Robots – Science and Fiction

Start

Everything stated in this presentation is explicitly NO financial advice. Because the stock market is unpredictable, but pace of innovation is the only thing that matters in the long run.

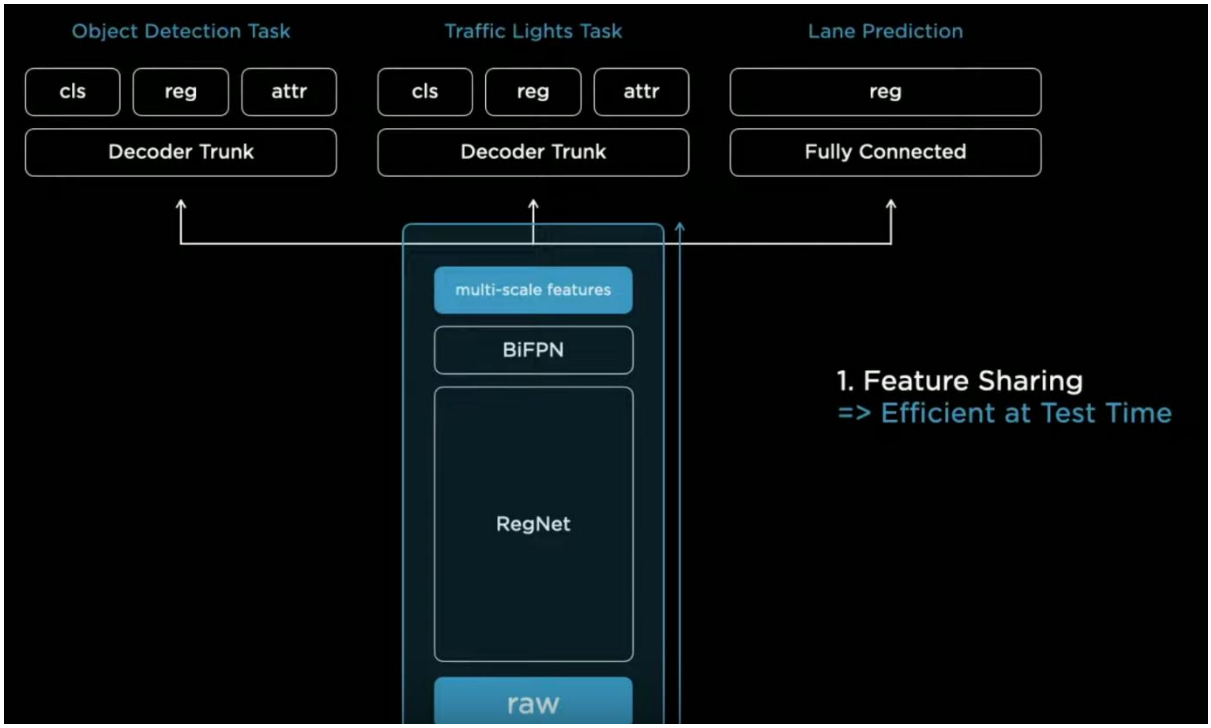
Or as a certain AI would put it:
"The data and analysis presented in this presentation are not a guarantee of future performance and should not be taken as financial advice. There is always room for surprise and disruption, especially in industries like technology and energy."

Training

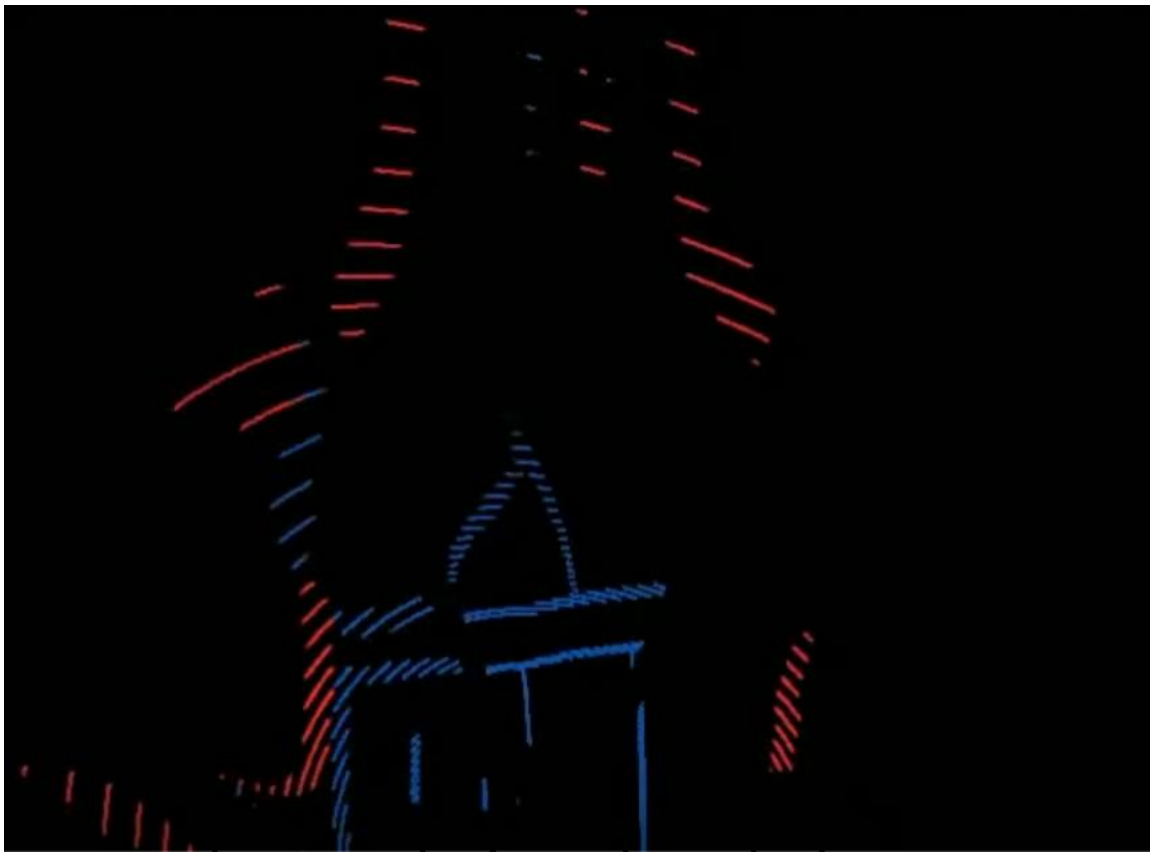
Some modern robots already function with a neural net. This is a simulation of a brain in 1 and 0. It works by creating neurons, which are value points that are interconnected, creating value strings. Those interact with each other and the data it has to process. It's a very complex algorithm that cant be fully understood by humans. And we don't have to. We only need to know how to train it. So, how does training work? We train it by letting it process data, generating better versions of it based on the results, letting them process data and making new versions of the best versions and so on. What sounds simple takes an enormous amount of compute power. But, since a NN works the same way as a human brain, it can do anything. From extremely complex calculations to playing chess and driving cars. Even human behavior is possible nowadays.

