Jensen Huang: NVIDIA - The $4 Trillion Company & the AI Revolution | Lex Fridman Podcast #494

145 MIN · YOUTUBE · HTTPS://WWW.YOUTUBE.COM/WATCH?V=VIF8NQCJVF0
https://www.youtube.com/watch?v=vif8NQcjVf0

- The following is a conversation with Jensen Huang, CEO of NVIDIA, one of the most important and influential companies in the history of human civilization. NVIDIA is the engine powering the AI revolution, and a lot of its success can be directly attributed to Jensen's sheer force of will and his many brilliant bets and decisions as a leader, engineer, and innovator. This is Lex Fridman Podcast. And now dear friends, here's Jensen Huang.

You've propelled NVIDIA into a new era in AI, moving beyond its focus on chip scale design to now rack scale design. And I think it's fair to say that winning for NVIDIA for a long time used to be about building the best GPU possible, and you still do, but now you've expanded that to extreme co-design of GPU, CPU memory, networking, storage, power cooling, software, the rack itself, the pod that you've announced, and even the data center. So let's talk about extreme co-design. What is the hardest part of co-designing system with that many complex components and design variables?

- Yeah, thanks for that question. So first of all, the reason why extreme co-design is necessary is because the problem no longer fits inside one computer to be accelerated by one GPU. The problem that you're trying to solve is you would like to go faster than the number of computers that you add. So you added you know, 10,000 computers, but you would like it to go a million times faster. Then all of a sudden you have to take the algorithm, you have to break up the algorithm, you have to refactor it, you have to shard the pipeline, you have to shard the data, you have to shard the model. Now all of a sudden when you distribute the problem this way, not just scaling up the problem, but you're distributing the problem, then everything gets in the way. This is the Amdahl's Law problem where the amount of

speed up you have for something depends on how much of the
total workload it is. And so if computation represents 50% of the problem,
and I sped up computation infinitely like a million times, you know, I only sped up the total
workload by a factor of two.

Now all of a sudden, not only
do you have to distribute a computation, you have to, you
know, shard the pipeline somehow. You also have to solve
the networking problem because you've got all of these
computers are all connected together. And so distributed computing
at the scale that we do, the CPU is a problem, the GPU is a
problem, the networking is a problem, the switching is a problem. And
distributing the workload across all these computers is a problem.

It's just a massively complex computer
science problem. And so we just gotta bring every technology to bear.
Otherwise, we scale up linearly or we scale up based on the capabilities
of Moore's Law, which has largely slowed because
Dennard scaling has slowed. - I'm sure there's trade-offs there.
Plus you have a complete disparate disciplines here. I'm sure you have specialists
in each one of these high bandwidth memory, the network and the NVLink,
the NICs, the optics and the copper that you're doing, the power delivery, the
cooling, all of that. I mean, there's like world experts in each of those. How do you get
'em in a room together to figure out- - That's why my staff is so large.

- What's the process—can you take me through
the process of the specialists and the generalists? Like how do you put
together the rack when you know the s- the set of things you have
to shove into a rack together? Like what does that process look
like of designing it all together? - Yeah. There's the first question,
which is: what is extreme co-design? You're, you, we're optimizing
across the entire stack of software from architectures to chips, to
systems, to system software, to the algorithms, to the applications. That's
one layer. The second thing that you and I just talked about is goes
beyond CPUs and GPUs and networking chips and scale up
switches and scale out switches.

And then of course, you gotta
include power and cooling and all of that because, you know, all
these computers are extremely, extremely power hungry. They do
a lot of work and they're very energy efficient, but they in

aggregate still consume a lot of power. And so that's one. The
first question is, what is it? The second question is, why is it, and
we just spoke about the reason, you know you want to distribute the
workload so that you can exceed the benefit of just increasing
the number of computers.

And then the third question is,
how is it, how do you do it? And that's the, that's kind
of the miracle of this company. You know, when you're designing a
computer, you have to have operating system of computers. When you're
designing a company, you should first think about what is it that you
want the company to produce. You know, I see a lot of companies organization
charts, and they all look the same. Hamburger organization charts,
soft organization charts, and car company organization charts.
They all look the same.

And it doesn't make any sense to me. You
know, the goal of a company is to be the company is to be the machinery, the
mechanism, the system that produces the output. And that output is the
product that we like to create. It is also designed, the architecture
of the company should reflect the environment by which it
exists. It almost indirectly says what you should do with the
organization. My direct staff is 60 people. You know, I don't have one-on-ones with 'em because it's
impossible. You can't have, you can't have 60 people on your staff if you're, you
know, gonna get work done and- - So you still have 60 reports.
You still have across- - More, yeah.

- More. And most stars at least
have a foot in engineering. - Almost all of them. There's experts in memory, there's
experts in CPUs, there's experts in optical. All, all— - That's incredible. - Yeah, GPUs and— Architecture,
algorithms, design, um— - So, you constantly have an eye on the
entire stack, and you're having to, like, intense discussions about the
designer of the entire stack? - And no conversation is ever one
person. That's why I don't do one-on-ones. We present a problem and all of us attack it. You know,
because we're doing extreme co-design.

And literally, the company is doing
extreme co-design all the time. - So, even if you're talking about a
particular component, like cooling, networking, everybody's listening in? - Yeah, exactly. - And they can contribute, "Well, this
doesn't work for the power distribution. This doesn't-" - Exactly. - "... This doesn't work for the

memory. This doesn't work for this." - Exactly. And whoever wants to tune out, tune out. You know what I'm saying? And the reason for that is because the people who are on the staff, they know when to pay attention. There's supposed... You know, it's something they could have contributed to, they didn't contribute to, "I'm going to call them out." You know?

And so, "Hey, come on, let's get in here." - So, as you mentioned, NVIDIA is this company that's adapting to the environment. So, at which point can you say, did the environment change and began adapting sort of secretly- ... in the early days from GPU for gaming, maybe the early deep learning revolution to we're now going to start thinking of it as an AI factory? What does NVIDIA do? It produces AI, let's build a factory that makes AI.

- Uh, I could, I c- you, you could- I could reason through what just systematically. We started out as a, as an accelerator company. But the problem with accelerators is that the application domain's too narrow. It has the benefit of being incredibly optimized for the job. You know, any specialist has that benefit. The problem with intense specialization is that, of course, your market reach is narrower, but that's, that's even fine. The problem is, the market size also dictates your R&D capacity. And your R&D capacity ultimately dictates the influence and impact that you can possibly have in computing. And so, when we first started out as an accelerator, very specific accelerator, we always knew that had- that was going to be our first step. We had to find a way to become accelerated computing. But the problem is, when you become a computing company, it's too general purpose and it takes away from your specialization. The tur- I connected two words that are actually have fundamental tension. The better computing company we become, the worse we became as a specialist. The more of a specialist, the less capacity we have to do overall computing. And so, that... And I connected those two words together on purpose, that the company has to find that really narrow path, step by step by step, to expand our aperture of computing, but not give up on the most important specialization that we had. Okay, so the first step that we took beyond acceleration was, we invented a programmable pixel shader.

So, that was the first step towards programmability. You know, it was our first journey towards moving into the world of computing. The second thing that we did was we created we put FP32 into our shaders. That FP32 step, IEEE-compatible FP32, was a huge step in the direction of computing. It was the reason why all of the people who were working on, on stream

processors and, you know, other types of data flow processors
discovered us. And they said, "Hey, all of a sudden, you know, we might be able to use
this GPU that's incredibly computationally intensive, and it's now, you
know, compliant with IEEE."

I can take my software that I was
writing, you know, previously on CPUs, and I can, you know, see about,
you know, using the GPU for that. And which led us to
create, put C on top of FP32, what's called, we
call Cg. The Cg path took us to eventually CUDA.
CUDA, step by step by step We... Well, putting CUDA
on GeForce, that was a strategic decision that was very,
very hard to do, because it cost the company enormous amounts of our
profits, and we couldn't afford it at the time. But we did it anyways
because we wanted to be a computing company. A computing
company has a computing architecture. A computing architecture
has to be compatible across all of the chips that we build.

- Can you take me through that decision? So,
putting CUDA on GeForce, could not afford to do? Can you explain that decision? Why,
why boldly choose to do that anyway? Can you explain that decision? - Yeah, excellent. That was, that was
the first... I would say that that was the first strategic decision that is
as close to an existential threat. - For people who don't know, it
turned out to be, spoiler alert, one of the most incredibly
brilliant decisions ever made by a company. So, CUDA turned out to be an incredible foundation for computation
in this AI infrastructure world. So- - Thank you - ... just setting the context. It
turned out to be a good decision.

- Yeah, it turned out to have been a good
decision. I think the... So, here's the way it went. So, we invented this
thing called CUDA, and It expanded the aperture of applications that, that we can accelerate with our
accelerator. The question is, how do we, how do we attract developers to CUDA? Because a computing platform is all
about developers. And developers don't come to a computing platform
just because, you know, it could perform something interesting. They
come to a computing platform because the install base is large.

Because a developer, like anybody
else, wants to develop software that reaches a lot of people. So,
the install base is, in fact, the single most important

part of an architecture. The architecture could attract enormous
amounts of criticism. For example, no architecture has ever
attracted more criticism than the x86.... you know, as a less than, less than
elegant architecture, but yet it is the defining architecture of today. It
gives you an example that in fact so many RISC architectures which
were beautifully architected, incredibly well-designed by some of the
brightest computer scientists in the world, largely failed.
And so I've given you two examples where one is, you know,
one is elegant, the other one's barely aesthetic, and so yet x86
survived and the reason for- - Install base is everything.

- Install base defines an
architecture. Not... Everything else is secondary, okay? And so there were
other architectures at the time. CUDA came out, OpenCL was here. There were...
You know, there's several other competing architectures. But the thing that...
The decision that we made that was good was we said, "Hey, look,
ultimately it's about, Install base and what is the
best way we could get a new computing architecture into the world?" By that timeframe, GeForce had
become successful. We were already selling millions and millions
of GeForce GPUs a year, and we said, "You know, we ought
to put CUDA on GeForce and put it into every single PC
whether customers use it or not, and use it as a starting point of
cultivating our install base."

Meanwhile, we'll go and
attract developers, and we went to universities
and wrote books and taught classes and put CUDA everywhere.
And eventually people discover... And at the time, the PC was the primary
computing vehicle. There was no cloud, and we could put a
supercomputer in the hands of every researcher in school, every scientist, you
know, every engineering school, every... or every student in school, and
eventually something amazing will happen. Well, the problem was
CUDA increased our cost of that GPU, which is a consumer
product, so tremendously, it completely consumed all of the
company's gross profit dollars.

And so at the time, the company was probably,
you know, worth, I don't know, at the time, eight... Was it like $8 billion or something? Like
six, $7 billion or something like that. After we launched CUDA, I recognized
that it was going to add so much cost, but it was something we believed in. You know, our market cap went down to like
one and a half billion dollars. And so we were down, we were down

there for a while and we clawed our way way back
slowly, but we carried CUDA on GeForce. I always say
that NVIDIA is the house that GeForce built, because it was
GeForce that took CUDA out to everybody. Researchers, scientists, they
discovered CUDA on GeForce because they were all, you know... Many of 'em were
gamers. Many of them built their own PCs anyways. In a university
lab, many of them built clusters themselves, you know,
using PC components. And, and so that, you know, that's
kind of how we got going.

- And then that became the platform and the
foundation for the deep learning revolution. - That was also another great,
great observation. Yeah. - That existential moment, do you
remember... Like, what were those meetings like? What were those discussions like,
deciding as a company, risking everything? - Well I had to make it clear to the
board what we're trying to do, and the management team knew our gross
margins were gonna get crushed. So you could imagine a world
where GeForce would carry the burden of CUDA and
none of the gamers would appreciate it and none of the gamers
would pay for it. You know, they only pay certain price and it doesn't matter what
your cost is. And so the... You know, we, we increased our cost
by 50% and that con- consumed... And we were a 35%
gross margin company, and so it, it was a... It was quite a
difficult decision to make. But you could imagine that someday this would
go into workstations and it would go into supercomputers and, and in those segments,
maybe we can capture more margin.

so you could, you could reason your
way into being able to afford this, But it still took... It took a decade. - But that, but that's more of, like, conversation
with the board convincing them, but you psychologically- ... as NVIDIA's
continued to make bold bets that predict the future, and in part,
especially now, define the future. So I'm almost looking for wisdom about how you're able to make those decisions,
to make leaps- ... like that as a company.

