could iterate on the metrics that matter secondly because we're vertically integrated we're taking that massive amount of compute and we're going to throw it at our users you know paying or not paying we're going to throw it at

03:06:05 our users so that they can get the best product experience and the highest quality results and then finally we're actually going to be able to push this out to our users in real time overnight and be able to get a pulse check on how this is going you know this is what we did for for M query and when we evaluate in production we can say you know thumbs up thumbs down and then hit the drawing board again back to that same cycle repetition and so you can start seeing how these pieces of compounding

03:06:32 technology come together right we've alluded to some of them today modeling infrastructure being able to retrieve but then it also includes things like as parsing indexing massive amounts of repos knowledge graphs parsing documentation look at websites online the list can go on and on and on but we're confident that we're solving these problems one piece at a time using that same iteration cycle that same idea that we're going to take the the distribution and knowledge that we have and that additional compute that we're willing to

03:07:03 afford each user to solve each one of these puzzle pieces and um I want to leave you with uh a parallel analogy so in my past life I had experience in the autonomous driving industry so to bring over a metaphor for from that industry in 2015 Tech crunch boldly predicted that that was going to be the year of the self-driving vehicle uh it was largely uh you know now we're in 2024 so we can look back in hindsight largely untrue right we were doing things like Sensor Fusion we were decreasing our polling rates we were running offboard

03:07:37 models all this in the effort of making turistic that would compensate for the lack of compute that was available because consumer graphics cards were not as popular or not as uh powerful as they are today fast forward today we're seeing 100x the amount of compute available to a vehicle you can take a wh around San Francisco which I encourage you to do it's a wonderful experience um but that means that we're actually able to throw larger models at these problems right more sensors higher frequency and

03:08:03 now 2024 Tech crunch has released another article that said will 2024 finally be the year of the self-driving vehicle and we can now look at this pattern and say driving performance was substantially better by throwing larger models being able to more and more data and so at codium we believe that this embedding based retrieval is theistic we should be planning for AI first products throwing large models at these at these problems so that AI is a first class citizen we're planning for the future and finally we also believe

03:08:37 that ideas are cheap you know I could sit up here and tell you all these different ideas about how you know we're going to transform coding and the way that the the the the theory behind uh possible solutions but what we believe at codium is that actually shipping actually showcasing this technology through a product is the best way to go and so if you agree with these beliefs you can come join our team we're based in San Francisco and you can download our extension it's free I'm not obviously uh uh what's it called I'm not

03:09:07 advertising uh the core product nearly as much we're kind of talking about the technology but you can experience this technology firsthand today by downloading our extension it's available on all the different plugins uh VSS code jet brains Vim emac uh and you can see how this infrastructure and the way that we've approached product development has shaped the experience for you as a user and then of course you can reach out to me on Twitter uh I put my handle up there I'll be kind of floating around

03:09:33 outside so if you have other questions or interested in what I had to say um but I hope that you learned something today I hope that you know you use codium you try it out and see what the magic can do for yourself thank you thank you so much Kevin and you guys can find Kevin outside the salon after the tracks have wrapped if you have any follow-up questions for him um next I'm very very excited to bring up to the stage Michael Tru he is the co-founder of cursor and previously created highlight at two Sigma which is an

03:10:09 artificial intelligence programming challenge back when he was an intern there uh and it still persists today inside T Sigma um cursor as you guys may all know is one of the most popular AI Cen tools out there right now I can't count the number of Twitter mentions I see of the tool every single day and on top of that they count amongst their users customers like Shopify Samsung open AI ramp replicate the list goes on and on um so very excited to be uh bringing up Michael to talk through this incredible

03:10:45 product great to be here uh can the audience hear me okay amazing uh great to be here I'm also here with swall my co-founder um thanks for having us uh we're going to talk through cursor and give you a high level sense of what we'd like to do over the next few years and then do a little bit of a deep dive into some of what we've built so far uh and then if there's time we can take audience Q&A at the end um and so here on the first slide this is to set up kind of the problem that we're nerd sniped by which is what does programming

03:11:15 look like in the age of AI and to frame this uh you know this is a little bit anachronistic because no one really wrote x86 and they didn't do it in terminal.app but on the left here you know sort of in the 1980s before then we had machine code and then over the next many decades Humanity invented things like highle programming languages and syntax highlighting and navigation features and lints to make building software much easier uh and this transformed developer productivity and so over the next 5 to 10 years we think

03:11:48 that level of a productivity jump is going to happen uh many times over in a much more compressed time frame and we're really nerd tonight by this problem of what is the equivalent of a high LEL programming language what is the equivalent of all of this tooling around programming languages in the AI age um and so just to frame the problem and talk a little bit about how some other folks are thinking through it I think that you know in the popular discourse there are kind of by and large uh two ways people are approaching the

03:12:17 problem of how does AI affect programming uh one is this kind of agent approach which uh seems to advocate for you know programming kind of goes away ceases to exist as kind of a high level profession anyone can build software and mostly the way we build software will be through you know prds or chat messages that then get turned into code bases or big code changes um and then on the other side of things you know there are folks who are building you know really useful plugins to existing coding

03:12:46 environments and are kind of nipping at the edges of you know we can make ghost Tex auto complete better and we certainly can um and you know we can we can optimize the developer experience in little ways and to contrast this with how we're thinking through the problem we think that programming is still going to exist in 5 years it will still be a profession programmers will still be paid a lot of money uh it will still be a technical discipline but it's going to change a ton and it's not going to

03:13:12 demand just a plug into an existing coding environment it's going to demand an entirely new tool for doing sofware engineering and so this focus on really pushing theing of the amount of work the tool can take on while keeping the program in the driver seat is our Focus as a company and so uh to talk through a little bit you know what we've built so far so we built you know our product is called cursor it's a code editor that's built for programming with AI and our goal is to be the best tool for

03:13:44 professional programmers to use Ai and so far we focused on a few areas mostly around code writing and Q&A and I want to talk through a couple of the the pieces of things we built in the code writing bucket uh because they also kind of illustrate why you would need uh a new Dev environment for this and not just a plug into an existing coding environment so one is we focused on um predicting the next move of a programmer in a code base and this started with you know great works a great work from Folks

03:14:13 at co-pilot with ghost text autocomplete and we took this idea to kind of the next level of you know if you're a programmer working within a code base you're not always just typing characters after your cursor sometimes you're jumping to a completely new place sometimes you're doing a diff you know you're deleting lines you're inserting code in different places and so we trained uh a model to predict your next edit within a code base and the next place you're going to jump to and the result is uh an auto complete system we

03:14:42 called copilot Plus+ um that that can predict these things and kind of the the second piece of the product I want to talk through is called command K uh which lets you go from instruction to code and select a part of the code base ask for it to be changed and then iterate with the AI on that block of code um and so I can talk through some of the technical details of how we went about building each of these both on the model side of things and also maybe on the editor side of things too um okay so uh copile Plus+ so it started off

03:15:13 with you know when we first wrote Our uh our vision for what we wanted to do the first line in there was like oh copal should do your next edit uh but over the next like couple of months every time we would prototype something it would include something like strikethroughs or it would move your cursor a ton when you were typing and that was super annoying or the model just wasn't accurate enough um uh around October and November of last year we had a couple of very interesting breakthroughs one of them

03:15:42 was learning how to use the trajectories of programmers so you know you go from one file to another file uh using commit histories to learn how like programmers do diff after diff after diff of these like coherent edits and then learning a model to predict them and combined with them are uh we had to come up with this side by side edit ux um and the nice thing about it is super easy to put parse uh but also like it's not something that's super intrusive when you're typing on the command K end of

03:16:15 things uh the super uh interesting problem for us was how to make it both low friction so you could if you have an instruction you want to type in a couple of characters you can do it very quickly but also just pull in context from across your repository or across your recent files so we don't make a mistake uh we still are on the journey for making both of them even more accurate and you know every couple of weeks we have either new model updates or new context updates uh you know if you Ed cursor like six months ago I'd

03:16:46 recommend trying it again and you'd find it both much faster and also much more accurate yeah and um both of these you know uh both of these pieces of code writing are just the beginning um they're also you know in addition to what swall said which is you know we're always optimizing the background context building for these things we're optimizing the models uh we're optimizing the ux in little ways uh these we think are also the start of a journey when it comes to the final form factor for what programming looks like

03:17:17 with I uh two directions that we're especially excited about in the future in addition to building off of these uh one is under the bucket of making programming feel like writing and reading pseudo code and so you're already seeing the right side of things here we're now the keystrokes people are typing in their code editor are not really corresponding to go and rust and types scrap you know people are writing things that are much more tur and kind of look a little bit like gibber and they're getting expanded into code by

03:17:44 the diff based auto complete or they're writing you know higher level instruction and they're getting turned into code changes by command K and then on the read side of things one thing we're experimenting with is you know are there times when it make sense to trade off the formalism of a real programming language with concision and you know sometimes it might make sense to give the programmer kind of a slider that lets them control the level of abstraction of the code base that they're looking at and lets them look at

03:18:09 something that looks a little bit more like pseudo code uh and let them both read and use that for navigation and then also edit that and have the change get made down at the source code by uh the AI and so this idea of you know still giving programmers control both control over the level of abstraction and also letting them you know gesture at Specific Instructions in a code base instead of stepping back and you know having to write something like a PRD which is you know uh very divorced from the code is you know an example of one

03:18:38 of the the things you know sort of the idea space that we're really interested in and then another bucket uh that we're super interested in for the future is letting the AI do kind of constrained tasks in the background uh like I mentioned before there's a lot of interest in agents having Bots do things end to end use tools maybe go from a PRD to an entire code base or a big set of code changes we think that for professional programmers for a long time the Tech's going to need to progress before we can really talk about endtoend

03:19:08 automation of PRS and instead in the meantime the way that this technology is going to be deployed is you're going to really constrain what the AI can do you know write the interface for a method or a class and ask the AI to go Implement that and then you can go use that method or class yourself um but you know give the AI more of a latency budget you know five minutes to go Implement that stuff instead of the you know 10 15 seconds that are required if you're working with it in the loop and have have a constrain

03:19:35 agent go and work on that code uh and already you know both here on the the pseud code side of things and especially on the constrained background agent side of things we have internal experiments that look very promising um and so maybe to talk a little bit through kind of some of the the tech aspects of what's been required of us uh to build both copal Plus+ Comm K you know these next action prediction and instruction of code respectively features and then also the Q&A and chat and debugging features that we built so

03:20:02 far uh a lot of tech has been required um to kind of do this this vertically integrated full stack AI product so as SW I mentioned you know we've done a lot of work on the next action prediction model that's powering copilot Plus+ the way that started was you know we started by trying out of the box models on this idea of predicting the next action uh that a user is going to take in in a code base and the problem was the models weren't that accurate they were expensive and they were slow and so we

03:20:28 started with you know having small curated data sets and doing a parameter fish and fine tune to you know take some out of the box models and make them better at this objective and then we went even further and we started to get the open source models uh to be you know good at this objective and now we have very small models uh that are very very good at this next edit prediction OBC Ive and we've also spent a bunch of time optimizing the inference for these models too where uh when you're rewriting

03:20:54 code often you know one thing you waste time on when you're rewriting code is the unchanged parts of the code and so there are things you can do in the inference environment to kind of jump over those unchanged pieces and then you want to talk through some of the other kind of technical points here uh one of the things that we found really interesting was every time you would open up a code base for embeddings uh for almost everyone things will be super duper slow uh you know if you opened up you know the sqlite repo or if

03:21:22 you opened up a much larger repo like lvm you would end up in a in a in the situation where like llvm would take hours and hours and hours to upload which is just just kind of unacceptable for uh for us uh so we had to build a very performant uh you know file syncing engine uh that that sort of syncs embeddings across the server you know it can generally something do something like a thousand files over a few minutes if you're like instacart and have like hundreds of thousands of files it will take like tens of minutes but it's not

03:21:55 something that would take days uh and when you do edits uh on that code base our embeddings sync extremely quickly and that means that your context is almost always very real time and it's not going to be something where you know you check something out and it's like not updated for for an hour or so the other things we've sort of done that iph find most interesting are are a lot of model caching tricks uh that rely on both the you know code editing features you know rely on how the models sort of relate to each other in so in in

03:22:31 in a general codebase a lot of the files have links that come from you know go to definition links that come from a lot of the semantic features of a language and using that to actually build context has been something that's been super useful for us yeah I guess I'll say one last thing about remote performance profiles which uh you know when you're building an editor you're not only working on the sort of AI side of things we found it incredibly important to both make cursor really fast and you know that's

03:23:02 something that both the VOD team works really hard on but that's something we've sort of developed our muscle in as well so if you you know hopefully will come work for us you don't not only work on the AI side of things but also on you know building a performant editor that shipped to hundreds of thousands of people which is just an interesting development problem in enough itself and maybe just to wrap it up here um you know as a Shameless plug we're always looking for brilliant people to join us uh we're really small talented

03:23:33 team very in person based in San Francisco and we're looking for both talented creative people on the design and product side of things and also people all on the other end on the research scientist side of things because as mentioned we're working on the full stack uh we want to build the tool when it comes to you know down to the interface and figuring out what programming is going to look like there and then also working backwards and building the most useful tack for people and sometimes that requires using the

03:23:58 biggest smartest models sometimes that requires using you know models that are really specialized to a particular task are very fast and very cheap and so you know to leave you with this in the next 5 years we really believe that it will be possible to build a tool that automates almost all of software engineering as it looks like today and transforms the discipline of programming into one where individual Engineers can build systems that are much more complex than even entire engineering teams can

03:24:28 build today and TBD will be the ones to execute on that opportunity but that's one that really excites us and you know we wake up every day passionate to try and solve it so if you want to join us uh the best way to reach us is at hiring at ere. Inc and thank you all for your time and happy to take questions if we have extra time too [Applause] mic back thank you um thanks so much guys for going through that um yeah we do have I think three minutes-ish left for Q&A um oh we have one already here

03:24:59 in the front okay I will I just want to first off thank you it's probably been the most productive year writing code in my life in part large partly because of cursor um for folks who have started playing with a little bit maybe use the chat you know ask it to write things paste in the chat what are some like more Advanced Techniques of using cursor that you've discovered beyond the obvious of just you know moving code from the chat into the or using command K Etc I don't know if I have okay uh it's

03:25:36 been really interesting seeing the different ways that people end up using the tool um I think there are cursor is a a pretty powerful and full feature tool at this point and there are lot of hidden features as you dig more and more into the tool so for instance on the chat side of things uh one thing a lot of folks do is these models often output code blocks in chat and they output like kind of shorthand code blocks with dot dot dots interpers and taking those code blocks actually putting them and

03:26:03 implementing them in your code base can be a bit tedious and so we have specialty models for instance that will apply those code box from chat um I would say kind of the two features we listed today and kind of dive deep into one of them uh a lot of people know about but maybe less than you know the entire user base uh is command K and so often that inline code editing is way more ergonomic than going to something like the chat and then you know having to apply changes from chat and going back and forth um there are also some

03:26:32 hidden debugging features too so you know if you get a stack Trace in the terminal uh we have this kind of a specialty Loop to help you debug that uh and if you run into really thorny linter errors we have a way to you know to B basically to bug those two you have to kind of hover over things to discover that but by and large right now the most useful parts of the product are this you know next edit prediction with co-pilot Plus+ uh command k for writing code and then chat for Q&A and uh asking

03:27:01 questions about a code base any more questions I think we have time for one more maybe one and a half uh again some work on the product um my question is do you run like personalized models uh I'm finding more and more that it feels like it knows what I want to do next hello oh okay my mic is enabled um we've done a ton of work on the the context building side of things uh so it's not uh that models are being edited in the weights per se but we're using the in context learning abilities of these models to find the parts of the

03:27:40 code base that are most relevant to your task or query uh and that goes into both the next action prediction next edit ition side of things where we're looking at your revision history we're trying to figure out the next thing you're you're going to do so if you're in the middle of a refactor that model is really good at figuring out hey out of the last 200 lines you changed what were the most important 15 that gives me a sense of your intent uh and ditto for you know chatting command K it's figuring out the

03:28:04 parts of the code base you know the building blocks that are most relevant for writing the code you wanted to write or answering the question that you have um for the future one thing I'd like to comment on is sort of a nerd snipe would be uh two problems we're very interested in in the sort of personalization end of things is one of them is can we uh run some sort of you know agentic Loop in the background that's building context for you while you're asking questions and then the second problem that we're