All this is achieved thru several layers within a neural net. In essence, there are several interconnected layers in a net, each performing a different task, such as translating words to 1 and 0, making sense of all that and generating a response. This is an abstract high level overview. We can already see this in action, as ChatGPT or Stable defusion for example. Many industry experts try to play those systems down, saying anyone could have done it. But in my opinion, this is not the point. The point is, it exists, and we are today on a level where it truly can – and will – grow exponentially. There are no boundaries anymore, not within what we as human beings can think off. Because artificial perception is solved. There won't be a stagnation anymore, since the system can now fully adapt.

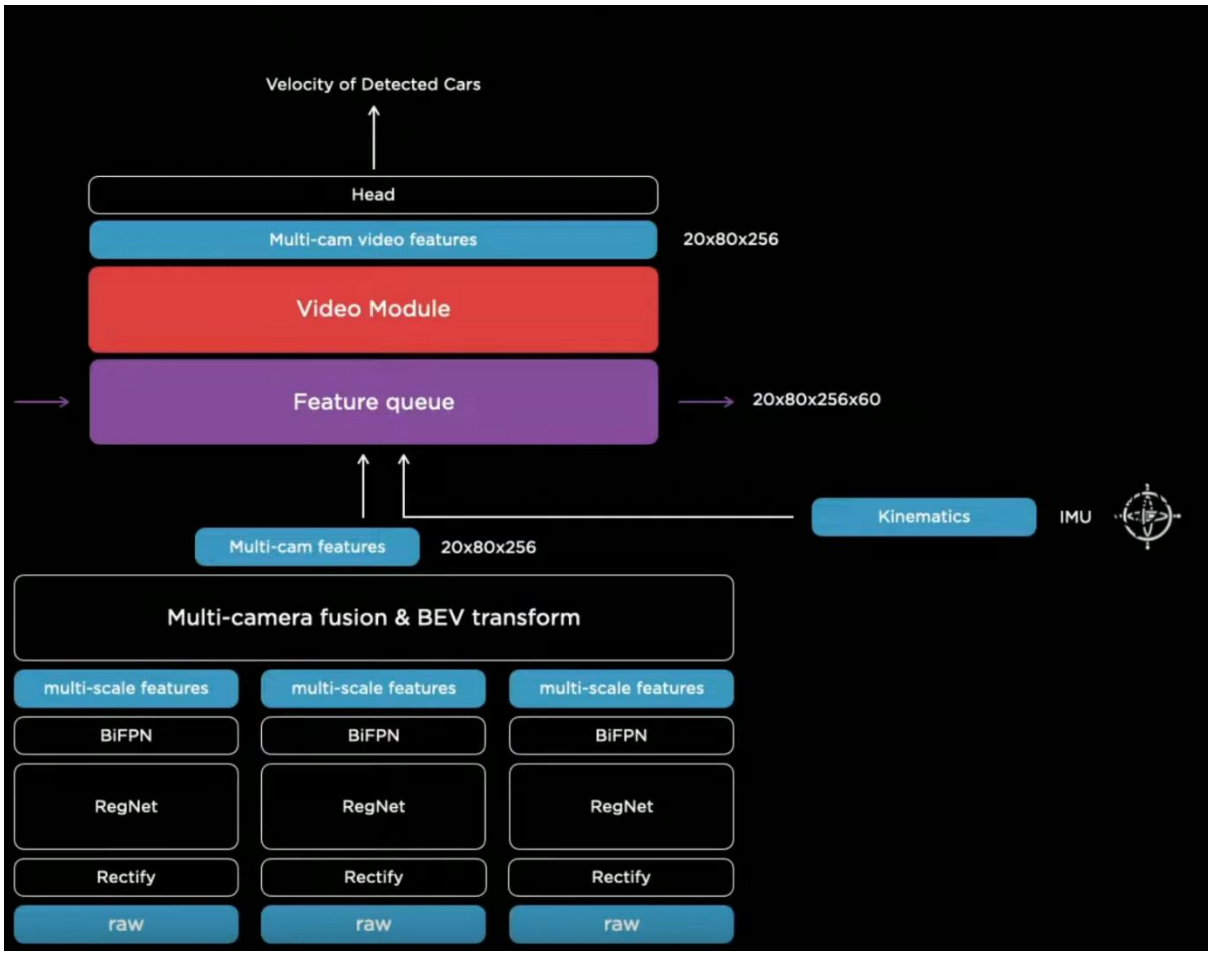
What do I mean by stagnation? Teslas progress in training its self driving car first went up exponentially, but then came to a dead end. This is due to the structure and training of the NN being too specific, to good at one thing, so it cant be good at everything, basically. Tesla faced this problem several times, which lead them to the decision to basically start all over again with FSD. So, why will FSD not bring the same problem again?
This is a simplified graphic showing the Autopilot stack, the old software:



