TELSA AUTOPILOT

BREAKING DOWN TESLA'S AI DAY PRESENTATIONS (PART 1I)

Gene Olafsen

OVERVIEW

- Part I
  - Tesla Vision
  - Planning and Control
- Part II
  - Manual Labeling
  - Auto Labeling
  - Simulation
- Part II
  - HW Integration
  - Dojo

ACKNOWLEDGEMENT AND REFERENCE

- The screenshots and content is largely taken from the AI Day presentations.

- Tesla AI Day - YouTube
- 1612.03144.pdf (arxiv.org)

DATA

- The story of datasets is critical.
- The hundreds of millions of parameters must be set correctly for the neural networks to make 'correct' predictions.
- Datasets in the vector space must be clean and diverse.

TOPICS

- Manual Labeling
- Auto Labeling
- Simulation
- Scaling Data Generation

LABELING HISTORY

- About four years ago, Tesla was using a third-party to obtain datasets.
- High latency to get the datasets and the quality was not 'amazing'.
- In the spirt of full vertical integration at Tesla, the data acquisition and labeling task was brought in house.

VERTICAL INTEGRATION

- Currently there is a 1,000 person labeling team that works very closely with the engineers.
- All the infrastructure that supports the labeling process was built from scratch.

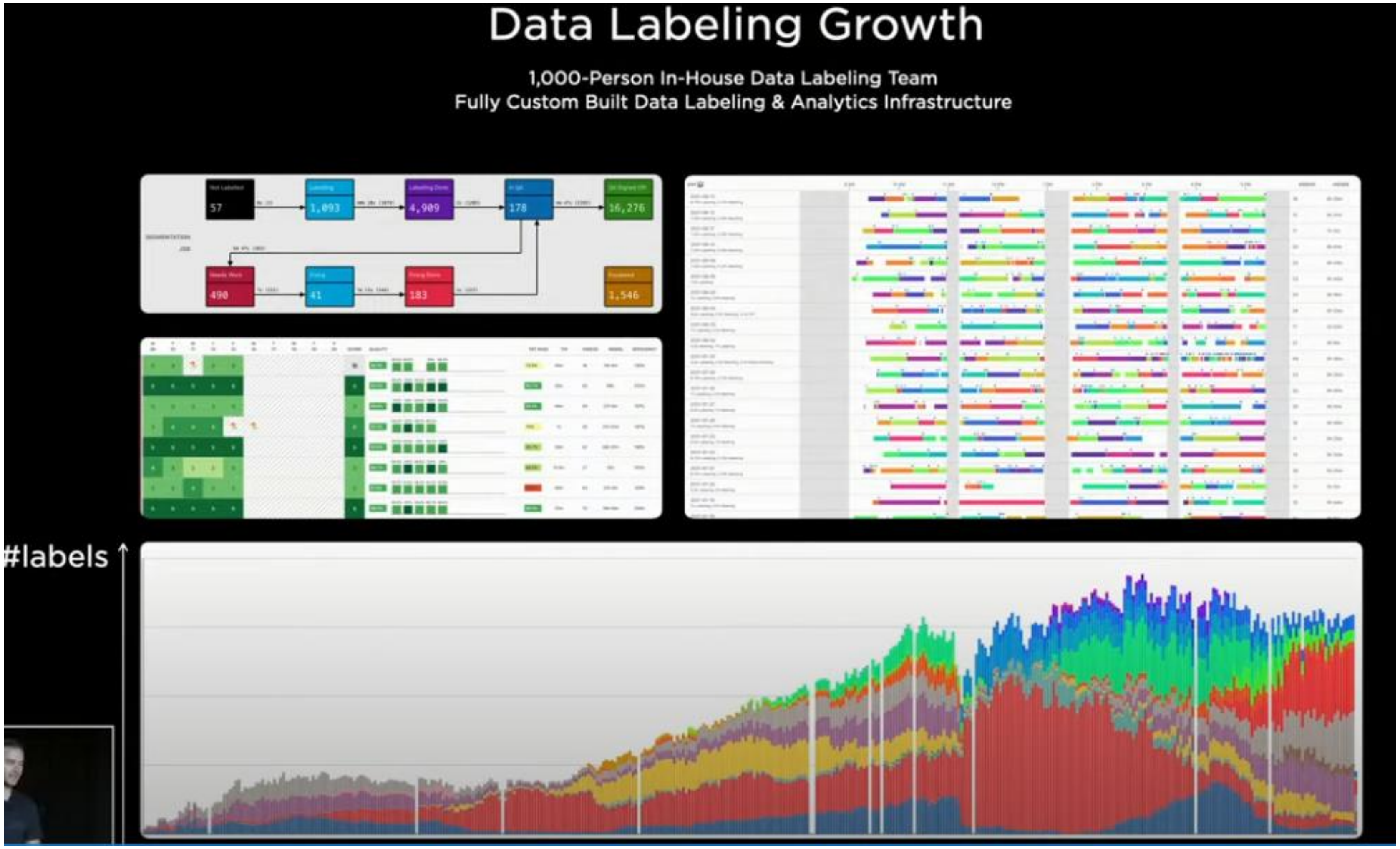


IMAGE SPACE LABELING

- Four years ago, most of the labeling was performed in image space.
- A lot of time is spent annotating and drawing bounding boxes around objects.

LABELING

- Directly labeling in vector space.
- This is a reconstruction of the ground plane on which the car drove.

HUMAN/COMPUTER COLLABORATION

- The labels are produced in vector space and being re-projected into the images.
- This system increases labeling throughput by 100X.
- However, this was not good enough because people are good at semantics while computers excel at geometry, reconstruction, triangulation, tracking.

AUTO LABELING

- The task of training the network requires many more human labeling experts.
- Tesla has invested in a massive auto-labeling pipeline.