03:28:29 really interested in can we make a model learn your codebase so if you have you know research ideas for you know actually learning a really large code base uh in which case you might be sort of start getting inaccurate when you do retrieval and uh we're we're super interested in if you have ideas awesome well thank you so much everyone uh for the time today we had some incredible speakers here um as mentioned please do feel free to find them after the fact um Rahul said that he'll be over at the Microsoft booth in

03:28:58 the Expo Kevin will be around here and I think the cursor guys will be around here as well um thanks again for for attending no I actually I already have so many web tools that I have to sign into some of those apps work better on some machines and others sometimes there are restrictions on what websites we can and can't visit from within the school but I can always send an email [Music] n [Music] [Music] [Music] everyone I'm an AI engineer B in San Francisco along with my teammates I created math matrix movies at a

03:31:36 hackathon and SF on May 11th 2024 today I'd like to show you what our project can do because I think it's really cool what it does is that generates really cool math explainer videos in a truly unique style that is able to get Concepts across visually this is something that I think is really unique that you may never have seen before so AI hackers let [Music] he [Music] [Music] n [Music] [Music] [Music] [Music] [Music] oh [Music] [Music] [Music] will [Music] [Music] [Music] my [Music] [Music] am

03:38:29 [Music] [Music] while [Music] [Music] am over [Music] [Music] [Music] [Music] [Music] [Music] is [Music] [Music] [Music] feel it's forever [Music] [Music] going [Music] [Music] [Music] [Music] [Music] [Music] we [Music] [Music] [Music] n [Music] she it sometimes [Music] [Music] [Music] [Music] a all [Music] hold [Music] [Music] got [Music] [Music] n [Music] B e [Music] bre [Music] [Music] true [Music] ring [Music] [Music] let [Music] [Music] [Music] feel [Music] true [Music] [Music] [Music] [Music]

03:57:27 [Music] [Music] [Music] [Music] [Music] you [Music] would you you you you you [Music] [Music] [Music] you you you you [Music] sh [Music] you you you [Music] you you you you you you [Music] [Music] we got [Music] [Music] an don't want to feel it your SES [Music] [Music] esep now Hest right [Music] now I feel [Music] [Music] [Music] AI [Music] I Need You [Music] Now broken want toen [Music] I need you f it out all the [Music] [Music] [Music] to holding our heart broken dreams I you [Music] br [Music]

04:09:13 you know [Music] [Music] [Music] [Music] [Music] [Music] know [Music] [Music] a [Music] [Music] [Music] for it's [Music] [Music] [Music] [Music] [Music] mind I know [Music] [Music] [Music] I'm [Music] [Music] [Music] [Music] [Music] [Music] my [Music] [Music] my I'm all I [Music] [Music] ready [Music] [Music] [Music] [Music] ready I [Music] and [Music] [Music] [Music] ready [Music] [Music] [Music] only [Music] in your [Music] feel it all come back in a moment [Music] [Music] [Music] [Music] feel

04:20:45 [Music] [Music] fire [Music] it slow mo [Music] [Music] [Music] it it all slow motion [Music] [Music] n n [Music] [Music] [Music] [Music] [Music] he he [Music] [Music] back [Music] [Music] we [Music] [Music] [Music] problem CL like [Music] super superl [Music] now

[Music] [Music] [Music] up as [Music] El [Music] [Music] [Music] is [Music] [Music] [Music] come us [Music] [Music] [Music] am [Music] [Music] [Music] so come [Music] [Music] [Music] [Music] [Music] we [Music] [Music] [Music] [Music] she

04:37:41 got it it [Music] [Music] [Music] re a [Music] [Music] [Music] how [Music] [Music] [Music] St [Music] B [Music] d a [Music] our [Music] [Music] up in [Music] you [Music] [Music] [Music] [Music] hor ch [Music] [Music] [Music] [Music] [Music] [Music] [Music] [Music] [Music] [Music] this you [Music] you would you you you you [Music] [Music] [Music] you you [Music] you you you [Music] you [Music] [Music] but you be with me right now feel [Music] we in down it'sing [Music] [Music] [Music] [Music] baby IEP

04:57:48 [Music] [Music] [Music] you right now I [Music] [Music] you [Music] [Music] [Music] heart trying to get on my feet CAU in the mad I feel youen [Music] [Music] I want to be next [Music] to you want to be next to me hold our Broken Dreams to know get [Music] [Music] know [Music] [Music] [Music] it was back in9 [Music] [Music] it [Music] [Music] [Music] are hold I'm [Music] [Music] [Music] bre right [Music] up [Music] [Music] I'm [Music] [Music] I'm [Music] [Music] [Music] you and me we were the only

05:09:13 on we were hold [Music] [Music] [Music] you come it open slow motion [Music] [Music] [Music] [Music] Tom heart [Music] [Music] always on my mind [Music] [Music] all right um thank you everyone for coming to the afternoon edition of Coen talks very excited to be an incredible group of speakers today for this afternoon session uh my name is Britney I'm a GP at CRV uh also known as Charles R Ventures we're an early stage Venture Capital firm that's been around since the uh' 70s so actually one of the the

05:12:45 older funds in the country um we do primarily seed and series a Investments and I specifically focus a lot of my time on software infrastructure hence being here with all of you today um to give you some interesting facts about uh Coen today and about what they're working on um have very exciting speakers here starting off with Quinn slack from Source craft morgante pel from Grit and lastly gunun Patel from pal Alto networks um with that I will have Quinn come up here on the stage um Quinn is the co-founder and CEO of

05:13:22 source graph he also previously co-founded blend um he has been building the company Source graph since 2013 so definitely an OG uh in the space lot of folks and most importantly he was also the first employee of Bleacher Report which for those of you who are in Sports uh is one of the primary sources for all of us uh today so with that I will bring Quinn up here to the stage to talk about how why the AI Emperor has no daus thanks all right thank you Britney it's great to see you all again I'm Quinn

05:13:57 slack I'm uh Co and co-founder here at source graph and yes I was at bleach report I think I learned I'm better coding than doing sports so I'm in the right role here uh I want to talk to you about why most devs still do not use code AI it's probably a mindboggling fact to all the people in this room but it's true and we'll have a good discussion about that uh I'm a coder I code all the time been building Source graph which is code search and code AI for Dev since 2013 I merged four PRS earlier today but I want

05:14:29 to talk mostly about the product and and what to build and how to get it in the hands of users because if you don't do that right then all the code that you write is for nil so three points I want to make one is that most devs don't use code AI I want this to scare you I want to scare you in this talk and then I want to share some of the mistakes we made and some of the tips from our experience doing all the things wrong and then trying to figure out how do we do it right at source graph as we've been

05:15:02 building Cody all right um first just show of hands here who uses who in the last week has used any kind of code AI tool while coding all right man I wish the whole world was like this room but we are in the heart of it in San Francisco and the whole world is very different most devs do not use code Ai and I'm going to give some numbers here I'm privy to a lot of private information and stuff that I've heard as well and all of this is in line with what I've been hearing so first number oh man this is a great statistic

05:15:34 92% of devs use code AI tools at work wow that means these tools must have tens of millions of users uh this was a study from GitHub uh 500 people well it turns out this is only in the US and they had a very broad definition of what it means to use a code AI tool so here's a case where the hype kind of outstrips the reality a common pattern this is an infographic from GitHub 70% of developers see a benefit to using AI coding tools at work surely if there's a benefit they must all be using these things right I mean if only

05:16:08 it worked that way all right some numbers for the broader Universe uh there are 1.3 million paid subscribers to get up co-pilot this was what Microsoft reported in January 2024 1.3 million is a lot but if you're just paying for something are you actually using it and also 1.3 million is not a lot compared to all the people in the world that write code and there's actually a way to get an even finer point on that number Get Up release to study saying uh if you look at the fine print that in this time range this year

05:16:42 935,000 devs received a suggestion not even accepted it so there's going to be some that saw that little ghost text in their editor never actually accepted it and also that's yearly active users I don't know of any companies that go and site yearly active users uh surely the monthly active users and daily active user figures are much lower than that so these numbers are you know pretty you could fill you could fill a city with these people but it's not that many people when you compare it to the 26

05:17:10 million professional developers in the world and I think this is an undercount um I will say that you know how do you get this number I Googled a bunch of different stats and they all were around this um so you know that's the Vibes citation there there's a lot more people that touch code in some way or another I again it's really hard to get a number on this but I'd say probably 100 150 million people in the world touch code in some way and that includes students and other people and so on So You Know

05:17:37 935k best case what is that an MA that's tiny compared to the number of people that interact with code so this is a scary thing my best case estimate is that around 5% of professional developers use code AI these are people that are paid a lot of money to write code and they do it all day every single day and all of us here in the room we know that there's this amazing new alien technology that's dropped on Earth and then it changes how we code and 5% are using it that's crazy that means room of 20 people 19 are not

05:18:10 using this it just absolutely boggles my mind and that's you know a huge problem and it's an even bigger problem because if you look at what they're actually using the vast majority of code AI use is just autocomplete it's the ghost text I mean yeah that's that's good but anyone who has used anything beyond that knows that AI can do so much more when coding it can write entire files it can answer questions it can fix bugs and all that stuff so this usage is absolutely tiny and we need to realize that we here

05:18:37 in this room we are all the Freaks and for us to be successful requires us to change a lot of Minds out there in the world so how do we change those Minds well you know we've been building Cody Cody is a Codi tool it does all of these things and so we encounter a lot of reasons why people are hesitant at first and I've taken the reasons from our own internal tools and rank them why don't most devs use codei the first one is just n they don't have a good reason uh there's some people in the world believe

05:19:09 it or not that have not used chat gbt that have not heard about it even devs uh there's a lot of devs who are kind of grumpy and they say well it's not perfect I tried it and it gave me a wrong answer yeah no but there's a lot of other people that have figured out how to get the right answer out of it uh a lot of people say I don't need it it didn't help me that much you know again you got to figure out how to make it useful for you a lot of people's company hasn't adopted it's too expensive or you know one thing we're

05:19:36 seeing a lot less of now is these security privacy and concerns I think it's a situation where every enough people are using it so that if it's illegal to use code AI then we're all so it's mutually assured destruction but there's a lot of reasons why people don't use it yet and this should really scare us because there's a lot of companies that are relying on a lot of people at a lot of Enterprises using Code AI there's a lot of people's retirement accounts or option trading accounts or whatever that

05:20:09 are also relying on that and if you think about how technology makes money I mean you know the the money that gets deposited into your bank account in your paycheck where do those dollars come from it takes you know kind of a long route uh just to explore the market a little bit here you know you've got these Foundation model companies like open Ai and anthropic at the top and then you have the AI infra companies and the AI infra companies you know they will send if they're serving and doing inference they'll send some of the money

05:20:40 back to the foundation model companies in some cases they're the same companies and you know what drives usage of that you have some experimental usage people being the looky lose going and trying it out that will turn into some experimental Revenue but all of that can happen without any real actual usage uh but that gives the perception of holy this stuff is hot this stuff is working and then you start to get some real usage some people using it uh you know maybe paying 20 bucks a month on their credit card you get some devs

05:21:10 using it at work and then the Holy Grail the way that software makes money is from Enterprises where there's recurring revenue from real usage that first year contract it doesn't matter it's got to be JP Morgan pays you a million dollars the first year and then they renew for $2 million the next year and all of the money in our paychecks ultimately comes from that if you're working at an AI info company if you've raised money to go and sell some AI product it's ultimately because someone out there

05:21:36 thinks that 5 years from now JP Morgan is going to be paying you $2 million a year so that's the only thing that matters everything else is Downstream of that and it turns out there's just not that much of that going on and also this is an even more lopsided pyramid than I've depicted here we got Nvidia and chipmakers at the top and so much investment writing on it so you know this is where we ultimately need to get to we need this person to be using our software and maybe it's not the coolest thing out there but that is the reality

05:22:06 of how technology makes money and want to put some numbers to this I estimate based on uh my internal information some of the information that GitHub has shared at this conference that the total recurring revenue from code AI usage is around 300 million ARR now look that's a lot of money if you had a company that was doing that then it could go public on its own so that's good but how much of that money actually goes to these other companies not that much uh from our own data at source graph where we

05:22:37 spend a lot of money on AI inference we spend less than 10% of our Revenue so if you take that number and by the way we're not even doing it in that optimiz of a fashion if you take that 300 million number 10% of it that's $30 million going back up to these Foundation model companies and AI infro companies it's a long way from where we are today that amount of Revenue to them making the kind of Revenue that's going to justify these massive multi-billion dollar valuations also another way to look at it is Salesforce salesforce's

05:23:08 annual revenue is $ 36 billion all of this stuff all this hype all this usage of code AI it's it's amounting to a tiny tiny fraction 1 120th of Salesforce so we have a long way to go usage needs to grow a lot or I could be wrong and maybe the doomers are right maybe not for the reason they thought maybe we should never have done this and this is all just going to be one massive height bubble and it's going to pop and we're all going to be miserable but I think because I use code AI every day obviously and you do too this stuff is

05:23:40 real it just it's really early and usage needs to grow a lot so if you're building a product keep that in mind this is what I'm betting on and let's talk about some of the lessons from our experience building Cody at source graph I wish I had this mindset going into it uh just quick background to establish you know why would you even listen to me maybe I should have done this in the first slide but hey here we are uh Source craft has been around as Britney said for 10 years we started out with Co search and then we found out

05:24:11 that if we we had this tool that all the devs used and had all the code in a company and it turns out that was amazing context to build a code AI on top of and so we took that and we built Cody it's got really great autocomplete and inline edits generating unit tests and chat and chat is where we really differentiate because that's where you can make the best use of context so you can figure out why is this broken or how do I change this or where should I start on this that's where we Excel we've got a lot of really big customers

05:24:41 we have four of the top five biggest banks in the country we have most of the Fang or Manana companies or whatever and a lot of other great customers including some that are presenting today and we are the number two code AI company in Revenue second only of course to GitHub co-pilot uh but because we're number two and because we're a startup and we're Scrappy we try harder all right so what have we learned one is that hype fools everyone hype fools us hype fools all of you hype even fools your

05:25:13 customers and the second thing is that AI code completion the thing that auto completes the rest of the line or the next few lines that's like the freakish kind of feature that comes along like one time in a 100 years and we kind of got spoiled it is so perfect so I'll get to that but first just you know on the hype how do you get away from being fooled by the hype I'll share a few tips that have worked for us the first and this is the most important thing if you're building a product and you are

05:25:42 not using it every single day it is not going to make it there is zero hope if you're building it on a team and the people all building it don't use it every day your customers will not but they might say it's really awesome and that's the dangerous thing about this hype customers don't know what they want every customer for about a year said I want fine-tuned models they said that all the time our salespeople would say hey we need to build fine tune models everyone was talking about it get a co-pilot has it

05:26:10 as a bullet you know all of our competitors have it as a bullet that's like italic and it's like coming soon it's some thing really what they just want is they want it to work well and they use some of these terms so don't listen to them because what they're saying is probably Downstream of them sitting at you know some conference like this and getting some cool ideas they want the product to work and if you can't describe why it works and why it's great without using the word AI then you're probably not going to make

05:26:34 it uh this is a really existential question so we make a product that makes developers a lot more productive like 20 or 30% more productive if it's so damn good why are we selling a product it's like the people that have the stock picks and sell newsletters why don't we go and buy a software Outsourcing firm and prove it and monetize our product by actually capturing that value directly that's a damn good question and I wish that we had done that a year ago and we're looking into doing that now and uh if

05:27:02 you don't have the confidence to go and do that to put you know a few million bucks behind that if you've raised a bunch of money then say your product's probably not ready yet and then finally I think we've all seen this um this is actually not even one of the worst ones this is you know a tweet that's got a lot of lot of activity right um just keep in mind that that does not translate into da so don't feel shitty about yourself all the time I wish that Elon would add something like this that would actually tell you

05:27:27 the diu of the products that are getting all this hype but until he does that you can imagine it all right I talked about autocomplete being this freakishly good feature here's how we think about features at source graph this uh kind of four box you know you want to be in the top right and that's where it's an AI feature that doesn't take a ton of time to see is that correct and it's you know really easy and it's used often AI autocab is great because it's literally every keystroke and you can like glance at it

05:27:56 you in a few seconds or even milliseconds 100 milliseconds you can figure out is this correct so that's a really good property of a feature and it just so turned out that the first feature of code AI happened to be smack dead in the top right it's pretty amazing but most other code AI features do not have this level of product Market fit and we need to realize that now we can work on it but you know where's chat not used nearly as often and it's harder to vet that long response edits you know similar I know a lot of people