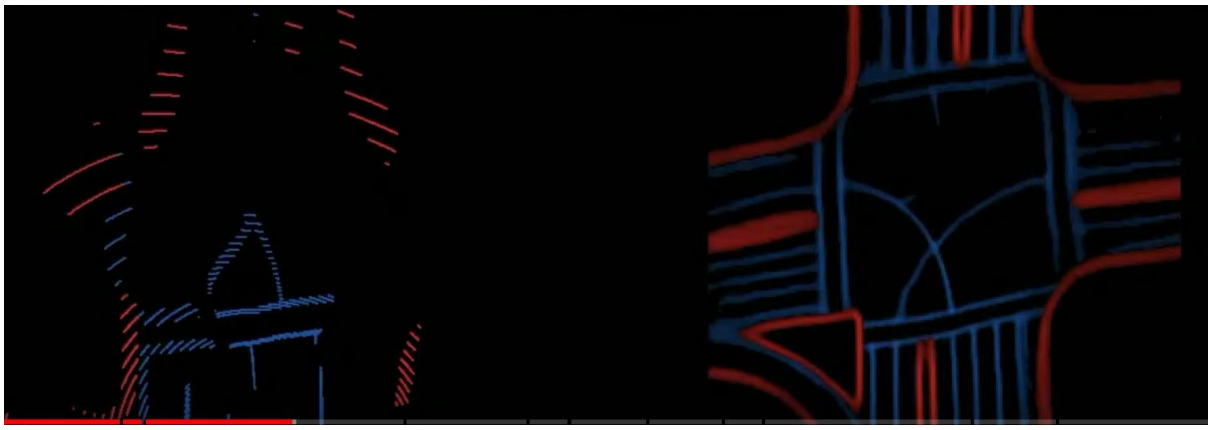
Basically, one NN backbone fed data to many “substacks”, and only did this picture by picture. Plus, pictures were evaluated in 2D space so the single backbone could even do the job, which made it hard for the car to understand its surrounding. As a result, this is what an intersection looks like to Autopilot in birds eye view:



In contrary, this is the simplified FSD stack:



It takes the opposite approach, which allows it to model the world in a 3D VECTOR space. This way, objects can be identified way easier and the car has basically the same view of the world as a human driver, meaning it can adapt to situations it has never been in before. This is the same intersection, Autopilot left, what FSD sees on the right:



Why is this important? It shows not only the development, but also that the car now really knows what's going on. It actually sees the world.

So, this has to somehow be trained. The main bottleneck for training besides compute power is data. You can only train a NN with data you actually have. So, how does this work?
I am going to compare Tesla and Waymo for this example.

Waymo gathers training data on secure, private test tracks with only few cars and only from time to time. This is limiting in many ways:

1. They can only get so much data in a specific time frame
2. Their training data does not contain real world traffic, nor edge cases.
3. This is why they still do not realize that vision is the key to real world AI

Therefore, they rely on LIDAR, which means they have to give their cars a precise map of the area they have to drive in. The car does not understand the world it lives in, nor its direct surrounding. It can only function in a premapped area, as long as this area does not change.

Tesla on the other hand gets constant feedback from every car they sell. And they have one of the most capable supercomputers for training, so they can handle this massive amount of data. Tesla has so much of it, that they were able to train a separate AI on making realistically simulated traffic situations, for getting even more data on edge cases, based on existing ones.