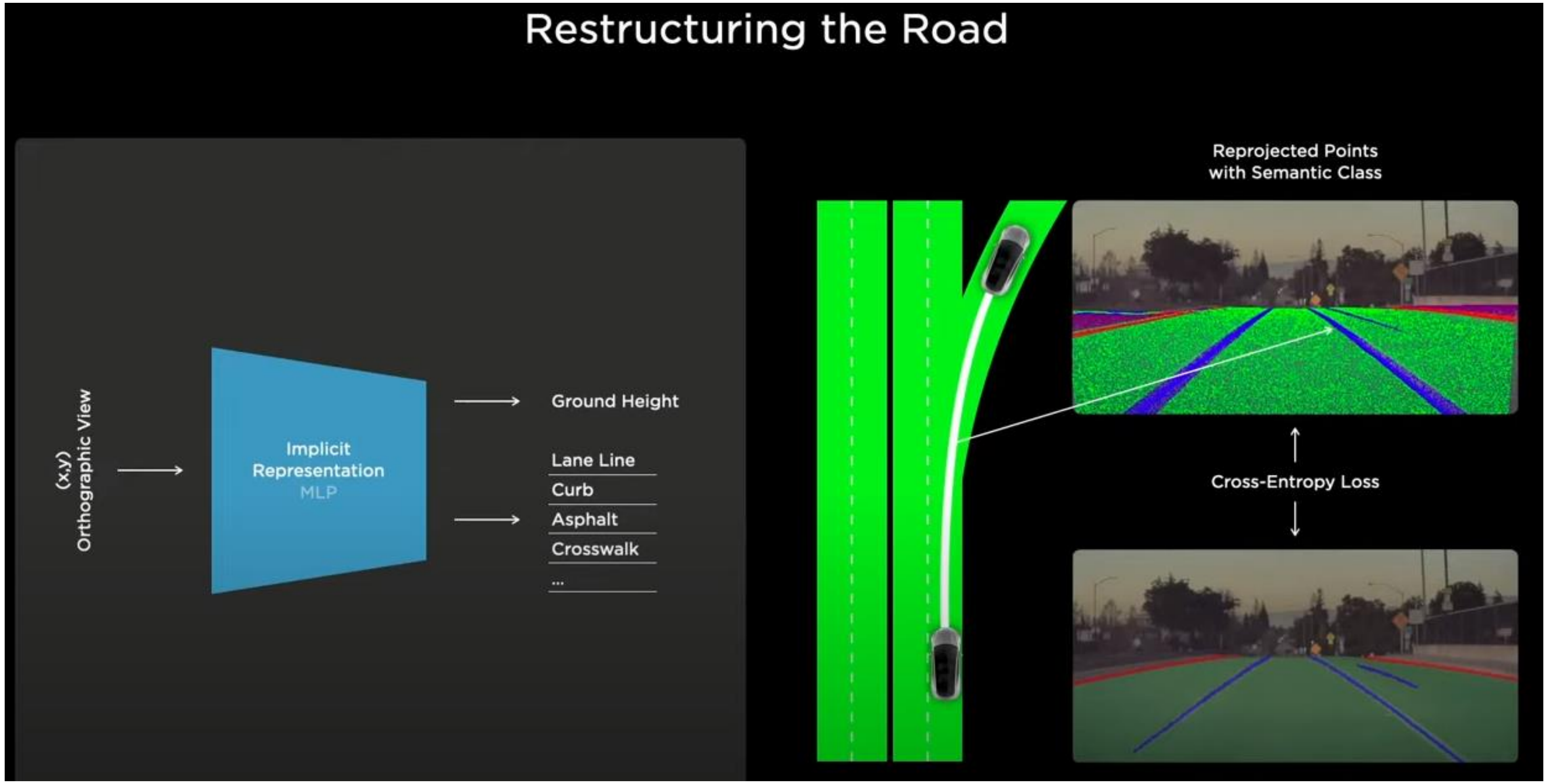
LIFE OF A CLIP

CLIP PROCESSING

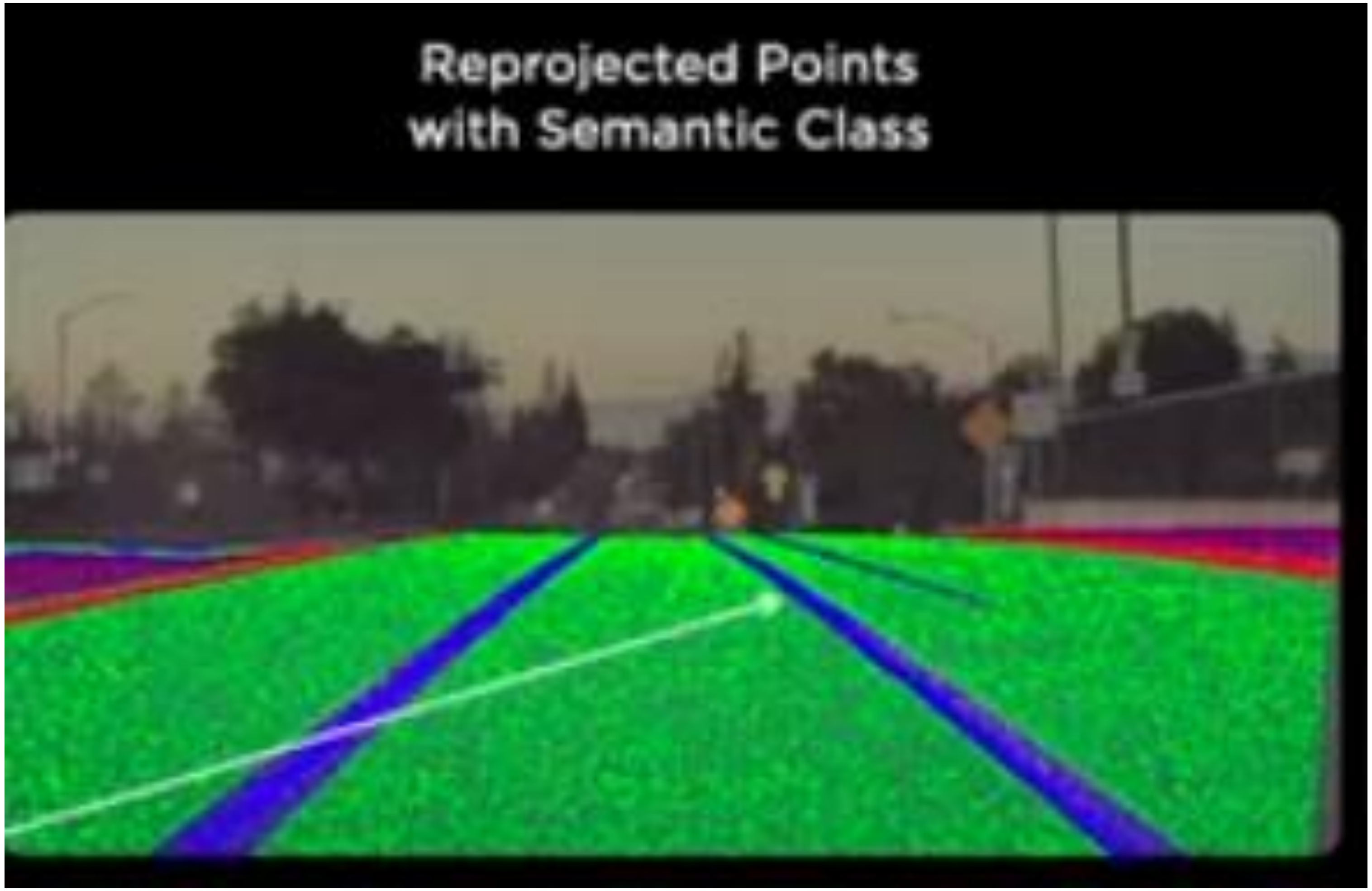
- A clip has dense sensor data and may contain up to a minute of video.
- Acquired from either engineering test cars or customer cars.
- Pipeline servers produce segmentation masks, depth, etc.
- Then on to other processing which creates the labels and produces the data to train the networks.

ROAD IDENTIFICATION

- The first task is to label the road surface, which can typically be represented by splines or meshes.
- Tesla uses a technique that queries XY points on the ground and asks for the height as well as various semantics: curbs, lane boundaries, etc.
- Given an XY, you get a Z and this 3D point can be projected into all the camera views.

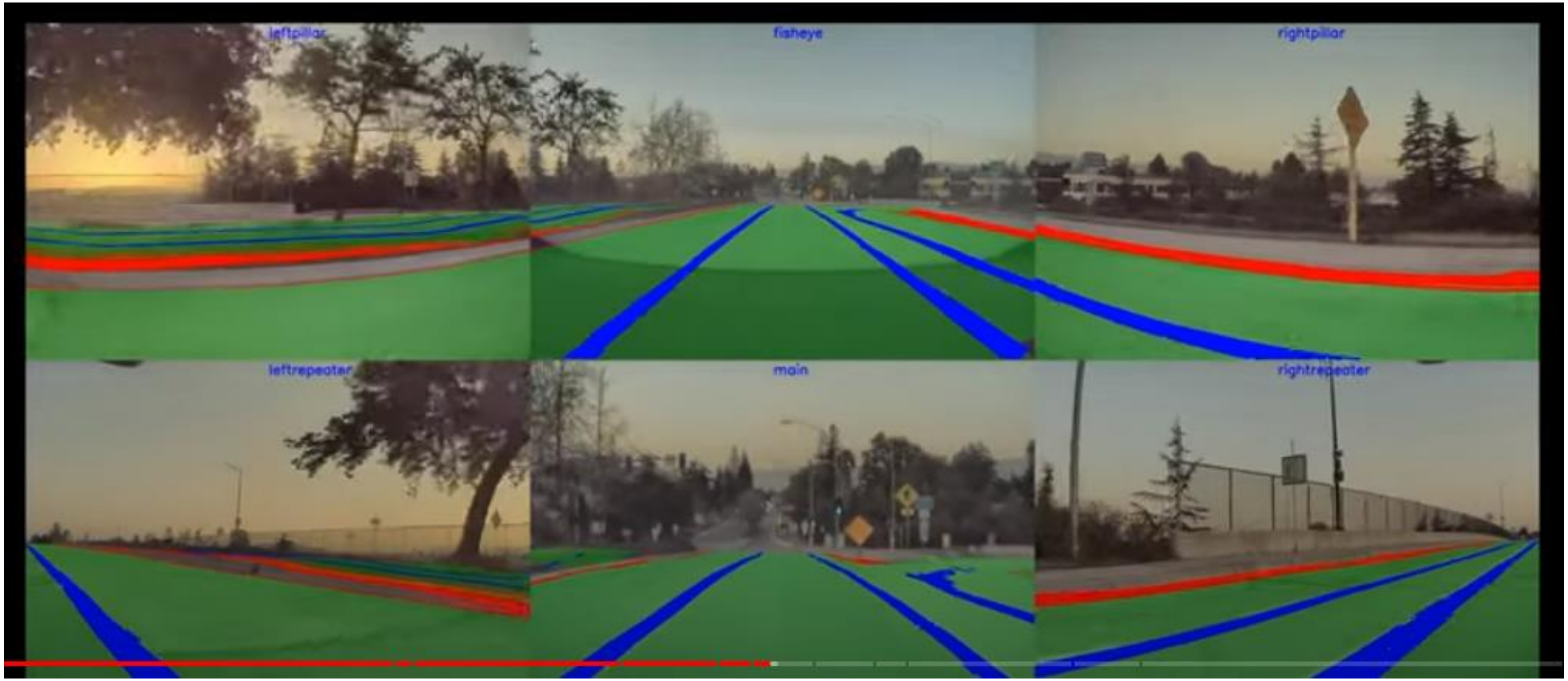


POINTS



- The system makes millions of these queries and calculates lots of points. And all the points are re-projected into all the camera views.

RECONSTRUCTING THE ROAD



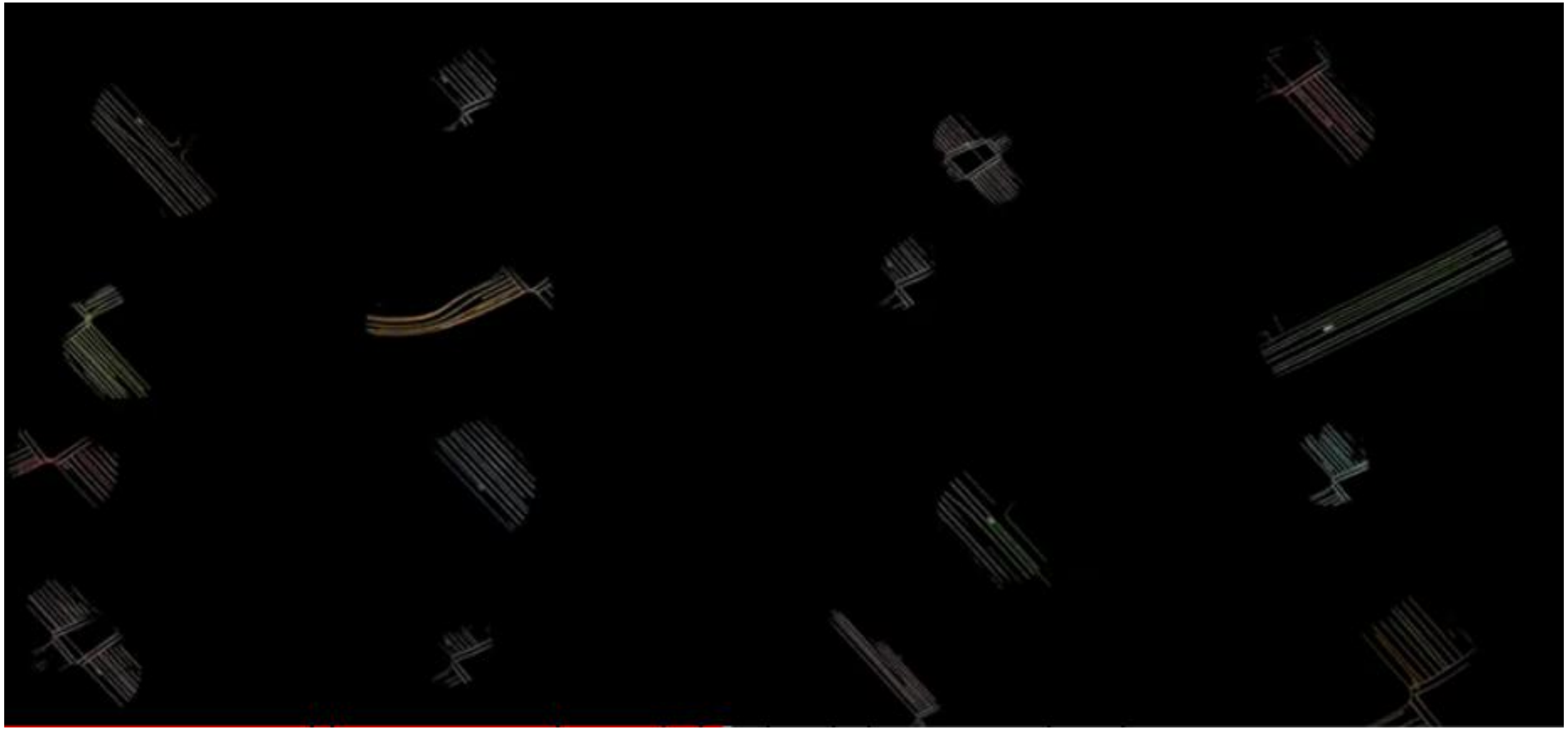
- The points are consistent across space and time.

