## EPISODE 1522

**[00:00:01] SPEAKER:** All right, you're ready to go?

**[00:00:02] GH:** Yup.

**[00:00:02] SPEAKER:** All right, George Hotz, welcome to Software Engineering Daily.

**[00:00:06] GH:** Good to be here.

**[00:00:07] SPEAKER:** So, I'm sure a lot of people know you from jailbreaking breaking devices or working on self-driving cars or marathon Twitch sessions. But for those that don't know who you are, can you give a little intro background on who you are and what you've been up to?

**[00:00:23] GH:** My name is George. I'm a programmer. I’m a programmer for 20 years.

**[00:00:28] SPEAKER:** Nice. I love it. So, for those of you don't know, George is a bit of a celebrity. How old were you when you jailbroke the iPhone? 18 or so, something like that?

**[00:00:38] GH:** 17. I didn't do the first jailbreak. I did the first unlock.

**[00:00:42] SPEAKER:** Okay. Okay, so –

**[00:00:45] GH:** The technique that some guy, “Oh, man he’s still on Twitter 20 years later. Bro, you stole my technique.” I’m like, “What you mean stolen? You got a copyright on that? Come sue me, bro.” Don’t sue me, don’t sue me. I’m getting more of that now.

**[00:01:00] SPEAKER:** Yeah, cool. So, jailbroke the iPhone, jailbroke PlayStation, all this cool stuff. And then what 2015, 2016 released this self-driving kit that you can just add to not any car, but almost any car. You can hook it up to a huge variety of cars. So, not starting from the ground up like Cruise, Waymo Tesla, things like that. But actually, modifying existing cars. That became comma.ai. You recently left comma.ai. But can you tell us a bit about, I guess, like how it went over the last five or six years? What did you learn? What changed? Are you bullish on the industry? What are you thinking?

**[00:01:37] GH:** Well, so first off, the main reason I left was there just wasn't that much for me to do. We have really good people in place who are executing on the plan. I go there, and I want to change things. Like I said, like, I'm a wartime CEO, I'm not a peacetime CEO. I mean, I think they're doing a great job. And I think that the way to write good software is to write it three times. So, everything gets rewritten, and everything gets improved, right? And it's a very slow process. The same thing is true about hardware, right? There's a reason the iPhone 14 is so good, it's because of the 14. It's just a slow grind.

**[00:02:23] SPEAKER:** So, do you like to write the first version, but don't love the second and third as much?

**[00:02:27] GH:** Yeah. I like the first. I feel like once I can – as long as I can conceptually improve things, I'm down to keep rewriting it. But once I kind of run out of conceptual improvements, and it just comes like, "Well, okay, you just got to do this", then I'd rather hand it off.

**[00:02:48] SPEAKER:** Yeah. A couple things that were different in your approach, at Comma compared to other places. One, I think was in sort of like the segmented approach versus a more end to end approach that you took. Can you talk about those differences? Do you still think that's the end to end approach that you took at Comma is the right approach?

**[00:03:07] GH:** Oh, unbelievably. I think we could even almost go more end to end today. So, what Comma really is, is we built a simulator and then we train our models in this simulator, right? This is second paradigm model. So, we talked about first paradigm models were these models that just we hand coded lanes and cars, it turns out with lanes and cars, you can do a lot of driving. But then it becomes really hard to do something like say drive to an intersection, right? Just like drive through an intersection because there's no lane lines there. You could say, "Okay, I'll detect the lane lines on the other side and interpolate the path", but it's not really good.

So, this was first paradigm models. We're like, "How do we fix this?" We fix this with second paradigm models. Second paradigm models are trained in a simulator. But what's notable about this simulator is it's not like a car law, Unreal Engine style simulator. We call it the small offset simulator, because it uses a real route, loads them in, and lets you make small deviations from what the car actually did. Right? So, you can load up a route –

**[00:04:08] SPEAKER:** Which way?

**[00:04:10] GH:** Well, laterally or longitudinally. There's like a real path of the ego car. And then you can change that by kind of just visually shifting stuff, right? You get the depth of everything, and then you can like re-project. And this solves a really fundamental problem. We call it behavioral cloning, but it means a lot of things. If you just take a model, and you just predict the human path, given the picture, this will not drive a car. And the reason it won't drive a car is because your samples are not IID. Right? When you're training a classifier, the first image is an image of a cat, and it says cat or it says dog. It doesn't matter if it says cat or dog because it doesn't affect the next images. This isn't true to driving.

If you have an image and you predict go left, the image afterward changes, your predictions affect the future. And this is why standard supervised learning techniques don't work at all on driving, this is why you have to train in a simulator. So, we try to simulate or some hybrid of supervised and reinforcement learning. And this is, I think, the most extreme of anyone in the industry. But I would like to go even more extreme. I would like to go where we at least train a really powerful foundation model, and we're moving in this direction. A foundation model, we train an autoencoder on the visual features. You can look like when you look at stable diffusion, the outside of stable diffusion is an autoencoder. Meaning you can do dimensionality reduction on the image in a completely unsupervised way.

And then the second part is a really powerful dynamics model, just a predictor. And the auto encoders will output things into discrete spaces, not continuous spaces. The cool thing about a discrete space, this is like language, right? So, you can break the picture down to say 64 words from a vocabulary of 1,000. And then you can use GPT style techniques to build a predictive model. GPT is also working on like a vocabulary of like 1000, and 64 words is very much within. So yeah, we're getting close to the true dream and we call this third paradigm models.

**[00:06:32] SPEAKER:** Okay. I want to talk a lot more about AI as we go. So, building a simulator, is the purpose of having a simulator just it's much cheaper to sort of have a simulated world and make changes as compared to like, driving it out in the world and trying to get better on that video?