- Well, first of all I'm informed
by a lot of curiosity. At some point, there's a
reasoning system that, that convinces me, so clearly this outcome
will happen. That this will happen. And so I believe it in my mind,
and when I believe it in my mind, you know, you know how it is.
You manifest a future and that future is so convincing,

there's no way it won't happen.

There's a lot of suffering in between, but
you've gotta believe what you believe. - So you envision the future- ... and you essentially, from a sort of
engineering perspective, manifest it? - Yeah. And you, you reason about how
to get there. You reason about why it, it must exist. And, you
know, I reason... We all reason it here. The management team would
reason about it. All the people that I... We spend a lot of time reasoning about
it. The thing, the thing that... The next part of it is probably a
skill thing, which is, you know, oftentimes in
leadership the leadership stays quiet or they learn about something,
and then they do some manifesto, and it's a brand-new year, and somehow
at the end of the year, next year, we're gonna have a brand-new plan.
Big huge layoff this way, big huge organization change this way, new mission statement... brand new logos
you know, that kind of stuff.

Um, we've just never, I never do things
that way. When I learn about something and it's starting to influence how
I think, I'll make it very clear to everybody near me that, you
know, this, this is interesting. This is going to make a difference.
This is going to impact that. And I reason about things step by step by
step. Oftentimes, I've already made up my mind, but I'll take
every possible opportunity, external information, new
insights, new discoveries, New engineering, you
know, revelations, new milestones developed, I'll take
those opportunities and I'll use it to shape everybody else's belief system. And I'm doing
that literally every single day.

I'm doing that with my board, I'm doing that
with my management team, I'm doing that with my employees. I'm trying to shape their belief systems such that
when I come
the day I say, "Hey, let's buy Mellanox," it's completely obvious to everybody
that we absolutely should. On the day that I said, "Hey guys, let's go all in on deep learning,"
and let me tell you why. I've already been laying down the
bricks to different organizations inside the company. Every
organization and everybody, many of the people might
have heard everything.

Most of the company hears,
of course, pieces of it. And on the day that I announce it, everybody's kind of bought
in to many pieces of it. And in a lot of ways, I like to

announce these things, and I imagine, that the employees are kind of saying, "You know, Jensen, what took you so long?" And in fact, I've been shaping their belief system for some time, and therefore leadership. Sometimes it looks like you're leading from behind, but you've been shaping their, you know, to the point where on the day that I declared it, 100% buy-in. But that's what you want. You want to bring everybody along. You know, otherwise, we announce something about deep learning and everybody goes, "What are you talking about?" You know, you announce something about let's go all in on this thing, and your management team, your board, your employees, your customers, they're kind of like, "Where's this coming from?

You know, this is insane." And so, so GTC effect, if you go back in time, you look at the keynotes, I'm also shaping the belief system of my partners in the industry and I'm using that to shape, you know, the belief system of my own employees. And so by the time that I announce something, like for example, we just announced Grok. We've been late... I've been talking about the stepping stones for two and a half years.

You just go back and go, "Oh my gosh, they've been talking about it for two and a half years." And so I've been laying the foundation step by step by step, so when the time comes you announce it, everybody's saying, "You know, what took you so long?" - But it's not just inside the company. You're shaping the landscape, the broader global landscape of innovation. Like, putting those ideas out there, you really are manifesting reality. - We don't build computers. We actually don't build clouds. We don't... As it turns out, we're a computing platform company. And so nobody can buy anything from us. That's the weird thing. You know, we vertically design, vertically integrate to design and optimize, but then we open up the entire platform at every single layer to be integrated into other companies' products and services and clouds and supercomputers and OEM computers and so the amazing thing is, I can't do what I do without having convinced them first. And so most of GTC is about manifesting a future that by the time that we... My product is ready, they're going, "What took you so long?"

- Yeah. So one of the things you've been a believer for a long time is scaling laws, broadly defined. So are you still a believer in the scaling laws? - Yeah, yeah. Yeah, we have more scaling laws now. - So I think you've outlined four of them with pre-training, post-training, test time, and agentic scaling. What do you

think, when you think about the future, deep future and the near-term future, what are the blockers that you're most concerned about that keep you up at night that you have to overcome in order to keep scaling?

- Well, we can go back and reflect on what people thought were blockers. So in the beginning, we were the first... The pre-training scaling law. You know, people thought well rightfully so, that the amount of data that we have, high-quality data that we have will limit the intelligence that we achieve. And that scaling law was an important, very important scaling law. The larger the model, the correspondingly more data results in a better... With a results in a smarter AI. And so that was pre-training. And Ilya Sutskever, Ilya said, "We're out of data," or something like that. "Pre-training is over," or something like that. The industry panicked, you know, that this is the end of AI. And of course, of course that, that's obviously not true. We're gonna keep on scaling the amount of data that we have to, to train with. A lot of that data is probably gonna be synthetic, and that also confused people, you know? And what people don't realize is they've kind of forgotten that most of the data that, that we are training that we teach each other with, inform each other with, is synthetic. You know, I...

It's synthetic because it didn't come out of nature. You created it. I'm consuming it. I modify it, augment it, I regenerate it, somebody else consumes it. And so, so we've now reached a level where AI is able to take ground truth, augment it..... Enhance it, synthetically generate an enormous amount of data. And that part of post-training continues to scale, and so the amount of data that we could use that is human generated will be smaller, and smaller, and smaller.

The amount of data that we use to train model, is going to continue to scale to the point where we're no longer limited... Training is no longer limited by... Data is now limited by compute. And the reason for that is most of the data is synthetic. Then the next phase is test time, and I still remember people telling me that, "Inference? Oh, yeah, that's easy. Pre-training, that's hard." These are giant systems that people are talking about. Inference must be easy. And so inference chips are gonna be little tiny chips, and ... you know, they're not, they're not like NVIDIA's chips. Oh, those are gonna be complicated and expensive, and, you know, we could make... And this is- in the future, inference is gonna be the

biggest market, and it's gonna be easy, and we're gonna commoditize it. You know, everybody can build their own chips. And, and that was always illogical to me because inference is thinking, and I think thinking is hard. Thinking is way harder than reading.

You know, pre-training is just memorization and generalization, you know, and looking for patterns in relationships. You're reading and reading, versus thinking, reasoning, solving problems, taking unexplored experiences, new experiences, and breaking it down into... Decomposing it into, you know, solvable pieces that we then go off, either through first principle reasoning, or, you know, through previous examples, prior experiences. You know, or just uh, exploration and search and, you know, trying different things. And that whole process of, of test time scaling, Inference, is really about thinking. And it's about reasoning, it's about planning, it's about search, it's about... And so how could that possibly be compute light? And we were absolutely right about that. You know, so test time scaling is intensely compute intensive.

Then the question is, okay, now we're at inference and we're at test time scaling, what's beyond that? Well, obviously, we have now created, you know, one agentic person, and that one agentic person has a large language model that we've now now, you know, developed. But during test time, that agentic system goes off and does research and bangs on databases, and it goes out and, you know, uses tools, and one of the most important things it does is spins off and spawns off a whole bunch of sub-agents. Which means we're now creating large teams.

It's so much easier to scale NVIDIA by hiring more employees than it is to scale myself. And so the next scaling law is the agentic scaling law. It's kind of like multiplying AI. Multiplying AI, we could spin off agents as fast as you want to spin off agents. And so, you know, I... You know, I have four scaling laws. And as we use the agentic systems, they're gonna create a lot more data, they're gonna create a lot of experiences. Some of it we're gonna say, "Wow, this is really good. We ought to memorize this."

That data set then comes all the way back to pre-training. We memorize and generalize it. We then refine it and fine-tune it back into post-training. Then

we enhance it even more with test time, you know, and the
agents, agentic systems, you know, put it out to the
industry. And so this loop, this cycle, is gonna go on and on and on.
It kinda comes down to basically intelligence is gonna scale by
one thing, and that's compute.

- But there's a tricky thing there that you
have to anticipate and predict, which is some of these components, it requires different kind of hardware to
really do it optimally. So you have to anticipate where the AI innovation's
going to lead. For example, a mixture of- - Perfect - ... experts with sparsity. - Perfect. - With hardware, you can't just
pivot on a week's notice. You have to anticipate what that's
going to look like. It has some- - So good - ... that's so scary and
difficult to do, right?

- For example, These AI model architectures
are being invented about once every six months. Right? And system architectures
and hardware architectures kind of every three years. And so you need to anticipate
what likely is going to happen, you know, two, three years from now. And there's a couple ways that you could do that. First of
all, we could do research internally ourselves, and that's one of the reasons why we have basic
research, we have applied research.

We create our own models.
And so we have hands-on life experience right here. This is part of
the co-design that I'm talking about. We're also the only AI company in the world that
works with literally every AI company in the world. And so to the extent that we
can, we try to get a sense of what are the challenges
that people are experiencing. - So you're listening to the whispers
across the industry, the AI labs. - That's right. You got to listen and
learn from everybody. And have a...

And then the last part is to
have an architecture that's flexible, that can adapt and move
with the wind. And one of the benefits of, of CUDA is that
it's, you know, on the one hand, an incredible accelerator. On the
other hand, it's really flexible. And so that balance,
incredible balance between specialization, otherwise we
can't accelerate the CPU, versus generalization, so that we
can adapt with changing algorithms, that's really, really important.
That's the reason why CUDA has been so resilient on the

one hand, and yet we continue to enhance it. We're
at CUDA 13.2, and so we're evolving the architecture
so fast that we can stay with, you know, with
the modern algorithms.

For example... When mixtures
of experts came out, That's the reason why we had
NVLink 72 instead of NVLink eight. We could now take an entire four
trillion, 10 trillion parameter model and put it in one computing domain
as if it's running on one GPU. Um, people probably didn't
notice, I said it, but if you look at the architecture of the Grace Blackwell racks,
it was completely focused on doing one thing, processing the LLM.

All of a sudden, one year later,
you're looking at a Vera Rubin rack. It has storage accelerators. It has
this incredible new CPU called Vera. It has Vera Rubin and NVLink 72 to run the LLMs. It also has this new
additional rack called Rock. And so this entire rack
system is completely different than the previous
one, and it's got all these new components in it. And the reason for that
is because the last one was designed to run MoE large language models, inference.
And this one is to run agents and agents bang on tools, and- - Obviously, the design of the system had
to have been done before Claude Code, Codex, OpenClaw. So you were
anticipating the future, essentially.

And that comes from what? From the whispers,
from understanding what all the state- - No - ... of the art is about? - No, it's easier than that. You just reason
about it. First of all, you just reason. no matter, no matter what
happens, at some point in order for that large language model to
be a digital worker... Let's just, let's just use that metaphor. Let's
say that we want the LLM to be a digital worker. What does that have to do?

It has to access ground truth.
That's our file system. It has to be able to do research. It doesn't know
everything. We don't have... And I don't wanna wait until this AI becomes,
you know, universally smart about everything, past,
present, and future before I make it useful. And so therefore, I
might as well let it go do research. It's obviously, if it wants to
help me, it's gotta use my tools. You know, a lot of people would say,
"You know AI is gonna completely destroy software. We don't need software anymore.
We don't even need tools anymore." That's ridiculous. Let's use the...
Let's use a thought experiment.

And you could just sit there,
enjoy a glass of whiskey, and, and think about all these things, and it
would become completely obvious. Like, if I were to create the most amazing- the most amazing agent that we can
imagine in the next 10 years. Let's say it'd be a humanoid robot. If
that humanoid robot were to be created, is it more likely that the
humanoid robot comes into my house and uses the tools that I have to
do the work that it needs to do?

Or does this hand turns
into a 10-pound hammer in one instance, turns into a
scalpel in another instance, and in order to boil water, it
beams, you know, microwaves out of its fingers? You know, or is it more
likely just to use a microwave, you know? And the first time it
goes up to the microwave, it probably doesn't know how to use it.
But that's okay. It's connected to the internet. It reads the manual of this microwave, reads it,
instantly becomes an expert.

And so it uses it. And
so I think the... I just described, in fact, almost all
of the properties of OpenClaw. You know, that it's gonna use tools, that it's
gonna access files, it's gonna be able to do research. It has I/O subsystem.
And when you're done reasoning through it, reasoning
about it in that way, Then you say, "Oh, my gosh, the impact
to the future of computing is deeply profound." And the reason for that is, I
think we've just reinvented the computer.

And then now you say, "Okay, when did
we reason about that? When did we reason about OpenClaw?" If you take the
OpenClaw schematic that I used at GTC, you'll find it two years ago. Literally,
two years ago at GTC, I was talking about agentic systems that exactly
reflect OpenClaw today. And, of course, the confluence of many things had to happen. First
of all, we needed Claude and GPT and, you know, all of these
models to reach a level of capability. So their innovation and
their breakthroughs and their continued advances was really important.