RECONSTRUCT THE ROAD FROM A SINGLE CAR

- Using this technique a car can map out the path around the car.

COLLECT DIFFERENT TRIPS IN THE SAME LOCATION

- A location can be collected by the same car or different vehicles.



COMBINED TRIPS

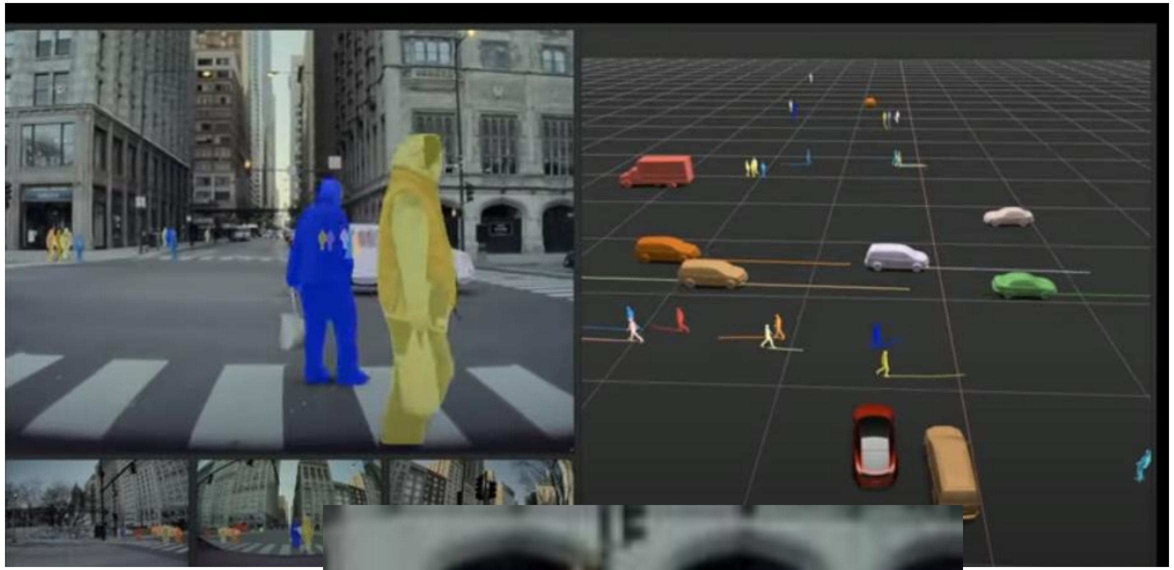
- Here 16 trips are composited together for the same intersection.
- This labels both where the car drove, but other parts of the road as well. A good way to check that the labels of the points from other vehicles are in agreement.
- Human labelers can then clean up any noise.

WALL, BARRIERS, ETC.

- A reconstructed 3D point cloud of walls and other obstacles.
- Notice the high density of the point cloud.

OCCULED ACTORS

- The advantage of working with the data offline, is the benefit of hindsight.
- The velocity of any actor can be tested by predicing the velocity, acceleration, direction, etc. and then comparing the guess to the actual values.
- The system can even predict actors that are occluded.
- The planner needs to know these possible behaviours, even if they are occluded.



COMBINED

- Putting it all together.
- Tesla trains on a million+ clips.

NO RADAR

- Truck dumps snow from roof on a moving car.
- The car does not 'remember' the car in front of it in poor visibility.
- Tesla removed RADAR within three months

SIMILAR EVENTS

- They had their fleet of vehicles find similar conditions.
- 10K similar clips were collected and labelled in a week.

VEHICLE PERSISTENCE

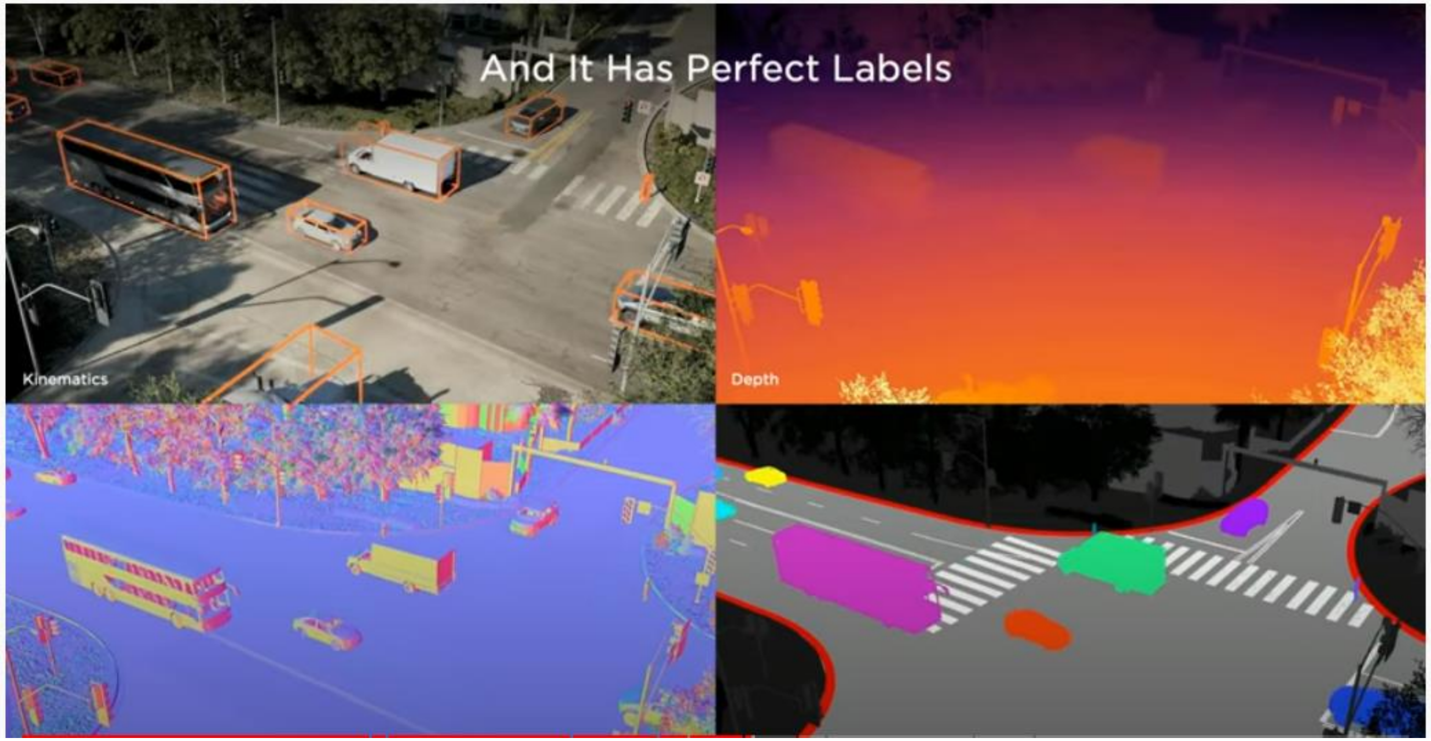
- The system now remembers when conditions deteriorate.

SIMULATION

- Point cloud data can be used to create a simulation from any camera angle.
- Autopilot is controlling the car with the icon over the roof.

PERFECT LABELS

- The simulation space has perfect labels.
  - Kinematics
  - Depth
  - Surface Normals
  - Segmentation



WHY SIMULATION

- Scenes that are rare or situations that need to be considered.
- Scenes that could take 'forever' to label.
- Vary situations with small adjustments.

ACCURATE CAMERA SIMULATION

- The simulation must match what the real cameras 'see'.
- Model properties of the camera.
- The simulation can even be used to help with sensor design and placement.