**[00:06:48] GH:** No, you can't. So, the problem is – okay, the simulator allows you to train on policy. If I gather – if I take a car out, right? Let's say hypothetically, I wanted to train on policy with the car. I could load my models onto the car, I could go out and gather data, and then I can take one more step in those models, and then load the next models onto the car. But the problem is you have to do this every time you train. We have tons and tons of data. We're not data limited at all. But the problem is all that data is off policy, when you're training a model, right? In order to get on policy data, you need to a simulator.

**[00:07:29] SPEAKER:** Gotcha. Okay. Sounds good. What do you mean off policy, on policy, just to clarify that?

**[00:07:36] GH:** Well, these are reinforcement learning terms, right? So, on policy, okay, so you can think of reinforcement learning is like you have a state at time T, and then you take an action. And then you have a state at time T plus one. So, when you're on policy, your actions came from your current model. When you are off policy, the action could come from well, actually the human who drove the car, right? So, it's much harder to learn off policy than on policy, because your model didn't take that action.

To be honest, like a lot of this stuff is like these – these are very standard. Everyone who knows reinforcement learning knows that stuff. But there's not many theoretical guarantees for anything about off policy. This is one of the things we really explore at Comma, because, okay, a lot of off policy learning depends on noise. If you are training on policy, let's say you have a model that knows nothing, and does completely random actions, it turns out with completely random actions, you can understand what those actions do. If you're off policy, and you have rollouts with no noise at all, well, okay, tell me what happens if I'm driving down the highway, and I go like that with the wheel? You don't have that in your data set. So, maybe it's a good action, right? You don't know. And this a problem with off policy. But on policy, your model, we'll just do that and be like, "Oh."

**[00:09:02] SPEAKER:** That was a mistake. So, you've talked about comma being like the furthest down the road, at least on this this simulator thing. What do you think of the other approaches, especially with Cruise, Waymo, and seeing more of their actual driverless cars in the world? What do you think of that?

**[00:09:19] GH:** The problem with Cruise and Waymo, and I've always said this, people think that like, my dispute is about LiDAR, or maps or something like that. I think LiDAR map are done. But how do you think that's the fundamental problem with Cruse and Waymo? The fundamental problem with

Cruise and Waymo is unit economics. Each one of those cars cost like half a million dollars, and what product is the car provide? Well, it's like Uber. But is it as good as Uber?

**[00:09:47] SPEAKER:** Probably not quite, for the customer. Yeah, a little slower.

**[00:09:50] GH:** And why not?

**[00:09:54] SPEAKER:** I would just say like, it's probably safer, a little more cautious in certain ways that don't actually benefit the driver experience. Maybe slightly over stopping at that certain things.

**[00:10:07] GH:** Yeah, it takes it takes 50% more time to get anywhere, because it has to come to a full stop at every stop sign because that's what's coded in the software. Where's my Uber driver can go full stop, in ever stop sign. Look, I mean, this isn't what – this is a very fundamental problem, right? This is not a problem that like can be fixed. These things are going to be slower. These things right now are more expensive. And you're like, "Okay, with scale, it could come down." But like, how much scale? I think that some six figure software engineers sometime forget how much money an Uber driver makes. In fact, a friend of mine drove for Uber, and I looked at her Uber stuff. And I'm like, "Yo, you're losing money. You understand you're paying more for gas." She's like, "I like it. I like it. I like it." Uber preys on people who are bad at math. You literally are getting drivers to subsidize.

**[00:10:59] SPEAKER:** Not thinking about depreciation and all the wear and tear they're putting on the car? Absolutely.

**[00:11:03] GH:** When you truly factor in everything, a lot of kind of hobbyist Uber drivers lose money. Which another way of saying lose money is give free money to Uber, right? So, I think that this is the main problem with way Waymo and Cruise. It's a very fragile managed system that costs a lot of money and may not provide a product that's competitive on the market.

**[00:11:26] SPEAKER:** So, it's two things, it's slower, but then it's also cost more. That slower aspect, would that also be true for Comma? Or do you think Comma will actually be a better driving experience?

**[00:11:39] GH:** We're not trying to be level five or level four, and then this is sort of a key distinction, right? If you ever don't like what your Comma is doing, let's say, it takes a term too aggressively or not aggressively enough. Take the wheel and change it. And that's the distinction. So, our policy doesn't have to be this ideal policy. It's just, it's an assistant system.

**[00:12:07] SPEAKER:** Yeah. Well, on that same note, I saw you on Twitter recently, going back and forth with Gary Marcus a little bit and saying, "L5 by the end of the decade." I mean, tell me about that. So, you don't think Comma's the one that's going to get there?

**[00:12:22] GH:** I think it's very possible. I kind of think that by the end of the decade, it's just going to be pretty easy. In the same way, right now, it's pretty easy to train like an unsupervised chess engine. I think that the techniques will just become a lot better and the software frameworks to train these things will become a lot better, the compute will become a lot more available. So, I don't think it's going to be particularly tricky. I do think by the end of the decade, assuming Comma continues on its path, we'll have pretty much solved it. I think Tesla's going to beat us by about one to two years. I think there's a decent chance we get second.

**[00:12:59] SPEAKER:** Based on what you thought two years ago, or maybe four years ago, do you think you're ahead of, behind, or sort of on track with how you predicted that?

**[00:13:07] GH:** In the very beginning, I was over optimistic. But I think I corrected pretty quickly on that. And then I think it's pretty much where I thought we'd be.

**[00:13:19] SPEAKER:** Cool. I want to talk a lot more about that state of AI stuff that you mentioned around hardware and software. But a few more things on Comma, what benefits or downsides do you get from having sort of like a post install kit, rather than building a car from the ground up? Is it mostly a price thing? Are there are other benefits in making the product evolve faster? What's that like?