And then, of course, somebody had
to create an open source, you know project that was sufficiently robust,
you know, and sufficiently complete and that we can all put to
work. And I think OpenClaw did for, did for agentic
systems what ChatGPT did for generative systems. And I just

think it's a very big deal. - Yeah, it's a really special moment.
I'm not exactly sure why it captured so much of the
world's attention, but it did, more than Claude Code and Codex and so on.

- Because consumers could reach it. - Sure, yeah. But there's also
so much of this is vibes. And Peter, I had a
podcast with him, he's a wonderful human being. So part of it is
also the humans that represent the thing. - Yeah, no doubt. - Part of it is memes and the– 'Cause
we're all trying to figure it out. There's really serious and
complicated security concerns about when you have such powerful
technology, how do you hand over your data so they can do useful stuff? But then
there's scary things associated with that.

And we, as a civilization, as individual
people and as a civilization, figuring out how to find that right balance. - Yeah, we jumped on it right away and we
sent a bunch of security experts this way. And we did this thing called
OpenShell. It's already been integrated into, into OpenClaw. - And NVIDIA put forward NemoClaw. - Yep, exactly. - They install super easy. It
makes sure that it's secure. - We give you two out of three
rights. Agentic systems can access sensitive information, it can
execute code, and it can communicate externally. We could keep things safe
if we gave you two out of those three capabilities at any
time, but not all three.

And out of those two out of three
capabilities, we also give you access control based on whatever rights that
you're given by enterprise. And then we connect it to a policy engine
that all these enterprises already have. And so we're going to try to do our best
to help OpenClaw become a better claw. - So you eloquently explained
how we have a long history of blockers that we thought were going to be blockers,
and we overcame them. But now looking into the future, what do you think might be the
blockers now that it's clear that agents will be everywhere?

So obviously we're going to need
compute. So what is going to be the blocker for that scaling? - Power is a concern, but it's not the
only concern. But that's the reason why we're pushing so hard on extreme co-design,
so that we can improve the tokens per second per watt orders of magnitude every single year. And so in
the last 10 years, Moore's Law would have progressed computing
about 100 times in the last 10 years. We progressed
and scaled up computing by a million times in the last 10 years.

And so we're gonna keep on doing that through extreme co-design. So energy efficiency, perf per watt, completely affects the revenues of a company. It affects the revenues of a factory. And we're just going to push that to the limit so that we can keep on driving token costs down as fast as we can. You know, the our computer price is going up, but our token generation effectiveness is going up so much faster that token cost is coming down. It's just coming down an order of magnitude every year.

- So power, that's an interesting one. So the way to try to get around the power blocker is to try to, with the tokens per second per watt, try to make it more and more efficient. Of course, there's the question of how do we get more power. - We should also get more power. - That's a really complicated one. You've talked about small modular nuclear power plants. There's all kinds of ideas for energy. How much does it keep you up at night? The bottlenecks in the supply chain of AI, like ASML with EUV lithography machines, TSMC with advanced packaging like CoWoS, and SK Hynix with the high bandwidth memory?

- All the time, and we're working on it all the time. No company in history has ever grown at a scale that we're growing while accelerating that growth. It's incredible. And it's hard for people to even understand this. In the overall world of AI computing, we're increasing share. And so supply chain, upstream and downstream, are really important to us. I spend a lot of time informing all the CEOs that I work with, what are the dynamics that's going to cause, The growth to continue or even accelerate? It's part of the reasons why to the entire right-hand side of me were CEOs of practically the entire IT industry upstream and practically the entire infrastructure industry downstream.

And they were all... There were several hundred CEOs. And I don't think there's ever been keynotes where several hundred CEOs show up. And part of it is, I'm telling them about our business condition now. I'm telling them about the growth drivers in the very near future and what's happening. And I'm also describing where are we going to go next so that they could use all of this information and all of the dynamics that are here to inform how they want to invest. And so I inform them that way like I inform my own employees.

And then of course, then
I make trips out to them and make sure that, "Hey, listen, I
want you to know this quarter, this coming year, this next year, these
things are going to happen." And if you look at the
CEOs of the DRAM industry, The number one DRAM in the world was
DDR memory for CPUs in data centers. About three years ago, I was able to
convince several of the CEOs that even though at the time HBM memory was
used quite scarcely, you know, and, and barely by supercomputers
that this was going to be a mainstream memory for data centers in
the future. And at first it sounded ridiculous, but several of the
CEOs believed me and decided to invest in building HBM memories. Another
memory was rather odd to put into a data center, is the low power memories
that we use for cell phones.

And we wanted them to adapt them
for supercomputers in the data center. And they go, "Cell phone
memory for supercomputers?" And I explained to them why.
Well, look at these two memories, LPDDR5, HBM4. The volumes are so incredible. All three of them had record years in history, and these are, these are 45-year companies. And so, you know, I... That's part of my job, is
to inform and shape, inspire, you know.

- So you're not just manifesting
the future and maybe inspiring NVIDIA, the different
engineers of the company, you're, you're manifesting the supply chain of the
future. So you're having conversations with TSMC, with ASML. - Upstream, downstream. - Upstream, downstream. So that's the thing. - GEV, Caterpillar. Yeah, that's downstream
from us. Yeah, yeah, there you go. - Yeah, the whole thing. I
mean, but that's so... There's so much incredibly
difficult engineering that happens in the entire semiconductor
industry, and it just feels scary how intricate the supply chain
is, how many components there are, but it works somehow.

- Exactly, the deep science. The
deep engineering, the incredible manufacturing, and so much of the
manufacturing is already robotics, but we have a couple of hundred
suppliers that contribute the technology that goes into
our 1.3 million component rack. Each rack is 1.3, one and a half million
components. There are 200 suppliers across the Vera Rubin rack. - So it's interesting that you don't list that as the
thing that keeps you up at night in the list of blockers.

- But I'm doing, I'm doing all
the things necessary to- - Okay - ... yeah, see? I can go to sleep because I checked it
off. I said, okay, you know, I go, I yeah, I can go to sleep and I go,
well, let's see, what re- let's reason about this. What's important for us? Um, because okay, let's
reason about this. Because we changed the system architecture
from the original DGX-I that you remembered to NVLink-72
rack scale computing- ... what's gonna... What does
that, what does that mean?

What does that mean to software? What does
that mean to engineering? What does that mean to how we design and test? How,
and what does that mean to the supply chain? Well, one of the things that
it meant was we moved supercomputer, supercomputer integration
at the data center into supercomputer manufacturing in the
supply chain. If you're doing that, you also have to recognize
you're gonna move one... And if, if you're, if you're, you
know, total footprint of whatever data center you're gonna build,
let's say you would like to have, you know, 50 gigawatts of supercomputers
that are running simultaneously, and it takes one week
to manufacture that 50 gigawatts of supercomputers,
then each week in the supply chain, the supercomputers are
gonna need a gigawatt of power. And so, so we're gonna need the supply chain
to increase the amount of power it has to build, test, to build and test the supercomputers in the supply
chain before I ship it.

Well, NVLink-72 literally builds
supercomputers in the supply chain and ships 'em two, three
tons at a time per rack. It used to be—they used to come in
parts and we used to assemble 'em inside the data center. But that's impossible
now because NVLink-72 is so dense. And so that's an example. And I
would have to go and to, you know, I've... Fly into the supply chain, go meet
my partners saying, "Hey," I said, "guess what? So here's what I'm going to do
with... This is the way we used to build our DGXs. We're gonna build them this way.
This is gonna be so much better because we're going to need 'em for inference."
The market for inference is, you know, coming. The inflection point for inference
is coming. It's gonna be a big market.

And so I first explain to them what's
going on, why it's gonna happen, and then I ask 'em to make several billion
dollars of capital investments each. And because they, you
know, they trust me and I'm very respectful of 'em, and I give 'em
every opportunity to question me and I spend time to explain things to people

and I reason about it. I draw on pictures and I reason about it in first
principles. And by the time I'm done with them, they know what to do.

- So it's a lot of it is about relationships
and building a shared view of the future. Uh, but do you worry about certain
bottlenecks? I mean, what are the biggest bottlenecks in the supply chain?
Are you, are you worried about ASML's EUV tooling? Are you, are you worried about
the packaging, CoWoS packaging of TSMC, about how fast it could
scale? Like you said, you're not only growing incredibly fast,
you're accelerating your growth. So it feels like everybody in the
supply chain, and those are certainly bottlenecks, would have to scale
up. Are you having conversations with them, like, how can you scale up
faster? Do you worry about it?

- No. - Okay. - Because, because I told 'em what
I needed. They understood what I need. They told me what they're gonna go do,
and I believe them what they're going to do. - Interesting. That's great to hear. So maybe if we can
just linger on the power for a little bit. What are your hopes for how
to solve the energy problem? - One of the areas, Lex, that I'm that I would love, I would love us to
talk about and just get the message out, you know our power grid is designed for the worst case condition
with some margin. Well, 99% of the time we're nowhere near the worst
case condition because the worst case condition is a few days in the winter,
a few days in the summer, and extreme weather. Most
of the time we're nowhere near the worst case condition and
we're probably running around, call it 60% of peak. And so 99% of
the time, our power grid has excess power, and they're just
sitting idle, but they have to be there sitting idle because just in case,
when the time comes, hospitals have to be powered and, you know, infrastructure has to be
powered and airports have to run and so on and so forth. And so the question that I
have is whether we could go and, Help them understand and
create contractual agreements and design computer architecture systems,
data centers, such that when they need, The maximum power for
infrastructure in society, that the data centers would get less.

But that's in a very rare instance anyways.
And during that time, we either have a backup generator for that little part of
it, or we just have our computers shift the workload somewhere else, or we
have the computers just run slower. You know, we could
degrade our performance, reduce our power consumption and provide for a, you know, slightly longer
latency response, you know, when somebody asks for, you know, asks
for an answer. And so I think that that way of using computers,
of building data centers, instead of expecting 100% uptime-...

and these contracts that are
really, really quite rigorous, it's putting a lot of pressure on the grid
to be able to... Now, they're gonna have to increase from their maximum.
I just wanna use their excess. It's just sitting there. - Yeah, that's not talked about
enough. So what's stopping there? Is it regulation? Is it bureaucracy? - I think it's a three-way
problem. It starts with the end customer. The end customer
puts requirements on the data centers that they can never not be available, okay? So that the
end customer expects perfection.

Now, in order to deliver that
perfection, you need a combination of backup generators and your grid power supplier to deliver on perfection.
And
so everybody's gotta have six nines. Well, I think first of all, right now, we ought to have everybody understand
that when the customer asks for these things, you have somebody in
your data center operations team disconnected from the CEO. I
bet the CEO doesn't know this. I'm gonna talk to all the CEOs.
The CEOs are probably not paying any attention to the
contracts that are being signed, and so everybody wants to sign the best
contract, of course. And they go down to cloud service providers, and the
contract, the... The two contract negotiators that are... You,
I could just see them now.

You know, negotiating these multi-year
contracts. Both sides want, you know, the best contract. As a result, the CSPs then have to go down to the
utilities, and they expect the nine, the six nines. And so I think
the first thing is just make sure that all of the customers,
the CEOs and the customers realize what they're asking for. Now, the second thing is we have to build
data centers that gracefully degrade. And so if the power, if the utility, if the
grid tells us, "Listen, we're gonna have to back you down to about 80%," we're
gonna say, "That's no problem at all."

We're just gonna move our workload around.
We're gonna make sure that data's never lost, but we can reduce the
computing rate and use less energy. The quality of service degrades
a little bit. For the critical workloads, I shift that
somewhere else right away so I don't have that problem, and so,
you know, whoever, whichever data center still has 100% uptime, and so... - How difficult of an engineering
problem is that,
that smart, dynamic allocation of power in a data center?

- As soon as you could specify, you

could engineer it. U- beautifully put. So long as it obeys the laws of physics
on first principles, I think we're good. - What was the third thing
you were mentioning? Um... - So the second thing is the data
centers. And the third thing is we need the utilities to also recognize
that this is an opportunity- ... and instead of saying, "Look
it's gonna take me five years to increase my grid capability," uh, if you, if you have, if you're willing to
take power of this level of guarantee, I can make them available for you
next month and at this price.

And so if utilities also offered more
segments of power delivery promises, then I think everybody will
figure out what to do with it. Yeah, but there's just way too much waste in
the grid right now. We should go after it. - Uh, you've highly lauded Elon and xAI's accomplishment in Memphis, in
building Colossus supercomputer, probably in record time in just
four months. It's now at 200,000 GPUs and growing very quickly. Is
there something that you could speak to the understand about his approach
that's instructive to, broadly to all the data center creators that's that enabled that kind of accomplishment? His approach to engineering, his approach to the whole management of
construction, everything?