PHOTOREALISTIC

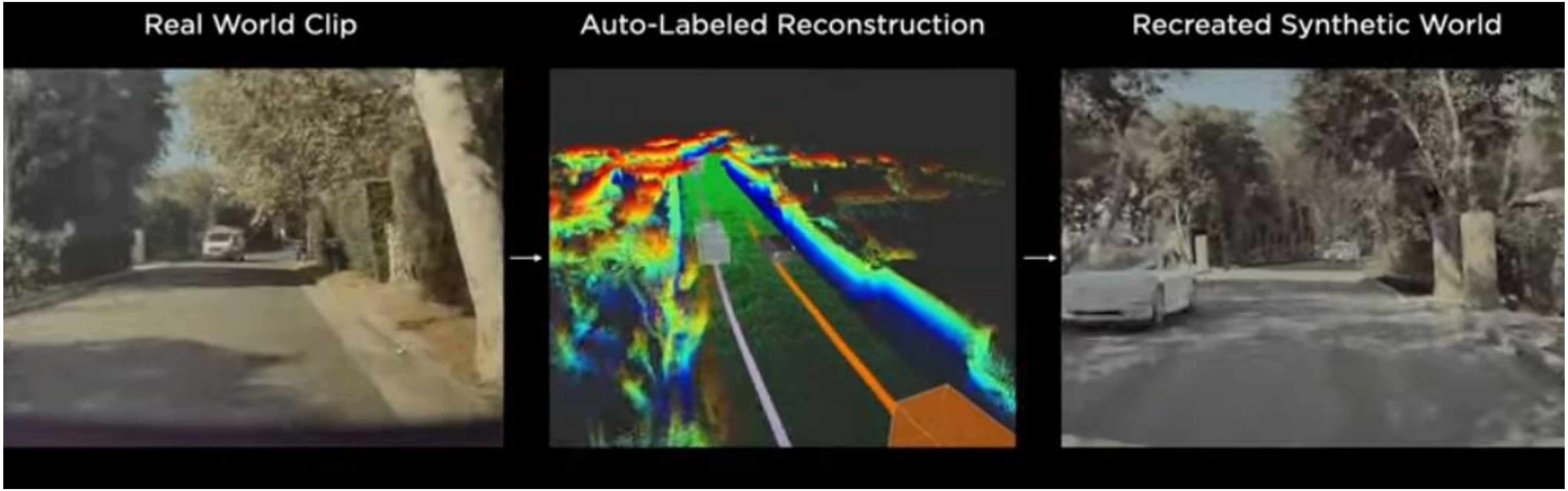
- The simulation is ray traced.
- The goal is to be visually indistinguishable from reality.
- This rendering system also has a NN stack to add more realism.

DIVERSE ACTORS AND LOCATION

SCENE GENERATON

- Scenes are generated by hand, proceduraly and ML-based adversarial.
- When Autopilot encounters a situation where it fails- many more scenes are created around the failure point to learn/train solutions.

REAL -> SYNTHETIC



- Building a pipeline to replicate scenarios and environments anywhere a Tesla vehicle has driven.

ENHANCE WITH NEURAL RENDER

- Lefthand side was captured by the cameras.
- Righthand side is rendered from the simulation pipeline.

SIMULATION TODAY

- A half-billion labels.

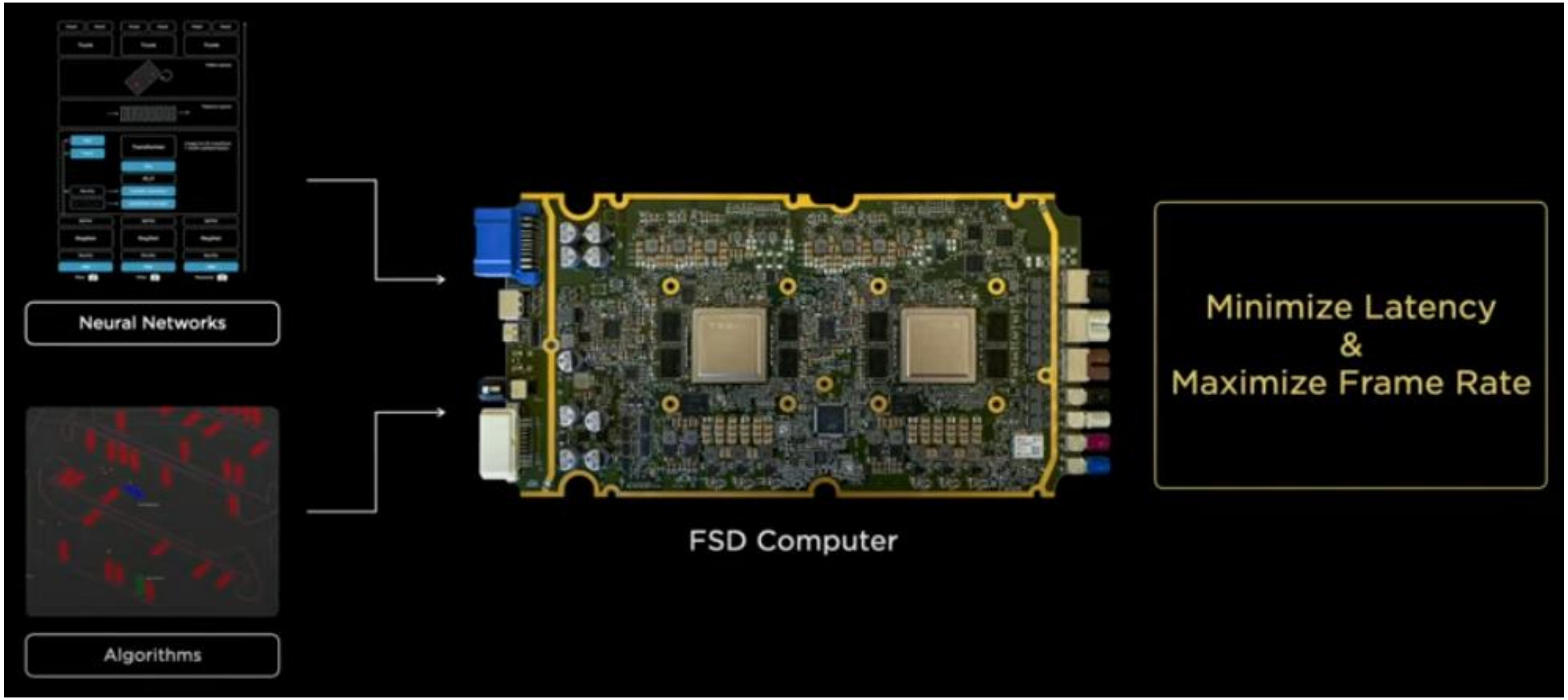
FUTURE SIMULATION

- In the next several months, these are the tasks that will be included.

SCALING DATA GENERATION

- To get rid of the RADAR sensor
  - 10+ billion labels
  - 2.5 million clips
- Compute was scaled across thousands of GPUs and about 20K CPU cores.
- Included in the comput loop were over 2,000 Autopilot system cores.
- This is Tesla's smallest comput cluster.