**[00:13:41] GH:** I don't care about this stuff. Comma's mission is to solve self-driving cars while delivering shippable intermediaries. We use the hardware to fund the company. It's very important for me to be profitable and to make a sustainable company. But like, aside from that, "What are you going to do when the cars like locked down?" Who cares? We're here to solve an AI problem. I think that's something that's lost on a lot of people who live in this world where their companies are trying to be on some, like, optimality curve all the time. I don't care. We raised a small amount of money from a bunch of people who like our latest round of investment, these guys are like, they're down for the mission. Right? None of them are here. “Look, I don't want to squander your money, we're going to try our best to seriously sell 100,000 of these devices for 100 million in revenue is pretty good.” But like, the point is solving self-driving cars. The point is not to make a business that makes money.

**[00:14:39] SPEAKER:** How have you been so capital efficient? Because like you're saying, you've raised, I think 18 million and for a long time, it was 8 million. And I see web-based SaaS apps that are raising 100 million and just shoving data in and out of a database. I guess, how have you done that while shipping hardware and iterating on a hard space?

**[00:14:55] GH:** Yeah, I mean, you're really – if you cut out all the useless people and stuff, you don't need that much. This is what hacking is. Hacking lets you drill down to the absolute minimum like, “Okay, what is this?” Look, we got like our boards, they're routed by one guy. He’s very, very smart. And he designs the things. And he does it slowly, over time, it takes time, but like, okay, why do I need a team of 10 people? See, you need a team of 10 people if you want a marginal speed increase, right? Say, you need the product to be ready 2x faster, then you might need 10 people. But this **[inaudible 00:15:40]**. People don't scale very well.

So, when a lot of these things are not a question of like, doing a lot of things, but just doing one thing very carefully and correctly, it doesn't help you to have more people than more people's a big cost of this. But everything else is just, look, I'm a middle-class guy from New Jersey. When I look at the price of something I'm like, It's $20,000 for a couch nowadays, bro. Where are some startups would be like, “Yo, we got to buy this couch. It's from Design Within Reach.” And I'm like, “Dude, we’ll go to Ikea. We’ll get a couple guys, we put in the back of the pickup, that's what we're going to do.”

**[00:16:19] SPEAKER:** How many people work for comma?

**[00:16:22] GH:** 23.

**[00:16:24] SPEAKER:** That’s amazing.

**[00:16:25] GH:** 22. I don’t work there anymore.

**[00:16:25] SPEAKER:** Yup, sure. Okay. So, tell us about what's next. You recently left Comma, what are you working on?

**[00:16:32] GH:** So, I'm working on tinygrad. I'm trying to build – So, I think like PyTorch and TensorFlow are the Fortran and COBOL of deep learning programming languages. I wasn't around for Fortran and COBOL. I can only really talk about what was there. But I think that these things are very – they're very clunky, and they're – I mean TensorFlow in particular, right? We switched from TensorFlow PyTorch, maybe two and a half years ago. The thing that really drove me over the edge was it was a parameter to the TensorFlow optimizer called clipnorm, Clipnorm is this thing you do, clips all the norms, and we were putting in a number and it wasn't changing behavior. And then we look, read the code, clipnorm is broken. They didn't add the cert, there's just broken, right? And then like, this has to go. We switched to PyTorch. But PyTorch has similar sorts of problems. It's very hard to go into the PyTorch codebase. And read, okay, "I want to do a rally. Where does the rally happen?" Well, it's like four layers of indirection. And all these specialization things and all these – and then, I think that software, I think this is pretty well established that the amount of bugs in your software correlates to line count. So, can I write a deep learning library that's competitive, actually faster than PyTorch with 1,000 lines of code?

**[00:17:57] SPEAKER:** That's amazing. I was looking through it. Just for, I guess, comparison sake, do you have a sense of how long PyTorch And TensorFlow are?

**[00:18:05] GH:** Yeah. So, the core parts of PyTorch, let's say 10,000 lines and 100,000 lines. Now, in reality, the repos are 100,000 and a million. But let's say PyTorch is 10x bigger, and then TensorFlow is 10x bigger than PyTorch.

**[00:18:23] SPEAKER:** So, it is TinyCorp and tinygrad, is that your main focus right now? Will you –

**[00:18:30] GH:** You can look at my GitHub, last month. I've been coding a lot since I left. So, whenever I'm on my GitHub screen, I'm happy.

**[00:18:36] SPEAKER:** Yup. Good. And will you hire other people? Will this be your own project? What are your plans for Tiny?

**[00:18:42] GH:** I don't think so. I think that people who want to contribute are welcome to contribute on GitHub. My hope is, if it starts to get adoption, the companies will start to contribute to tinygrad. And then effectively, in fact, we say this about openpilot too. There are probably more people being paid full time to work on openpilot outside Comma than inside Comma now. This is really cool. I mean, this is how you know you're starting to succeed as an open source project, right? Maybe when I was younger, I was more a fan of like GPL style things, but I switched everything to MIT.

And like, what I see about open source is the reason that somebody wants to upstream their thing, you're like, "Okay, fine. Why would I upstream you?" Because we'll maintain it for you. If you're a company, let's say you're a car company, you want to get your thing, let's say small car company, right? You want to get your thing upstream to openpilot, so we maintain that code, and we do refactors. Look at the beauty of this Linux kernel and say, "Why? You don't need GPL to do this." When someone refactors the PCIe subsystem, you want your PCIe driver to keep working, and now it's someone else's responsibility once you've upstream it. This is the beauty of open source and it can work in such a way that it's worth it for companies to contribute back to these projects.

**[00:20:10] SPEAKER:** Which companies outside of Comma are paying engineers to work on open pilot to get it set up?

**[00:20:16] GH:** So, there are several, several Chinese car companies. There's another one, some guys down, I think they're in Vegas, doing like full self-driving sort of stuff. But it's just like, everybody who – there's no reason today that you should start a data system from scratch. Everyone is doing it as an idiot and they should use openpilot. It's really good. Not only is the model and stuff really good, but the infrastructure the whole thing is really good. We built – like Ross is, a lot of people use Ross for robotic stuff. Ross is, we thought about using Ross. We've been looking at like two core components of Ross. There's a serialization and there's the messaging.