- First of all, Elon is deep in
so many different topics. Um, Yet he's also a really
good systems thinker. And so he's able to think through
multiple disciplines, and, and he obviously pushes things, questions everything, where they're,
number one, is it necessary? Number two, does it have to be done this way?
And then numbers, you know, does it have, does it have to take this long? And,
and so, so he, he has, he has the a- he has the ability to question everything, To the point where everything is down to its minimal amount that's necessary, you can't take
anything else out. And yet the, the necessary capabilities
of the product remains, you know? And so he's, he is as minimalist
as you could possibly imagine, and he does it at a system scale. I
think... I also love the fact that he is he is represented. He is
present at the point of action.

You know, he'll just go there. If
there's a problem, he'll just go there and then, "Show me the problem." You know,
when you do all of this in combination, you overcome a lot of previous,
"This is just the way we do it." "Um, you know, I'm waiting for them. Uh," You know, I mean, it's just,
everybody has a lot of excuses. And so, and then the last thing is
when you act personally with so much urgency it causes everybody
else to act with urgency, you know? And every supplier has a lot

of customers going on. Every supplier has a lot of projects going on, and he made it his... He makes it his business that he's the top priority of everybody else's, you know, projects. And so he does that by demonstrating it.

- Yeah, I've been in a bunch of those meetings. It's just, it's fun to watch, 'cause really, not enough people ask the question like, "Okay, so, can this be done a lot faster, and how? Why does it have to take this long?" - Yeah, right. - And then in the... That becomes an engineering question often. And yes, I think when you get the ground truth of actually... I remember, one of the times I was hanging out with him, he literally is going through the entire process of how to plug in cables into a rack. He's, he's working with an engineer on the ground that's doing that task, and he's just trying to understand what does that process look like so it can be less error-prone. And just building up that intuition from every single task involved in, putting together a data center— ...you start to immediately get a sense at the detailed scale and at the broad systems scale of where the inefficiencies are, and so you can make it more and more and more efficient.

Plus you have the big hammer of being able to say, "Let's do it totally different-" - Yeah. That's right. - "... and remove all possible blockers." - That's right. - Is there parallels in the NVIDIA Extreme Systems co-design approach that you see in the way Elon approaches systems engineering? - Well, first of all, co-design is an ultimate systems engineering problem. And so we approach, we approach the work that we do from that first, from that The other thing that we do and this is, this is a philosophy, a thought, a state of mind, I guess, a method, That I started 30 years ago, and it's called the speed of light. The speed of light is not just about the speed. The speed of light is, is my shorthand for what's what's the limit of what physics can do. And so every single, everything, everything that we do is compared against the speed of light. Memory speed, Math speed, power, cost, time, effort, number of people, manufacturing cycle time. And when you think about latency versus throughput, When you think about cost versus throughput, cost versus capacity, all of these things, You test against the speed of light to achieve all of these different constraints separately. And then when you consider it together, you know you have to make compromises because a system that achieves extremely low latency versus a cheap, a system that achieves very high throughput are architected fundamentally differently.

But you want to know what's the speed of light of a system that achieves high throughput, what's

the speed of light of a system that achieves low latency? And then when you think about the total system, you can make trade-offs. And so I force everybody to think about what's the, what the first- the first principles, the limits- ... the physical limits, For everything before we, you know, before we do anything. And we test everything against that. And so that's a good frame of mind.

I don't love the other methods, which is continuous improvement. The problem with continuous improvement, it... First of all, you should engineer something from first principles at the speed, you know, with speed of light thinking. Limit it only by physical limits, and physics limits. And after that, of course you would improve it over time. Um, but I don't like going into a problem and somebody says, "Hey, you know, it takes 74 days to do this today-" "... Right now. And we can do it for you in 72 days."

You know, I'd rather strip it all back to zero- ... and say, "First of all, explain to me why 74 days in the first place. And let's note, let's think about what's possible today. And if I were to build it completely from scratch, you know, how long would it take?" Oftentimes, you'd be surprised. It might come to six days. Now, the rest of the six days, the 74, could be very well-reasoned and compromises, and, you know, cost reductions, and all kinds of different things. But at least you know what they are.

And then now that you know that six days is possible, then the conversation from 74 to six, surprisingly much more effective. - In such incredibly complex systems that you're working with, is simplicity sometimes a good heuristic to reach for? I mean, if I can just... I mean, the pod, the Vera Rubin pod that you announced is just incredible. We're talking about seven chips, seven chip types, five purpose-built rack types, 40 racks, 1.2 quadrillion transistors, nearly 20,000 NVIDIA dies, over 1,100 Rubin GPUs, 60 exaflops, 10 petabytes per second of scale bandwidth.

That's all just one... - That's just one pod. - That's just one pod. - Yeah, that's just one pod. - I mean, in- ... so you have the... And then even the NVL 72 rack alone is 1.3 million components, 1300 chips, 4,000 pods crammed into a single 19-inch wide rack. - And Lex, we're probably gonna have to crank out about 200 of these pods a week, just to put it in perspective. - The amount of different components,

I suppose simplicity is impossible, but is that a metric that you kind of
reach for in trying to design things?

- You know, the phrase, the phrase
that I use most often is, we- we need things to be as complex as
necessary, but as simple as possible. And so the question is, is
all that complexity there necessary? And we ought to test for that. And we got to challenge that. And then
after that, everything else above it, you know, is gratuitous. - But it's still almost incredible. Semiconductor
industry broadly, but what NVIDIA is doing, some of the greatest engineering in
history. So these systems are just truly, truly marvels of engineering.

- It is the most complex computer
the world has ever made. - Yeah, the engineering teams, I mean- ... I don't, it's not a competition, but I
don't know. If it was like an Olympics of engineering teams, I mean, TSMC does
incredible engineering. Like I said, ASML at every scale, but NVIDIA is
gonna give them a run for their money. Just incredible, incredible teams. - Well, it's gold medalists in every
single, in every single sport, all assembled right here.

- And have to work together. And report directly to you. This is
wonderful. You recently traveled to China. so it's interesting to ask
you China's been incredibly successful in building up its
technology sector. What do you understand about how
China's able to, over the past 10 years, build so many
incredible world-class companies, world-class engineering teams, and
just this technology ecosystem- ... that produces so many
incredible products?

- A whole bunch of reasons for, well, first
of all, let's start with some facts. 50% of the world's AI researchers
are Chinese, plus or minus, and they're mostly in China still. We have many of them here, but
there's amazing researchers still in China. They, their tech industry showed
up at precisely the right time. At the time of the mobile
cloud era their way of contributing was software, and so this
is a country's incredible science and math. Really well-educated kids.

Their tech industry was created
during the era of software. They're very comfortable
with modern software. China is not one giant economic country. It's got many provinces
and cities with mayors all competing with each other. That's
the reason why there's so many EV companies. That's the reason why there's so many
AI companies. That's the reason why there's so many, every company you could imagine,

they all create some of them. And as a result, they have insane
competition internally.

And, you know, what remains
is an incredible company. They also have a social culture where
it's family first, friends second, and company third. And so, the amount
of conversation that goes back and forth between... They're essentially
open source all the time. So the fact that they contribute
more to open source is so sensible because they're probably, "What are we
protecting?" You know, my engineers, their brothers are in that company, their
friends are in that company, and they're all schoolmates. You know, the schoolmate
concept. There's a, you know, one schoolmate, you're brother
for life. And so they, they share knowledge very, very quickly.

And so there's no sense keeping technology
hidden. You might as well put it on open source. And so the open
source community then amplifies, accelerates the innovation process. So you
get this rapid, incredible great talent, rapid innovation because of open source
and just, you know, the nature of friends, and insane competition. Among comp- among the company, what emerges
is incredible stuff. And so this is the fastest innovating country in the world today, and this is something
that has everything that, everything that I've just said is fundamental to just
how the kids were grown, the fact that they have excellent
education, the fact that they, parents want them to do well in school, the
fact that they, their culture is that way. These are, you know, these are
just the thing about their country, and they showed up at precisely the time
when technology is going through that exponential.

- Plus culturally, it's pretty
cool to be an engineer. It connects to all the components
that you're mentioning... - It's a builder nation. - It's a builder nation. - Yeah, it's a builder nation.
Our country's leaders, incredible, but they're mostly
lawyers. Their country's leaders, and because we're, they're trying
to keep us safe, rule of law, governing, their country was built
out of poverty. And so most of their leaders are incredible
engineers. Some of the brightest minds.

- To take a small tangent, because you
mentioned open source, I have to go to Perplexity here, who you
have been a fan of a long time. - Love it, yeah. - And thank you for releasing
open source Nemotron 3 Super, which you can also use inside

Perplexity to look stuff up. Now, which is 120 billion parameter open weight MoE model. What's your vision with open source? So you mentioned China with DeepSeek and MiniMax, with all these companies really pushing forward the open source AI movement, and NVIDIA is really leading the way in close to state-of-the-art open source LLMs. What's your vision there?

- First, if we're gonna be a great AI computing company, we have to understand how AI models are evolving. One of the things that I love about Nemotron 3 is it's not a, just a pure transformer model, it's transformer and SSMs. And we were early in, Developing the conditional GANs, which, that progressive GANs, which led step-by-step to diffusion. And so, The fact that we're doing basic research in model architecture and in different domains gives us visibility into, you know, what kind of computing systems would do a good job for future models. And so it is part of our extreme co-design strategy. Second, um, I think we rightfully recognize that on the one hand, we want world-class models as products, and they should be proprietary.

On the other hand, we also want AI to diffuse into every industry and every country, every researcher, every student. And if everything is proprietary, it's hard to do research and it's hard to innovate on top of, around, with. And so...Open source is fundamentally necessary for many industries to join the AI revolution. NVIDIA has the scale and we have the motives to not only skills, scale, and motivation to build and continue to build these AI models for as long as we shall live.

And so therefore, we ought to do that. We can open up, we can activate every industry, every researcher, you know, every country to be able to join the AI revolution. There's the third reason, which is from that, to recognizing that AI is not just language. These AIs will likely use tools and models and sub-agents that were trained on other modalities of information. Maybe it's biology or chemistry or you know, laws of physics, or you know, fluids and thermodynamics, and not all of it is in language structure.

And so somebody has to go make sure that weather prediction, biology, AI, AI for biology, physical AI, all of that stuff stays, can be pushed to the limits and pushed to the frontier. We don't build cars, but we wanna make sure every car company has access to great models. We don't, discover drugs, but I wanna make

sure that Lilly has the world's best biology AI systems, so that they can
go use it for discovering drugs. And so these three
fundamental reasons, both in, recognizing that AI is not just
language, that AI is really broad, that we wanna engage everybody into the
world of AI, and then also co-design of AI.

- Well, I have to say, once again,
thank you for open sourcing, really truly open sourcing
Nemotron 3 and ... - Yeah, I appreciate you were saying that. We open sourced
the models, we open sourced the weights, we open sourced the data, we open sourced how we
created it. Yeah, it's pretty amazing. - Uh, it's really, it's really incredible. You're originally from Taiwan and
have a close relationship with TSMC. So I have to ask, TSMC I think also is a legendary company in terms of
the engineering teams, in terms of the incredible engineering
work that they do. What what do you understand about TSMC
culture and their approach that explains how they're able
to achieve this singular unmatched success in everything
they're doing with semiconductors?

- You know, first of all, the deepest
misunderstanding about TSMC is that their technology is all they have.
That somehow they have a really great transistor, and if somebody shows
up another transistor, game over. It's the technology and of course, you know, I don't mean
just the transistor, the metallization systems, the packaging, the 3D
packaging, the silicon photonics, the, you know, all of the technology that
they have. That technology is really what makes the company special. Their
technology makes the company special.

But their ability to orchestrate the demands, the dynamic demands of
hundreds of companies in the world as they're moving up, shifting out, you know, increasing, decreasing,
push, pushing out, pulling in changing from customer to customer,
Wafer starting, wafer stopping, Emergency wafer starts,
you know, all of this dynamics of the world's
complexity as the world is shape-shifting all the
time, and somehow they're running a factory with high
throughput, high yields, really great costs,
excellent customer service.

They take their work, they take their
promises seriously. They, when your wafer, because they know that they're helping
you run your company, when the wafers, when the wafers were promised to show up, the
wafers show up, you know, so that you could run your company appropriately.