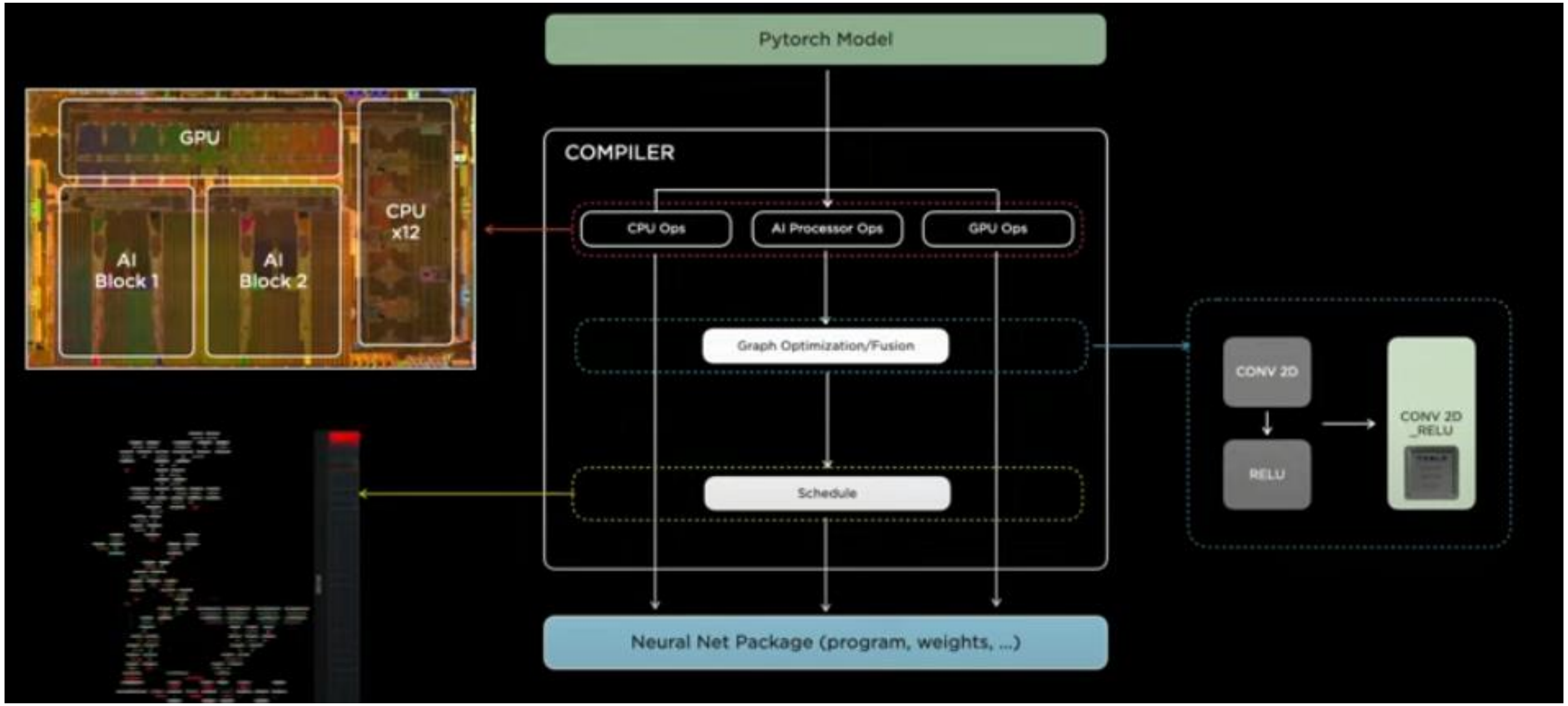
HARDWARE INTEGRATION



- Minimizes Latency and maximizes framerate

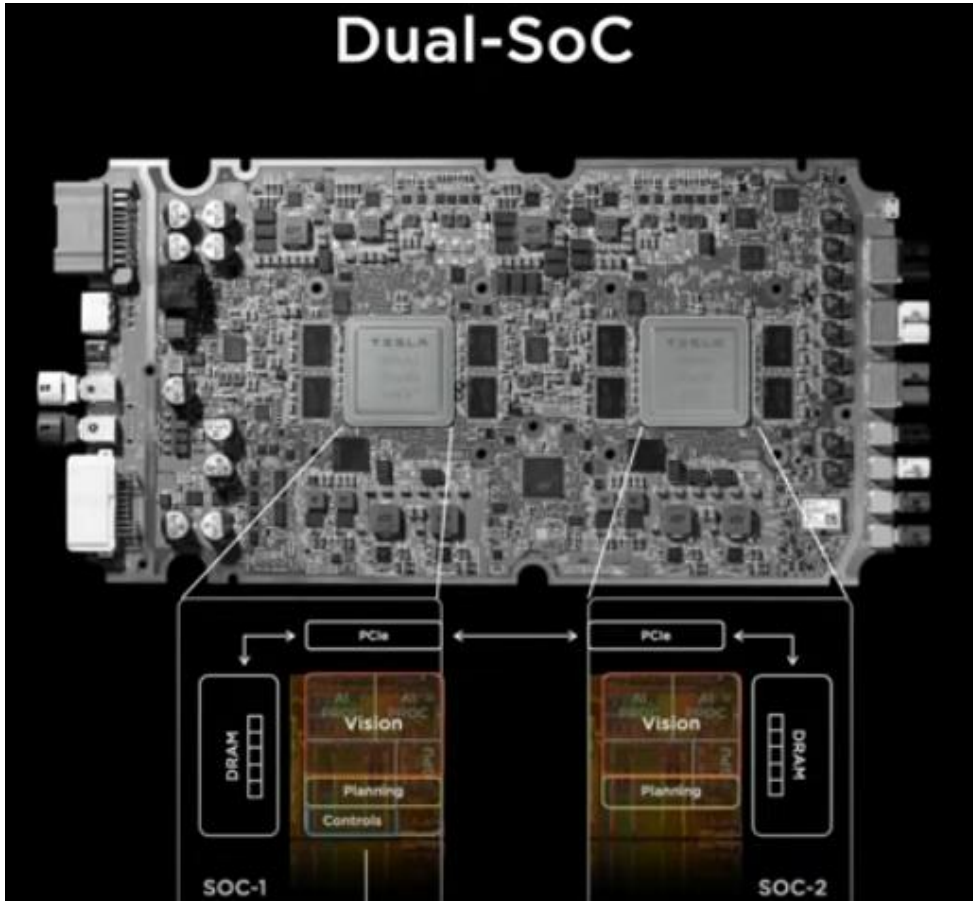
NEURAL NET COMPILER

- The jobs that need to be run on the vehicle are scheduled for throughput.



TWO COMPUTE ENGINES

- Only one has control of the car at a time.
- The other is used as a compute extension.
- Those roles are interchangeable.

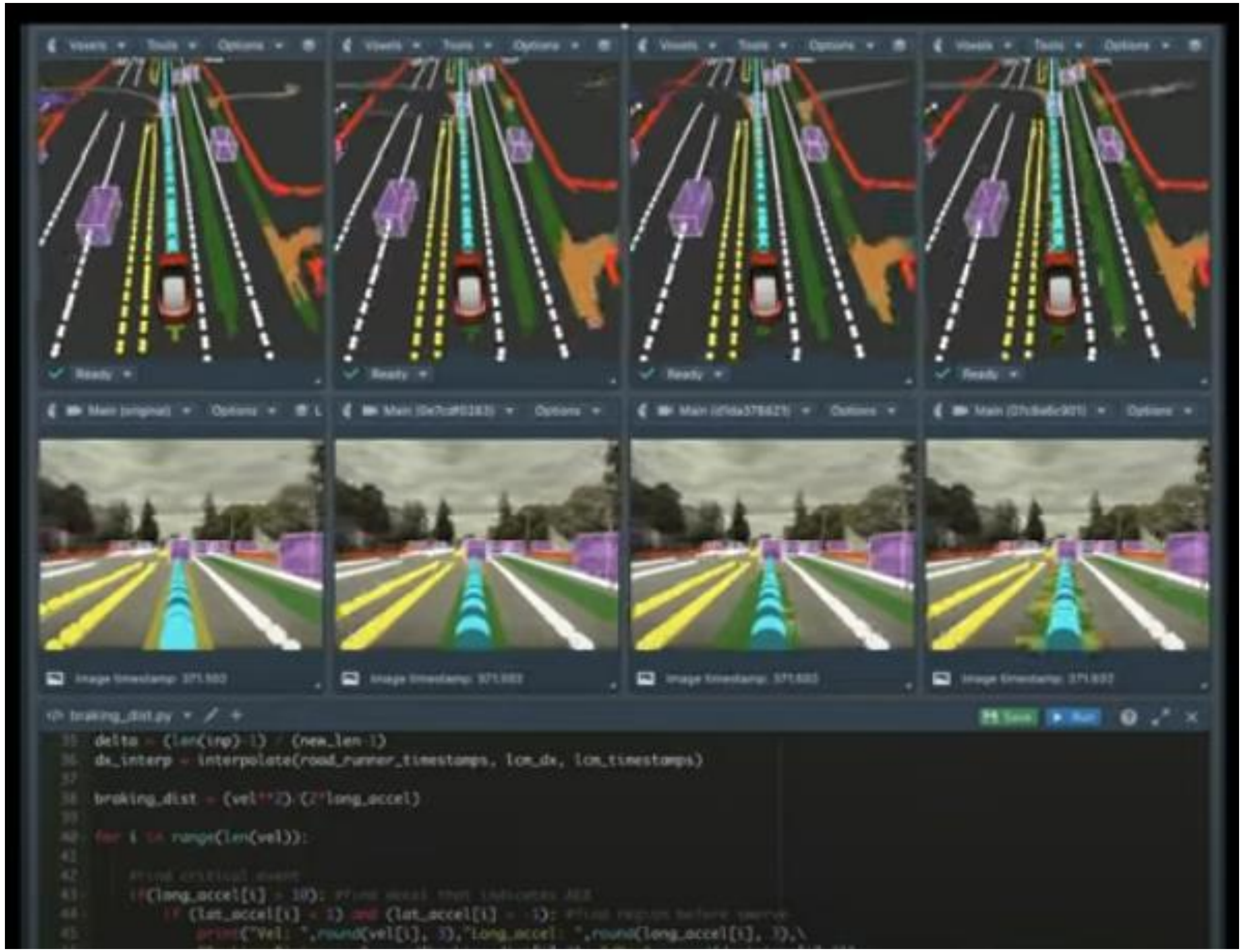


AI EVALUATION INFRASTRUCTURE

- Tesla runs a million evaluations a week for any code change that the team produces.

TOOLS

- This tool compares the output of code revisions to iterative video clips.

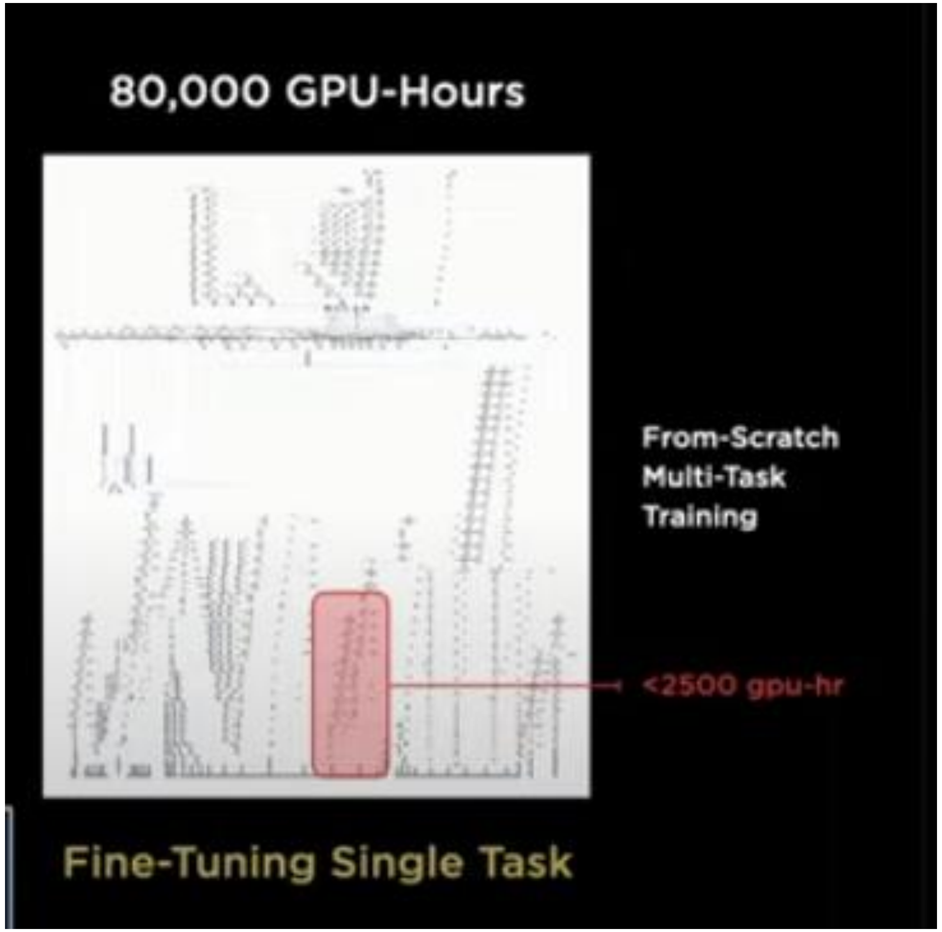


TRAINING COMPUTE

- Just shy of 10K GPUs.
- Which is more than the top 5 supercomputers in the world.

INTRODUCING DOJO

- A super fast training computer.