So, Ross, build custom versions of both of those things. We start out in openpilot, with saying, "Okay, we're going to use Cap'n Proto." Cap'n Proto is like protobuf, but I think protobuf is really bloated. And then we're going to use the ZMQ, right? The MQ with Fairbridge node is a very common socket library. We actually moved off the ZMQ, because if two processes want to communicate, ZMQ uses sockets, and you're going through the kernel. You're making a copy. So, we switched to a custom thing called message queue, where you can process, B, can access a shared memory of process A. Process A creates a ring buffer, and then it pumps up.

But yeah, it's compatible. Right? You can, you can still run all of openpilot with ZMQ equals one and it'll fall back to ZMQ. These messaging, and serialization things are the best on the market today. Everybody should start using them. There are a few people out there now. I think somebody might be good dog.ai. They're like doing robotic stuff and it’s like, use openpilot.

**[00:22:08] SPEAKER:** So, you mentioned a bunch of companies using openpilot, and then on the TinyCorp, tinygrad website, you've sort of mentioned that you'd be willing to embed with a company to sort of – if they wanted to sponsor you. Do you have any ideal projects or companies that you need?

**[00:22:24] GH:** We had a discussion about a contract with a large AI chip company. The contract kind of fell. These companies are bloated and hard to work with. But I think as the – there are companies, multiple companies now. Cerebus, Tenstour, Graphcore that have done tape out of these chips, and they don't have software. They don't have software, they don't have a function PyTorch port. In fact, PyTorch to ship MPS, which is metal performance shaders, and I found a bug. I found a bug in their matrix multiply. That's not a major wrong. But let's move on things like transpose and it didn't take the transpose into consideration. I found this on stream when I was doing my tinygrad, my stable diffusion port of tinygrad. And I'm like, “Tinygrad’s got to be wrong. There's no way by PyTorch is wrong.” Wait, it was wrong. Only for MPS. Obviously, when you go back to CPU, if you're on an x86 and Nvidia, this is not going to be a problem. But it's because it's really hard to do a port of PyTorch to a custom accelerator. You have to port tons and tons of operations, write a lot of code. Whereas, tinygrad, it's really one operation.

**[00:23:37] SPEAKER:** Interesting. And it works with all these different accelerators?

**[00:23:41] GH:** Well, it could. It's really easy to do a port. The amount of code you have to write to port tinygrad to new accelerator is 100x less than the code you have to write for PyTorch. I mean, 100.

**[00:23:54] SPEAKER:** Okay, tell me about the state of AI generally, which I think factors into why you're doing tinygrad. I guess, what are the hard parts in AI? Is it software related? Is it hardware related? Is it engineering related? What's hard there?

**[00:24:11] GH:** Well, everything's hard and everything's easy, right? If you look, this is an Eliezer Yudkowsky thing. I love him. If you look at factoring algorithms, would you rather have a 1970s factoring algorithm on modern hardware, or a modern factoring algorithm on 1970s hardware?

**[00:24:28] SPEAKER:** Former.

**[00:24:30] GH:** So, you think you'd rather have a 1970s factoring algorithm on modern hardware?

**[00:24:36] SPEAKER:** That's what I would think. Yes. But am I wrong on that?

**[00:24:38] GH:** You are. But not by much. Not by much. Factoring algorithms have come a – back in the ‘70s, you’re using maybe something like Polar Drone, or you use a fancy general number field C. I think the exact same thing is true for SAT solvers as well. Would you rather have a 70 SAT solver on modern hardware or modern SAT solver. And modern SAT solvers are much, much better. In fact, I just, Art of Computer Programming 4B and half the books about SAT solvers. And I'm like, “Oh, this is great.” SAT solver is this amazing thing to think about.

So, yeah, I think like factoring in SAT like these hard problems. SAT, we know is in NP. I don't think factoring is in NP. But the algorithms have progressed faster than the hardware, but not by much. So, I think the same thing is going to be true about AI. It's interesting that like, they're pretty close and a software has a shorter ripple time. So, I think you can make progress on software faster than you can hardware but not by too much.

**[00:25:35] SPEAKER:** Yeah.

**[00:25:36] GH:** Yeah, I think the same thing is true about AI. I think that that the software is bad, the hardware is bad, the algorithms are bad. But what do you mean by bad? Sure, just because they'll be better in the future, it doesn't mean they're terrible. They're a lot better than what we had, “Oh, man, trying to do this stuff. You tried to this stuff 20 years ago? What are you?” GPUs are like not general purpose. You have no frameworks. You’re hand coding all your forwards and backwards passes. Your algorithms are stochastic gradient descent. You don't have Adam, you don't have batch norm, you don't have drop out.

**[00:26:13] SPEAKER:** What does the development process look like when you're sort of iterating on AI? I sort of think of like, build up this model, and then go set it to train for hours and hours and hours. Is that true? Or is that like an outdated – what is iteration look like?

**[00:26:26] GH:** That's true. What do we do at Comma? We spend most of our time writing tests. Now, of course –

**[00:26:32] SPEAKER:** How do you test that? What do the tests look like?

**[00:26:35] GH:** All sorts of things. So, tests kind of fall into, we can say two buckets, right? There are tests of overall performance, are there are tests of like specific performance, right? Maybe the same things are like, maybe you can say their integration tests and unit tests. It's the same basic ideas. A lot of Comma – like I say these things. And these are like, one of the things is, I just got bored saying the same things over and over again at Comma. And I hope the that people work, they don't feel the same way. I hope that people – I don't know. This is me. This is something about me, but like, we've been saying, I mean, I feel like we figured it all out. This is now a question of doing it. But a lot of the stuff is like testing 2.0 methodologies, right? How can we say one model is better than another model, right? With two models, how do we say which one is going to drive the car better?