And so their system, their manufacturing system is completely
miraculous, I would say. Then the second thing is their culture.
This culture is simultaneously, Technology focused on one
hand, advancing technology, simultaneously customer service
oriented on the other hand. A lot of companies are very customer service
oriented, but they're not very technology excellent. They're not
at the bleeding edge of technology.

There are a lot of companies who are tech, at the bleeding
edge of technology, but they're not the best customer service oriented company. And so it
just depends on somehow they've, they've balanced these two and
they're world-class at both. And then probably the third thing
is the technology that I most value in them that they
created this, you know, this, Intangible called trust. I trust them to
put my company on top of them. That's a very big deal.

- When they trust, I mean, there's a really close
relationship there that you've established, and that trust is established based on many years
of performance, but there's human relationships involved there as well. - Three decades, I don't know how many
tens, hundreds of billions of dollars of business we've done through
them, and we don't have a contract. That's pretty great. - Amazing. Okay, there's this story ... ... That in 2013, the
founders of TSMC, Morris Chang offered you the chance to
become TSMC's chief executive, And you said you already had
a job. Is this story true?

- Story is true. I didn't dismiss it. Um but
I was deeply honored and, and of course, of course uh, I knew then as I
know now, TSMC is one of the most consequential companies in history. And, and Morris is one of the
highest regarded executive and, and business and personal friend
that I've had in my life. And, um ...Uh, for him to ask is, uh um,
I was humbled and, and really honored.

But, but the work that I'm doing here
is really important, and I've seen, you know, in my mind's eye, what NVIDIA was going to be and what the
impact that we could have. And uh, it was really important work. And it's my responsibility, you know,
my sole responsibility to make this happen. And so I uh, I
declined it, You know, not, not because it wasn't
an incredible offer. It's an unbelievable offer but,
but I simply couldn't take it.

- I think NVIDIA, both
NVIDIA and TSMC are two of the greatest companies in the history
of human civilization. And running either one, I'm sure, is incredibly
complicated effort and takes... You have to truly be all in. Uh, everybody at every scale, not just
at the CEO level. Everybody is really truly all in- - Yeah. Yeah, no doubt - ... To, to accomplish
this kind of complexity. - So now I can help both companies. - Exactly. So NVIDIA is now the most
valuable company in the world.

I have to ask, what is the
NVIDIA's biggest moat, as the folks in the tech sector say? The edge you have that protects
you from the competition. - Our single most important
property as a company is the install base of
our computing platform. Our single most important thing
today is our, is the install base of CUDA. Now, the reason why, uh, 20 years ago, of course,
there was no install base.

But what makes... And if
somebody came up with a GUDA or TUDA it wouldn't
make any difference at all. And the reason for
that is because it's never been just about the technology.
The technology, of course, was incredible visionary. But it's the fact
that the company was dedicated to it, stuck with it, expanded its reach. Um, it wasn't three people that
made CUDA successful. It was 43,000 people that made CUDA
successful. And the several million developers that believed in us, That trusted that we were going to
continue to make CUDA 1, 2, 3, 13, that they decided to port and dedicate
their software on top of it, their mountain of software on top of
it. And so the install base is the number one most
important advantage.

That install base, when you amplify
it with the velocity of our execution at the scale that
we're talking about, no company in history had
ever built systems of this complexity, period. And then to
build it once a year is impossible. And that velocity combined
with the install base, in the developer's mind, you just
go now, take the developer's mind. From the developer's
perspective, if I support CUDA, tomorrow it'll be 10 times better. I
just have to wait six months on average.

Not only that, if I develop it on CUDA,
I reach a few hundred million people, computers. I'm in every cloud, I'm in every

computer company, I'm in every single industry, I'm in every single country. So if I create an open source package and I put it on CUDA first, I get these both attributes simultaneously. And not only that, I trust 100% that NVIDIA is going to keep CUDA around and maintain it and improve it and keep optimizing the libraries for as long as they shall live.

You could take that to the bank, and that last part, trust. You put all that stuff together, if I were a developer today, I would target CUDA first. I would target CUDA most. And that's the reason that I think in the final analysis is our first, that's even our first- - core advantage. Our second one is our ecosystem. The fact that we vertically integrated this incredibly complex system, but we integrate it horizontally into every single company's computers.

- We're into Google Cloud, we're into Amazon, we're in Azure. You know, we're ramping up AWS like crazy right now. We're in new companies like CoreWeave and Nscale. We're in supercomputers at Lilly. We're in enterprise computers. We're at the edge in radio base stations. You know, I mean, it's just crazy. One architecture is in all these different systems. We're in cars, we're in robots, we're in satellites, we're out in space. And so the fact that you have this one architecture and the ecosystem is so broad, it basically covers every single industry in the world.

- Well, how does the, how does the CUDA install base evolve into the future with AI factories as a moat? What do you... Do you think it's possible that NVIDIA of the future is all about the AI factory? - Well, the unit of computing used to be GPU to us. Then it became a computer, then it became a cluster. Now it's an entire AI factory. When I see a computer, when I see what NVIDIA builds, in the old days, I would, you know, I visualize the chip.

And then when I announced the new product, new generation, like, "Ladies and gentlemen, we're announcing Ampere today," I'd pick up the chip. That was my mental model- ... of what I was building. Today, I wouldn't... Picking up the chip is kind of still adorable. But it's adorable. It's not my mental model of what I'm doing. My mental model is this giant gigawatt thing that has power generations connected to the grid. It's got cooling systems and networking of

incredible monstrosity, you know. 10,000 people are in
there trying to install it, hundreds of networking engineers in there,
thousands of engineers behind it trying to power it up.

You know, powering up one of those factories,
as you know, it's not somebody going, "It's on now." It takes thousands
of people to bring it up. - So mentally, you're actually... When you're thinking
about a single unit of compute, you're like literally, when you go to bed at night, you're
thinking now about collection of racks, so pods, not individual chips. - Entire infrastructure. And I'm hoping my next
click is when I'm thinking about building computers, it's, you know, planetary
scale. That'll be the next click.

- Well, what do you think about
the space angle that Elon has talked about, doing compute
in space for solving some of the... It makes some of the energy
issues in terms of scaling energy easier. - Cooling issues is not easy. Yeah. - Cooling. Well, there's a large number of
engineering complexities involved with that. So, so what... You know, NVIDIA
has also announced that you're already thinking about that. - Yeah, we're already there. NVIDIA
GPUs are the first GPUs in space.

And I didn't realize it,
it was so interesting to... I would have declared it maybe. We're in space. You know, little, little
astronaut suit on one of our GPUs. But we've been in space. It's the
right place to do a lot of imaging. You know, because those satellites have,
have really high resolution imaging systems, and they're sweeping the Earth,
you know, continuously now. And, um, you want, you know,
centimeter scale, imaging imaging that is done continuously
for the world, so that, you know, you'll basically
have real time telemetry of everything. You don't wanna beam that
back down to earth. It's just, you know, petabytes and petabytes of data. You
gotta just do AI right there at the edge, throw away everything you don't need, you've
seen before, didn't change, and then just keep the stuff that, that you need.
And so AI had to be done at the edge.

Obviously we have, we have a 24/7 solar,
if we put it at the polars. And uh, but, you know, there's no
conduction, no convection. And so, you know, you're
pretty much just radiation. And uh, but, you know, space is big. I
guess, you know, we're just gonna put big, giant radiators out there. - How crazy of an idea do you think it is?

Like is this, is this five years out, 10 years out, 20 years out? So we're talking about blockers for AI scaling.

- You know, I'm just so much more practical. I look for where, where, um my next, next bucket of opportunities are first. Meanwhile, I'm cultivating space. And so I send engineers to go work on the problem. We're starting to... We're learning a lot about it. How do we deal with radiation? How do we deal with degrading performance? How do we deal with a continuous, Testing and attestation of, of de- defects? And you know, how do we deal with redundancy? And how do we degrade gracefully and things like that? And so we could, we could do a...

What about software? How do you think about software and redundancy and performance out in space? Make it so that the computer never breaks, it just gets slower, you know. And I... So we could start doing a lot of engineer exploration upfront. But in the meantime, my, my favorite answer is ge- eliminate waste. You know, we've got all that idle power, I want to evacuate it as fast as possible. - Yeah. There's a lot of low-hanging fruit here on Earth- ... That we can utilize for the AI scaling. Quick pause.

Quick 30-second thank you to our sponsors. Check them out in the description. It really is the best way to support this podcast. Go to lexfridman.com/sponsors. We got Perplexity for curiosity-driven knowledge exploration, Shopify for selling stuff online, LMNT for electrolytes, Fin for customer service AI agents, and Quo for a phone system, like calls, texts, contacts, for your business. Choose wisely, my friends. And now, back to my conversation with Jensen Huang.

Do you think NVIDIA may be worth 10 trillion at some point? Let's ask it this way. What does the future of the world look like where that's true? - I think that NVIDIA's growth is Extremely likely, and in my mind, inevitable. And let me explain why. We're the largest computer company in history.

That alone should beg the question, why? And the reason of course... Two reasons. First, two foundational technical reasons. The first reason is that computing went

from being a retrieval-based, file retrieval system. Almost everything is
a file... We pre-write something, we pre-record something. You know, we draw
something, we put it on the web, we put it in a file. And we use a
recommender system, some smart filter, to figure out what to
retrieve for you. And so we were a pre-recording, human
pre-recording, and file retrieving system. That's what a
computer is, largely.

To now, AI computers
are contextually aware, which means that it has to process
and generate tokens in real time. So we went from a retrieval-based
computing system to a generative-based computing system. We're gonna need a lot more processing
in this new world than in the old world. We need a lot of storage
in the old world. We need a lot of computation in this new world. And so, so that's the first part of it. We
fundamentally changed computing and the way how computing is done. The only
thing that would cause it to go back..... is if this way of computation, this way of computing generating
information that's contextually relevant, situationally aware, that is
grounded on new insight before it generates information, this
computation-intensive way of doing computing would only go back if it's not effective.

So if... For the last 10, 15 years
while working on deep learning, if at any single moment I would have come to the conclusion
that, "You know what? This is not gonna work out. I think this is a dead end." Or, "It's not gonna scale,
it's not gonna solve this modality, not gonna be used in this application." Then, of course,
I would feel very differently about it, but I think the last five years
has given me more confidence than the last ten year, previous
ten years. The second idea, is computers, because it was a storage
system, it was largely a warehouse.

We're now building factories.
Warehouses don't make much money. Factories directly correlates with
the company's revenues. And so, the computer did two things. Not only
did it change the way it did it, its purpose in the world changed. It's
no longer a computer, it's a factory. It's a factory, it's used
for generation of revenues.

We're now seeing not only
is this factory generating products, commodities that
people want to consume, we're seeing that the commodities are so interesting, so valuable, so,
to so many different audiences that the tokens are starting to
segment, like iPhones. Mm-hmm. You have free tokens, you have premium tokens, and you

have several tokens in the middle. Yeah. And so intelligence,
as it turns out, you know, it's a scalable product.
There's extremely high intelligence products, tokens that you could... that
are used for specialized things, people be willing to pay. You know, the
idea that somebody's willing to pay $1000 per million tokens is just around
the corner. It's not if, it's only when.

And so now we're seeing that the commodity that this factory makes is
actually valuable, and is revenue generating and profit generating.
How, now the question is how many of these factories
can, does the world need? How much, how many tokens
does the world need? And how much is society willing
to pay for these tokens? And what would happen to the world's economy if the productivity were to
improve so substantially?

What would happen... Are we, are we
gonna discover new drugs, new products, new services? And so when you take these
things in combination, I am absolutely certain that the world's GDP is
going to accelerate in growth. I'm absolutely certain the percentage of
that GDP that will be used for computation will be 100 times more than
the past... because it's no longer a storage unit. It's
a product generation unit.

And so when you look at it in that context and then you back into what
is NVIDIA's, what does NVIDIA does NVIDIA do and how much of that new economics, new industry
would we have to benefit to address, I think we're
gonna be a lot, lot bigger. And then the rest of it, to me, is is it possible for NVIDIA to be
a, you know, $3 trillion revenues company in the near future?
The answer is of course yes. And the reason for that is because it's not
limited by any physical limits.

There's nothing that I see
that says, you know, gosh, $3 trillion is not possible. And as it turns out, NVIDIA's supply chain is,
the burden is shared by 200 companies. And the fact that we scale out on the backs
with the partnership of this ecosystem, the question is do we
have the energy to do so? And surely we will have the energy to do
so. And so all of these things combined, that number is just a number, you
know? And I still remember, NVIDIA was a, NVIDIA was a, the first
time we crossed a billion dollars, I was reminded of a CEO who told me,
"You know, Jensen, it's theoretically impossible for a fabless semiconductor
company to exceed a billion dollars."