05:28:27 love doing the inline edits like uh you know meta K option K and Cody and cursor and things like that it's great but we got to be prepared for what actually works being a very different form from what exists today then there's a lot of other stuff like the agentic stuff I mean look it's obviously the future but it's just not there yet who here has used the code AI agent to actually merge a PR in the last week all right cool thank you for helping us you know push the world forward but it's not there

05:28:56 yet and you know other features you really got to go for that top right and drive any feature you make to the top right and if not it's just not going to make it so what are we doing at source graph to address this well we are searching for that next great code AI modality what's the next autocomplete there's some ideas like the next edit suggestion or like co-pilot Plus+ you know I'm I'm skeptical of that uh chat Steve Yi who is at source graph he just wrote this great blog post about chat oriented programming chop where it

05:29:28 turns out there's a lot of devs who have a chat session running in like Cody or even chat GPT or something like all the time while they're coding and it's just one ongoing conversation it's a totally new way of coding and what's weird is it's like the Gen Z's who do it and it's like the 50-year-old you know disgruntled programmers who do it so I love that it's got kind of you know uh some usage in both Generations um there's new ways that people are going to be using Code aai that we have not even thought about and it's so early to

05:29:54 the previous point we always got to remind ourselves to build the manual and explicit thing first we got spoiled by autocomplete it's automatic and you know it's uh it it's just like triggers every keystroke no first you got to make something work in manual and explicit mode if you have chat you got to make it so people manually at mention the context they want before magically inserting the context if you've got an agent put it in the editor and make it work in the editor so that if it's wrong

05:30:22 the dev can just change it right in their editor instead of having to go into code spaces or some other totally different UI so make it manual and explicit first and really easy for the dev to go and fix it and then you can add the magic probably you know you're going to be bogged down enough with uh stuff to make it a great product and you won't actually get to adding the magic for a while when you're pitching always explicitly dehype and set expectations low no matter how much you dehype the person receiving it they will hype it up

05:30:50 in their minds so you got to remove the hype do this so aggressively and uh here's an example from Sam Alman this is a tweet I don't know like a year ago when chat GPD had probably tens of millions of Da so he's right we know he's right and if he's saying this and his product has way more validation than your product well you probably should be saying it like 10 times more and 10 times more intensely also it's kind of cool marketing and IT you know makes you seem cool um yeah you got to become here it's

05:31:23 covered up uh you got to become a Dau yourself or kill the product and then the last thing is just that as a ecosystem all these Foundation model companies all these AI infra companies all these AI applications we all live or die together and dropic and open Ai and mixl and fireworks and all these great companies that we use are not going to get paid if we're not in business making a ton of money on Cody so we all need to work together and realize that we all have so much to benefit from this stuff actually being used from turning

05:31:54 down the hype and building great products that get actual devs using it all the time and that starts with you if you're the one building the product so thank you and if you uh want to reach out to me uh there's my contact info and happy coding thanks so much Quinn for running through that I love to see a presentation that actually DEH Hypes this stuff versus hyping it up more um next we're gonna have morgante pel come to the stage he's the founder and CEO of grit he actually also has a Wilderness first aid

05:32:33 certificate uh so a man of many talents um one exciting recent devel velopment with grit they open sourced uh grit ql in March and it's seen tremendous adoption in that time frame sure he's going to go into that and many other elements of the grit product that they're building so with that I will leave you to it [Applause] morgante cool so I'm morante I'm the founder of grit I'm going to talking about code generation and maintenance at scale uh or CPUs still matter and what it takes to actually make one of these

05:33:13 agentic workflows work in production Quin was talking about how most people have not merged them that's 100% true uh grit has probably merged more PRS than any other company in the world at this point because we focused very narrowly and have done a lot of work above the model layer and we're going to talk about how we did that uh it's helpful to know why I started grit my background is all developer tools I've been working at Google Cloud for five years and built a lot of stuff uh on the devops layer

05:33:40 right thinking about kubernetes and how do you orchestrate very large scale systems uh Working On Tools like customize or terraform templates and one of the biggest things I learned from this was how rare it was for a customer to come to us and ask for a brand new application right people didn't come and say I want to build a new app on Google Cloud it sounds cool 90% of the time customers came and said I have this line of business application that is doing $100 million in Revenue how do I run that on kuber dens right and that's what

05:34:08 all of our templates did that's what everything we built uh in the sort of pre-ai era of automation was all on how you ran existing applications and that's why we started grit because every demo that you usually see the type you know one of these ones on Twitter it's usually type a prompt get a new application from scratch right build something brand new it's exciting it goes really on Twitter that's not what devs do day in and day out to helpers spend most of their time modifying huge applications so that flights run on

05:34:35 time and this goes into three sort of cies developer tools right there's ID developer assistance this clearly has the most product Market fit today it's really easy right like is saying you just do autocomplete right that's very simple thing then there's AI agents that are focused on lowering the floor right they're allowing people to do tasks that they otherwise don't have the skills for right allowing a product manager or other non-technical user to build an application that they don't have the

05:34:59 skill set for this is powerful but I actually am pretty skeptical that that's how most software is going be built in the future it requires a real thinking of how do you actually spec things out how do you think about edge cases basically how we train as Engineers that's required to build great software which is why with grit we focus on raising the ceiling of what great Engineers can do right principal Engineers the most high level Engineers that you work with they're primarily limited by time right how they can't be

05:35:24 in 10 places at once but AI agents can be in 10 places at once if there's the right engineer controlling them that's what we focus on is supercharging the productivity of the top 1% of Engineers it also hopefully gets around the problem of 95% of Engineers not using AI uh the great thing about grit is 95% of our customers 95% of their Engineers don't touch it right there'll be 100 Engineers on the team they're not using grit there's one engineer who is deeply embedded with grit and it's generating

05:35:50 hundreds of PRS with their agents but to do this tools need to change right the ID that you have today it's a scalpel right it's focused on editing individual lines of code and it's great for that but it's not focused on editing hundreds of repos at once it's not focused on how do you open a thousand files and make those changes in them and that's why we built great is because we want to have bulldozers for code right if you're generating huge quantities of code how do you push that around in an effective way uh when

05:36:17 you're not editing individual lines when you're working on a higher level of abstraction and this is super necessary right we've seen an explosion how much code is being generated a lot of our customers are seeing uh 20 to 30% more code is coming out of their teams now just because there's more PRS there's more CI there's everything that's running because you have Cen and this is going to accelerate right once we go from 5% to maybe 50% of people are actually using AI there's be way way more code in the world and we need

05:36:41 better tools for managing that code once it's in production so just an example of what this looks like in practice uh this is a real C customer that we had uh they have been around they've got thousands of repos they've got thousands of developers and they wanted to use open Telemetry instead of logging right this is traditionally a massive effort right you have to coordinate across hundreds of teams to get them to understand open Telemetry to understand how to instrument their code you have to get

05:37:07 them to do actual code changes to swap out their logging Library you have to do a bunch of Education efforts and it's actually very much a people and process problem usually right something where you have a program manager who has a massive Excel spreadsheet that's what I say like grit we compete with is actually Excel not any other AI Dev tool it's that you go have these spreadsheets where you can monage these changes right and tens of thousands of developer hours go into a change like that so a lot of

05:37:30 companies just say you know what that's not worth it right I'm not going to migrate to open Telemetry I'm not going to go into the cloud I'm going to stay in my older ways you know people still have millions of lines of cobal because it's just so much work to do this kind of coordinated change with grit you don't have to do that coordination effort right because you can have one engineer who is actually coordinating that change is driving individual AI developer agents to do the changes you don't have to have a bunch of meetings

05:37:53 because it's just one person telling their their little agents what to do right and you can do it with under 100 develop hours because they're just doing high level orchestration and then thousands of uh compute hours that the AI is doing is that's healing these changes and we've seen this is little project that they had postponed for years because it just was not feasible they couldn't get the get enough on the road maps they got it done with grit in a week right it just open up a thousand PRS across all the repos fix them uh

05:38:19 iterate on the changes merged and migrated over so how do we actually do a change like that uh it's sort of a three-level process um planning is a big part of it right so we index the entire code base we do both semantic indexing so understanding uh embeddings and understanding the intent of each file but we also do a lot of traditional more static analysis indexing so we understand what's the structure of the code what's being imported from where what's the dependency graph right this is all the sort of thing you need to

05:38:44 know to actually do really high quality agentic flows then once we have the plan of how we're going to make changes we execute the plan right so we use large language models that are going to take that change delegate it to a sub agent the sub agent is going to make a modification in one file uses something called gql as well as diff generation from language models grit K is our custom query engine that's able to actually index uh millions of lines of code and find the right places to modify things and then finally push it up for

05:39:12 PR review and be able to uh both have developers who are the director of it right so a typical scenario is that there'll be the Principal engineer who's driving the change they'll review the PRS then IND developers their primary interaction with grit is just seeing a PR land in their repo and they might leave like one line comment that grit will learn from but they don't necess open up the grit UI ever because they're just responding to the changes that come from gr so a little bit more about how we

05:39:37 find code right so our goal here is to find all the error logs in the codebase because we want to migrate those over to open tet TR uh the naive approach that you go to many of the workshops yesterday would be all right just chunk it put it into a rag uh you have a bunch of embeddings uh and you know that you know theoretically could work for maybe some document use cases I'll tell you that absolutely will not work for this problem right if you just go to try to find stuff that looks like looks like a

05:40:00 logging error uh it's going to find a lot of irrelevant context right it's going to find anything looks like logik it llm has a hard time differentiating between a user facing log like an alert in a UI and an actual log that you want to be putting into open Telemetry right there's also unpredictable thresholds right you don't actually know how much code you're looking for you can't do you know retrieve the top 10 uh closest matches in some cases you want to retrieve 10,000 in a lot of cases developers don't even actually know how

05:40:25 big of a change it is until grit starts to propose it for them right so that's where we built the grit ql query engine it's our own Quest and query system that combines the best of static analysis with the best of AI so you've got this query here that's looking for uh logger with some set of arcs right so we're just look for a function call basically and that's a syntactic query so we're just looking for all of our function calls across our entire code base uh and then we're going to say that our args

05:40:50 should be like an error occurred right and that's just we're giving an example of like what's an error message that we might be trying to look like like there is a magical word that converts it into a semantic representation so we want to say what's some code that you know embedding search the coine similarity is sufficiently above a threshold that this is an actual log message versus some other function call that we wouldn't be wanting to modify and then we can finally do a imported from we've got a

05:41:15 built in library to able to understand the whole dependency graph so we can do things like make sure this is actually imported from log for J in this example we wanted to make sure that we're only substituting our log for J logs that will go and Traverse uh the import graph earlier in the program so that brings us to finding the code that we want to change but once we've actually found the code how do we make reliable changes uh and unfortunately really smart models uh still have a hard time doing this

05:41:42 completely autonomously uh just to give an example I just used Claud Sonet today uh 3.5 it's a really good model uh and it uses uh we put put the entire files uh so BN put a bunch of context into the context window 100,000 tokens from grits vs code extension and some of our linter outputs uh and we wanted to just write a function that's going to convert from our lter Json output uh and puts it into diagnostics for the great vs code extension right pretty simple task I promised that everything that's required

05:42:10 was in the context window right it's not something where it had to go retrieve additional information it was all there uh came back with a pretty reasonable completion converts eslint to LSP Diagnostics uh this looks reasonable to me like I I imagine if you look at this code you wouldn't be able to tell anything that's wrong with it I certainly couldn't tell anything that's wrong with it from eyeballing it right uh but this is wrong right and this is one of the main things to understand is that uh humans also can't look at this

05:42:37 code understand what's wrong with it right this is why we have systems that allow us to uh type check lint things right that allows us to understand code that even looks kind of correct is in fact incorrect will fail in production but I went back and uh Claude to fix the code for me I said this broken production I tried to put in my vs code extension it broke and I just ask it why uh as you can kind of Imagine doesn't do any better than I do of just looking at looking and eyeballing the code and understanding why it's wrong

05:43:02 right it just comes in spe says some totally irrelevant answer of how to fix it because again he not not grounded in what the actual errors are uh fortunately uh we have a great tool for typ script called TSC right we can compile this and it's going to go and tell me actually uh grit positions and grit ranges have a slightly different type signature uh than LSP ranges right and this is you know y compilers isic great is we can actually get that information really close in the de Dev loop with this information tread

05:43:28 back into an llm it's able to correct that mistake uh no problem right that's a pretty easy change it uses the convert LSP range to grit range which by the way was in the context window it could have used that before it just didn't realize that it needed to use that until it had the compiler error forcing it to right so this is already how I think I it's important to see that IDs are already making us superum and we need to make sure that all of our AI agents have access to the same tools that make them

05:43:54 uh super AI uh so compilers rock right this basic flow of prompt get some code uh build it type check it and then fix that output based on the LM this is actually really powerful this is probably uh half of what you need to do to build a really good Agent is make sure that you have this flow working reliably uh but they're really slow when you're talking about Enterprise code basis uh so this is real numbers from like one of our customers uh it takes them 10 minutes to build their application from scratch and that's just

05:44:24 for type checking it's not even prod pushing a production build right and this is actually pretty typical if you look at very large scale Enterprise code bases that's why large companies have had to build a lot of caching because it's hard to build a large codebase from scratch which this is completely different than what people usually expect for AI people usually think inference takes a long time right you're waiting for the AI and this is actually a pretty long prompt 30 seconds right we're using a huge model to generate

05:44:46 this code takes 30 seconds but that's dwarfed by the 10 minutes to build the application right this basically destroys our entire agenta flow if we're waiting 10 minutes for every single change to valid it it's correct but this is even more compounded if we're trying to do that in a loop right if we're trying to do a single change it might take a day if you're just doing this naively there's some agent projects that in fact do take a day to make very basic changes because you don't have haven't

05:45:11 done this optimization level they may ask like how are you able to make changes in your Ide at a fast rate right you're not waiting 10 minutes every time you make a single keystroke to get a compiler error uh it's much faster than that it's because there's been a lot of work with language servers uh to solve this so that you can do a bunch of upfront prep so you can build the index in memory have that inmemory queriable index and then only rewrite the parts or only recheck the parts that you've

05:45:34 modified right on every keystroke uh most tools like TS server for example in typescript uh is doing live reconation of figuring out that specific file right and this is much much faster you can do the 30C prompt then one second recompute from the TS server then 30 seconds to fix it and this is a much more reasonable flow right so you obviously want to be using the same kind of language server tools that you'd be using as a human uh not CLI based tools which often don't have the same her istics in place to be able to

05:46:02 optimize and then you know ideally you do this in a nice Loop you eventually get to the point where you can commit and get a fresh PR to do that migration to open

Telemetry this is what it looks like in theory in practice at some point it hits an error that it can't fix right it hits an error that gets into a loop and it's conly trying to fix the same error it uses five different techniques then goes back and your context Windows completely polluted with the wrong errors right everyone often says like

05:46:28 agents don't work uh this is probably half of the reason the agents don't work is that you just have compounding failures right we found anytime we actually have more than 10 prompts in a row uh our chance of having successful PR is dramatically lower right uh so the way we work around that is instead of trying to repeatedly fix an error uh we should actually just save our original state revert back to that uh and then continue to edit from there right so if we went down a path there was just a bad

05:46:56 path right we got stuck in a row we want to go back to a known good checkpoint uh and then build from there this is actually how we're able to do this quickly we don't want to spend 10 minutes recomputing each time uh we want to actually build our inmemory graph that we were talking about with TS server we want to save that we want to take a snapshot of memory uh so we use firecracker it's a VM manager uh that's used for a slim up but we can actually use it for Dev environments too uh and we can actually take the in memory State

05:47:20 snapshot that and then Fork it into 10 different isolated environments that all have everything pre-computed you can try 10 different changes in them and then figure out the correct change that is most likely to yield good results from there in fact this becomes massively parallel you can end up with an AI system that looks more like a distributed database than it does a traditional agent or something that you're running on your laptop right we actually have flows where we often have uh six up to 10 different agents working

05:47:47 in parallel all working from a known good state they're supposed to report back once they're done and then we'll actually look at the different evaluations we look at uh both some LM based evals but also heuristics like how many errors are there currently in the code base uh how many unit tests are currently passing uh and then actually compute like what of these which is the Quorum right it's actually similar to uh again a database system where you would have a voting of like what's the new Master uh here it's like what's the new