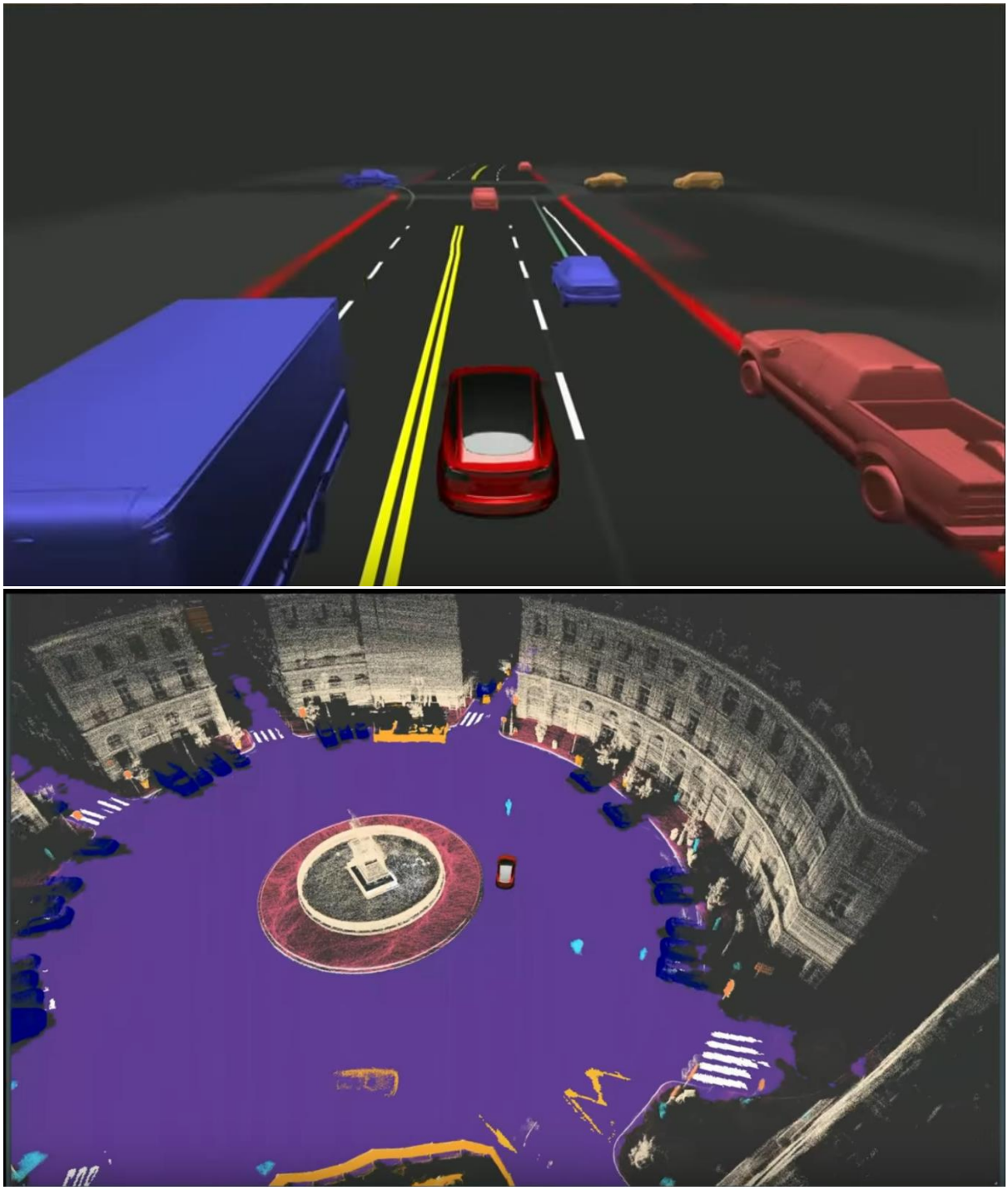
In essence, you need real world data from a fast feedback loop.