And I won't bore you
with why, but the end, of course it's illogical and there's a
lot of evidence we're not. And then, somebody told me, "You know, Jensen,
you'll never be more than $25 billion because of some other company." Somebody
told me that, "You'll never be, you know, because..." And so the, those
aren't first principled thinking. And the simple way to think about
that is what is it that we make and how large is the
opportunity that we can create?

Now, NVIDIA is not in the market share
business. Almost everything that I just talked about don't exist. That's
the part that's hard. You know, if NVIDIA was a $10 billion company trying to take market
share, then it's easy to see for shareholders that, oh,
yeah, if they could just take 10% share, they could be this much larger. But it's hard for
people to imagine how large we could be because there's
nobody I could take share from.

You know? And so I think
that that's one of the challenges.... for the world
is, is the imagination of the future. But I got plenty of time, and I'll
keep reasoning about it, and I'll keep talking about it, and every single GTC
will become more and more real. You know, and then more and more people will
talk about it, and one of these days, you know, we'll get there. But
I'm 100% we'll get there. - Yeah, this view of you know, token
factories essentially, this token per second per watt, and every
token having value. Like it's an actual thing that brings value,
and it brings different kinds of value, different amounts of value to different people
with value. That's the actual product, is really could be loosely thought of as the token.
And so you have a bunch of token factories.

And then it's very easy, first principles,
to imagine a future, given all the potential things that AI can
solve, that you're going to need an exponential number more
of token factories. - And what's really interesting, the reason
why I was so excited about it, the iPhone of tokens arrived. - What do you call it? Wait, are you saying
OpenClaw's iPhone? That's interesting. - Agents. - Yeah, agents. True. - Agents in general. The iPhone of tokens
arrived. It is
the fastest-growing application in history. It went straight
up. Went straight up.

- That says something. - Yep, there's no question OpenClaw

is the iPhone of tokens. - Is there something truly, as you know, something truly special happening from about December, where people have really woke up to the power of Claude Code of Codex, of OpenClaw. Um, I mean, I'm embarrassed to admit that on the way here in the airport, I've ... It's the first time I've done this in public. I was programming, quote unquote, by talking to my laptop.

And I was embarrassed because I was pretending like I'm talking to a human colleague. Uh, I'm not sure how I feel about the future where everybody- ... is walking around talking to their AI, but it's such an efficient way to get stuff done. - And it's more likely that your AI is bothering you all the time. And the reason for that is because it's getting stuff done so fast. It's reporting back to you, "I got that done."

"You know, what do you want me to do next?"
You know, it... That's the part that I think- ... most people don't realize is they're
The person who's gonna be chatting with them, texting them most, is their claws or lobster. - What an incredible future. Uh, I read that you attribute a lot of your success to your ability to work harder than anyone and withstand more suffering than anyone. So we can list many of the things that entails. I mean, dealing with failure, the cost and engineering problems we've talked about. The human problems, uncertainty, responsibility, exhaustion, embarrassment, the near-death company moments that you've mentioned, But also the pressure.
Now, as the CEO of this company that economies and nations strategize around plan their, Financial allocations around, plan their AI infrastructure around, how do you deal with this much pressure?

What gives you strength, given how many nations and peoples depend on you? - I'm conscious about the fact that, NVIDIA's success is very important to United States. We generate enormous amounts of tax revenues. We established technology leadership for our nation. Technology leadership is important for national security.

National security not just in one aspect of national security, all aspects of national security. When our country's more prosperous, we could do a better job with domestic policies and helping social benefits. Because we're generating so much re-industrialization in the United States, we're creating mountains of jobs.

We're helping shift, how we, how we build things back to
the United States in so many different plants, chips,
computers, and of course, these AI factories. I'm completely aware that, that... And I have the
benefit, and this is a real a real gift with mainstream investors, teachers, policemen who have
somehow, for whatever reason, invested in NVIDIA or because
they watched Jim Cramer bought some stock and now are millionaires.
And I am completely aware of that circumstance. I'm aware of
the circumstance that NVIDIA, is central to a very large network of ecosystem partners behind
us and downstream from us.

And so the way I deal with that is
exactly what I just did. I reason about what is... what is it that we're doing? What is it causing? What's the
impact that has on other people benefit, you know, positively or even, even through great burden,
for example, to supply chain? And the question is, therefore,
what are you gonna do about it? In almost everything that I feel, I
break it down, I reason about, "Okay, "what's the circumstance? What
has changed? What's hard?

And what am I gonna do about it?" And I
break it down, decompose the problem, and the decomposition
of these circumstances turns it into manageable things that I
can do. And the only thing after that I could do is, "Did you do it?
Did you either do it or did you get somebody else to do it?
And if you didn't do it, you reasoned that you need to do it, and you didn't
do it, and you didn't get anybody else to do it, then stop crying about it." You
know? And so- ... so I'm fairly Tough on myself. And, but I
also break things down so that, so that I don't panic. I can go to sleep because I've made the list
of things that needed to be done, and I've made sure that everything that
could put our company in harm's way, could put my partners in harm's way, put
our industry in harm's way, I've told somebody. Everything that I feel could put anybody in harm's way,
I've told someone. And I've told that someone who could do something about
it. And so I've gotten it off my chest or I'm doing something about it. And so
after that, Lex, what else can you do?

- So given all the insane,
intense amount of suffering on the journey of building up NVIDIA,
have you hit low points psychologically? - Oh, yeah. Oh, yeah. Sure.
All the time. All the time. - And there— - All the time. - ... you just break down
the problem into pieces? - Yeah. Yeah. - See what you could do about it? - And, and part of, and, you know, Lex,
part of it, part of it is forgetting.

One of the most important attributes
of AI learning, as you know, is, right? Systematic forgetting. You,
you need to know when to forget some things. You can't memorize everything.
You can't keep everything and, and, you know, you, you want to— you don't want to
carry everything. One of the things that I do very quickly is decompose the problem,
I reason about the problem, and I share the load with it. When
I say I tell everybody, I'm essentially sharing that
burden. As quickly as possible.

Whatever worries me, tell somebody
else. Don't just keep it. You know, don't freak them out. Decompose the
problem into smaller parts and get people to, so, and, and inspire them to
be able to go do something about it. But part of it is just, just
forgetting. You know, like, a lot of it is you gotta be tough on
yourself. You know, just come on, stop crying about it. Let's get going. You know?
And, and then you get out of bed. And then the other part is, is you,
you're attracted to the next shiny light, the next future,
you know, the next opportunity, the next, "Okay, that's behind
us. Let— what's next?"

It's a lot, I think, you know, you
watch this with great athletes. They, they just worry about the next point. The last point is behind them. The
embarrassment, the, you know— the setback. You know, and, and then, and because
I do so much of my job publicly, you know? Lex, you do a fair amount of your
job publicly too. And so, so I do a lot of my job publicly. And so you
know, I say a lot of things that, that seem sensible
at the time or funny at the time, mostly it's just because it's funny to
me at the time. And then, you know, you reflect on it, it's
less funny, but, but...

- Yeah. No, trust me, I know. But
you basically allow yourself to be pulled by the light of the future.
Forget the past and just keep- - That's right. - ... keep working towards that. I mean,
you did say, there's this kind of famous thing you said that if
you knew how hard it would be to build NVIDIA it turned
out to be, what is it? A million times more hard than you
anticipated that you wouldn't do it.

- Yeah, right. - Um, but isn't... You
know, when I hear that, that's probably true about
everything worth doing, right? - Exactly. That is, by the way, what I

was trying to explain, is that there's a, there's a incredible superpower of being, being have a, the mind of a child. You know? And I say to myself oftentimes when I look at something, and almost, almost everything, My first thought is, "How hard can it be?"

You know? And so you get yourself into that mode, how hard could it be? And, and nobody's ever done it. It looks gigantic. It's gonna cost hundreds of billions of dollars. It's gonna take, you know, all this... And you just go, "Yeah, but how hard could it be?" You know? How hard could it be? And so you gotta get yourself into that state of mind. You don't wanna, you don't wanna actually over simulate everything and all the setbacks and all the trials and tribulations and all the disappointments. You don't wanna simulate all that in advance. You don't wanna know that.

You don't, you wanna go into a new experience thinking it's gonna be perfect, it's gonna be great, it's gonna be incredibly fun. And then while you're there, you know, you need to have, you need to have endurance, you need to have grit, so that when the setbacks actually happened, and those setbacks are gonna surprise you, the disappointments are gonna surprise you, you know, the embarrassments are gonna surprise you, the humiliations are gonna surprise you. You just can't let... Now you just gotta turn on the other bit, which is just forget about it. Move on, keep moving. And, and to the extent that, to the extent that my assumptions about the future and why the future is gonna manifest, so long as those assumptions and that input doesn't change or didn't change materially, then I should expect that the output won't change. And so my simulated output of the future is still gonna happen. And if it's still gonna happen, I'm still gonna go after it. I believe it's gonna, you know, and so there's a combination of two or three human characteristics, the ability to go into a, into an experience fresh-minded, the ability to forget the setbacks, the ability to believe in yourself, you know, to believe what you believe and stay, stay true to that belief.

But you're constantly reevaluating. This combination of three, four, five things I think is, is really important for resilience. And, and... and, you know, I'm fortunate that, that whatever life experiences led to this, I've got kind of those four, five things. You know, I'm always curious, always learning. I'm always learning from everybody, you know? I'm always asking my... And because I'm humble about everything, I'm always thinking, "Gosh, they did that so nicely. They

did that so wonderfully." You know, I wonder what they're thinking through.
How do they... You know, so I'm simulating everybody. In a lot of ways,
you know, I'm emulating almost everybody I watch, right? You're empathetic
towards everything that they do that you're observing and
respect. And, and so you're constantly learning and, you know.

- You're now one of the wealthiest
people on Earth. One of the most successful humans on Earth. Is it harder to be humble and to
be able to... Do you feel the effect of money and power and fame
in making it harder for you to sort of be wrong in your own head? Enough to hear out an opinion of
somebody else when they disagree with you and learn from
them? Those kinds of things.

- Um, surprisingly, no. And I would,
I would actually go the other way. Because I do so much of my
work publicly, when I'm wrong, pretty much everybody sees it. - You get humbled. Fair enough. - And when I'm wrong, when I'm
wrong or it didn't turn out that way or you know, I mean, most of the
things that, that I say outside I'm fairly certain about. And the reason for
that is because, because it's gonna impact somebody else and I want to be
quite concerned about that and quite, circumspect about that. For stuff that,
that I'm reasoning about inside a meeting, you know a lot of things could
turn out differently. And so, but it doesn't ever stop me from
reasoning. The way that the way that I manage and lead, I, you know, I'm constantly reasoning in front of
people. And even when I'm talking to you, you can kind of see me kind of
reasoning through things.

And I want to make sure that you understand
what I'm saying not because I told you- ... because I'm so humble about
what I'm about to tell you. I kind of show you the
steps that I got there. And then you can decide whether you believe
what I said in the end. And so I'm doing that all day long in meetings. With all of my employees, I'm constantly
reasoning through, "Let me tell you what, how I see it." And then I reason through
it. It gives everybody the opportunity to intercept and say,
"I disagree with that part."

The nice thing about reasoning through
things and letting people interact with it is that they don't have to
disagree with your outcome. They can disagree with
your reasoning steps. And they could pull me in different
directions, and then we can reason forward. And so we're kind of, you know, collective path searching method.

And
it's really fantastic. - Yeah, you have this way about you of ... When you're
explaining stuff, I can feel you actually reasoning on
the spot about it with a constant open-mindedness where you
could ... I could feel like I could steer your thinking.

And that's a, that's really beautiful that
you've been able to maintain that after so many years of success, and pain. I think sometimes pain makes you close, closes you down a bit. - Mm-hmm. Yeah. - And I think to maintain- - Yeah. Tolerance for embarrassment, I think is... - Yes, that's ... The tolerance
... I mean, that's a real thing. Is many years of embarrassing
yourself. Even those meetings knowing that there's people around
you where you declared one idea and it was shown that that idea was wrong- ... and be able to admit that and to grow from
that. That's not, that's very difficult on a human level.

- Yeah. Well, you know. They
knew that recently my first job was, you know, cleaning toilets, so. - I'm glad you maintained
that same spirit of Denny's the, the work. I mean, that, that was beautiful.
Your whole journey from, starting from Denny's is a beautiful one. Let me ask you about
video games. So I'm a big gaming fan. So I have to say thank you to NVIDIA
for many years of incredible graphics. - By the way, GeForce is
our still, to this day- ... our number one marketing strategy.