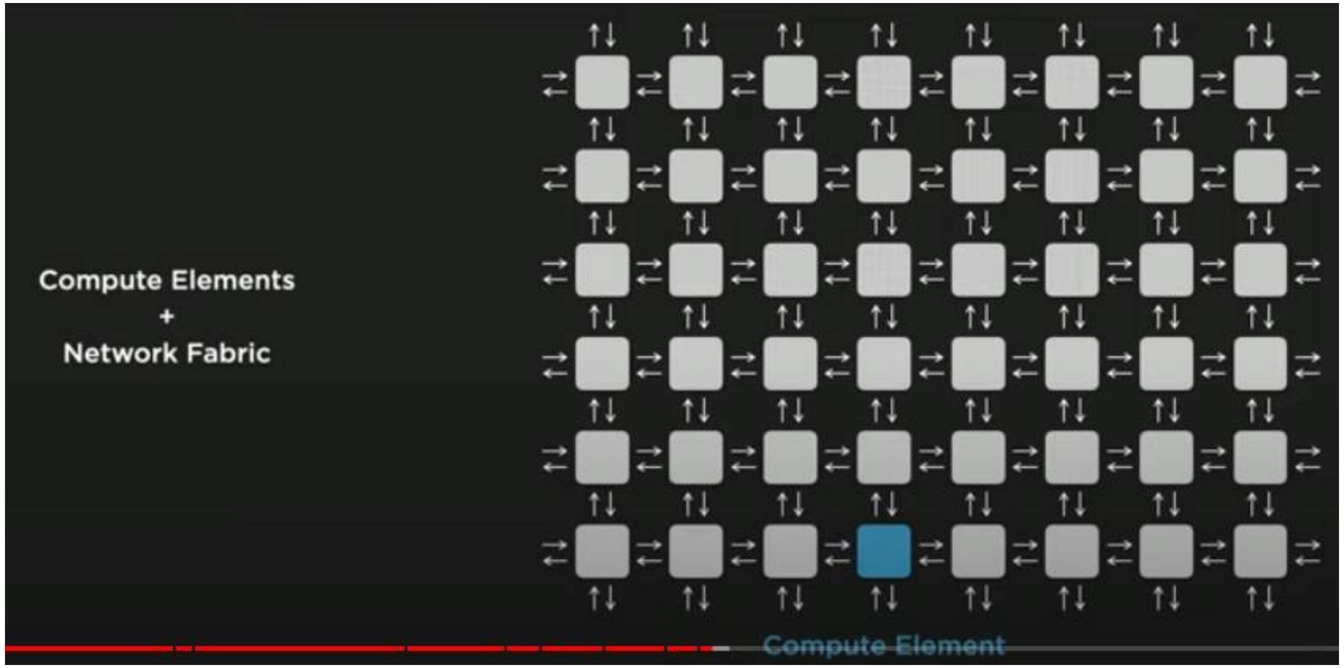


GOALS

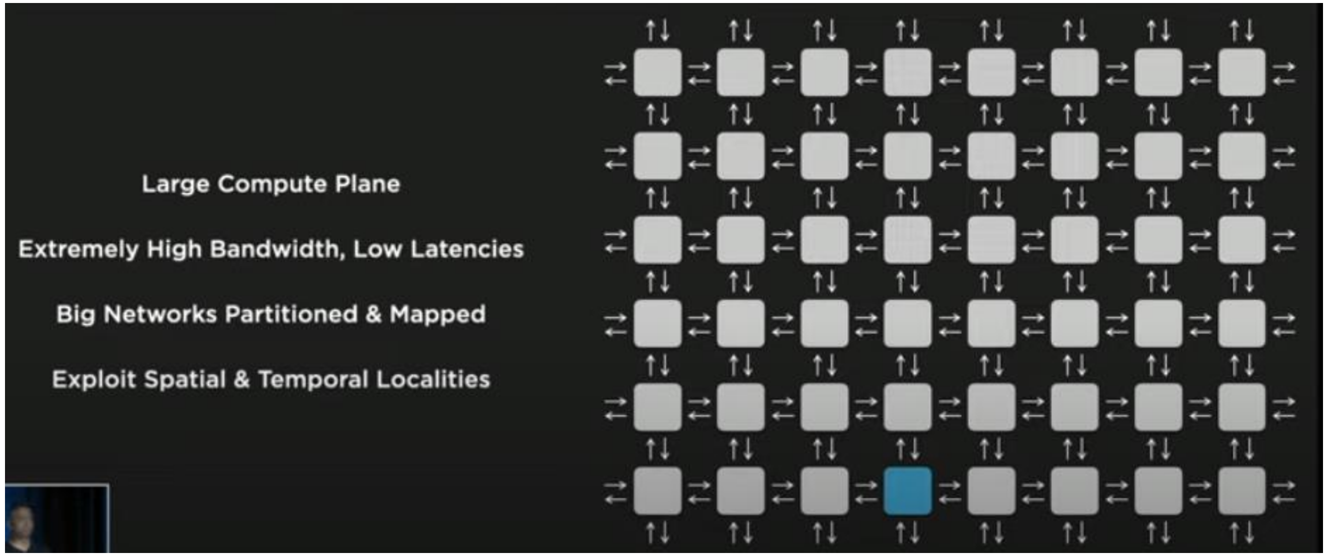
- Achieve best AI training performance
- Enable larger and more complex NN models
- Power efficient and cost effective compute

DISTRIBUTED COMPUTE ARCHITECTURE

- Very easy to scale the compute.
- Difficult to scale bandwidth.
- Extremely difficult to reduce latency.



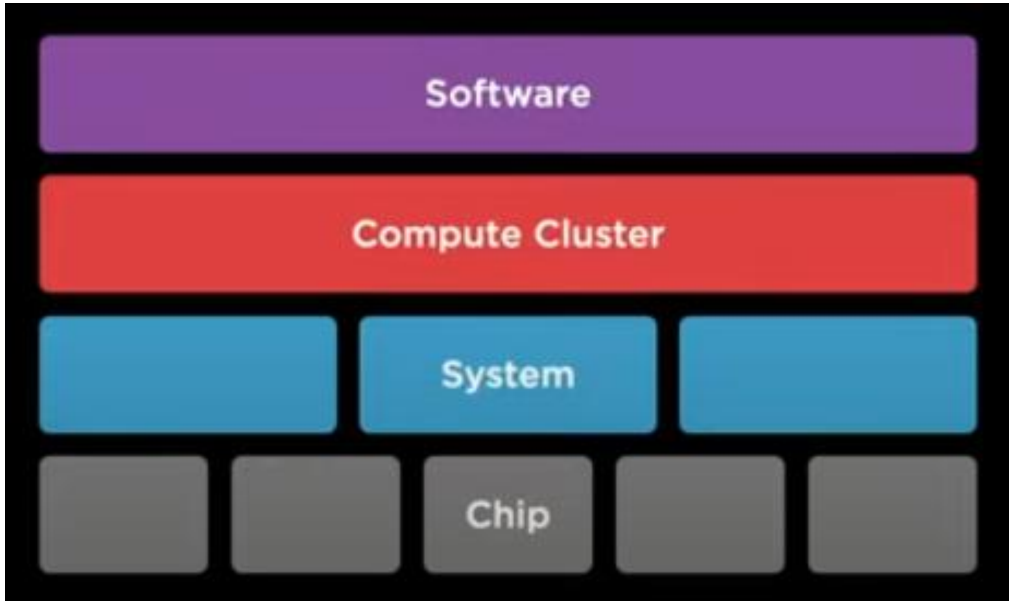
DOJO ARCHITECTURE



- Large compute plane filled with robust compute elements, backed with a large pool of memory and interconnected with high bandwidth/low latency fabric. (2D mesh)
- Big neural networks are partioned and mapped to extract parallelism.
- Neural compiler (of Tesla's design) will exploit spacial and temporal locality to reduce communication demands- bandwidth communication can keep scaling as the compute plan grows.

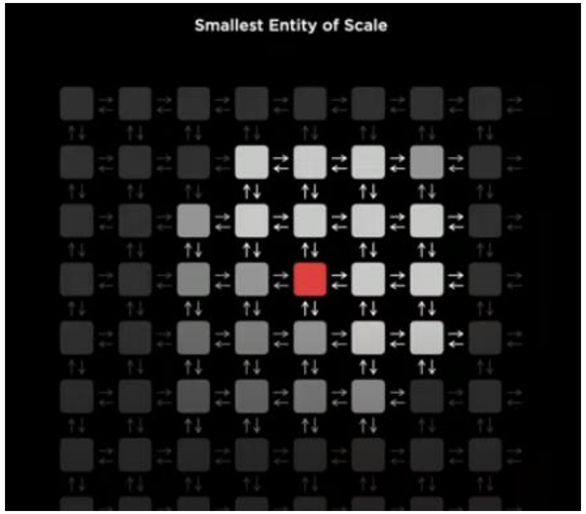
PERFORMANCE AND ALL LEVELS

- Tesla wants to attack the problem at all levels.



TRAINING NODE

- The smallest compute entity and is designed to provide seemless scaling.
  - If it is too small, it runs fast, but synchronization and software will not scale.
  - If it is too big, it will complex to implement, and produce memory bottleneck issues.