05:48:11 good state that we want to Fork from uh you there four here that have similar states that we want to use that as our new known good State uh save that as our known good State and then Fork from there going for it right and this ends up being much much for arrival because we have an entire PR that yes we've done 30 or 40 uh different Generations on it but in the final chain there was only four different Generations right because we had one then we got went back to know good state then the second one that's

05:48:38 all operating from that Quorum at each checkpoint but these edits get pretty expensive right if you're doing uh 40 different edits make a single PR across very large files uh that's a lot of money that you're spending on inference uh this is common problem with making good edits everyone naively just ask for Generate the whole file again right it's the simplest approach you definitely should start with that if you're building your own AI Tool uh but then you run into the classic problem of laziness so this is actually still from

05:49:04 uh Sonet it still said You know the start of the function Remains the Same right left this comment in because it didn't want to Output that code and it's just because output tokens are fundamentally more expensive if you look at gbd4 it's 5 to15 ratio of input tokens to Output tokens uh Cloud 3.5 Sonet is 3 to 15 uh this is pretty consistent across the board and then response limits are not growing at the same level of context size right we've got models out there that have 1.5 million tokens two million tokens in

05:49:30 their context window and still only outputting 4,000 tokens at a time right because it's autoaggressive it gets R more expensive so you really don't want to Output entire large files as you're making edits you want to find a good edit format uh so whole edit format works well uh it's very expensive though uh you can do diffs right you can say like generate a unified diff for this try to apply that um there's some problems with this one is like line numbers uh LM are still not very good at knowing what the right line number is

05:49:56 even if you give it them they just not that good at the math part uh and it's also off distribution right real world code that trained on is largely not trained on diffs right it's trained on actual full files uh you can do will search and replace with function calls uh the problem with this is function calls are underneath for the most part Json uh escaping code in Json format is terrible you end up using a lot of tokens just for uh just Escape characters right it's just not a very good format to use so that's way we

05:50:22 actually developed a gql loose search and replace so we could actually do something that's similar to what you would have on the model of being just a before snippet and this is something you might have like in a tutorial which is like replace this with that right this is what we and this is actually the exact same output that comes from the LM we'll do a m match we'll do a loose match to try to find what's the code that looks like that and replace it with the code that looks most similar to that

05:50:43 afterwards right and this works really really well because we don't have to you can alide irrelevant details like what's currently inside the make match function and just give enough detail to make the replacement cool uh and just want to leave you with where we're going next this is our current UI it's still is very traditional right it's still is uh building it you know what's a AI workflow looks kind of like your CI even though it's thousands of Agents executing I'm really excited about where

05:51:07 go next with this uh figure out like what does it look like to manage an entire qu base I think of like Sim City is like the ultimate where you can zoom in and out and understand different levels of granularity and edit things there cool uh thanks so much M and we are hiring so uh scan the QR code thank you so much morante um next we're going to bring to the stage G Patel he is a director of engineering at po Alto Network specifically focused on their generative AI efforts especially around autonomously identifying and

05:51:46 fixing security vulnerabilities which is what he's going to talk about here today he previously worked on open- Source container and Cloud networking as well as security and he is also a prankster um so also a man of many talents uh just like morgante and with that I will uh hand it over to gjn hi thanks Britney for the intro um my name is gjan Patel um I'm a director of engineering at Balter networks uh like many of you in the last year and a half or two years I've pivoted to working on generative AI um The Talk today is about

05:52:28 self- evolving code I'm not here to sell you anything none of this is a product from our company uh this is um a side project that I've been exploring before we get started with this talk um I would like to dedicate this talk to my close friend nikil who passed away two weeks ago um it was unexpected uh but um he was very excited for me to come and do this talk uh and he was very enthusiastic about this conference itself um so I want dedicate this talk to him okay let's start with the coding flow uh if you have read the book called

05:53:11 The Flow um it talks about this Zone where uh your skills on one axis you your skills and how challenging the problem is on the next uh next axis as you're starting to work on a project you should be somewhere within that green zone to stay in the Flow State in reality what happens is in the beginning you have an idea you start writing some code initially crank out some code then you have a little bit of a difficult problem so it cranks up the uh Direction a little bit to the anxiety may even go a little bit into

05:53:51 anxiety and then you have to write unit test and then it goes straight to the boredom part uh because it's uh your skill levels are quite higher but nobody likes writing unit tests uh or code documentation so what's the problem there as a developer you when you're writing code uh when you're developing software writing code is a small portion of that entire flow right uh you have to work with a lot of stuff like you know uh building kubernetes uh like manifest building Docker file Etc deploying stuff

05:54:34 writing unit test so then there is this co-pilot duct tape yes uh having something like a co-pilot that would help you get through a lot of technical hurdles on uh coding side whenever you see something challenging you can overcome that um but we are trying to Outsource more and more of that to do offline right so you have you as a developer have more time to write code so the key part in this talk is about Outsourcing a boring task uh uh to CI um I had to insert this Meme here um that like on both ends of the spectrum

05:55:17 people use chat GPD and the middle the mid uh such as myself is like juggling all these different tools trying to figure out how to write the code okay let's get into this uh so what's the difference between co-pilot and uh I just Callin this term ghost pilot uh co-pilots are made for quick just in time uh decisions right uh co-pilot examples are uh Source graph Codi like when just talked about earlier uh and GitHub co-pilot they work with developer lives developers are impatient you need response in seconds

05:55:56 that how good of a tool it is partly measured by how quick the respones are right um code review uh code Reviewer is made to have a deliberate thinking right when you write code and then uh code Reviewer is reviewing your code it's deliberate thinking it's not just guess the next word uh at this point uh you need to reflect on the answers that uh that you're providing or uh like human code review or uh being aware of the full context and uh to update the answer right so some of those things required the need time they

05:56:37 require iteration uh so one one example is uh this book called Thinking Fast and Slow uh so if you think about co-pilots as Thinking Fast uh the slow system is this uh ghost pilot system so this is the high level architecture uh I'm talking about here developer writes um code using uh a co-pilot or ID assistant uh tool they check in the code in the in CI the during the pull request process uh the first step is uh improving code VAR uh code comments and variable names second step is adding and running

05:57:26 running unit tests in a loop uh and I'll go into each of these steps in more detail uh adding unit test here early on uh sorry let me talk about the first F doing the first step properly means adding unit test makes more sense right like you need AI to understand what is the intent of this piece of code uh adding unit test early on sets a baseline behavior for the code that this is if this is the input this is the expected output so it's set it's setting the Baseline then third step is based on the

05:58:00 environmental context identify security issues security best practice that are not followed um and fourth step is fixing them and after fix proposing a fix not fixing them proposing a fix for human to review uh run the unit test again so let's go into each of these steps so first one why improve variable names uh and code comments uh yesterday there was a session here from Manu uh that he he wrote this line llms are cultural technology right llms are built based on all the knowledge in the world not just

05:58:41 coding knowledge like most of us here we know a lot more about coding than art or how humans behave uh psychology Etc but LMS have all that knowledge stuff into them we should figure out how to utilize that completely right so an example is if you're working at a finance company for example and use a variable name rev it makes sense to you like Revenue in that con small context of your code uh but improving it to annual revenue and then in the subsequent steps if uh llm is run uh using different um different

05:59:24 llm providers it will pull in its existing Knowledge from all the finance world so there may be some Corner cases that may not have been discovered if you just use the variable name rev versus annual revenue um the second part is uh improving good uh improving code comments right uh code comments can get out of date quickly if someone is in a hurry makes change to piece of code without improving the code comment uh then it's out of date right like it doesn't State the developer's intent so improving that um to make sure that llms

06:00:06 understand what is intended behavior of this code even though even if the code may not reflected and that's why we have bugs right this is the intent nobody intends to have a bug uh so making that intention of the code block clear that's why we improve the code comments so this is a small example of that um you can see uh in CI it's adding high level flow of what is expected from this code uh code block um here like it's a server so I just put port and it improve to port number because it could be Port name or

06:00:46 port number uh I don't think about those things but uh when it comes to adding unit tests AI Genera unit test it it becomes important second one is adding unit test so adding unit test as I said earlier it's to set the Baseline behavior um uh then once we have the Baseline Behavior we can add second layer of unit test uh for covering Corner cases third one is to get context from historical bugs uh what looking at JRA fetching for this project we have seen these kind of bugs in the in the past so adding additional

06:01:26 precautions to make sure those cases are covered so what are the steps like it's not asking an llm to say generate unit test for this code it doesn't do a very good job uh and this is where having iterations comes into play right so first one we do is um because we have all the time in the world people are not rushing for CIA jobs to complete I mean yes a lot of people are impatient uh but it's not like when you're writing code uh with an ID based assistant you need response quickly so what we do in this one is set

06:02:03 the Baseline so uh llm out puts the expected behavior of this code not the person who wrote it llm will read it and say okay this is the expected Behavior Uh it will take things into consideration like uh code comments read me file and maybe a PRD document it may not go into function level details but overall intent is there uh then in the next step uh we added adversarial mindset uh you can use the same model or a different model and say uh system prompt is the context looks like uh context is used in the uh step

06:02:43 three but things like what is the cloud provider what is a service where is it deployed is it a company internal tool or a customer facing tool is there piia data in it um is it a front end or backend uh application networking PRD where where is the gr project to pull all the bug reports and uh more information on this project slack channnel now this is important I haven't implemented this part yet but where it could go and ask followup questions to the user saying hey I'm not sure about this context it's

06:03:17 missing this context similar to uh if you use perplexity uh and use perplexity Pro you ask a question it's not clear we'll ask you a question back uh so that's what this is designed to do uh and your company security policies um guidelines from your infosec team if you have any then second part of the context is built uh using AI so AI INF first this context uh what language is this is this ISC code or is this backend code uh shell code Etc uh because you'll have different priorities based on that uh

06:03:55 what are the focus areas that comes from historical bugs uh SQL injection timeout errors concurrent users uh whatever issues and escalations youve had in the past um here uh Quinn has done a great job on open context. org uh that's a potential integration Point here uh as well for bringing in additional context finding security bugs now uh security issues so security best prati this is not just looking it's different from running static code analysis and uh different running SAS based tools it's

06:04:34 finding SEC with security best practi are not followed that your company may have policies around uh logical flow based issues uh I'll show you an example in a bit uh and prioritize which ones need fixing right not just identify because when you ask AI find issues it's going to find issues it's never going to say no it's fine I mean it's a small issue but it's going to say something is wrong so how do we do that this is this is a little bit of a weird side now right uh what I'm doing here is simulating three AI employees right um

06:05:16 one is a red team engineer one is a python developer assuming this is python code and an engineering manager uh engineering managers are useful for something uh other than uh talking in meetings uh so what context am I providing the red team engineers security policies and the code that's written or the code that's being reviewed python developer will assume the identity of the person who wrote this code uh and the PRD uh product requirements engineering manager will have business side of things uh

06:05:52 business context or uh Zoom meeting transcripts uh slack Channel conversations and PRD now there is a prompt to have these three people take turn multi-turn and then debate among each other on each of the security issues on identifying what is the risks associated with it and which ones need to be fixed and what is the effort for it and at the end after they're done with the debate uh self-reflect on the answer and say okay this is you know and that could be a separate model it could look at the answer and say okay this is

06:06:31 matching the original intent or not uh some fun things you could try um I was listening to a podcast recently and they talked about Odyssey Journey which is like you think about like three potential paths you could take uh and then time travel back and then talk to your past self now that's not really possible in real life but in here we can do that um with AI it's it's all fun um but it uh there is some research paper that show that this is actually more effective than just using llms uh as a standard tool

06:07:09 uh this is an example uh this was a kubernetes bug uh in the goord it was a critical severity bug uh and I made sure to find this issue from after the training end so GPD 3.5 uh training date end uh this was this wouldn't have been picked up in any of the SAS tools because it's logical thing this if statement needs to go from here to here and that caused the uh security vulnerability now that's something that's when I run it through this it's it finds that issue last one is fixing the security issues so this is outcome based uh and

06:07:50 it's again suggestion uh of what to fix and then human reviewer comes in at the end and uh decides which ones to actually fix for each of the fix explain the reasoning why it's prioritized uh that all the conversation that the the three virtual employees had have had uh it's summarized in three bullet points uh so human reviewer has the context and security uh policy uh citation and at the end make sure before a human comes in run the unit test that establishing the Baseline early on so human doesn't have to waste their time

06:08:29 uh with verifying the fix um this is not a great example but a simple example of what it would look like in a code comment uh hardcoded API key I mean most people wouldn't have that uh what is the risk score what is the effort score uh what is the recommendation which is the code snippet and then fix it after I feel you somehow don't let me go I need you right now I want to be next to you you want to be next to me holding our Paper Hearts fading now Broken Dreams I want to be next to you you want to be next to me

06:09:12 hold it on paper heart fing our Broken Dreams I want to be next to you [Music] you tell me that you want to stay baby just don't walk away I Need You Now fade it out all the time we spent AI fighting through the fire don't let me down I need you now I'm feeling worn out it's getting to me lost some heart trying to get on my feet caught in the madness I feel you somehow don't let me oh I need you right now I want to be next to you you want to be next to me holding our Paper Hearts fading our Broken Dreams I want to be

06:10:49 next to you you want to be next to me holding our Paper Hearts feing our Broken Dreams I want to be next to you [Music] want to be next to you you want to be next to me holding our heart fing our Broken Dreams I want to be next to you you want to be next to me hold our paper heart reading out Broken Dreams I want to be next you you [Music] [Music] [Music] [Music] [Music] [Music] [Music] [Music] [Music] a [Music] [Music] [Music] [Music] let

[Music] we got an insomniac with eyes wide shut we got everything we need and then a little too

06:15:45 much I know that you're starving for something you can't touch but you be Hest with me right now there's something in the under car and I can feel it cing up don't you want to feel it taking over your senses don't you a fe Technologic FES baby come escape with me I'll come sweep you off for your Fe don't you to feel it don't you don't you think there's something in my bag that's weighing me down oh it's just the weight of the world now I'm calling it out we're a little starving for some Lightning Love can we speak

06:16:36 honestly right now there's something in the undercurrent I can feel it coming up don't you want to feel it taking over your SES don't you ever feel it [Music] teolog baby come esape with me I'll come sweep you off your Fe don't you want to feel it don't don't [Music] [Music] I'm falling right in and I'm ready to go I found what I want and I know that we're on top so I'll tap and I'm ready to hold my breath and I'm ready to go I catch you laughing and I'm ready to go you're holding the it we are a Sumer

06:17:41 storm feeling you can ignore do you ever stop to feel it CAU in the after I'll come back to your door to know that you believe Sumer all that I want you know Sumer we got it all holding my breath and I'm ready to go I'm falling right to I what I want and I and I'm breath and I'm ready to go I catch laugh and I'm ready to go you're hold stri it and my soul [Music] we light it up again the sky and our [Music] Sil dancing on the pavement caugh in aect stone you those eyes again when I least expected said you're all that I

06:19:01 want we know together we got it [Music] hold my bre I'm right I'm go I found I want and I know so and I'm ready to my breath and I'm ready to go I catch you laugh and I'm ready to your I'm it my soul and I'm ready to if I find myself at your door would you follow me to better if I find myself at your the keys let's go I want to taste if up my I would follow you to better places if you show up let's go let's go oh holding my breath and I'm to go I'm falling right in and I'm ready to go I found what I

06:19:57 want and I know we're on top so I and I'm ready to hold in my breath and I'm ready to go I catch you then I'm ready to go [Music] hold I'm ready to [Music] hold and I'm ready to my breath to iatch I'm to I'm ready to [Music] I'm ready to [Music] oh [Music] [Applause] go [Music] down down [Music] [Music] a down [Music] night days you and me we were the only one we were holding nothing back from the greatest nights we ever [Music] hadat the driving slow in your car singing Al every night must to play that song 100 times

06:24:36 made of fire we were Li and the summer bre dancing the rec in on mind you the and TI always on my I feel it all come back in the moment spin away like the ocean so if you want to come with do it all play it all in slow motion [Music] feel like a melting up the night when I'm alone when I hear this words you made [Music] driving SL in your heart singing every night must to play that song 100 times made fire and the summer bre dancing Rec like on mind the music the mind child always on my mind I feel it all come back in the

06:26:42 moment SP the oce so if you want to come with it all in slow motions MO [Music] feel it all come back in the moment I can SP away like the oce so if you want to come it open it all back [Music] [Music] [Applause] [Music] know [Music] [Applause] [Music]