Right. People learn about NVIDIA
while they're in their teenage years. And then they go to college and they
know who NVIDIA is and then in beginning is just, you know, playing Call of Duty, you
know? You know, Fortnite. And then later they're using CUDA, and then later they're
using NVIDIA and, you know, Blender and Dassault and Autodesk. - Yeah. I mean, I should say I
mentioned to a friend that I'm talking with you. He said, "Oh,
they make great gaming GPUs."

- Yeah, exactly. - It's like- - Exactly. - You know, there's more
to it, but, yeah, yeah, people really love the ... It really brought
a lot of joy to a lot of people. The, the, the hardware really
brings these worlds to life. There was some controversy
around this with DLSS 5. Can you explain to me the
drama around this? Uh, I guess people, the gamers online were concerned
that it makes games look like AI slop.

Uh, what do you think of this drama? - Yeah. Uh, I think their

perspective makes sense and I could see where they're coming from,
because I don't love AI slop myself. You know, all of the AI
generated content increasingly, um, looks similar and
they're all beautiful, and I can... So I can... I'm empathetic
towards what they're thinking. That's just not what DLSS 5 is trying
to do. I showed several examples of it. But DLSS 5 is 3D conditioned, 3D guided. It's ground
truth structure data guided. And so the artist determined the geometry.
We are completely truthful.... to the geometry maintains
in every single frame.

It's conditioned by the textures, the
artistry of the artist. And so every single frame, it enhances but it doesn't change anything. Now, the question is,
the question about enhancing, DLSS 5 also lets, because it's, the system is open, you could train
your own models to determine, and you could even in the future prompt
it. You know, I want it to be a toon shader. I want it to look like this
kinda, you know, so you can give it even an example. And it would
generate in the style of that, all consistent with the
artistry, you know, the style, the intent of the artist.
And so all of that is done for the artist, so that they can create
something that is more beautiful, But still in the style that
they want. I think that they got the impression
that the games are gonna come out the way the games are
shipped the way they do, and then we're gonna post-process it. That's
not what DLSS is intended to do.

DLSS is integrated with
the artist, and so it's, it's about giving the artist
the tool of AI, the tool of generative AI. They could
decide not to use it, you know? - I think people are very
sensitive to human faces. And we're now living in this moment,
which I think is a beautiful one, which is people are sensitive to AI slop. It puts a mirror to ourselves to
help us realize that what we seek is imperfections. What we seek is
sometimes not perfect graphics.

It helps us understand what we find
compelling in the worlds we create. And that's beautiful. And as long as it's
tools that help us create those worlds- - Yeah, that's right - ... it's wonderful. - That's right. Yet, yet another
tool, and they want the generative, models to generate the
opposite of photo real. Yeah, it'll do that too. And so it's
just yet another tool. I think the the gamers might also appreciate that,
that in the last couple of years, we introduced skin shaders to the game developers. And many
of those games have skin shaders that include subsurface scattering that

make skin look more skin-like. And so the industries, you know, game developers
are looking for more and more and more tools to express their art. And so this is just yet more, one more tool,
and they get to decide what to use.

- Ridiculous question. What do you think
is the greatest or most influential game ever made? Maybe from
NVIDIA's perspective? - Doom. - Doom, unquestionably. That
was the start of the 3D. - I would say Doom, from an
art, the intersection of the cultural implication as well as the
industry, turning a PC into a gaming device. That was a very important moment. Now, now
of course, flight simulation companies were before it. And but they just didn't have
the popularity that Doom did to have made the industry turn the PC from an office
automation tool into a personal computer for families and gamers and things like that.
And so Doom was really impactful there.

From an actual game technology
perspective, I would say Virtua Fighter. And so we're great friends
with both of them, you know? - And then there's games more
recently, I mean, Cyberpunk 2077, really nice GPU-accelerated
graphics. Like- - Fully ray traced. - Fully ray traced. Also, I like, I
personally, I'm a huge fan of Skyrim, uh, Elder Scrolls, and the, you know,
it's, it's been released a long, long time ago, but people release mods and- - We love mods - ... they create these,
these inc- I mean, it-
... it's like a different game and it just allows me to replay the game
over and over and get i- It makes you realize that
you can re- experience in a totally new way the world
you already love. So- - That's right - ... I do that all the time. One of my
favorite things is just walk across Skyrim.

- Uh, we created this thing called
RTX Mod. Yeah, it's a modding tool. - Awesome. - It allows the community to inject the
latest technology into an old game. - Of course, like what makes a great video game
is not just graphics, it's also story and character development, but- - That's right - ... beautiful graphics
can add to the immersion. The feeling like it's another
place you're transported to. Ah, what you said, I think
accurately, that the AGI timeline question rests on your definition of AGI.

So, let me ask you about
possible timelines here. Let's, this ridiculous definition
perhaps of what AGI is, but an AI system that's able to
essentially do your job. So, run, no, start, grow, and run a successful

technology company that's worth- - A good one or a one? - No. It has to- It has to be worth more than a billion, more, more than a billion dollars. So, you know, you know how hard it is to do all those components.

So, how far are we away from that? So, we're talking about OpenClaw that does all the incredibly complex stuff that are required to to, first of all, innovate, to find customers, to sell to them, to manage, to build a team of some agents, some humans, all that kind of stuff. Is this five, 10, 15, 20 years away? - I think it's now. I think we've achieved AGI. - Do you think you could have a company run by an AI system like this?

- Possible, and the reason for that is this. You said a billion, and you didn't say forever. And so for example, uh... It is not out of the question that a Claw was able to create a web service, some interesting little app that all of a sudden, you know, a few billion people used for 50 cents, and then it went out of business again shortly after. Now, we saw a whole bunch of those type of companies during the internet era, and most of those websites were not anything more sophisticated than what OpenClaw could generate today.

- Interesting. Achieve virality and monetize that virality. - Yeah. It's just that I don't know what it is, but I couldn't have predicted any of those companies at the time either, you know? And - - You're gonna get a lot of people excited with that statement. It's like, what do you mean? I can- I can just, uh - ... launch an agent and make a lot of money. - Well, by the way, it's happening right now, right? You know that when, when you go to China you're gonna see, you're gonna see a whole bunch of people teaching their, getting their Claws to try to go out and look for jobs and, you know, do work, make money.

And I'm not, I'm not actually... I wouldn't be surprised if some social thing happened or somebody created a, a digital influencer, super, super cute or some social application that, you know, feeds your little Tamagotchi or something like that, and, and it become an out of the blue an instant success. A lot of people use it for a couple of months and it kind of dies away. Now, the odds of, you know, 100,000 of those agents, Building NVIDIA is zero percent.

And then the one part

that I will, I won't do, And I, I want to make
sure we all do, is to recognize that people are really
worried about their jobs. And I just want to remind them
that the purpose of your job and the tasks and tools
that you use to do your job are related, not the same.
I've been doing my job for 33 years. I'm the longest running
tech CEO in the world, 34 years.

And the tools that I've used
to do my job have changed continuously in the last 34
years, and sometimes quite dramatically, you know, over the
course of a couple, two, three years. And the one story that I really wanna make sure that everybody
hears is the story that the first job that computer
scientists said, AI researchers said was gonna
go away was radiology. Because computer vision was going to
achieve superhuman levels, and it did.

CV... Computer vision was superhuman in
2019, 20, maybe a little bit later, 2020? Okay? And so it's been a long time since
computer vision has been superhuman. And so the prediction was radiologists
would go away because studying radiology scans was a thing of
the past. AI will do that. Well, they were absolutely right. Computer vision is completely
superhuman. Every radiology platform and package
today is driven by AI, and yet the number of
radiologists grew. And so the question is why? And we now have a
shortage of radiologists in the world.

And so, one, the alarmist warning went
too far and it scared people from doing this profession that
is so important to society. And so it did harm. Now, why was it
wrong? The reason why is because the purpose of a radiologist, the
purpose is to diagnose disease and help patients and
doctors diagnose disease. And because we're able to
study scans at so much faster now, you could study more scans, you
could diagnose better, you could, you could inpatient faster, you can see people more. The hospitals
are making more money. You have more patients in the hospital. You
need more radiologists. I mean, the amazing thing is, it's so
obvious this was gonna happen.

The number of software engineers at
NVIDIA is gonna grow, not decline. And the reason for that is because

the purpose of a software engineer and the task of a software engineer
coding are related, not the same. I wanted my software engineers to solve problems.
I didn't care how many lines of code they wrote, you know? But their job, their
purpose of their job didn't change. Solving problems, working as a team, diagnosing
problems, evaluating the result, looking for new problems to
solve, innovation, connecting dots. You know, none of that
stuff is gonna go away.

- Do you think it's possible that... Let's
even take coding. Do you think the number of programmers in the world
might increase, not decrease? - Yes. And the reason for that is this.
What is the definition of coding? I believe it is... The definition of
coding, as of today, is simply specifying, specification, and maybe if you want to be rather directive, you could even give
it an architecture of the software that you wanted to write. So the question
is, how many people could do that?

Describe a specification for a
computer to go... Telling the computer what to go build. How many
people? I think we just went from 30 million to probably 1 billion. And so
every carpenter in the future will be a coder, except a carpenter
with AI is also an architect. They've just increased the value that they
could deliver to the customer. Their, their artistry just elevated tremendously.

I believe that every accountant is,
you know, also your financial analyst, also your financial advisor. So,
all of these professions have just been elevated.... and, and
if I were a carpenter, I sees a, I see AI, I would just
completely go berserk. You know, the services I can bring
to my clients if I were a plumber, completely go berserk. - And the people that are currently
programmers and software engineers, I think they're at the cutting edge of understanding
intuitively how to communicate with the agents using natural language in
order to design the best kind of software.

- That's right, exactly. - So over time they'll converge, but I think there's still value in getting,
I think learning how to program, like learning what programming
languages are. The old, the old kind of programming, what are good practices for programming languages, what are
design principles for programming- - That's right - ... Languages for large software systems? - And the reason for that, Lex, and you know, as you're
saying for the audience, I think the goal of, the goal of

specification, the artistry of specification, the goal
and the artistry of it, Is going to depend on what
problem you're trying to solve.

When I'm thinking, when I'm thinking
about giving the company strategies and formulating corporate directions
and things that we should do, I describe it at a level
that is sufficiently specific that people generally
understand the direction and it's actionable. It's specific
enough that they can take action on it, but I under specify it on purpose, so that enable 43,000 amazing people to make
it even better than I imagined.

And so when I'm working with engineers
and when I'm working with people, I think about who, what problem am I
trying to solve? Who am I working with? And the level of specification, the
level of architecture definition relates to that. And, and so everybody's going to have to learn how,
where in the spectrum of coding they want to be. Writing a specification is coding. And so you might decide to be quite
prescriptive because there's a very specific outcome you're looking for.
You might decide that, you know, this is an area you want to be much more
exploratory, and so you might under specify and enable you to
go back and forth with the AI to even push your own
boundaries of creativity.

And so this artistry of where you are in
the spectrum, this is the future of coding. - But just to linger on it outside of coding,
I think a lot of people, rightfully so, are worried about their jobs, have a lot
of anxiety about their jobs, especially in the white-collar sector. I don't
think any of us know what to do, With tumultuous times that always come
when automations and new technology arrives. And I just...
First of all, I think we all need to have compassion and the
responsibility to feel sort of the burden of what the actual suffering feels like for
individual people and families that lose their job. I think whenever you have
transformative technology like that's coming with artificial intelligence,
there's going to be a lot of pain, and I don't know what to do about
that pain. Hopefully, it creates much more opportunities for those same people,
for the same kind of job as the tooling evolves and makes them more
productive and makes them more fun, hopefully, as it does in the programming.
I've been having so much fun programming, I have to say. Like, I've never had this
much fun. So hopefully it makes their job, automates the boring parts
and makes the creative parts, the ones that the human beings are responsible

for. But still there's going to be a lot of pain and suffering.

- So my first recommendation before...
And this is now how I deal with anxiety. In fact, we
just talked about it earlier. Enormous anxiety about the future, enormous
anxiety about the pressure, enormous anxiety about uncertainty, I first break
it down, and then I'm gonna tell myself, "Okay, there are some things you can do something
about, there's some things you can't do anything about. But for the stuff that you
can do something about, let's reason about it and let's go do it."

If we were to hire a new college graduate
today, and I have a choice between two, one that have, that is no clue what AI is and one that is expert in using AI, I would
hire the one who's expert in using AI. If I had an accountant,
a marketing person, the one that is expert in
using AI, supply chain, customer service, a salesperson,
business development, a lawyer, I would hire the one who
is expert in using AI.