Well, if you're in a very early stage, you can actually go out and try and both on the car. But this doesn't work, as your models get better. If you have models, that one model makes a mistake every 1,000 hours, and one model makes mistake every 10,000 hours, well, you're not going to ask that right?

**[00:27:45] SPEAKER:** That’s what you’re going to do.

**[00:27:46] GH:** Yeah, you got to drive for 1,000 hours, right? And that's okay. So, you have to come up with better testing methodologies that can discover these things without needing a car.

**[00:27:56] SPEAKER:** I mean, what are those look like? Do you run it just through simulator? And certainly, mistakes it makes or what's that look like?

**[00:28:03] GH:** So, simulators are one tool. We use tons of different tools. We definitely have the simulators called analyze LAT. Analyze LAT runs it on 2,000 segments and tells you how it does.

**[00:28:19] SPEAKER:** How do you – maybe this is a dumb question. But like you have all these simulators, but you also have a bunch of people with Comma installed in their cars. Do those match up pretty well, where like, “Hey, if you run through a simulator, this is a big improvement. We've moved it from 1,000 to 10,000 errors.” Do you find that's true when it gets out in the real world?

**[00:28:36] GH:** For the most part, yes. And then whenever we are surprised, whenever we are surprised whenever we find that we ship something to the real world, and we put these on our master branch, and there's maybe like 10% of the Comma fleet running the master branch, and they're on Discord. And they're like, “Bros, I mean, everything's great. But every time I go through an intersection, it slams on the brakes.” Well, that's not a real example. But a lot of the stuff is way more subtle events, like exit splitting is something. Like take exits on the highway when it's not supposed to.

So, whenever we get these complaints, we figure out how to validate them in simulation, and then we add that to our test suite. So, whenever we're surprised, now we have a new test, right? And now it has to pass this entire complicated test suite before we ship it. So, you have the whole problem. The actual iteration on the model is not most of the work, right? It’s the model, it doesn't matter. It really doesn't matter. It's about testing.

**[00:29:42] SPEAKER:** How is the AI hardware market right now? I know like Nvidia, the big one in the room, but we also got GPUs. We got Apple now. What does the hardware market look like?

**[00:29:53] GH:** I'll say one quick thing about testing and I'll go into that. My point is like if you have a test that's good enough, well, that kind of is all you need. Because, okay, I'll just do a random search across a model space, right? If I have a test that can always tell me which model is better, okay, ship it out train 10,000 models and tell me which of this one is. Great. And I can do automatic search or I can search in a space if I have a good evaluation function. That good evaluation function is not hard at all.

The hardware market, we love low stock price Nvidia. High stock price Nvidia is like, “Oh, we’re going to gouge people and it's going to be terrible and we're not worried about anybody.” But low stock price Nvidia is like we're going to open source our kernel driver. The 4090s are amazing. Oh, man, they built an amazing chip. Well, a few things about the 4090s are so incredible. So, one of the biggest problems in the 3090 was it only had six megs of L2 cache. The A100 had 40 megs.

So, a lot of the benefits from A100 actually just came from the larger cache size. But the 4090 has 72 megs. So, they did not memory bandwidth that much, but they added this big L2 cache, which effectively gives you a lot more memory bandwidth, especially at the size of models we're training. So yeah, that's really exciting. I mean, the chip is five nanometer, four nanometer. It's a massive chip too. Things like 600 square millimeters. So yeah, I mean, these chips are amazing, and they're not that overpriced, like a 4090, you can get for $2,000.

Right now, Nvidia is okay. TVs are useless, right? And you know why TVs are useless? Because what am I going to get? I'm going to go Google Cloud?

**[00:31:40] SPEAKER:** Yeah. Is it still a licensing thing? I know, I was listening to you on WEX about that, where they sort of have restrictive things that you can do with TPUs. Is that still true?

**[00:31:47] GH:** No. A friend of mine at Google is like George's can't possibly be true. But he's actually is true, and they fixed it. The idea that like, the TPU is like if you train self-driving cars on our TPU –the term was only meant to apply. I think, someone from Google really has to adjust this. But that term was only meant to apply to their like, machine learning hosted thing, they have like access to a cloud suite of Google models. If you train your own model, and you rent bare metal hardware, it doesn't apply. But I don't know. I just don’t want to deal with any of this.

**[00:32:20] SPEAKER:** It's scary enough and uncertain enough that you just don't want it.

**[00:32:24] GH:** Yeah. We build the compute cluster in our new office. It’s so good. It's hundreds of GPUs, thousands of cores, petabytes of storage, and it's like ours. You can go touch it.

**[00:32:38] SPEAKER:** So, all your training done on prem, in your office?

**[00:32:41] GH:** Yeah. All trainings on prem. All are like, you can download the Comma app. It's a cloud dash cam, that's all Azure, but for training, GPUs have – the cloud is a good deal, as long as all you want are CPU cores, right? The minute you need storage or GPUs, the clouds a huge rip off, right? And what is training, storage and GPUs.

**[00:33:07] SPEAKER:** Yup, yup, that's interesting. My brother-in-law, he does machine learning stuff and he's found the same thing where he's just building his own boxes and running server play spaces, because AWS is just killing him on that.

**[00:33:19] GH:** If you can pay for low priority CPU cores, most of your bandwidth is ingress, then the cloud is a phenomenal deal. Love the cloud. But yeah, the minute you need anything specialized for storage, or egress bandwidth.

**[00:33:37] SPEAKER:** Yeah, exactly. They'll kill you there. Tiny also talks about training on the edge in that aspect. Why is training on the edge so important and necessary?