[Applause] [Music] w [Music] [Music] [Applause] you know you you know you [Music] [Applause] [Music] [Applause] [Music] [Applause] [Music] [Applause] [Music] feel me [Music] heav Echo secrets that we know door set open for us in a moment keeping light on ring

06:30:29 roll keep everything we want we catch our breath in the middle of it all Chasing Echoes sun is cut up over the rest is coming [Music] Crystal like a True Believer we walk on the w i can see on the horizon all thees we can [Music] feel for the trees I'm keeping watch all the storming waking up and turning on keeping light ridings keeping our sights on everything we want we catch our breath in the midle of it all chasing ech sun is coming up over the rest is coming [Music] Crystal like a you true

06:32:06 looking on the I can see on the horizon all we can feel it Chas [Music] the chasing the light all we know can't stop and I won't let it go can't stop I Won't Let It Go like a fever up we're on true belever we looking on the I can see it on the horizon all our we can feel it chasing [Music] we True Believer weing the I can see on the horizon all we can feel [Music] [Applause] [Music] oh a [Music] w [Music] oh [Music] oh [Music] [Music] [Music] [Music] [Music] [Music] [Music] [Music] [Music] [Music]

06:39:43 [Music] [Music] me [Music] you you you you you you you would you would you would you you you you [Music] you you you you you you [Music] me [Music] you you you you you you what you what you you you you you you you you you you you you you you you you [Music] [Music] [Music] [Music] [Music] [Music] [Music] [Music] [Music] [Music] one more breath beside you so I could find strength to divide us It All We Got and I know we did the best we could if I could go back UND the mess I would memorize your face before I

06:46:35 go but this is how we grow got to give it up sometimes as go KN when to kill your pride no theame nothing really stays the same this is how we [Music] gr hold on to let go [Music] hold there is lost us and oh I know you have your reasons some days I'm a mess but I know there's a rainbow over all the past your head on my shoulder but I know better on but this is how we go got to give it up sometimes is G KN when to kill you right there's no to blame nothing really stays the same this is how we grow hold let

06:48:16 [Music] go let sometimes we we hold [Music] how got to give it up know to kill your pride there's no to blame nothing really stays the same this is how we grow this is how we sometimes we hold on let go [Music] [Music] B [Music] [Music] [Music] B [Music] [Music] [Music] d [Music] I was watching you watch the sun come up vage t-shirt through highs these nights tast like sweetsession show me as comes we were out the night like we we our clothes dancing right through the fire while we watch it singing on r

06:53:03 we give up our go as a new morning comes through the windows we riding all new Burning through the page tearing past all the you're wearing man we our problems underneath cles like super like superheroes it's coming over now title we crash down a Harmony of that only we can hear a super CR you want to feel like us it's forever America under your influence full moon Waxing now I couldn't see until you show me how feels like we're insane we blame it all on love saturated so we can't get we were out the night like we wear

06:54:14 our clothes dancing right through the fire while we sing an we up ours as a evening comes through the windows it's coming over it's w down a Harmony that only we can super CR you want to feel like us it's all forever America coming over me electric Sy every night on fire I KN on Master super CR you want to feel like it's forever you're in [Music] America holdon going black so come with us don't hold back tonight is all we have the sky is going so come with [Music] us don't hold tonight is all we have the

06:55:51 is going so come with us don't hold back tonight is all we have the sky is going so us it's coming over now it's down a Harmony that only we can super CR you want to feel like us it's forever you're in America it's coming over me electric Sy every night on fire Aon Masterpiece super Crush you want to feel like us it's forever in [Music] America don't hold tonight we going so come with hold back [Music] T come us hold [Music] back it wasmer back in we were kids falling in love for the first time H your hand you look me in

06:57:41 the eyes kind of feeling you get Once in a li but now something went wrong you're moving on I found myself on The Blind Side now you won't call we lost it all you fade away I'm picking up my heart from every piece that's broken been trying to get back to myself but don't have a clue I'm looking for some luck can't find the door that's open I'm losing all my feel like I'm left here because I'm missing you because I'm missing you oh because I'm missing you because I'm missing you because I'm missing

06:58:58 you because I'm missing you I was chasing all the wrong sides trying to hold on to something that I couldn't find which you didn't Captivate my mind now I know we've in the sunsets in Paradise but now something went wrong you're moving on I found myself on The Blind Side now you won't call we lost it all you fade away I'm picking up my heart from every piece that's broken been trying to get back to myself but don't have a clue I'm looking for some luck can't find a door it's open I'm losing All My Hope feels like I'm left

06:59:55 here two because I'm missing you because I'm missing you oh because I'm missing you missing you because I'm missing you because I'm missing you up my heart every piece that's broken been trying to get back to myself but don't have a clue looking forck can't find it's open I'm losing on my like I'm [Music] [Music] tell me that you to stay baby just don't walk away I Need You Now f it out all the time we SP alone fting through the don't me down I need you now I'm feeling out it's getting to me lost some heart

07:01:30 trying to get on my feet [Music] caught in the madness I feel you somehow don't let me go I need you right now I want to be next to you you want to be next to me holding our Paper Hearts fading our Broken Dreams I want to be next to you you want to be next to me holding our Paper Hearts feing our Broken Dreams want to be next to you you [Music] you you you you you you [Music] tell me that you want to stay baby just don't walk away I Need You Now fade it out all the time we spent alone fighting through the fire don't let me down I

07:03:03 need you now cuz I'm feeling worn out it's getting to me lost some heart trying to get on my feet caught in the madness I feel you somehow don't let me go I need you right now I want to be next to you you want to be next to me holding our Paper Hearts fading our Broken Dreams want to be next to you you want to be next to me holding out paper hearts feing out Broken Dreams I want to be next [Music] to I want to be next to you you want to be next to me holding our paper heart fing our Broken Dreams want to be next

07:04:09 to you you want to be next to me holding our paper heart feeding out Broken Dreams I want to [Music] [Music] [Music] [Music] [Music] [Music] oh [Music] [Music] [Music] [Music] f it out all the time we spent AI fighting through the fire don't let me down I need you now I'm feel out [Music] [Music] [Music] [Music] we got an ins with eyes wide shut we got everything we need and then a little want to be next to you want to be to meing broken [Music] [Music] ladies and gentlemen the keynote presentations are starting now please

07:09:16 take your seats if there are empty seats closer to the middle of your row please move inwards to open aisle seats for others thank you ladies and Gentlemen please welcome back to the stage your host Benjamin duny [Music] how do I look Booth Bingo who's been playing let's see the show hands who's been playing both Booth Bingo Booth Bingo Booth Bingo we got a few people so the folks at open pipe they sponsored this lovely bag and I came up with this great idea that I'm going to use like now and forever we give pins limited number to

07:10:25 all the sponsors and you fill out the Bingo board you get a prize scan the QR code anyone want to fill that up with some good stuff it's right there do we have a good time all right so many speakers so many sessions only one of each of you for now let's get those AI clones going soon and I don't know some Matrix downloads slow down a little bit um great so I don't want to take up too much of your time we're going to get to the closing Keynotes yeah we all ready we're all are we all that exhausted are we

07:11:16 ready please join me in welcoming to the stage the CEO of Bot Dojo Paul Henry [Applause] [Music] ah I got logged in cool so hello my name is Paul Henry I'm the founder of Bob dojo and as a previous CTO I was working with teams deploying llms applications for hundreds of thousands of customers and like many of you guys know it's super easy to hook up a vector database um with an llm over the weekend but really hard to get it production ready and so that's what we're do we are an AI enablement company and we let companies

07:12:00 deploy AI to prod live demo time all right so today I'm going to show you a demo of a product we're going to take uh synthetic data that we're going to generate and we're going to combine it with evaluations to see how we can improve the performance of a chatbot or at least that's what I hope happens all right so I'm going to open up our template of our uh a chatbot and we have customers live that are using this template it's kind of battle tested um so let's test it out how do I create a vector index in

07:12:42 bot Dojo okay and as you can see all the little nodes are lien up as they execute um we're taking the question we're looking at the chat history we're going to the vector database to retrieve the information and then we're answering it with a AI model so if I pulled this up you can kind of see in our Loc code uh editor this is the prompt that we're sending to the llm we're getting the results out here and we also support uh J uh Json schema so if the model supports Json output like um grock um Claude and all that stuff then we just

07:13:18 conform to that um one key thing is you can pull a trace of each node and see exactly what we sent to the llm what came from the retriever the exact you know data which has been super useful for debugging apps all right and cool we have an image it's got citations we should ship it that was supposed to be a joke but all right um so this is where evaluations come in so I'm going to demonstrate um the evaluations that I previously Ran So we have a a feature in bot Dojo uh called batches which allow

07:13:54 you to run a whole bunch of questions through your chatbot or your AI flow and um run evaluations to kind of see how things are doing so if you can see this we have a few uh five evaluations that we ran there's a little bit of red um that's because uh we don't have enough information from our Vector database um it also checks for things like hallucinations so let's try to fix that and so I'm going to clone this batch I'm going to rename it with generated data I'm going to increase the throughput a

07:14:26 little bit because of time and um I had I don't have enough time to generate all the data for this demo so um the previous ran was filtering out the generated data and so I'm going to remove the filter that we're passing into the uh flow so it takes in the generated data you can also change the model and all that kind of stuff to see how it performs all right so while that guy is running I'm going to open up another flow and so this is the actual flow that we uh generated that uh synthetic data

07:14:59 and so let me uh let me run this one real quick and so this particular flow takes in multiple inputs and so I'm going to paste in uh some Jal from a previous run and what this is going to do is it's kind of a trick that's been working well for customers is where you take um you extract questions and answers from support tickets so these are live agents talking with customers and you use this as a test data to send it through your chatbot and um we take relevant information from the existing index and

07:15:33 we have it write a document um and so it it uses the same writing style and it um you know and then we do an inline CIT uh evaluation to where we check to see if the document has enough information to answer the question and then we also have a code node here where you know a lot of times when you're using these low code editors there's like situations where you have 40,000 different um boxes and so when you have to do write code we support um tiripon python but you can see that hey we're getting the information and we're right

07:16:05 into the vector index all right running out time okay let me go back to the support chatbot a moment of truth so I'm going to compare um the the batch that we ran before with the new stuff and 20 seconds oh you do it you do it 15 times and it doesn't work 10 nine we're also hiring so if you're an AI engineer help help us fix this all right there he comes okay all right one second left it's all green so it improved the uh you know measely improved something so uh thank you um bot dojo.com check us

07:16:48 out thanks [Music] Lally ladies and Gentlemen please welcome co-founder and CEO of emergence AI SAA Nita good afternoon everybody I'm here to tell you a little bit about what we're doing at emergence in the field of AI agents but first I want to tell tell you guys a few things about Who We Are so we are an R&D Le AI company advancing the science and development of agents and uh we come from some of the world's uh top AI Labs uh founding team out of IBM research uh and then we attracted uh Talent from

07:17:44 places like Google brain Alexa the of AI meta Microsoft Etc but even more importantly we have built and deployed some of the most scaled AI deployments on the planet from the IBM Watson platform to Alexa to the backend recommendation engines behind Amazon Prime Amazon video Twitter even bright Etc so we really think of ourselves as a distributed systems meets AI R&D team uh and our goal is to work on AI agents and to enable all of you to build agents to transform the world so we're all here today because we are excited about

07:18:22 what's actually coming out in AI so the long promise of AI over this last several decades has always been that AI will perform actions for us this what science fiction authors have told us since the 1940s and 50s and I think the time's finally here and in particular what we're very excited about is AI that will operate things like uh you know web browsers and uh various other Enterprise systems and software and in the process uh Drive Great productivity benefits for everybody both in consumer and

07:18:52 especially in Enterprise which is our Focus so we're interested in Enterprise because the most interesting workflows are actually in Enterprise and this will really stress and push the limits of what AI can do in particular what autonomous AI can do and uh so what we're building to enable that future to come to pass uh with the with the help of all of you uh are two very infrastructural platforms the first of which will be G in August uh Early Access is already uh live right now so you can go sign up for it this is

07:19:25 uh called an orchestrator agent so first of all it's an agent and uh in the sense that it acts it plans it basically also verifies so it it finishes that agentic Lo between planning acting and verifying it remembers and improves our time but what does it do it actually allows you to orchestrate across multiple agents and Stitch them together in complex workflows uh a simple version of it is you could simply orchestrate across multiple llms journalist and open source llms but more complex versions of it

07:19:56 will basically enable you to solve very complex Enterprise workflows like claims processing Etc now an orchestrator is only as good as the agents that will orchestrate too so one of the exciting things that we're doing here is integrating the orchestrator with a project that is currently being developed in the open source Called Agent e um agenty is a project that uh uh my my colleague Tamer had a session on earlier this afternoon it's a web agent it's basically meant to control the web like a human would and currently

07:20:30 it's the best web agent on the planet it's stopping the web Voyager Benchmark and uh it's designed to basically uh be used to build multiple Enterprise workflows and work seamlessly with agent with the with the orchestrator agent so I'm going to play a short video that will show you a little bit more about both these agents and then I'll WRA wrap the talk up okay we don't have sound so maybe I'll talk over it so our first product is the orchest and task specific llm and agent all using appropriate guard rails

07:21:04 the developer dashboard helps analyze prompts optimize cost and latency create new models with your data and enhance existing ones build with confidence Knowing

Your solution is future proof and can migrate to the latest llms on demand the orchestrator is an intelligent agent that improves with use connecting to agents like agent e our web automation agent in development in our R&D Labs agent e is an open- Source agent designed to learn how to autonomously operate the web and automate complex workflows for example get us a

07:21:41 reservation for 15 people near the office around 6:00 p.m. we like Asian Mediterranean and Mexican send the details to Levi please at emergence our mission is to advance the science of AI agents by tackling core AI problems like planning and self-improvement to enable the full transformation of AI and benefits the world okay so just in conclusion uh in our R&D Labs our area focus is around self-improvement agents this is our core focus and in the process we will advance things like AI planning and reasoning

07:22:14 and also solve things like uh you know how agents should be stitched together in really interesting ways through something called Agent oriented programming and we're doing all of this in the context of Enterprise workflows like RPA and document processing access Etc so that's my time thank you so much see us at our boo please welcome to the stage member of technical staff mid Journey changlu [Music] hello um this is a small visualization of our Lord and savior matrix multiplication I was asked to make a

07:23:13 cool demo so here it is um this is a single fragment Shader drawn fully on the GPU there's no imported asset no triangle meshes just purely a few hundred lines of GSL um Shader R is a niche digital R form and I highly recommend you to check it out um but the GPU was wasn't supposed to be abused this way but then again the entire domain of machine learning is enjoying a Renaissance thanks to it so how did that happen um today I would like to explore a little bit of these kind of second order effect

07:23:43 and why things happen with unintended consequences and how you can more reliably predict the future okay so deep blue be gas Gasper robing chess 99 96 and 2015 Alpha go beats the famous Lisa Doling go if you were a pessimist as many were back then you'd say that this is over for chess and go you know what's even the point of playing those anymore right if human is not at the top but if you check what actually happen subsequently this is a graph of professional goal players uh decision quality over time you know

07:24:22 guess where Alpha goal happened and on a completely unrelated topic this is Conway's Game of Life it has black and white cells and a few simple rules to how the cell actually interact and at first glance it's no big deal but as you start building macro patterns with those you get cool things like these and here's Conway's Game of Life implemented in Conway's Game of Life to incomplete uh these are examples of emerging behaviors produced by an order of magnitude uh quantity and quality and or performance increase the domain of

07:24:55 machine learning is pretty familiar with this phenomenon and generally speaking emerging behavor behavior is mostly MP complete so you can compute it easily so to create these patterns people have to zoom out a level um at the and consider high level macrodynamics a new set of rules plus various heuristics and errors uh those folks work more like biologists than mathematicians or physicists and what I'm trying to say is we cannot easily predict the emerging behavior of even a simple system that scale beyond

07:25:26 our low-level intuitions so in this talk I would like to provide a few personal thought processes I used to predict some interesting second order effect of AI aka the ripple effect caused by more direct consequences of this era of AI and as a famous sci-fi author once said good Sci-Fi predicts cars great sci-fi predicts traffic right so the first I would like to use is uh broadly called who is learning are you learning or is the machine learning and do you care uh I'm mostly talking about what people

07:25:56 want so for example chess didn't die it only got better right because it turns out that the crowd Dynamic of it is that when you have free chess teachers anytime you want instead of having to seek out that one dude in the village that teaches chess well everyone ends up knowing chess and when that crowd knowledge becomes distributed widely enough you get to have an audience to sustain more professional playing because ultimately it is you who want to do the learning and the audience this is disregarding whether the machine learns