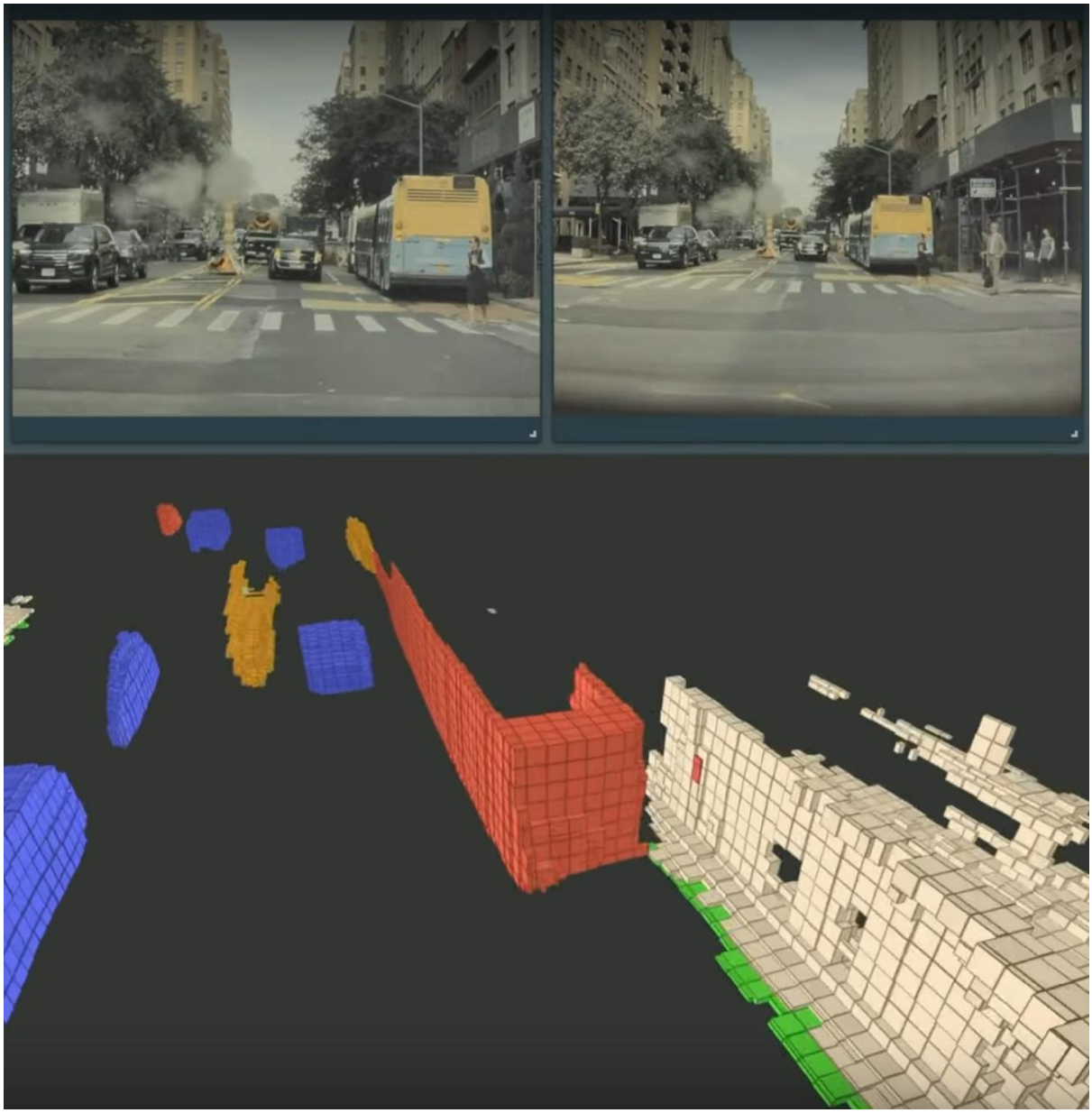
This creates the perception I just talked about and which we will have a closer look at now.

So, lets get into perception for a bit:

Like in our example, there is a difference between perception and simple processing. While you can write or even train software to measure distances, compare them to a DB and act accordingly, the right neural net has an understanding of what's going on. The electrical interaction between the neurons creates a complex, self-thinking system that is able to make intelligent decisions and handle almost any given situation.

We can see this very clearly when we look at Tesla's AI. These pictures show, what three different specific layers of FSD see in real time:

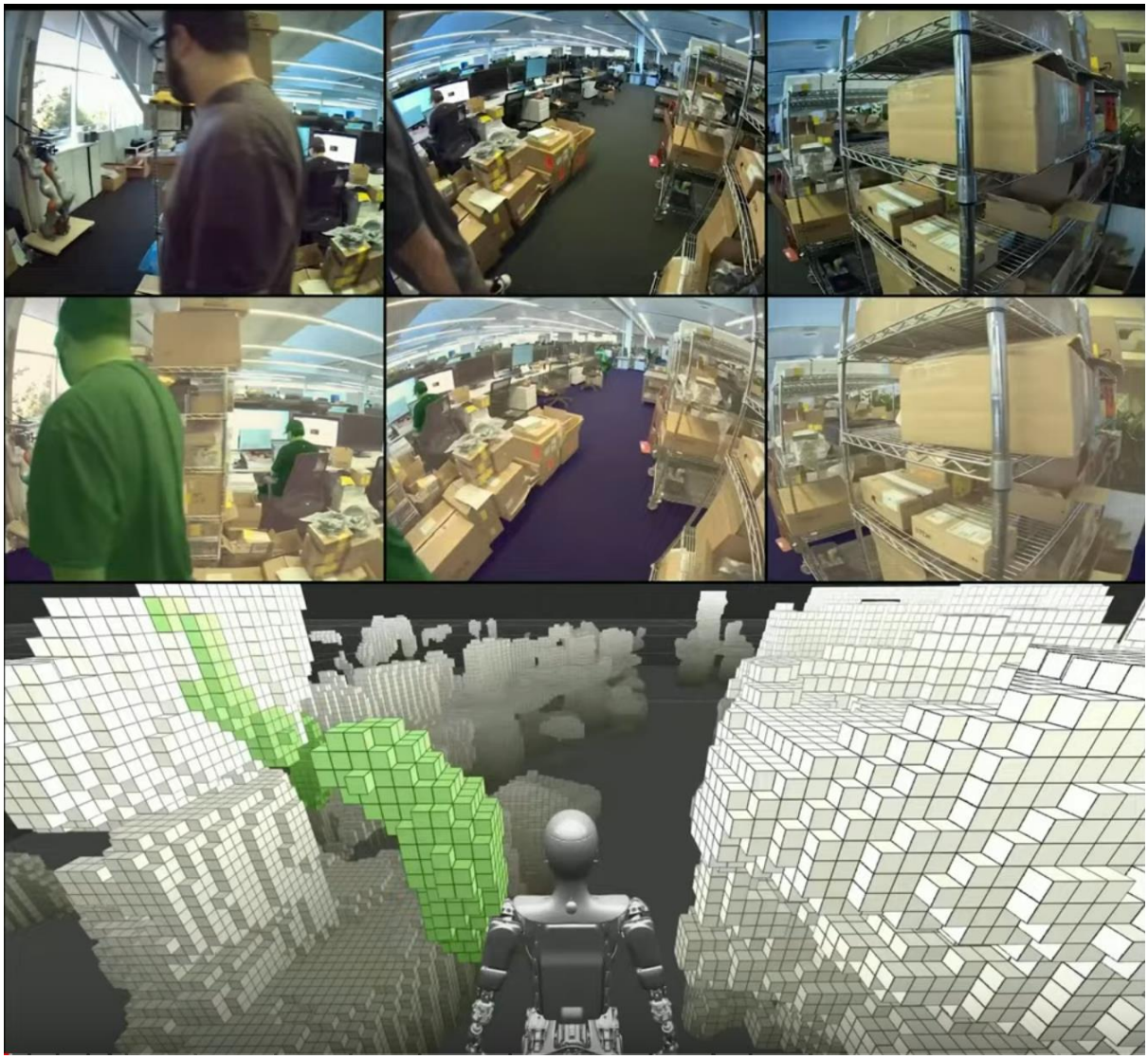




Note that it can detect speeds and distances based on its know-how and that it color codes objects and also works with forms and relative size. It does not need to know this specific model of a bus, it just recognizes a big, heavy, moving object and knows how to handle it.
This is not a visualization for presentations sake, this is what the software actually sees.
The FSD stack sees things in a 3d vector space. So it moves thru a room of relative distance and speed among various objects of different types.

ChatGPT works basically the same way in terms of the basic thinking process. It does not know every possible combination of words, but it can create them at any time.

To visualize the scalability of this approach, let's have a look at the Tesla bot. This thing can navigate in a much more complex world than a car needs to, and it can interact with its surrounding. This is what the robot sees:



It works quite similar to FSD. This is due to the fact that it is the same NN, with different training. Again, this is what the machine actually perceives. This is real world AI. The robot can interact with objects or even entire types of objects it has never seen before, and it also profits from fleet learning, just like the cars. This NN even runs on the same chip as the one in the car.

The Waymo equivalent here is Boston Dynamics for example, who solved precise movement already, but cannot fulfill almost any real world use case.

Production

There are basically two parts in production of an AI: Hardware, and software.

Hardware

State of the art

Fist, what's the state of the art in AI hardware?

In terms of training supercomputers, there is Dojo. Lets get into that.
Dojo is based on the D1 chip, which has equivalent power to an A100 chip used in modern GPU superclusters. Other than a GPU cluster, Dojo is a "sea of nodes". What does this mean? A D1 has the same compute power as an A100, but it has 800 MB of SRAM. For comparison, the highest

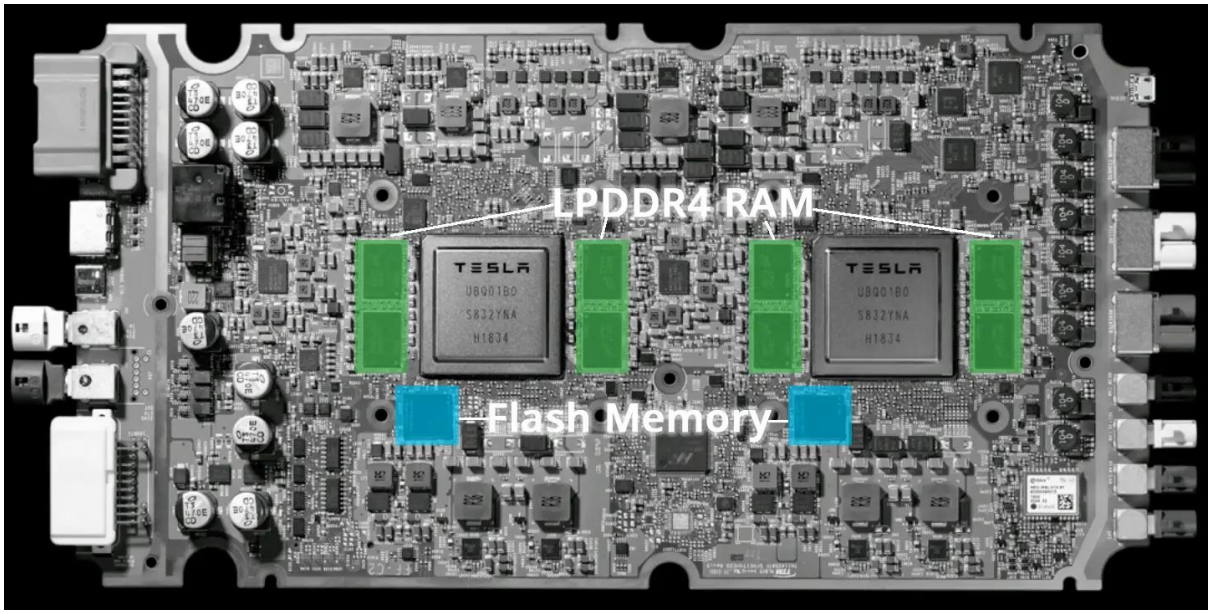
amount of SRAM in flagship consumer CPUs today is 80MB And the L2 cache of the 8th gen Nvidia A100 is 40MB. In a GPU cluster which was the best solution before Dojo, Many GPUs split their work and share the results via low bandwidth, high latency connections for final compiling. This way, each GPU profits from its high DRAM, but has to wait very often and for a very long time. In Dojo, 25 chips sit on a tile, featuring extremely fast connection between them and 20GB SRAM as well as 1 Petabyte compute per tile:

Those tiles sit on a bigger equivalent, which interconnects tiles with extremely low latency, and those are stacked in racks. 1 Exaflops in compute power, equal to 10'000 human brains, or 10 to the power of 18 1 and 0 per second, which is one quintillion bits per second or one billion billion bits per second, takes only 1 rack of space, and is faster than any other exaflop in the world, due to the ability of each chip to talk to its friends fast and efficient. This is also modular and scalable. Tesla plans to scale its Dojo from 1 to 7 exaflops within the next several years.

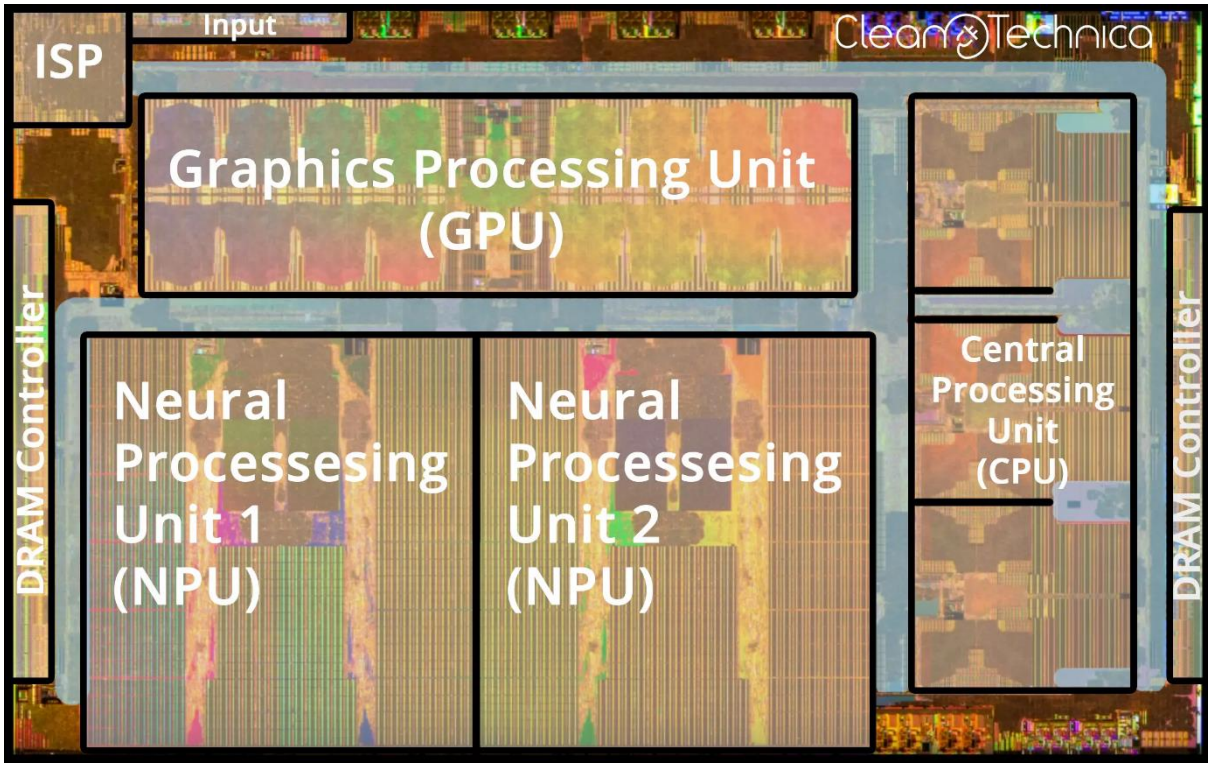
In short, how can one amount of operations per second be faster than another with same flops? That's because the data processed by this amount of FLOPS can be communicated faster between chips.

But what's up with the hardware that actually runs AI in the real world?

This takes way less compute power than training, since it does not have to build new nets, but simply execute on existing ones. What kind of hardware is needed depends on the use case. Stable Diffusion needs graphics processing power, since it generates images, while the C++ code might as well run on a CPU. But to stay with our example, Tesla designed a chip that is featured in every car they build, called hardware 3:



It features two identical, redundant chips on one single board. The chips got one piece of flash memory right next to them, shown in blue, as well as two times two LPDDR4 DRAM chips with a bandwith of supposedly 68 gigabytes per second on their side, shown in green, and feature 32MB of SRAM each. Note that this is not a simple processor, as you can see here, it is actually a Soc, a system on a chip:



Like in a smartphone, just on steroids. This is important, because graphic intense tasks can be done by the GPU, which is capable of 600 GFLOPS, or 600 billion bits per second, equal to 60 human brains. There are also some general-purpose processing tasks unsuitable to the neural processor that are done by the CPU. The way Tesla explained it, there are 12 ARM Cortex A72 64-bit CPUs in the chip running at 2.2Ghz. This seems strange, but remember that this hardware is already several years old, and HW4 is just about to be released.