**[00:33:47] GH:** So, this is a shift that that I see happening. It's going to be still 5 to 10 years. But the fact that there's any distinction between training and inference, I think is absurd. Humans don't have this distinction. Humans train all the time. Your systems are always going to be rigid and inflexible, if they're not lifelong learners. If they're not constantly training, if there's like, "Okay, we're just going to freeze the weight to this thing." Imagine a turn on like, let's say it's like a blind turn, right? And let's say when you're looking at it, it's kind of unpredictable. It's not exactly what they want to do. So, it's not that Comma is not going to be able to do this turn. But every time it takes this turn, it's going to need to like – it's going to miss it and then overcorrect. And that turns out to be what humans do every time the first time they see the turn. But the key is, it's only the first time. You drive that commute every day. Okay, maybe the second day, you mess it up as well. But by the third day, "Oh, yeah, I remember this." And that's because you're a lifelong learner. So, I think that we're going to have to move towards lifelong learning on device and that's one of the things that I really want tinygrad to support.

**[00:35:02] SPEAKER:** That feed back into like the main centralized model that then gets pushed out to sort of mastermind everyone? Or does it stay on that device, some of that learning?

**[00:35:12] GH:** Well, yeah. So, we really have some on device learning, right? We have something called params learner. Actually, we just shipped or we're shipping a big upgrade to it for Black Friday. Params learner can – okay, so here's an interesting fact. The wear on your tires is a factor in how much torque you have to put on the wheel. And that's pretty obviously true when you think about it. If you have fresh new tires, well, they grip the road a lot better. So, you turn the wheel, you put less torque on the wheel to make a turn, right? But if they don't grip that well, just imagine like some tire slide, right?

So, one of the things we learn on device, this thing called params learner, is the tire stiffness and the tire wear, my vehicle dynamics is not great. But my point is, we have this on device learner. But this on device learner is learning a bunch of hand coded features. Now, the real thing that you want is to get rid of those hand coded features. You want to say, “Okay, here's the desired path. Here's the actual path.” All right, you got a reward function to match those things. Now, make that stable, it's hard.

**[00:36:17] SPEAKER:** If you're doing a bunch of on device learning too, will it sort of pick up even my preferences or different things that are different from person to person? If I'm a little more cautious on some of these things or –

**[00:36:30] GH:** It can. It depends what the reward function is, right? So right now, the params learner, which again, it is a learner. It's using the derivative and it's updating a set of parameters based on the derivative of the error. At some point, when is it – it's not a deep learning, it's learning, shall learn. So, the reward function there is only it's only trained when the computer's engaged. The weights are not updated when you're driving the car. So, if you want the weights to update when you drive the car, see, the eventual final like model for Comma is disengagement is negative reward. You taking over, you correcting the car is actually the only thing you need in a reward function to solve self-driving cars, it's really cool. But that kind of stuff probably works better at Fleetscale. To Try to learn that from you, maybe that gets even fancier than what I'm asking for now. But yes, eventually, eventually, sure you can have a conversation with it like, “Bro, slow down.” “I got you. Don't worry. I got a fancy LLM modeling.” Bro, slowdown and betting in the vector space.

**[00:37:48] SPEAKER: W**hat's holding back stuff at the edge right now? Is it software? Is it hardware? Is it tinygrad isn’t complete yet?

**[00:37:58] GH:** It's not one bottleneck. Ask the same question, like what's holding back factoring? What's holding back SAT? That's hardware and software, its algorithms. It's all these things. So yeah, maybe you can really break it down into those three things. You can break it down into hardware, which is straight up, like how much flops can we shove through this thing. We can break it down to like software. And when I say software, I mean infrastructure. Infrastructurally, how can we maximally utilize the flops? How easy is it to code things in this? How easy is it to debug in these sorts of spaces? How easy is it update window accelerators come out? So, that's like the infrastructure component. That's what tinygrad is. Tinygrad is not a hardware company. Tinygrad is an infrastructure company.

And then there's algorithms, right? And then algorithms are – Comma is also really an infrastructure company. We try to say – we have a saying at Comma like don't do research, because you are not going to compete on the research with these overfunded AI labs, Google and Facebook and stuff. Read their papers, implement it, apply it and ship it. So yeah, then there's this quadrant of algorithms, right? And algorithms are currently dominated by like deep minds made a whole bunch of advances, OpenAI, the big labs, and where you get the advances there.

**[00:39:13] SPEAKER:** Yeah. Speaking of the big AI labs, there's a lot of, I would say excitement recently about language, art stuff, with Dall-E's stable diffusion, things like that. How does that interact with self-driving? Are those totally different domains? Are they helping each other?

**[00:39:31] GH:** I mean, there's a few cool things. I spent a weekend – that's one of my better streams. I spent a weekend implementing stable diffusion in tinygrad. It's just cool to go through, and like see what all the things are. I don't know. I'm not impressed with these things. I think they're like, I've seen classical systems from 20 years ago that could do similar stuff. Some cool things they can do, but they're not superhuman, in the way that I'm – one of the AI things that – and I'm not like a total AI hater. I'm just not that impressed with this compared to something like MuZero or AlphaGo. MuZero is one of my favorite papers to come out of all time. They're like, you can from nothing with little access to a simulator, learn how to win chess, Atari, and shogi in the same, like architecture. Now, this is incredible. Another paper came out, Iris. At first, I read it and I'm like, "Oh, this is like our research writeup." But it is a research writeup. They didn't come up with new ideas. But they did it well. It's Atari games, which learn a simulator and then train in that simulator. And this is cool stuff. Right? Whereas like, yes, great your language model, look, it understands poetry. Who cares? You can you can give these people a Markov model. Tell them it's some fancy AI. You've got such deep insights.

**[00:40:59] SPEAKER:** Yeah. All right. That's good. I want to switch back a little bit. You've, you've hired some people. I know at Comma you said, it's hard to get into Comma, it's hard to stay at Comma. What types of skills or backgrounds have you seen that work well in AI? Work well at Comma? Do you need this hard math background? Do you need good software engineering background? What do you see?