07:26:24 better than you or not you go the to the equivalent of mental gym because no matter how much the machine goes to the gym you won't get better unless you do the same is true for drawing imagine a novice learning to draw a blank canvas is actually a very daunting challenge right but very soon you'll you'll able to have this equivalent of what we call a stroke autoc completions so imagine a conceptual slider like this one where on one side nothing happens right on the other side the full drawing is made for

07:26:54 you what's interesting is that now we can have a learned behavior where we can slide that into the middle so what you prompt the system and that that you want to draw a chair the system goes oh okay you're curving this way so I guess you want a Victorian era chair right or when you use a shade of blue the system goes oh I guess you want to draw the reflection of the sea on her face but there's sand actually so it should a little be a be a little bit more green than this so if you try to learn coloring you know how long of a feedback

07:27:22 loop this actually is to master this and now you can dial that up and down per your need uh with immediate feedback and ironically over time your slides actually go way more to the left all the way to the end where you basically stop using AI because you've internalized everything and the skill came back to you similarly for music yes we can now generate full songs in one shot for utilitarian ends but AI could also help in a different way for you to learn and what I'm interested in is uh in music

07:27:53 most of the time you're using direct manipulation in UI speak uh of the instrument right uh the impotence mismatch between pressing a piano key and hearing the expected sound is almost none but that lack of indirection is also a tradeoff so this is a therin and already we're seeing a little bit more of an indirect manipulation so I was wondering what if you use your fingers to create and manipulate a music spectrogram right obviously your fingers aren't fine enough but if the AI has enough World model knowledge to Super

07:28:21 sample it for you so to speak maybe we'll end up with a new form of instrument and when you gesture more like a puppeteer indirectly and create new kinds of music that analog music manipulation couldn't achieve so um here's another reason I'm giving these examples uh my domain is a user interface mostly nowadays uh if you think about who's learning you might end up with a conclusion that direct manipulation of user interfaces is actually about learning for yourself akin to going to the gym for yourself or

07:28:51 learning to draw for yourself learning abstract instrument for yourself so AKA once the classic user interface um um of tapping this and tapping that gets increasing automated away once the utilitarian ends have been met all that's left is the kind of Lifestyle user interfaces where you use them not because you're more efficient than the machine but because you are the one who's trying to learn them for whichever self-fulfillment reason um and so in that regard we might end up with more artisanal quirky Niche interfaces for

07:29:20 luxury lifestyle or other purposes as a second order effect and the second category I want to talk about is the um is um the idea of widening the information bandwidth which is a trick I use quite often so the other day I was looking at some new research result from enthropy regarding sparse Auto encoders uh but tangentially there was this simple visualization of a cluster and just purely from a visual perspective it kind of reminded me of uh the movie arrival by Den where human Humanity learns an alien language that allows

07:29:51 them to unlock their full potential and I thought why not take it further right and make it one language per person right widen the whole information bandwidth so up until this point human language is this somewhat standardized communication interface and it's a very very narrow bandwidth one and very lossy one we learn relatively few languages mostly standardized and stuff our entire uh fuzzy ether of information into hoping that most of it isn't Lost in Translation now that AI basically solved

07:30:23 translation so why not go a step ahead and translate one English to another English say I'm arguing with someone and I say I feel blue and this is coming from my perspective really so it's unclear that this intent Ares to the other person intact maybe I for that particular listener I should have translated I feel blue into I'm feeling purple right and what if I can just show it what if my C chat speech bubble is much more Dynamic and much more nuanced because the AI understands the other person's aesthetic preferences right

07:30:53 what if things are fast enough that every sentence can be personalized into a dynamic R piece in 4D or something way more information dense and I can just hand it to the person in AR right much denser than static emojis and a few Bas curves right and what if you're communting AR and these gets machine translated into like some kind of cloud around you for the receiver ajusting time uh individual specific language translation mechanism free of the compromise of a one-size fit out uh low bandwidth text language right maybe

07:31:27 verbal conflict resolution end up taking in the order of seconds instead of minutes or hours in 5050 years so uh some more examples uh here's one for the iPhone the the uh the hardware user interface and the next one for the iPad uh for canvas apps the act of pressing the pencil against the tablet usually means drawing a line but it is overloaded to be selection moving resizing Etc right but the reality uh is of a much higher bandwidth so for example if someone multi-tabs on the screen with a pencil maybe right before

07:32:03 they said uh why is this part red right uh can we can we change it or maybe they drew a stroke they said yeah this probably goes there instead before that right and didn't feel like you know uh hunting for the lasso tool selecting it coming back and and drawing a circle long tap to hold the object move it double tap pencil back to the previous pen tool and you do all these kind of acrobatic because you want to move an item right so if you want to use a traditional design to categorize and overload the single stroke gesture then

07:32:33 you'll inevitably end up with more confusing behaviors with an implicit rule set so traditionally if your current stroke is conditioned on the current selected tool State the object under your pencil and maybe the action one second before uh if we need to undo the stroke in F uh in favor of interpreting as a tab but this is very messy right it's a it's very fine design and craftsmanship but this in this new New Era uh that line shouldn't be only conditioned on the beginning of its basic path right it should be

07:33:06 conditioned on the entire world so the tap and the stroke Behavior should be as learned as in machine learn as possible and some people's short press you know you know sometime they're just slightly too long and Trigger the wrong gesture and all sort of bad action that a human Observer would have corrected within a second so why can't machine learning just do it right locally too um so um the last uh thought process I like to use often which is extrapolating uh a certain quantity or quality to the

07:33:38 extreme uh which causes all all sorts of fun emerging Behavior you can try to guess like previously mentioned Conway Game of Life for example and and then I can reason from first principle there and see what kind of uh uh new things we can get from this um so uh if anyone's doing into programming languages this is small talk uh programming language environment uh from the' 70s that's the grandfather of the original object oriented uh programming which Inspire Objective C and other languages uh one

07:34:09 of its main characteristic is message passing as in sending commands maybe even to another remote Small Talk object on a different computer somewhere else through land or later on internet um I'm going to spare you of the detail but Alan K one of its in uh inventors said the the inspiration is um basically uh well it's inspired by cells right and that each object is basically a full computer you can examine and poke into and and you can you can like do things with it and and recursively it's a it

07:34:41 might be onetoone mapping to a computer it might be be that one computer has many object Etc and also he did say uh somewhat more obscurely that sending the message sending the command to another computer that's easy but finding the receiver that's hard so um each Small Talk object can theoretically smartly go to the Internet do stuff come back with an answer like a little self-directed intelligent agent um if this sounds familiar to this audience uh but Small Talk had a big problem which is that uh when an agent

07:35:18 is as smart as it can be uh it's also um as resource intensive arbitrarily as it can be so when each agent takes up enough resource uh you only get to have a single or double digit of them right by the law of numbers so you missed out on an entire category of emerging Behavior because you try to be too smart at the lower level uh in this case emerging behavior from quantity and collaboration so uh on the other hand look at this uh multi-layer perceptual uh that's a graph so interestingly it kind of solved the discovery of the

07:35:54 receiver problem because you can make it fully connected or whatever uh and because the weights are learned you'll propagate and you know some connections are more important than others right uh the biggest difference between this and a swarm of Agents is that uh the node are as dumb as they get and when they're dumb and simple you get have millions of them and when that happens you get leverage emerging behavior of the Aggregate and create a new media completely so there are quite a few um Asian Focus talk in the domain of ML and

07:36:25 I like to take this opportunity to um uh use this method to to offer some interesting food for thought so for example the more agents you have the more you zoom out to care more about the aggregate rather than lower level agents right just like people and civilization and the more you zoom out the less you actually care about each individual agent so in in an alternative reality not in this one uh we invented a couple of agents who got sent to scour the internet called Wikipedia and came back with some Snippets of information

07:37:01 however thankfully in our reality we send billions of Dumber notes to read Wikipedia and aggregated all of them together to sit in a phone coordinated by a single top level smart top level process so here's my last example um this was actually freshly picked uh this is the Apple TV Mac OS UI pretty decent looking um I'd like you to pay attention to this part uh the circle red part which is a more button here when you click on it to see the full description what do you expect well uh where does the description expand to right turns

07:37:37 out that you get a very atypical Apple UI when you click on it you get this which is literally a big UI text view right uh very onapple so it looks like a unfinished notepad in fact you can kind of Select it and do things with it which is uh weird um the thing is this Apple TV Mac app is actually a catalyst app which uh for those who don't do iOS development it it means it's a direct Port of their IOS app here um on iOS if you tap the description and get uh uh such a new view then things don't look

07:38:10 too out of place in fact it's rather idiomatic now you might say that the problem here is that that uh lack of UI uh uh design and lack of care lacks of craftsmanship but for the sake of making a point uh for this talk I would like to provide a perspective that this might actually be a lack of literally needing more UI a lack of more UI so what would the world look like if we extrapolated that qu uh quantity if we raise the order of magnitude and have way more UI two order of magnitude more right what

07:38:42 does that even mean so let's start with a simple 12 column grid right we first list out all the discrete pieces of information we want to potentially show across this entire view or maybe across the entire app right now that we have ai nowadays a design time not a runtime a design time we could generate thousand of these permutation of layout for shows UI screens right we're not shipping these just using U AI to generate a bunch of potential candidates so previously this task wasn't achievable through

07:39:12 traditional means unless you're in a particular Niche um since we didn't have a way to pay attention to the semantic relationship between say a Show's title and the Box's uh size and position in relation to other items right you could still generate plain boxes through traditional heuristic and generative algorithms but you'd have a hard time tagging each box with a right piece of information for example so after our first pass we can use a scoring mechanism either traditional or um uh or some fancy AI

07:39:42 driven scoring heuristic aesthetic things to uh eliminate undesirable layout at data generation time and of course You' involve the designer here too um this is done at design time offline not a runtime so we can use an algorithm that's as slow as we need and the designer can take as much time or as he or she needs and patiently curate the subset which is uh quite large um so the key here is that you've generated not 10 right you generated thousand of these through through smarter generative and semantic filtering techniques so

07:40:20 we're raising the order of magnitude right you're not you're not a designer making a single- digit number of design uh moving boxes yourself in figma and waiting for your boss to go uh can we move this box somewhere else instead and uh and just one more adog design please I promise it's just one more you'll solve everything right um and of course you want to involve the designer in this particular stage too um um the yeah so um maybe at some point you just uh you also decide to like throwing a little

07:40:48 bit of diffusion again of uh rough draft time to using some control n or whatever to generally some rough website to make the boss uh give boss more of an immersive feeling right to say like this this layout can work it's not just like boxes right and at app runtime now um which is the power I'm personally interested in um right now uh llm Generations they generate at writing time and then you they gener like two right or three and then you pick one and then maybe you ship that and then it's a traditional

07:41:21 web app but the thing is if your bottleneck is the web part even in the AGI cannot help you make your JavaScript faster than C+ plus plus right so we have to swap out some of these items with an actual neuronet if we have to if we want to advance and use a web platform for example or any other platform for that matter so at runtime you have for example a quick decision tree to choose the right layout so again for modern web development this roughly has this one single heuristic for you to select the right design which is called

07:41:55 media queries which all it does is depending on the width of your window you might show or hide some items but this entire space could actually use some help from learn algorithms so for example what if the user onboarding why is that a different concept when you high show uh or show some different boxes it's just another set of boxes right what if the user is a super user maybe you progressively show them a different set of layouts right maybe they require different screens and uh what if the user is in a different

07:42:29 country right uh different age search query so to be clear um big companies like uber and Facebook already do this on a daily basis right when you you use Uber the app in India or China uh it looks drastically different right but it's currently uh thousands of Engineers of effort right for big companies right and they create an entire mold out of the fact that they have a few more design uis plus a business logic to be fair and uh and it's very rdle right you cannot see everything and the algorithm

07:43:02 is as is basically less controllable than even a simple decision tree a classifier so um if a user is fuzzy searching right this might be a better example um you know hey what movie did did uh Den make right this goes into decision tree and shows the accurated layout um if the user is instead saying hey what movie did Den V make and with home whom then you show this curated layout instead so maybe you're asking a chatbot in which case the layout is even more contextual right if you do a napkin calculation of the generated and curated

07:43:37 uh uh number of UI you ever need uh they might actually be in the Thousand not tens right fortunately thousand can still be curated thanks to AI so essentially it's not an auto regressive problem it's not a diffusion problem it's a simple classification problem because we have discrete categories here so here we go Dynamic uis so let me sumarize this for a little bit uh second order effect are pretty uh unpredictable and there are many ways to U tame thinking about them if you think about these points among others then I

07:44:09 think you'll be decently prepared when the time comes and of course you know read from history and uh you know you got to do things uh don't forget that the best way to predict the future is to invent it thank you [Music] ladies and Gentlemen please welcome VP product spreadsheets are all you need edio isan Anand [Music] hello hope you're all having a good conference and I hope you're ready because if you came to this conference or the AI engineering field without a machine learning degree then this is

07:45:17 going to be your crash course in how machine learning models actually work under the hood let's let's bring up uh the slides there we go thank you okay so I'm isan and I'm dressed in Scrubs because today we're all going to be AI brain surgeons and our patient will be none other than gpt2 an early precursor to chat GPT and our operating table will be a table but it will be a table of numbers it will be an Excel spreadsheet this Excel spreadsheet implements all of GPT T2 small entirely in pure Excel functions no API calls no

07:46:06 python in theory you can understand gpt2 just by going tab by tab function by function through this spreadsheet but you want to hold on to those vlookups because there's over 150 tabs and over 124 million cells for every single one of the parameters in gpt2 small I will give you the abbreviated tour so we'll do three three things today in our little med school first we'll study the anatomy of our patient how he's put together then we're going to put him through a virtual MRI to see how he thinks and then finally we're

07:46:41 going to change his thinking with a little AI brain surgery okay let's start with Anatomy you're probably familiar with the concept that large language models are trained to complete sentences to fill in the blank of phrases like this one Mike is quick he moves and as a human you might reasonably guess quickly but how do we get a computer to do that well here's a fill-in the blank that computers are very good at 2 + 2 4 right they're really good at math in fact you can make it very complex and they do it

07:47:11 very well so what we're going to do in essence is we're going to take a word problem and turn it into a math problem in order to do that we take our whole sentence or our phrases and we break them into subword units called tokens and then we map each of those tokens onto numbers called embeddings I've shown Don it for Simplicity here is a single number but the embedding for each token is many many many numbers as we'll see in a bit and then instead of the simple arithmetic shown here we're doing

07:47:39 the much more complex math of multi-headed attention and the multi-layer perceptron multi-layer perceptron just another name for a neural network and then finally instead of getting one precise exact answer like you used to get in elementary school we're going to interpret the result as a probability distribution as to what the next token should be so here's setup we get input text we turn that text into tokens we turn those tokens into numbers we do some number crunching and then we reverse the process we turn

07:48:10 the numbers back out into tokens or text and then we get our next token prediction so this handy chart shows where each of those actions maps to one or more tabs inside our friendly patient spreadsheet let's take a look so the first thing you do is we get our prompt right here the prompt is Mike is quick he moves and then it will output after about 30 seconds since we're running in a spreadsheet don't use this in production the next predicted token of quickly so the first step is to split this into tokens now you see that every

07:48:42 word here goes into a single token but that's not always the case in fact it's not uncommon to be two or more tokens let me give you some examples so here's another version of the sheet let me Zoom this up so you can see it a little better right I've put actually some fake words reinjury is a real word but funology isn't a real word uh but you know what it means right because it's the word fun withy put together those are the morphemes as linguists like to call them and the tokenization algorithm

07:49:09 actually is able to recognize that in some cases whoa there we go right there you see fun split into a fun anology if we Zoom that one up there we go but it doesn't always work so notice how reinjury got split up right here it's rain and jury and that's cuz the algorithm is a little dumb it just picks the most common subword units it finds in its iterations and it doesn't always map to your native intuition and so in practice machine learning experts feel like it's a necessary evil um and then the next step

07:49:45 is we have to map each of these tokens to the embeddings so let's go back to the original one and that's in this tab here so we have each of our tokens in a separate row and then right here starting in column three is where our embeddings begin so this is row right here the second row is all the embeddings for Mike now in the case of gpt2 small the embeddings are 768 numbers so we're starting column 3 so that means if we go to column 770 we will see the last end of this and so there's the end of our embeddings for