And so I would advise that
every college student, every teacher should encourage
their student to be, to go use AI. Every college student
should graduate and be an expert in AI. And everybody,
if you're a carpenter, if you're, you know, electrician, go use AI. Go see what it can do to
transform your current job, elevate yourself. If I were a
farmer, I would absolutely use AI. If I were a pharmacist, I would
use AI. I wanna see how, what it could do to elevate my job so
that I could be the innovator to revolutionize this industry myself.

And so that would be the first thing
that I would do. And then I would also, I would also help them... it is the case that the technology will dislocate and will eliminate many tasks if... And because it will
automate it, if your job is the If your job is the task, then you're
very highly going to be disrupted. If your job's purpose
includes you, certain tasks- ... then it's vital that you go learn
how to use AI to automate those tasks.

And then there's the world
of spectrum in between. - And by the way, the beautiful thing
about AI, so the chatbot versions, is you can break down... You have anxiety and you can break down the problem by talking to it. Like, I've recently... It's really

just incredible how much you can think through your life's problems, and through...
And I don't mean, like, therapy problems. I mean, like, very practically, "Okay,
I'm worried about my..." Literally, "I'm worried about my job. What are the skills? What
are the steps I need to take? How do I get better at AI?" Everything you just said, you could
literally ask and it's going to give you- ... a point-by-point plan. I mean, it's
just a great life coach, period. This- - I don't know how to use AI, and the
AI goes, "Well, let me show you."

- Exactly. It's very meta, but it's- It's kind of incredible. So
people definitely should- - You can't walk up to Excel and say, "I
don't know how to use Excel." You're done. - I mean, that's really what AI has
done for me in all walks of life, is that initial friction of being a beginner
of using a thing for the first time. I can literally ask
about any single thing, "What are the first steps I need to take?"

- That's right. - And that handholding that it
does, removing the friction of all the experiences that
the world offers is... You know, like I mentioned to you offline,
you mentioned, "I'm going to China and Taiwan." - So awesome. - Just ask, "Where do I-" - So excited for you. -
"Where do I—what do—" "You know, where do
I go? How do I..." All of those questions— ... immediately answered,
and it's beautiful. - Well, when you, when you go to
Taiwan, just ask AI- ... "What are Jensen's favorite restaurants
in Taiwan?" And it'll actually- - You don't know?

- Oh, yeah. - Is it accurate? Okay. - Yeah. - All right. - It's all over Taiwan. - Well, you're a rockstar over
there. And like we also mentioned offline, maybe our paths will cross, which
would be really wonderful in computing. - COMPUTEX. NVIDIA GTC Taiwan. - Uh, do you think there's some things about
human nature, about human consciousness that is fundamentally non-computational?
Maybe something a chip, no matter how powerful, can never replicate?

- I don't know if the chip will ever get
nervous. And that's the, you know, of course, the conditions by which that causes anxiety
or nervousness or whatever emotion. Um, I believe that AI will be
able to recognize those and understand those. I don't
think my chips will feel those. And therefore, the... How
that anxiety, how that feeling, how that excitement,
how that, how that, you know...

All of those feelings manifest in

human performance. For example, extremely amazing human performance,
athletic performance, you know, average or lesser than
average. That entire spectrum of human performance
that comes out of exactly the same circumstances
for different people, manifesting a different outcome, manifesting
a different performance. I don't think there's anything about anything that
we're building that would suggest that two different computers being presented with all of exactly
the same context would perfo- Of course, it would produce statistically
different outcomes, but it's not because it felt different.

- Yeah, the subjective... Boy, there's something truly special
about the subjective experience that we humans feel. Like I mentioned to
you, I was pretty nervous talking to you. Like I mentioned to you, that,
the hope, the fear, the anxiety, and just life itself, the richness of
life. How amazing everything is. How deeply we fall in love, how
deeply our hearts get broken, how afraid we are of death and
how much pain we feel when our loved ones pass away. All of that, the
whole thing. I know it's very hard to- ... think AI being able to... A
computational device being able to do that.

But there's so many mysteries about
this whole thing that we're yet to uncover, that I am open to be surprised.
I've been surprised a lot over the past- ... few months and few years.
Scaling can create some incredible miracles in the
space of intelligence. Has been truly marvelous to
watch, so I'm open to surprise. - And it's just really important
to break down what is intelligence. You know, the word,
that word we use all the time, it's not a mysterious word. Intelligence
has a meaning, you know?

And it's a system that... You
know, it's something that we do that includes perception
and understanding and reasoning and the ability to do
plan. And, you know, that, that loop, that loop, is the...
Fundamentally what intelligence is. Intelligence is not one word that
is exactly equal to humanity. And that's, I think it's really important
to separate the two. We have two words for that. I'm not... I don't
over-fantasize about, and I don't over-romanticize about
intelligence. Intelligence is... And people have heard me
say it before, I actually think intelligence is a commodity. I'm
surrounded by intelligent people.

And I'm surrounded by intelligent people more
intelligent than I am in each one of the spaces that they're in. And yet, I have a role in that circle. It's
actually kind of interesting. They're more educated than I am. They
went to better schools than I did. They're deeper than, in any of the
fields that they're in. All of 'em. I have 60 of 'em. They're
all superhuman to me. And somehow, I'm sitting in the
middle orchestrating all 60 of 'em.

And so you gotta ask yourself...
Uh, what is, what is it about a dishwasher that allows that dishwasher to sit in the middle of
superhumans? Does that make sense? And so, but that's my point. My point
is intelligence is a functional thing. Humanity is not a, not specified
functionally. It's a much, much bigger word. And, and our life experience,
our tolerance for pain, our determination, those are, those
are different words than intelligence.

And so the thing that I wanna
help the audience understand, if I could give them one thing, is, is intelligence is a word that we've
elevated to a very high form over time. - The, the word we should
really elevate is humanity. - Character, humanity. - All those things. - All of those things.
Compassion, generosity, all of the things that you said just now, I believe those are superhuman
powers. And that now intelligence is gonna be commoditized.

Because we've spoken about it, the most
important thing is your education. The most... Now, even, even when they said
the most important thing is your education, when you went to school, there's more
than just knowledge that you gained. And so, but unfortunately, our society
has put everything into one single word, and life is more than one word. And I'm
just telling you, my life would suggest that being lower on the intelligence
curve than everybody around me doesn't change the fact I'm the
most successful. And so, and I think, I think that kind of is I'm
trying to hopefully inspire everybody else that don't let
this democratization of intelligence, this
commoditization of intelligence, you know, cause you anxiety.
You should be inspired by that.

- Yeah. I think AI will help us
celebrate humans more. And certainly humanity and human first, and I, I think
what makes this world incredible is humans forever will be so, and just AI is
this incredible tool that makes us- - That's exactly right. - ... humans more powerful. - That's exactly right. - Uh,
so much of the success of NVIDIA and the lives of millions of people

that I mentioned depend on you.

But you're just one
human, like we mentioned, a mortal like all of us. Do you think about
your mortality? Are you afraid of death? - I really don't wanna die. Um, I have
a great life. I have a great family. Uh, I have really important work. Uh,
this is, this is not a once in a, once in a lifetime experience suggests that it has been experienced
by many people, just not one person. This is a once in a humanity
experience, what I'm going through.

Uh, NVIDIA is one of the most consequential
technology companies in history. We're doing very important work. I take
it very seriously. Um, And so some of the, some of the
things that, that of course are, are practical things, like how do we
think about succession planning? And, I'm famous in saying that I don't
believe in succession planning. - Man. - And the reason, the reason for that,
the reason for that isn't because I'm immortal. The reason for that is
because if you're worried about succession planning, if you're worried, all that anxiety of succession
planning, then what should you do about it?

Then you break it all the way back down. The most important thing you should do
today, if you care about the future of your company, post you, is to pass on
knowledge, information, insight, skills, experience as often and continuously
as you can, which is the reason why I continuously reason about
everything in front of my team. Every single meeting is
about a reasoning meeting. Every moment I spend inside a
company, outside a company is about passing on knowledge to
people as fast as I can. Nothing I learn ever sits on my desk
longer than, you know, a fraction of a second. I'm passing that
information, that knowl- oh my gosh, this is cool. Before I even finish
learning all of it myself, I'm already pointing it to somebody else. "Get on
this. This is so cool. You're gonna wanna, you're gonna wanna learn this." And
so I'm constantly passing knowledge, empowering people,
elevating the capability of everybody around me, so that the
outcome that I, that I seek, that I hope for, is that I die
on the job, you know? And, and hopefully I die on the job
instantaneously, you know?

And there's no long periods of
suffering, you know? It's, uh — - Well, from a fan perspective
given your, your extremely, um, your enormous positive impact on

civilization, of course, I hope you keep going. But also it's just fun to watch what NVIDIA is doing, you know. It's just the rate of innovation. And I'm a huge fan of engineering. There's so much incredible engineering being continuously being done by NVIDIA. It's just fun to watch. It's a celebration of humanity, a celebration of great builders, a celebration of great engineering. So, it represents something special. So I hope you and NVIDIA keep going. What gives you hope about this whole thing we got going on, about humanity, about the future of humanity? When you look out, when you think about the future quite a bit, when you look out 10, 20, 50, 100 years from now, what gives you hope?

- I've always had a great confidence in the kindness, uh, the generosity, uh... the compassion, the human capacity. I've always been extremely confident of that. Sometimes more so than I should. And, and I get taken advantage of, but it doesn't, it doesn't ever cause me not to. I start with, always, That people want to do good. People want to, um help others. And, vastly, I am proven right. Constantly proven right. And, and often it exceeds my expectations.

And, and so I have complete confidence in the human capacity. I think the, the thing that, the things that give me incredible hope, Is what I see as, as I extrapolate, as I, what I see now is possible, and as I extrapolate, Based on the things that we're doing, what will very likely happen. And, and that there's so many things that we wanna solve.

There's so many problems we wanna solve. There's so many things that we wanna build. There's so many good things that we wanna do that are now within our reach, and within the reach of my, my lifetime. You just can't possibly not be romantic about that. You know what I'm saying? - What an exciting time to be alive. Like, truly- - How can- - ... truly so. - How can you not be romantic about, about that? The, the fact that, that there is a, there, it's a reasonable thing to expect the end of disease. It's a reasonable thing to expect. It's a reasonable thing to expect that pollution will be drastically reduced.

It's a reasonable thing to expect that traveling at the speed of light is actually in our future. And then, you know, not, not for long distances, but short distances. You know, and people ask me how. Well,

first of all, very soon, I'm gonna put a humanoid on a spaceship, and it's gonna be, you know, my humanoid, and, and we're gonna send it out as soon, you know, as soon as possible, and it's gonna keep improving and enhancing along the flight. And then when it's time, all of the, all of my consciousness has already been, you know so much of my life has been uploaded in the internet. Take all my inbox, take everything that I've done, everything I've said. You know, it's been collected and becoming my AI.

And I'm just, you know, when the time comes, you know, we'll just send that at the speed of light, catch up with my robot. - Oh, that's brilliant. I mean, but for me, that's sorta application-focused. But also, for me, the curiosity- ... Maxing perspective, I just, all of those mysteries. There's so much- ... fascinating scientific questions there. - Understanding the biological machine is right around the corner. It's, it's not 10 years. It's five years probably. - And then your biological machine, the human mind and cracking physics, theoretical physics open. It's so exciting.

- Explaining consciousness, that one would be awesome. - And it's all within our reach. Jensen, thank you so much for everything you've done over the years. Thank you for everything you're doing for the world. Thank you for being who you are. I can tell you're a great human being, and I wish you incredible success this year. I can't wait. As a fan, I can't wait to see what you do next, and hopefully I'll see you in Taiwan and thank you so much for talking today.

- Thank you, Lex. I had a great time. And also, if I could just say one more thing. - Yes. - And thank you for all the interviews that you do, the depth, the respect that you go through with and the research that you do to reveal, you know, for all of us, The amazing people that you've interviewed over the years. I've enjoyed I've enjoyed them immensely. And as an innovator, to have created this long form, unbelievable, and yet, you know, it's just captivating. So anyways, thank you for everything you do.

- It means the world. Thank you, Jensen. - Thank you, Lex. - Thank you for listening to this conversation with Jensen Huang. To support this podcast, please check out our sponsors in the description, where you can also find links to contact me, ask questions, give feedback, and so on. And now, let me leave you with some words from Alan Kay. "The best way to predict the

future is to invent it." Thank you for listening, and hope to see you next time.