**[00:41:24] GH:** Our interviews are – I'll just ask you a bunch of rapid-fire questions. And they're not like hard questions. I ask you like, what's the complexity of matrix multiplication? What's the speed of light? Basic things that you should know, or even stuff like what's 13 in binary, right? And if you don't, like right away, say, 1101, whatever that is, it's like a fluency with computers. And that's what gets you hired, right? You've spent a lot of time doing this. When you're in the shower, you're thinking about this stuff, right? I hate the word, but it's kind of passion. Someone that can say all they want, "I'm passionate." Anyone who writes in their email, I'm passionate, immediate, immediate. But you show me your passion when you get excited when I ask you questions like that. Do you know a huge breadth of knowledge on our infrastructure interview? And I can give away the whole interview, here's the question. I asked you, I typed google.com into a web browser, what happens?

Now, I like the people right away who were talking to me about keyboard interrupts, and how that ends up in the browser, right? And then, there's some people who like – and then it does a DNS lookup. Okay, what's a DNS lookup? Well, just like a DNS server, and it connects to it. Okay, how does it connect to it? You better talk to me about port 53 and sys calls. You don't know that stuff like that, you can work at Comma, but that's not a good job.

**[00:42:58] SPEAKER:** I think of you as a passionate person, and just like, extremely high energy, and just – I'm just kind of amazed that you can go on Twitch and live stream for 10 hours or so. I don't know you're getting older. You're still young, right? You're 33. But do you sense a change in your stamina or passion? Are you still like just as fresh as you always am?

**[00:43:20] GH:** No. I think this stuff starts to decline, maybe in your 40s. I got many years. But you look at like John Carmacl, that guy has got more stamina than I do, right? You're watching pretty much – people are like, "How do you stream on Twitch for 10 hours?" I don't understand. Do people have jobs where they work eight hours a day? I'm just baffled by this. Sure, if I stream on Twitch for 48 hours, you're like, "How does this guy do it?" 10 hours isn't that much, bro. And it's not like I'm spending 10 hours every minute working. You watch me take breaks and go off on Internet tangents. Yeah, I would probably say I get six hours of work done in those 10 hours. Is that not what people do at work?

**[00:43:58] SPEAKER:** Yeah, not anymore. I guess. We've all gotten fat, and spoiled, and lazy.

**[00:44:03] GH:** Yeah. I think that's just like weird to me. Yes, if you work at Comma, you're probably going to be in the office 10 hours a day. You probably work six of them. That's just the most normal job to me.

**[00:44:16] SPEAKER:** I forgot to ask you. You mentioned what gets people hired. What gets people fired? Because you've said you've hired people as well.

**[00:44:23] GH:** Gets people fired. A whole bunch of things. I mean, the easiest way to get fired is just kind of stopped doing work.

**[00:44:34] SPEAKER:** Do you see that a lot?

**[00:44:37] GH:** Do we see that a lot? I mean, yeah. People, not everyone is – and I respect the people who quit. I respect the people who say, “Look, I'm not really feeling this anymore. I quit.” Some people are not that self-aware or are nervous and then don't quit and then I'll fire you. So, there's that. Soft skills can get people fired. You're just arguing with people and you don’t like to work with, they get fired. Or you can't ship things. You've worked at Comma for six months, and you haven't shipped anything. That's not good. You don’t ship things. I'm not saying, like, we value super teammates who communicate well. I'm not saying that. But if you're pissing people off, you're going to get fired, right? Normal. Don't piss people off, ship things, and work hard, and that's it.

**[00:45:31] SPEAKER:** Yup. All right, I want to talk to you about some tech current events. Let's talk crypto a little bit just because we've got the FTX, SPF. I don't know where you are in crypto. I know you've done a little bit of stuff, including like, helping some team that you met at a restaurant and you solved some mega problem for them. I guess, where are you on crypto? What do you think on the future of it?

**[00:45:54] GH:** Look, I I've been around crypto ever since I was buying drugs on the Silk Road. It is exactly what it is. I don't have strong opinions on it. Is it going to go up or down? I think it's the same as every other asset. It's realistic. Everything is priced exactly where it should be. I don't know if any magical insights into the market. I almost bought a house when I was 19 and my boss at Google talked me out of it. In retrospect, I should have bought that house. It was 24,000, now it's worth like 2 million. But my boss at Google said this to me. He’s like, “George, you don't know anything about the real estate market. You don't know if it's going to go up or down. You don't know if it's a good buy or not. And a house is a very illiquid asset.” I’m like, “You know what, you're exactly right.”

The same thing is true about crypto. I don't have any special insights. Is it going to go up? Is it going to go down? Is this coin good? Is this coin bad? I mean, I could probably talk about some that are bad, but I could talk about some that are obvious scam. So, you see something like Luna and Terra. You’re like, “George, you’re only predicting the one that went under. Look at synthetics.” You look at something like synthetics, and it's trading these synthetic stocks backed by the snacks token, what's the snacks token? What's a token that you can stake and gives you 18% returns? This is a Ponzi, right? I don't mean to call it synthetics, right? I'm not saying that's even bad. I'm not necessarily saying Ponzis are bad. I'm just saying this is only sustainable in bull mark. This fundamentally can't work.

So, you can do some, like economics, one on one like that. Like, yes, a Ponzi scheme is defined as if new cash flows are paying out old money, new money is paying out old money, and usually companies, this can be cryptos, it could be anything, but just be aware of Ponzis, and then you don't ask the question, right? Like a dumb investor says, “That's a Ponzi. I'm not going to invest.” The smart investor says, “Am I early?” So yeah, that's kind of my take on that.

**[00:47:55] SPEAKER:** Apart from the investment side of it, you obviously did interesting work with that one company and helped that. Does the crypto space interest you technologically? Or are you just like, “Not nearly as cool as AI.”