I think low latency, all in one chips, pared with some form of redundancy is the way to go for running complex real world AI, so it can make big decisions within milliseconds.

Software

"The moving parts"

So, we already saw what a NN looks like on paper. And on the machine this is just an enormously complex flow of 1 and 0, we all get that. But there is actually more to it. First, there is the software that makes and trains a NN. I got some code tinkered together that does not actually do anything meaningful, it probably does not even run, but it gives us a good enough picture of what things might look like. This is written in PyTorch, an open source Python Library for NN. This code supposedly creates an AI and also trains it. Those who are interested can get this from me as a .txt file.

NNs (and their helpers(not?))

But wait, there is more. NN are not yet the ultimate superpower. For communication between stacks and control over hardware, it still takes human written code.

There is still no "single stack solution". AI generally consists of multiple stacks that are interconnected thru handwritten code, such as C. They also talk to the hardware thru handwritten adapters. First steps are being made into the next big thing, as parts of Teslas AI for example already produces output that can be directly understood and executed by the hardware itself. This allows for even lower latency, and is the first step to an AI system that can truly and fully control its robot.

State of the art

Another part is the compiler, which has to be able to handle AI on your specific supercomputer. For Dojo, Tesla wrote its own compiler that is able to handle the whole Dojo as one processing unit and distribute the work load as needed. Then, there is the NNs structure. As I showed before, there are many ways to build a stack, each with its own disadvantages. The main problem here is that we as human beings are simply to primitive to fully grasp even the potential, let alone the question what the best structure would be. In the future, AI will have to build better AI for us.

Why is this important for us?

There are several steps that we still need to make for getting to AGI. The first step is software 2.0, or *English as a programming language.* This means, you don't have to actually do the programming or the image refining, but you can just tell the AI what the result should be and it gives you 20 outstanding results within seconds. Whoever masters this, whoever knows how to use this for super productivity, will be able to do much more in less time than everyone else. This step is important to learn more about how to make AGI and how to handle it, but also to gather data for training, and proceed even faster, thru AIs help.

The second step are "*single stack solutions",* Where the whole thing is one software stack to rule them all. It is not yet fully clear if this is a good idea, but even if not, the basic principle stays the same. Less parts, less processes, more efficiency. This is crucial for our hardware to be able to

handle future AI systems, even tho our hardware already scales exponentially. You saw why at the beginning.

The third step has to happen simultaneously to step 1 and 2. Its what I call hardware 2.0. Modern hardware does not get better thru smaller chip architectures, since this is not physically possible any more, or any further. Some manufacturers solve this with way higher clock speed, and more DRAM. However, chip designers working for AMD or Tesla, found a more efficient, more scalable way Besides AI acceleration, which is very effective in itself: They give their chips as much SRAM as possible. The flagship Intel consumer processor features over 6000 Mhz DRAM speed and 32 MB of cache. The flagship AMD consumer processor features below 6000 Mhz DRAM speed, but 80 MB of cache. Both products perform almost equally, yet the AMD chip only uses 50% of the power that is needed for running the Intel chip, plus AMDs solution is much more scalable, which already today comes to play when one does so. Tesla used to train its neural nets on a NVIDIA GPU cluster, with high DRAM on each card and high latency between cards and in the cards themselves. Their new in house designed Dojo super computer features comparable performance per chip, but with less DRAM and way more cache per chip, while the chips are gathered on tiles, which in themselves work as a higher level chip, but with 20 GB combined cache. This monster reduces training time from several months to 2-3 weeks, despite “only” offering 1 Exaflops.

So the industry needs to and will shift to high cache low latency chips, which again is crucial to understand for creating training scripts and low level system software.

The final step is merging. Maybe to the “one stack to rule them all”, or to a kind of a stack of stacks that can do everything. By then, it is crucial to understand how things work, so we can guide this development in a direction we like, as a species.

And on the way, mastering AI helps you stay relevant in IT.

Why is it important (TLDR)

To reach AGI, there are four essential steps. The first is software 2.0, which enables you to program using English, making it more efficient and saving time. Second, "single stack solutions" streamline the software stack and increase hardware efficiency. Third, hardware 2.0 uses chip architecture with more cache, which reduces power usage and makes AI systems more scalable. Finally, merging various software stacks into a comprehensive system is the last step. It's important to understand these steps to guide AI development towards a beneficial direction for humanity. Additionally, mastering AI is necessary to remain relevant in the IT field.

End

The integration of robots and neural networks marks the beginning of a new era in technology. With limitless potential, we have the power to shape the future and create a world that is more efficient, creative, and compassionate. Let us embrace this opportunity to push the boundaries of what is possible and strive towards a better tomorrow, powered by AI.

The pictures shown in this document are screenshots of parts of Teslas past presentations. I highly recommend watching those.