07:50:16 Mike and let's go back and each one of these again is the embedding for uh each token okay then we get to the layers this is the heart of the number crunching so so there are two key components there's attention and then the neural network or multi-layer perceptron and in the attention phase basically the tokens look around at the other tokens next to them to figure out the context in which they sit so the token he might look at the word Mike to look at the antecedent for its pronoun or moves might look at the word quick

07:50:49 because quick actually has multiple meanings quick can mean movement in physical space it can mean smart as in quick of wit it can mean a body part like the quick of your finger nail and in Shakespearean English it can mean alive or dead like the quick or the dead and seeing that the word moves here helps it disambiguate for the next layer the perceptron that oh we're talking about moving in physical space so maybe it's quickly or maybe it's fast or maybe it's around but it's certainly not something about your fingernail so let's

07:51:18 see where this is all happening so these are our layers now there's 12 of them so this is block zero all the way to block 11 each one's a tab and then if you go up here we can't go through all of this in the time we have but this is one of the attention Heads This is Step seven this is where you can see where each token is paying attention to every other token and you'll notice that there's a bunch of zeros up at the top right and that's because no token is allowed to look forward they can only look

07:51:43 backwards in time and you'll see here that Mike is looking at Mike 100% of the time higher values mean more attention these are all normalized to one uh here is the word he or the token he I should say and you'll notice 0.48 so about half of its attention is focused on its the antecedent of its pronoun now this is just one of many heads if I scroll to the right you'll see a lot more uh there aren't always as directly interpretable as that uh but it gives you a sense of how the attention mechanism works and

07:52:10 then if we scroll further down we'll see the multi-layer perceptron right here if you know something about neural Nets you know they're just a large combination of multiplications and additions or a m Matrix multiply and so I don't know if you can see this in the back there's a m mult which is how you do an Excel Matrix multiply and that's basically multiplying it its weight and then here we put it its activation function to get the next prediction okay let's keep going okay next we have the language

07:52:39 head and this is where we actually reverse the process so what we do is we take the last token and we uned it and reverse the embedding process we did before and we probabilistically look at which are the tokens the closest to the final last tokens un embedding and we interpret that as a probability distribution now if you're at temperature zero like we are in this spreadsheet then you just take the thing with the highest probability but if your temperature is higher then you sample it according to some algorithm like beam

07:53:12 search let's take a look and we'll go here so again I don't know if you can see in the back but this function here is basically there we go this function in the back basically is taking block 11 the output of the very last block it's putting it through a step called layer Norm then we multiply it another m m times The unembedded Matrix and these are what are known as our logits and then to predict the next most likely token we just go to the next one and if you can see this function it basically

07:53:50 is looking at Max of the previous column you saw in the previous sheet um and it's taking the the highest probability token just like that and that's our our predicted token we get a token ID then we look it up in the Matrix and we know what the the next likely token is okay so that's the forward pass of how gpt2 works but how do all those components work together so let's take our patient and put them through a virtual MRI so we can see how he thinks before we do that there's something I forgot to mention

07:54:19 these are called residual connections inside every layer there's an addition operation what this lets the model do is it lets it route information around and completely skip any part of these layers either a tension or the perceptron and so you can reimagine the model is actually a communication Network or communication Stream So the residual stream here is every one of those tokens and information is flowing through them like an information Super Highway and what each layer is doing is we've got

07:54:48 attention moving information across the lanes of this highway and then the perceptron trying to figure out what the likely token is for every single Lane of the highway but there are multiple of these layers so there they really reading and writing to each other information in this communication bus what we can do is we can do a technique called loit lens we can take the language head we talked about earlier and stick it in between every single layer of the network and what was it thinking at that layer so that's what

07:55:13 I've done in this sheet so I give it the prompt if today is Tuesday tomorrow is and the predicted token is Wednesday and gpt2 does this correctly for all seven days what you see in this chart is ESS ually The Columns here from 3 through 9 are all those Lanes of the information Super Highway and for example here at block three this is the top most predicted token at the last token position so it predicted not the second most likely word was going to be still then it was going to be just these are all wrong so

07:55:48 let's look for what we know is the right answer Wednesday so over here at block zero we see Wednesday it's at the bottom of the Tuesday stream for some reason on that Highway well it makes sense it'd be close to Tuesday and then it completely disappears and then oh over here towards the last few layers suddenly we see tomorrow forever Tuesday Friday it knows we're talking about time we're talking about days and it gets Wednesday but it's still the third most likely token and then finally it moves it up to the

07:56:14 final position and then it locks it into place so what's going on here well a series of researchers uh basically took this logit lens technique on steroids and isolated that only four components out of the entire network were responsible for doing this correctly over all seven days what they found was that all you needed was the perceptron from layer Zero attenion from layer 9 and actually only one head uh the perceptron from layer 9 and then attenion from layer 10 and that's kind of what we saw in the sheet right at the

07:56:46 top we saw Wednesday and then it disappeared until the later layers pulled it back up and up in probability at towards the end of the process so it's an example of where you can see each layer acting as a communication bus trying to jointly figure out and create what they call a circuit to accomplish a task okay we are now out of med school and ready for surgery so you may have heard about uh the pioneering work that anthropic has done about scaling monos semanticity this gave rise to what was known as

07:57:15 Golden Gate clae it was a version of clae that was very obsessed with the Golden Gate Bridge to some it felt like it thought it was the Golden Gate Bridge uh conceptually here's how this process worked you have a large language model and then you have this residual stream we talked about earlier and then you use another AI technique an auto encoder this one's a sparse autoencoder and you ask it to look at the residual stream and separate it out into interpretable features and you then try and deduce

07:57:44 what each feature is and then you can actually turn up and down each of these features back in the residual stream in order to amplify or suppress certain Concepts it turns out a team of researchers led by Joseph Bloom Neil Nanda and others are are building out sparse Auto encoder features for open source models like gpt2 small so here for example is layer 2's feature 7650 I don't know if you can see it in the back it's basically everything Jedi So Gone to our friendly patient again and I've taken the vector

07:58:22 for that feature while we wait for Excel to wake up there it is that first row is essentially what they call the decoder Vector corresponding to Jedi and then I've basically multiplied by a coefficient and then I've basically formatted it so that I can inject it right into the residual stream so this is the start of the block you can see that steer block too it's basically just taking that Vector I showed you and adding it into the residual simple addition now we go to our prompt and originally normally you ask gpt2 Mike

07:58:53 pulls out his makes sense he pulls out his phone but if we turn the Jedi steering vector on I'll give you one guess what he's probably going to pull out let's see okay so now we hit calculate now um and this is where you get to witness the 30 seconds it takes um and while we wait for it to to run a couple notes so first of all the way anthropic did their steering was slightly different but similar in spirit there's a few other ways to do this kind of steering one of those is called representation engineering where the

07:59:24 steering Vector is deduced via PCA or principal component analysis and there's another technique called activation steering where what you do is you'd take the thing you want to amplify like Jedi and You' run the model through just on that token and then you'd run on something you might want to suppress like in this case phone and then you'd create a phone a Jedi minus phone vector and inject that into the residual stream okay there it is there it is Mike pulls out his lightsaber there we go we have

07:59:53 done it our operation has been a success we've created the world's first gpt2 Jedi stick that on lmis Arena okay uh well hopefully I've given you a little better insight into how large language models work but also why they work but the root message I want to leave with is that to be a better AI engineer it does help to unlock the Black Box partly this about just knowing your tools and their behavior and their limitations better uh but also we're in a very fast moving field and if you want to understand the

08:00:26 latest research it helps to know how these work and then last but not least when you communicate with non-technical stakeholders there's very often a perception of magic and the more you can clear that up the more you can clear up misunderstandings I'll give you just one example of where this bubbles up where architecture bubbles up to how you use them so this is the uh instructions for RW KV which is a different type of model but the template for a normal Transformers at the top the template for

08:00:51 an RW KV uh prompt is at the bottom and what's interesting is that they recommend you swap the traditional order of instructions and context because the attention mechanism or the pseudo attention mechanism in RW KV can't look back the same way a regular Transformer can so it's a great example of where model architecture matters all the way up to prompting okay here are the references for the research we talked about today and then if you want to learn more you can go to spreadsheets or all need. and

08:01:20 you can download this spreadsheet and you can run it on your own device if you want to see me go through every single step of this spreadsheet I just launched a course on Maven today um and the link to it is on that website as well um and that's it thank you please welcome to the stage CEO of llama index Jerry Lou [Music] great um hey everybody I'm Jerry co-founder and CEO of llama index and I'm excited to be here today to talk about the future of knowledge assistance so let's get started um first

08:02:18 you know everybody's building stuff with LMS these days uh some of the most common use cases we're seeing throughout the Enterprise include the following uh it includes like document processing tagging and extraction it includes knowledge search and question answering if you followed our Twitter for the past like year or so basically you know we've talked about rag probably 75% of the time uh and also just you start generalizing that question answering interface into an overall conversational um agent that can not only you know do a

08:02:48 One-Shot quering search but actually store your conversation history over time and of course this year um a lot of people are excited about building a gench workflows that can not only synthesize information but actually perform actions and interact with a lot of services to basically get you back the thing that you need so let's talk about specifically this idea of building a knowledge assistant which you know we've been very interested in since the very beginning of the company the goal is to basically

08:03:16 build an interface that can take in any task as input and get back some sort of output so the input forms could be you know a simple question it could be a complex question it could be a vague research task and the output form could be a short answer it could be a research report or it could be a structured output rag was just the beginning uh last year I said that rag was basically just a hack and there's a lot of things that you can do on top of rag to basically make it more advanced and sophisticated if you build a knowledge

08:03:45 assistant with a very basic rag pipeline you run into the following issues first is a naive data processing uh pipeline you know you put it through some basic parser uh do some sentence splitting chunking do top K retrieval and then you realize you know even if it took you 10 minutes to set up that it's not suitable for production it also just doesn't really have a sense of being able to understand more complex broader queries so query understanding and planning there's also no uh kind of more sophisticated way of interacting with

08:04:17 other services and it's also stateless so there's no memory so in this setting we have said you know rag is kind of boring uh if it's just the simp rag pipeline it's really just a glorified search system on top of some retrieval methods that have been around for decades and there's a lot of questions and tasks that naive rag can't give an answer to and so one thread that we've been pulling a lot on is basically figuring out how to go from simple search and naive rag to building a general context augmented uh research

08:04:49 assistant so we'll talk about these three steps with some cool feature releases you know in in the mix um but the first step is basically Advanced Data and retrieval modules even if you don't you know care about the fancy agentic stuff you need good core data quality modules to basically help you go to production the second is Advanced single agent query flows building some agentic rag layer on top of existing data services as tools to basically enhance the level of query understanding that your QA interface provides and then the

08:05:19 third and this is quite interesting is this whole idea of a general multi-agent task solver where you extend beyond even the capabilities of a single agent towards multi-agent orchest ation so let's talk about Advanced Data and retrieval as a first step the first thing is that any llm map these days is only as good as your data right garbage in garbage out if you're an ml engineer you've heard that uh kind of statement many times um and so this shouldn't be net new but it applies in the case of

08:05:50 llm app development as well good data quality is a necessary component of any production grade LM application and you need that data process in layer to translate raw unstructured semi-structured data into some form that's good for your all map the main components of data processing of course are parsing chunking and indexing and let's start with parsing so some of you might have seen these slides already but basically the first thing that everybody needs to build some sort of proper rag pipeline is you need a

08:06:20 good PDF parser okay or a PowerPoint parser or some parser that can actually extract out those complex documents into a well struct Ed representation instead of just shoving it through Pi PDF if you have a table in a financial report and you run it through Pi PDF it's going to destroy and collapse the information blend the numbers and the text together and what ends up happening is you get hallucinations and so one of the key things about parsing is that even good parsing itself can improve performance

08:06:48 right even without advanced indexing retrieval good parsing helps to reduce hallucinations a simple example here is we took the Cal train schedule right the weekend schedule for Cal Train parsed it through llama parse one of our offerings and through some well structured document parsing format because the llms can actually understand well spatially laid out text when you ask questions over it I know the text is a little faint it's totally fine I'll share these slides later on you're able to actually uh get back the correct

08:07:16 train times for a given column versus if you shove it into Pi PDF you get like a whole bunch of hallucinations when you ask questions over this type of data so that's step one you want good parsing and you can combine this of course with Advanced indexing modules to basically you know model heterogeneous data within a document uh one announcement we're making today is you know we opened up llama parse a few months ago it has like tens of thousands of users tens of millions of pages processed gotten very

08:07:43 popular and in general if you're an Enterprise developer that has a bucket of PDFs and wants to shove it in and not have to worry about some of these decisions uh come sign up uh this is basically what we're building on the wac cloud side The Next Step is Advanced single agent flows so you know we have good data retrieval quality or sorry good data retrieval modules but in the end right now we're still using a single llm promp call so how do we go a little bit beyond that into something more interesting and

08:08:14 sophisticated we did this entire course with uh you know Andrew Ang at deeplearning.ai and we've also written extensively about this uh in the past few months but basically you can layer on um different components of Agents on top of just a basic rag system uh to build something that is a lot more sophisticated in query understanding planning and Tool use and so the way I like to break this down right because they all have trade-offs is on the left side you have some simple components that come with lower cost and lower in

08:08:42 latency and then on the right you could build full-blown agent systems that can you know operate and even work together with other agents some of the core agent ingredients that we see that are pretty fundamental towards building uh QA systems these days include uh function calling and Tool use uh being able to actually do query planning whether it's sequential or in some style of a dag and also maintain uh conversation memory over time so it's a stateful service as opposed to stateless we've pioneered this idea of a

08:09:13 gentic rag where it's not only just you know rag as a single LM prompt call where the whole responsibility is to just synthesize the information but to actually use the LMS extensively during the query understanding and processing phase where not only are you just directly feeding the query to a vector database in the end everything is just an LM interacting with a set of data services as tools right and so this is a pretty important framework to understand because at the end of the day you're going to have in any piece of llm

08:09:42 software llm interacting with other services whether it's a database or even other agents as tools and you're going to need to do some sort of query planning to basically figure out how to use these tools to solve the tasks that you're given we''ve also talked about AG reasoning Loops right probably the most stable one that we've seen so far is some sort of while loop over function calling or react but we've also seen fancier agent papers arise um that basically deal with like dag based planning planning out an entire dag of

08:10:12 decisions or tree based planning you know you plan out an entire set of possible outcomes and try to optimize there the end result is that if you're able to do this uh you're able to build personalized QA systems um that are capable of handling more complex questions for instance comparison questions across multiple documents being able to actually maintain the user State over time so you can actually revisit the thing that they were looking for being able to for instance look up information from not only unstructured

08:10:40 data but also structured data by treating everything as a data service or a tool but you know there are some remaining gaps here first of all you know we've kind of had some interesting discussions with other people in the community about this but a single agent generally cannot solve an infinite set of tasks um if anyone's Tred to give like a thousand tools to an agent the agent is going to struggle and generally fail at least with current model capabilities and so one principle is that specialist agents tend to do better

08:11:10 if the agent is a little bit more focused on a given task uh given some input and then the second Gap is that agents are increasingly interfacing with services that you know maybe other agents actually and so we might want to think about a multi-agent future so let's talk about multi-agents and what that means for this idea of knowledge assistance multi-agent task solvers first of all why multi-agents well we've mentioned this a little bit but they offer a few benefits Beyond just a single agent flow first they

08:11:44 offer this idea of being able to actually specialize and operate over a you know Focus set of tasks more reliably so that you can actually stitch together different agents that potentially can work together to solve a bigger task another benefit or set of benefits is on the system Side by being able to have you know multiple copies of even like the same Alm agent you're able to paralyze a bunch of tasks and um and able to do things a lot faster the third thing is that actually with a multi-agent framework instead of having

08:12:15 you know a single agent access like a thousand tools you could potentially have each agent operate over like you know five to 10 tools and therefore use a weaker and faster model and so there are actually potential costs and latencies savings there are of course some fantastic multi-agent Frameworks that have come out in the past few months and many of you might be either using those or kind of building your own and in general some of the challenges in building this reliably in production include uh one being able to you know um

08:12:43 either let the agents kind of operate amongst themselves and build some some sort of like unconstrained flow or actually being able to inject some sort of constraints between the agents you're basically explicitly forcing an agent to operate in a certain way given us ear input the second is when you actually think about having these agents operate in production currently the bulk of agents are implemented as functions in a Jupiter notebook and we might want to think about defining the proper service