**[00:48:09] GH:** I mean, I'm pretty happy with what I did. You can go look at Can. It's good open source code. I embedded a MIPS processor on chain that lets you verifiably run programs on the Ethereum chain. And you're like, “Oh, well, that's going to run really slowly”, but not exactly. Because you don't need to run the program, you don't need to run all N steps of the program. It's played as a multi-party game. So, two people can – it's like a challenge response kind of thing. And with a challenge response, you can get this O of N down to O of login, because there's only one step in which they disagree. You can treat it as one step, where they disagree, and you only ever have to play that one step out on chain.

So, you can get two parties to agree verifiably on compute, full and complete, kind of MIPS processor, and its login steps. Which yeah, it's cool technology. You know what? The repo has been sitting there. They're not doing anything with it, because decentralization doesn't matter. Decentralization theater matters and this is the problem with crypto. Don't actually make it decentralized. God, that's hard. We're not front about how it's decentralized. And then get people to invest in your Ponzi.

**[00:49:19] SPEAKER:** Yeah, get out why you can. All right, in terms of other current events, we have – one of your competitors, I would say, in self-driving. Elon, buying Twitter and that whole thing going on. Any thoughts on – I know, you're not a huge Twitter user, but what do you think of that?

**[00:49:36] GH:** You know what? I came back on Twitter because Elon bought it. You know what? I will say this, I'm not even going to say whether it's good or bad, it's entertaining. Elon's making Twitter fun. I'm like, excited to be on Twitter. What's going to happen tomorrow? It's cool. There's like a seriousness about these social networks that I hated that. It wasn't like that. The Internet didn't used to be like that. No one took like MySpace seriously. Then people started taking this stuff to put your address information. That's just rude. I can't take this. Elon buys Twitter and he's like, "YOLO, man." So, I'm entertained and I support entertainment.

**[00:50:20] SPEAKER:** Yup. Interested in some of Elon's tech side, SpaceX and Starlink and do those technological problems excite you?

**[00:50:30] GH:** Internet SpaceX, I got offered a job. Again, I kind of felt like SpaceX was at a stage where similar kind of where Comma is now, is when I was at SpaceX, maybe almost 10 years ago now. Where there's a lot of smart people working on a really hard problem, and it's just a slow and steady grind, right? There's not any magic I can do or big architectural things I can fix. I mean, here we are years later, they're landing rockets. It's amazing.

**[00:51:03] SPEAKER:** Yeah. Will you ever work at a big company again, SpaceX, Google, Facebook for different periods of time? Is that just not your vibe?

**[00:51:15] GH:** I mean, I think those three companies are all very different. I think, again, SpaceX is kind of like – the companies that I do work for are companies who are over capitalizing on paying me way too much money to solve problems that they should have fixed internally. So yeah, I don't know if I can do that at SpaceX. Google, sad. Facebook or aka Meta. I listened to John Carmack and he told Zuckerberg, he's like, "Look, make me the czar of the metaverse, I will make it good." Zuckerberg said, "Well, that's not really how we do things here." He left. How stupid. You're spending God knows what on horizons world? It's terrible. Carmack would have showed – he could have hit something cool, right?

So, again, you're literally like one of the best guys in computer gaming history wants to come around your thing, and you're like, "No, that's not how we do things. We do things by committee. We got to make sure everyone's represented." As long as companies are still on this, like, we got to make sure everyone's represented. Come on. I'm not trying to exclude people, but like, there's good ideas and bad ideas. I'm not trying to say anything about who comes up with the ideas. I don't care. But until we can say that this is smart, and this is done, it's not a place for me, right? If we're going to treat every idea like it's equally good. all right, you're going to end up with committee mesh. You're going to end up with –

**[00:52:50] SPEAKER:** Yeah. What about Carmack? What do you think of his thoughts on AI and agents and different things like that?

**[00:52:59] GH:** I mean, I'm excited to see what he does. I had met him maybe three years ago at a conference thing. Yes, he's very, very, very passionate talking about stuff, and I think the ideas are like, they're undeveloped, they're undeveloped. So, it remains to see, for me, say anything, I'd have to hear concretely what the idea is.

**[00:53:22] SPEAKER:** Yeah. Would you be interested in working on that sort of thing, or you just need to hear more on that?

**[00:53:28] GH:** I mean, I want to think if it's plausible. I got to hear way more about the idea and things like – I don't know. I don't know what it is public. I'm not going to say anything. But like, when he releases something, I'll evaluate it on its merits.

**[00:53:42] SPEAKER:** Yup. What motivates you in work? Is it purely like, man, what's the coolest biggest problem I can work on. You've talked about other companies throwing money at you, like, what do you care about?

**[00:53:53] GH:** I mean, yeah. I like the idea of, like, if people want to hire me as a contractor to solve something. I like solving problems. I like feeling like, "Man, I wrote like good code and this is good and I I did this better", not just in like a structural way, in like, your team wrote something 10x to complicate it. I came in and just rewrote it and doing like this is better.

**[00:54:25] SPEAKER:** Cool. I love it. George, this has been a fun interview. It's been fun to catch up on self-driving, AI and tiny and all that you're working on. If people want to find you, what's the best way for them to come find what you're up to?

**[00:54:40] GH:** Github.com/geohot. That's really what I'm doing. Everyone, you can find anywhere else. But look at my commands. Some are good, some are bad. You can see what I'm doing.

**[00:54:50] SPEAKER:** Yup. All right. That's good. I know. Geohot's blog. Also on GitHub. GitHub pages somewhere. Your Twitch stream, all that stuff. So, people want to see you in action, they can get a little bit of that stuff. But always got the cool stuff. You got it set up. So, George, thanks for coming on Software Engineering Daily.

**[00:55:07] GH:** Cool. Thanks for having me.

[END]