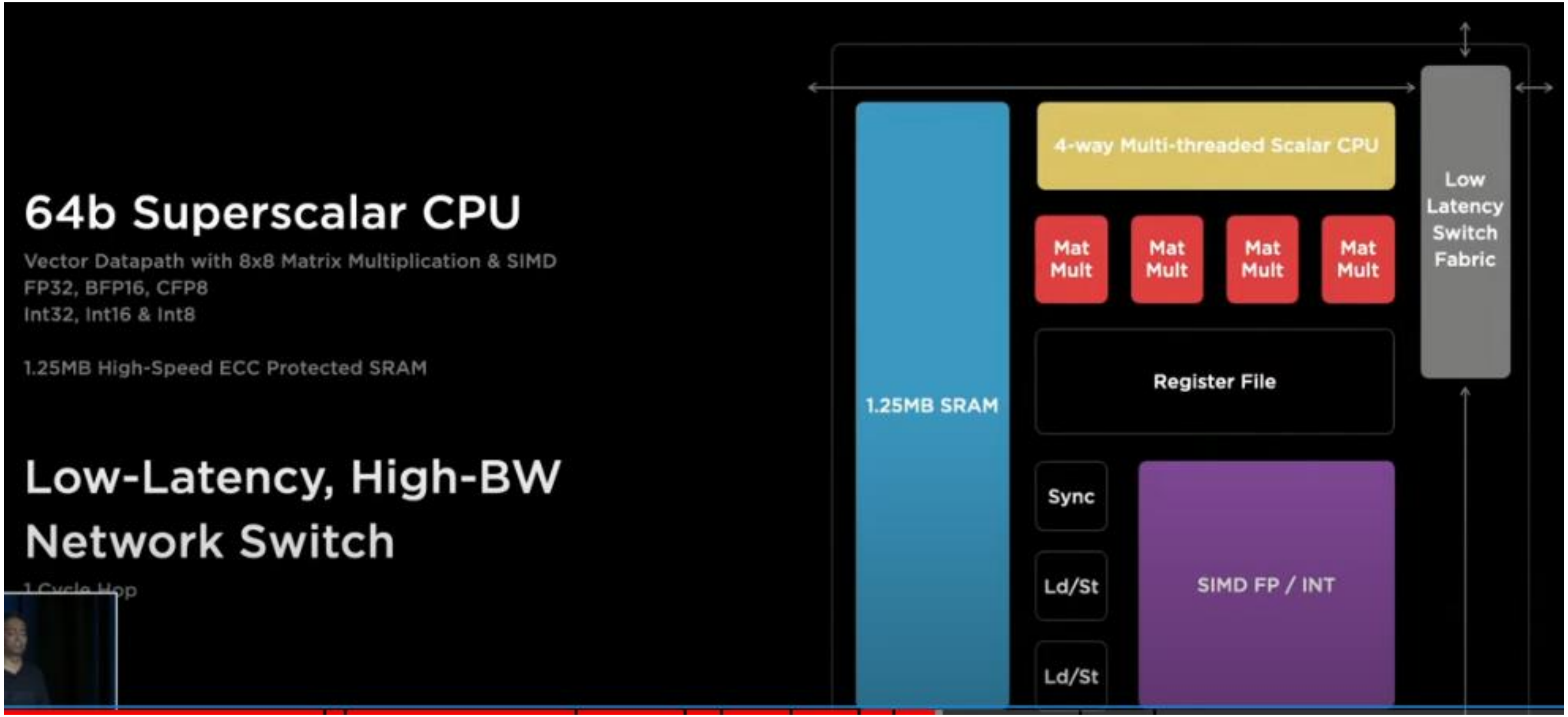


OPTIMIZING BANDWITH AND LATENCY

- Picked the farthest a signal could travel in a very high clock cycle (2+ GHtz)
- They 'drew' a box around that distance and filled it with wires, giving the highest bandwidth that can feed the box.
- Then added ML compute and a large pool of SRAM.
- Finally a programmable control core.

HIGH-PERFORMANCE TRAINING NODE

- Smallest entity of scale defined.

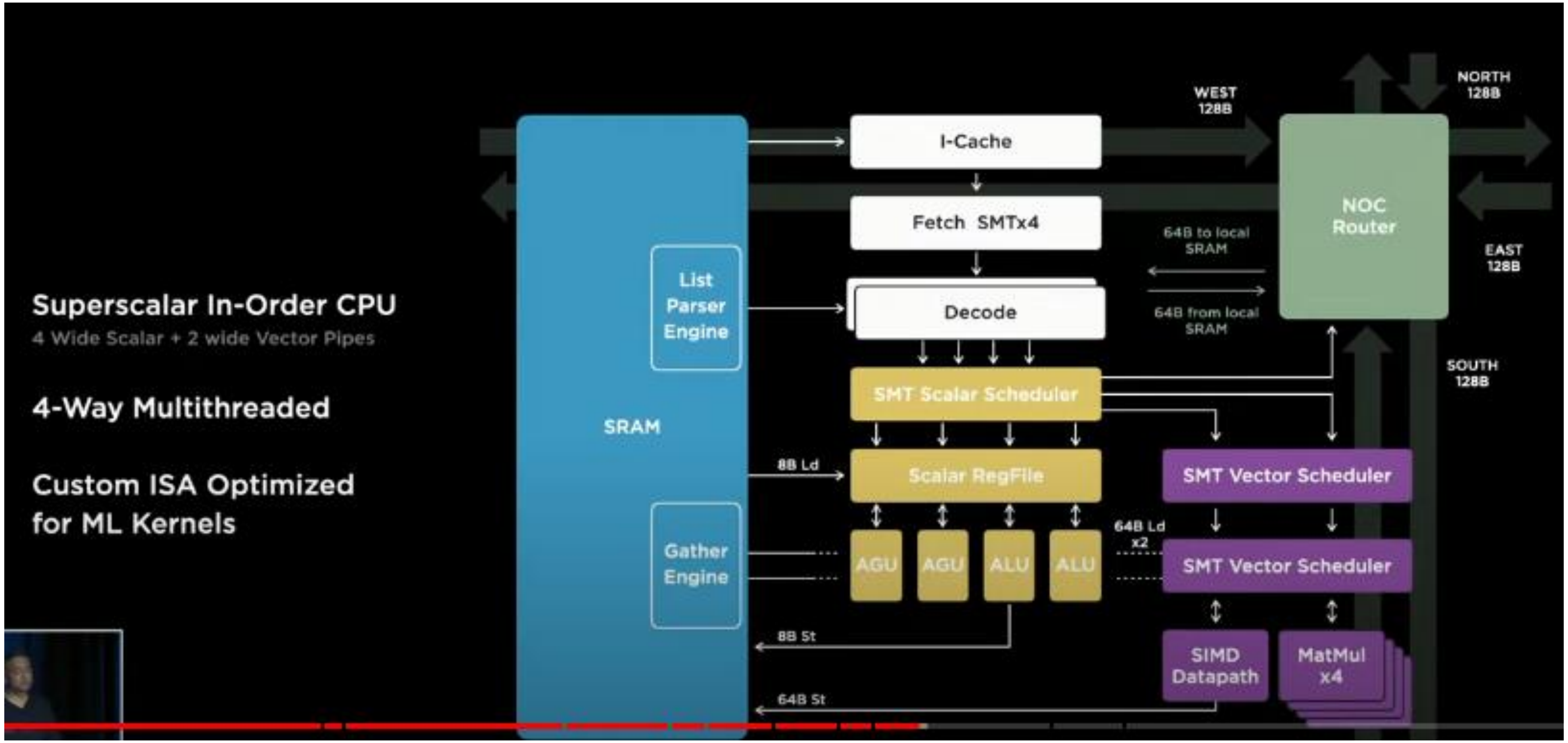


PACKING A PUNCH

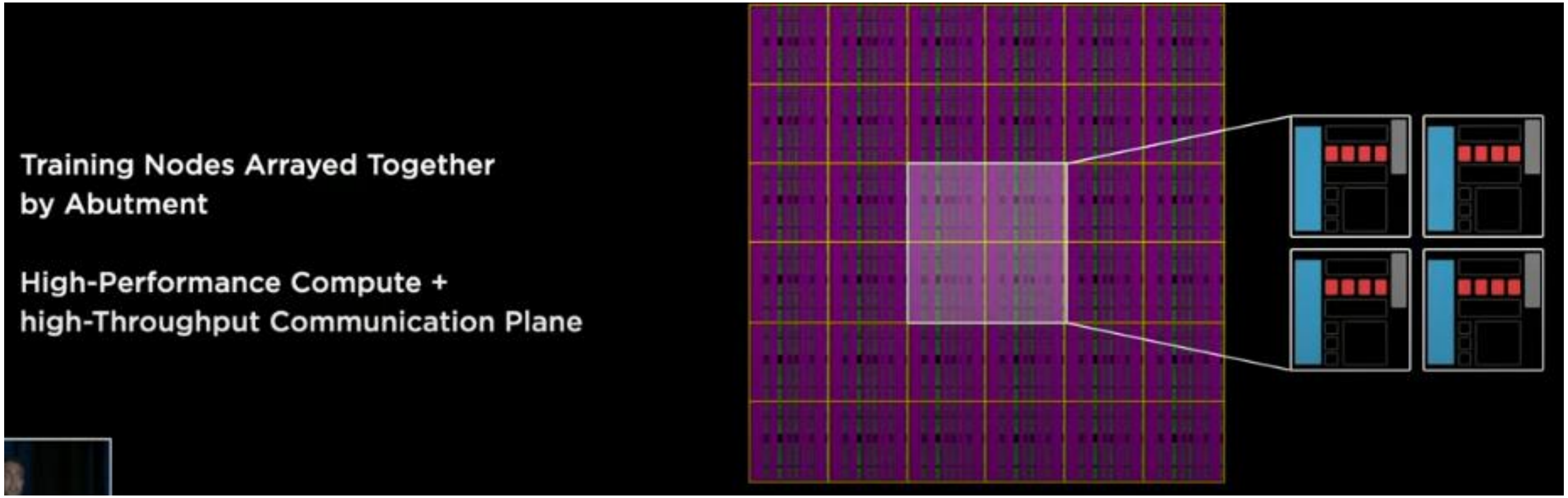
- More than one terra-flop of compute (BF16/CFP8)
- 64 GFLOPS for FP32
- 512 GB/s transfer in each cardinal direction

ARCHITECTURE

- A capable architecture that can do compute and data transfer simultaneously.
- Fully optimized for ML workloads.