08:13:09 architecture for agents in production and what that looks like so today you know I'm excited to launch a preview feature of a new repo that we've been working on uh called llama agents um and it's an alpha feature but basically it represents uh agents as micro Services right so you know in addition to some of the Fantastic work that a lot of these multi-agent Frameworks have done the core goal of llama agents really is to think about every agent as just like a separate service and figuring out how

08:13:40 these different Services can operate together communicate with each other through a central uh API you know communication interface and then also uh work together to solve a given task um that is you know scalable can handle multiple requests at once um is easy to deploy to you know different typ of services um and basically each agent can encapsulate a set of logic but still communicate with each other and actually be reused across different tasks so it really is really thinking about how do you take these agents out of a notebook

08:14:11 and into production and it's an idea that we've had for a while now but we see this as a key ingredient in helping you build something that's production grade uh a production grade knowledge assistant um especially you know as the world gets more agentic over time so the core architecture here is that you know every agent is just represented as a separate service um you can write the agents however you want basically you know with uh llama index with another framework as well and we have some of the interfaces to basically

08:14:40 build a custom agent and then you're able to deploy it as a service and basically the agents can interact with each other via some sort of message CU and then the orchestration can happen between the agents via like a general control plane right we took some of the inspiration from you know existing resource allocators for instance like kubernetes or just like other kind of like open source like um systems level projects and the orchestration can be either explicit so you explicitly Define these flows between services or it could

08:15:08 be implicit right you can have some sort of llm orchestrator just figure out what tasks to delegate to uh given the given the current state of things and so one thing that I want to show you basically is uh figuring out or just showing you how this relates to this idea of knowledge assistance right uh cuz we think that multi-agents are going to be a core component of this and this is basically a demo that we whipped up showing you how to run llama agents um uh on a basic rag pipeline this is a pretty trivial rag pipeline there's like

08:15:42 uh a query rewriting service right and then also some sort of uh default agent um that basically just does rag like search on retrieval um and you can also add in other components and services like reflection you could have other tools as well or even a general tool service and the core demo here is really showing that you know given some sort of input they're communicating through uh with each other through some sort of like API protocol and so this allows you to for instance launch a bunch of different

08:16:09 client requests at once handle you know task uh requests from different directions and basically have these agents operate at um as like an encapsulated microservice right and so the query rewrite agent takes in some sort of query processes it rewrites it into some uh new query and then you know second agent will basically take in this query do some search and retrieval and um basically output a final response if you built a rag pipeline all this stuff like the actual logic should be relatively trivial but the goal is to

08:16:38 basically show you how you can turn something even that's uh even something that's trivial into a set services that you can basically deploy right um and this is just like another example that's basically a backup slide that basically again highlights the fact that you can have multiple agents right and they all operate and work together um to BAS achieve a given task so you know the QR code is linked first of all this is in Alpha mode right and so we're really excited to basically share this with the community we have

08:17:09 we're very public about the road map actually so check out the discussions tab about what's actually in there and what's not we're launching with uh dozens of uh a dozen basically initial tutorials to show you how to basically build a set of like microservices that basically help you you know build that production grade uh a gench tech knowledge assistant workflow and uh there's also a repo linked that I think should be public now um you know in general we're pretty excited to get feedback from the community about what a

08:17:37 general communication protocol should look like how we basically integrate with some of the other you know awesome work that the community has done and basically uh help achieve this core mission of again building something that's production grade and a multi-agent assistant and this is just the last component um which uh I already mentioned but basically if you're interested in like the data quality side of things like let's say you didn't care about agents at all and you just care about data quality uh we're opening up a

08:18:04 weight list for llama Cloud more generally so that you're able to you know deal with all those decisions that I mentioned the parsing chunking indexing and ensure that you know your bucket of PDFs with embedded charts tables images is processed and parsed the right way um and if you're an Enterprise developer with that use case uh come talk to us so that's basically it thanks for your time and hope you enjoyed it [Applause] ladies and Gentlemen please welcome to the stage via recording founder of small

08:18:42 Ai and the co-founder of the AI Engineers Summit and World's Fair [Music] swix hi everyone I've been thinking a lot about about borders recently for no particular reason uh and borders are a very innately human thing if I don't have the right piece of paper I cannot cross this Line in the Sand like it's a very very real problem that I face and many many people face every single day and borders are just one kind of constraint that humans just make up and I think that's very interesting that we respect borderers so much but AI does

08:19:19 not AI is a border disrespect it is very very easily multilingual so if you trained in LM on mostly mostly English Text corpus going to learn other languages just as a side effect it's going to be very natively multimodal because it can you can turn llama into a vision language model with just like 100 bucks of just post training there's it's very um disrespecting of ground truth borders because it can just doesn't know the difference between hallucination and memorizing from a world model and it

08:19:48 also doesn't respect copyright which is a whole other topic that we won't get into today but it's also super fascinating and how does that relate to do with AI engineering right like I think a lot of you here are here because you are interested in that concept at least maybe you identify as an AI engineer maybe you're trying to hire an AI engineer so there a lot of definitions floating around and I I confess that you know I've I've contributed to that um is the engineering an API line right that's

08:20:12 the that's the line that a lot of people have and that's come under some debate recently and yeah that that's one form of AI engineering and I think that is useful to some people for understanding like where the responsibilities in a team might stop an in and start with uh the other people in the team and maybe it's there different subtypes right like last AI engineer Summit I talked about the three types of AI engineer that I was seeing emerge the AI enhanced engineer the AI products engineer and the

08:20:38 non-human agentic AI engineer or it could be a job description that we try to sort of list out and this is something that on the Laten space podcast we recently went through with elicit talking about the different roles that they see within their teams as well so okay if I broadly have any of these three things do I have I nail down a good definition of engineer that is workable yes right but is that something that we're happy with is there something that we can is is there nothing left to explore I think

08:21:05 the answer is no I think there's more to explore I think the very easy copout as well for people discussing this is that you have your opinion I have my opinion you come from your point of view I come from my point of view we agree to disagree or we agree that you know they different different strs different folks and we move on um I don't really like that just because there's no shared agreement on the things that is ground truth to everybody so I want to raise that challenge a little bit more I was in a

08:21:33 podcast with rahh who's one of the speakers today talking about what this conference is and why this conference is does what it does and I always say that a engineer conferences are effectively my highest Stakes expression of what I think the state of a engineering is so this time last year 2023 AI years at two times of human years um uh we had a few tracks and we had a few topics that were up for debate we had rag Coen and then agents and multimodality um all those tracks are repeated here today I have

08:22:03 some speakers Illustrated here just for illustrative purposes these obviously are not every not everyone involved um but I think just like the Inside Out metaphor that uh I've been thinking a lot about um as the a engineer matures so does the number of concerns that you have to juggle in your head so this year you know after you're a competent AI engineer this year you're now faced with like okay I have to migrate to open models I have to build up my evals maybe I should have done that first but it's a

08:22:29 whole topic of discussion maybe I should scale up my inference or maybe I should deploy it to the Fortune 500 and maybe on the management side of things I should be hiring teens of AI engineers and managing AI strategy for my company um I think the last track you know like I talking about the nine tracks in in a Engineers it's always about the network the the community the network that we're building um that's probably the single most important part of this conference and that's the part that we cannot sell we cannot I cannot

08:22:53 put on the website hey we have good Community because no one will believe us you have to see for yourself but please for those of you who've been uploading to the Google photos album who've been tweeting out your photos uh and sharing them on LinkedIn please keep doing that that's a way for myself and everyone else who's not at the conference to try to join in on the fun um the reason I'm not comfortable with any of these tracks because I is because I know how they were made because I made them up and I know that

08:23:19 because I was looking at um the the original document for the rise of the AI engineer it's we're celebrating the one-year anniversary today and uh just down in the document somewhere I just listed out the you know disciplines that I thought the E engineer would have and those eventually became mostly mapping to the tracks that I have been exploring in these conferences and the meetups that I do and it's arbitrary like why is there a separate agents track from Coden why is there a separate rag track from

08:23:46 open models like these are all related um what you know they're they're all of a kind and obviously as an competent Ager you should be familiar with all these things and that brings us back back to this mindset of having boundaries and borders right these are all made up by someone I made them up for this one but you know you're going to live in a world where your boss made it up at your company and these are not reflective of how reality actually has to operate right if if you start with all the rules and a different group of

08:24:15 people came in would they agree on the same rules probably not just because they're made up but the laws of nature are hard to make up because the reality actually works that way if you had an alien civilization come down to earth they would discover that gravity works the same the sun uh the energy from the Sun works the same the magnetic field works the same so what are the laws of AI engineering we we'll come we'll come to that via definitions of software engineering and real engineering so I

08:24:39 went and looked up definitions of software engineering and i e talks about the application of engineering to software with the design implementation testing and documentation of software Google developers talks about mostly the same thing they also mentioned software life cycle management okay like really reasonable definitions uh of kind until you look at real Engineers real Engineers talk about applying Natural Science Sciences uh utilizing them for benefit of humanity both the I and the National

08:25:07 Association of Engineers agree on that uh and it's curiously missing from software engineering like the benefiting Humanity part the understanding natural laws part completely missing from software engineering so my proposal to you is that AI engineering is somewhere in between right the the the the the software engineering but then encountering a lot of the more of the real world constraints than you would in a typical software engineer career so if we know the laws of Earth and they are independently derived they cannot no

08:25:31 matter what point of view you're looking at they all are the same laws then what are the equivalent laws of AI engineering I have a few you can come up with more but I'm just going to propose some to start off debate there's constants so for example if you're designing for humans you should respect the fact that humans only speak at 80 words per minute but they read at 200 words per minute right so there's an inherent disparity there there's also con contingent facts that things that are true for now instead of

08:25:56 forever and true for now facts are for example like the Apple intelligence when they ship a local model on every phone then that inference speed of 30 tokens per second that they're advertising becomes the Baseline speed limit or speed barrier of what intelligence that is too cheap to meter should look like and these things they're not set in stone they're not actual physical laws so they also Trend over time due to forces and momentum and I want to establish a little bit like I think it's it's very beneficial for AI Engineers to

08:26:26 understand what the mor laws of thei is so that you can plan for them so that you don't have to make the bad bets that are not going to last just obviously just because of overwhelming evidence the first bet is the improving of context right um a year ago I was interviewing Mosaic and talking about MPC 7B with they whopping 60,000 70,000 token context with a lot of loss today sitting in the audience we have people who have trained million token context windows and we've also uh have from anthropic which just released 3 five

08:26:57 yest last week um that the fact that you know we have complete utilization it's not just about the length of context it's also about the utilization of context and I think Greg who saiding in the audience as well would be very happy with how Claud is improving on their utilization of their very very long context Windows there's also the cost of intelligence the commodification of intelligence so in the past two years we've seen a 99.55% decline in the cost of gpt3 level intelligence the cost of GT4 level

08:27:22 intelligence has probably come down maybe 90% maybe maybe 80% um from from GT4 to llama 3 and all the the newer models that finally I think it's worth commenting a little bit on where AI engineering stands in contrast to other AI philosophies there's EA versus eak and maybe we're in the middle like we we care about safety but we also want to accelerate right it's kind of like a weird combination of of the two things my proposal is that that one dimension isn't enough to express how a engineer differs from the other philosophies and

08:27:51 actually need to add a second dimension to talk about utility we are utility Maxis above all else we see what's out there and we want to use it to benefit Humanity so my message to everyone at the world spare is to try to disrespect borders a little bit try to avoid your own dogmatic beliefs lazy consensus of other people or passive reactions and in other words try to disagree disagree more disagree with your own conclusions disagree with each other productively and disagree with the status quo and I

08:28:16 think there you will find that this conference becomes more of a useful landmark in your careers rather than just the party which it can very well be uh but my final analogy which I really like is is that AI Engineers are the kind of person that looks at shogo and sees instead of a monster that cannot be tamed they want to turn them into Mass Rapid Transit and the kind of person that looks at that looks at you know force of Nature and wants to turn it into tools that are useful for people is the kind of engineer that I would love

08:28:42 to speak to and welcome at conferences like this one so that's my view of what borders and Engineering Without Borders should look like I I very much encourage you to jump between tracks to jump between friend groups to jump between disciplines and modalities because here's the one place that you can do that outside of your work um and to mingle with everyone else that we've gathered so I hope you enjoy doing that um I wish I was there in person but just share them online and I hope to see you in person at the next one

08:29:14 bye all right what a way to close out the day um one more time for swix he's watching on the live stream so let's make sure he can hear us from [Applause] here we miss you swix I know you're watching that is some great fruit for Thought um we're about to head into the pine cone after party so I'll just be very brief I just want to thank everyone for coming to the second day the first day of sessions the second day of this three-day event um so I'm not going to keep you from the pine cone Afterparty

08:29:44 but we do have a representative from Pine Cone here to just say a few words before we head to the Pine uh the pine cone Afterparty so please join me in welcoming for the last word head of community at Pine Cone Joselyn Matthews [Music] hi welcome how's everybody doing good yeah that was my attempt to get you all on my side did it work so I'm Jon Matthews I am the head of community at Pine Cone and you may already be familiar with us we are the leading Vector database uh we're also proud to be the sponsors of the official

08:30:25 afterparty of the AI engineer uh World's Fair so um I tried I used to be an engineer myself and I tried to think about like it's the end of a long day what do you want on the way to a party is it a pitch no so instead I'm going to talk about our new announcement which is something that you can Tinker with and not a product that we're selling currently yesterday we um yesterday we launched pine cone assistant in beta so Pine con assistant is an API service for answering complex questions about your proprietary data and it answers

08:31:04 them accurately and securely within your applications uh this is the QR code if you want to sign up for the beta preview and the code will come up again so if your phone's at the bottom of your backpack don't worry you're going to see it again um with pine cone assistant you're going to get Simplicity high level results and full control over your data um I think actions speak louder than words so I've got this little animated gif and you can see here so developers still really struggle to build AI assistance

08:31:39 that can accurately answer questions about private data I think we all know that and we also all know that publicly available models are unaware of this data and providing it can pose security concerns uh most teams also might lack the time or the Deep AI expertise to be working uh with rag so pine cone assistant gives you pine cone assistant gives you a better way you just upload your PDF or your text files and then you start asking questions you can prototype quickly with the drag and drop method as

08:32:12 seen in the GIF and then add it to your applications in a matter of minutes with the chat completion compatible API um and then in response you're going to receive relevant answers that are grounded in your data with references uh this is what the code looks like you can see it's it's very comprehensible very accessible for a wide variety of Engineers um the answer quality will get better over time across more domains and will eventually support complex queries over multimodal data um I also thought that you might be

08:32:50 interested in seeing some actual numbers so um you can see that you are able to provide more accurate results with less effort um and that's why we built Pine con assistant with a focus on delivering the highest quality and most Dependable answers today it already performs better in beta than other assistant apis for text Heavy articles um text Heavy technical data and financial and legal documents um I also know from my work in the community uh where I deal with thousands of Engineers uh a question

08:33:28 that always comes up is around data protection so you're able to protect and control your data the data that you upload is encrypted and isolated your data is used as confer your data is used as context and reference for answers in real time it's not used to permanently fine-tune it's not used to train the underlying language model and in essence you control what the assistant knows and you control what the assistant forgets uh you can prototype and ship AI assistants in minutes and we welcome you

08:34:04 to you know try it out for yourself Tinker with the Tinker with it um if it takes you longer than minutes please email me my my email is Jocelyn Pine con. um actually no send it to the group inbox send it to community Pine con. you can um easily create an assistant inside the console and like I said you receive relevant answers which are grounded in your data and which do have references um all of the infrastructure the operations and organization are basically handled for you and that includes everything you see

08:34:39 on the screen which would be the chunking the embedding file storage Etc um the beta Starts Now with more to come we announced it for the first time yesterday and this event is the first time that we've talked about it publicly directly live with Engineers so you're kind of in a way hearing it here first um we're releasing it in beta as I said I want to stress the beta part and we want you to try and share your feedback as you're working with it the starting limits are one gig and I believe it's

08:35:12 100 queries per month those might or might not change during the beta period um and you can also expect rough edges because it is a beta but you can also expect rapid Improvement based on the feedback That We Gather from you so so that brings me to the end of my talk and let the party start thank you so much for being here [Music] [Music] are [Music] do [Music] [Music] [Music] [Music] up [Music] [Music] [Music] it bre [Music] [Music] [Music] let's go I want to up [Music] [Music] I to IAT and I'm ready to