MODULAR



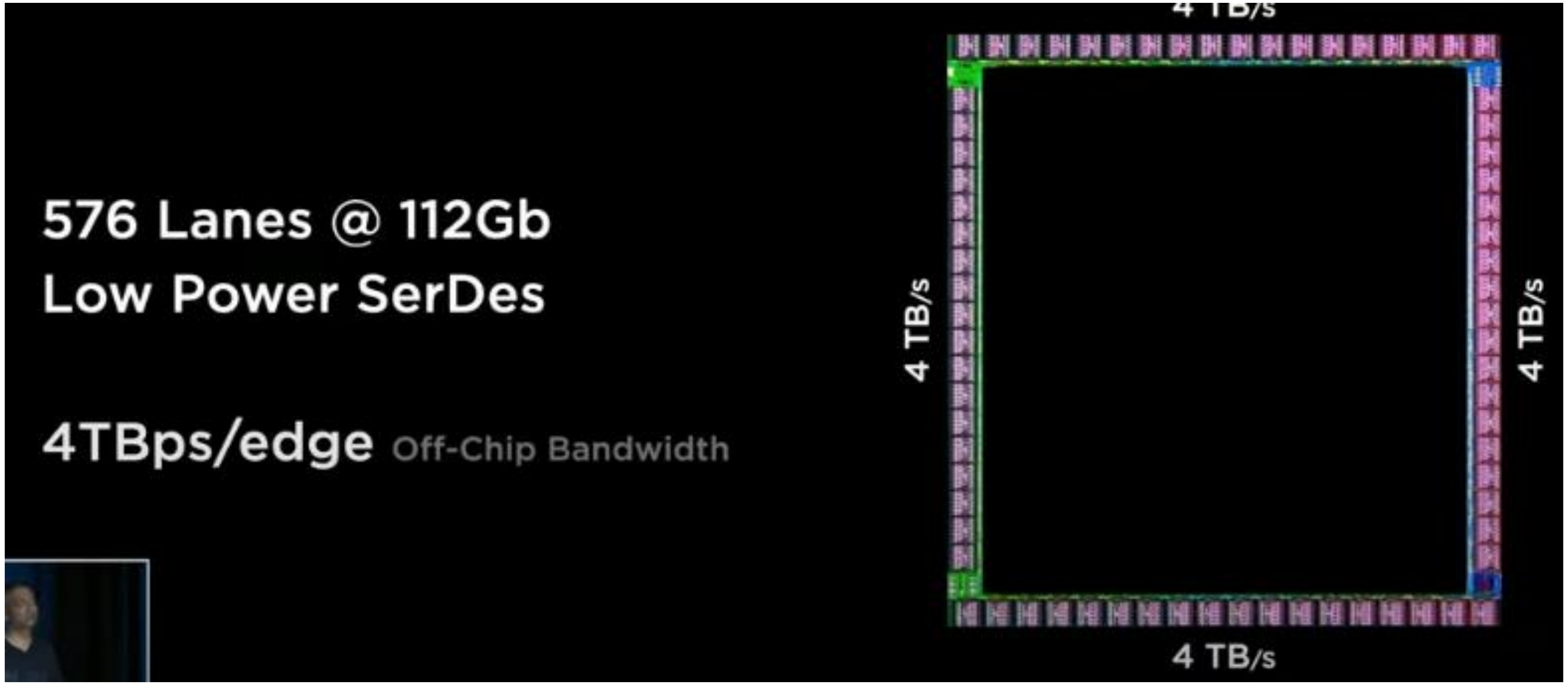
- Physically, the training node is designed to reside next to other training nodes in any direction.

COMPUTE ARRAY

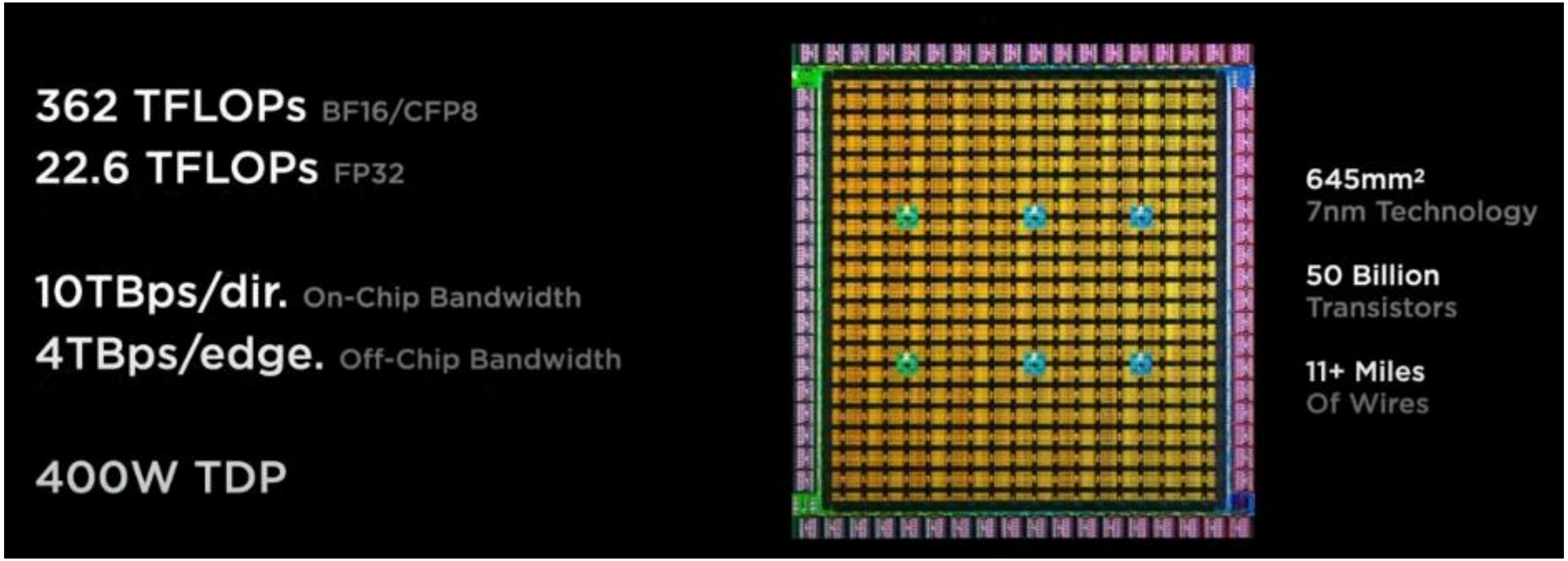
- The high bandwidth fabric supports 10TBps bi-directional throughput on a chip.

I/O RING

- More than two times the throughput of the state of the art network switch chips.

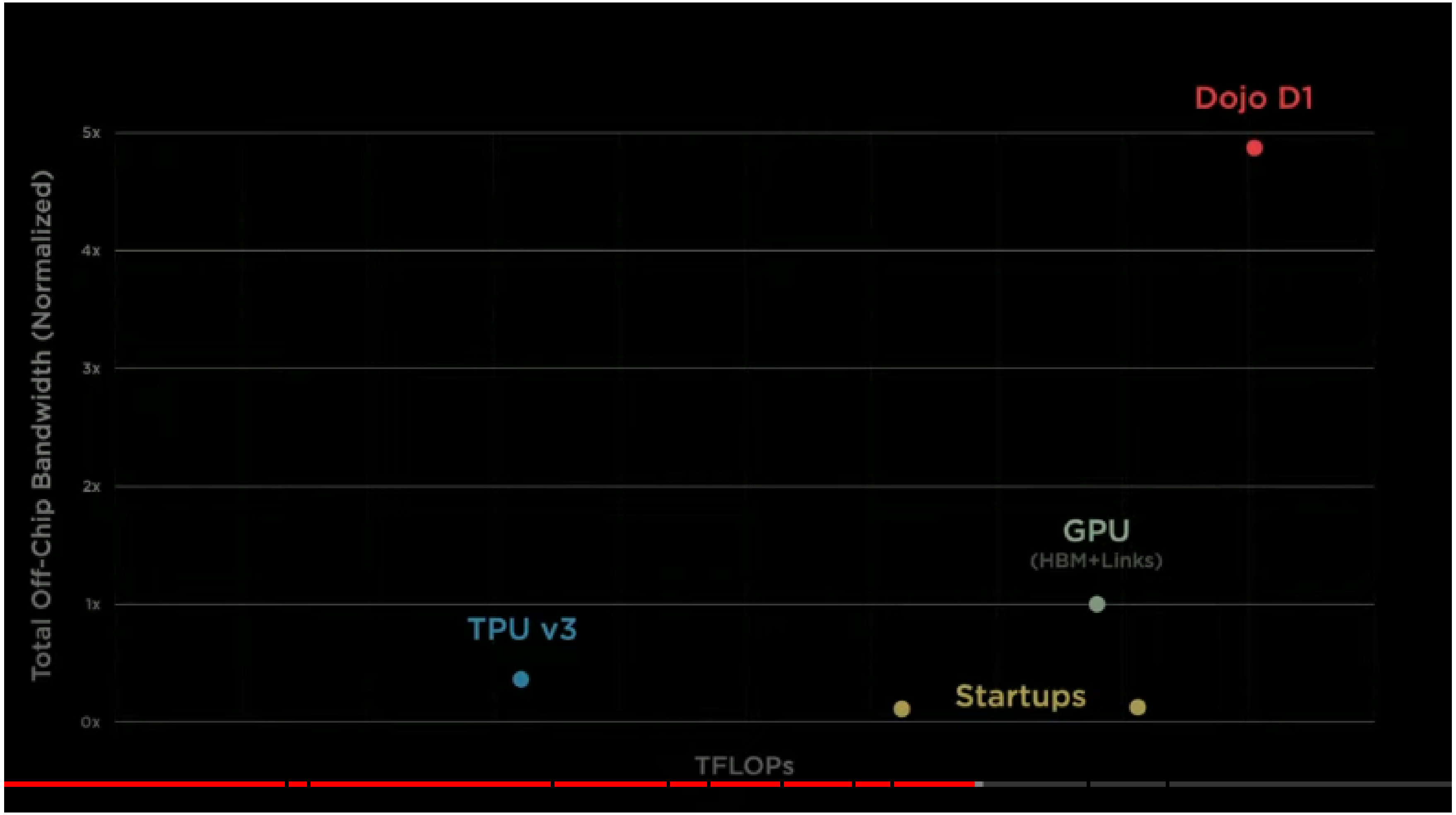


D1 CHIP

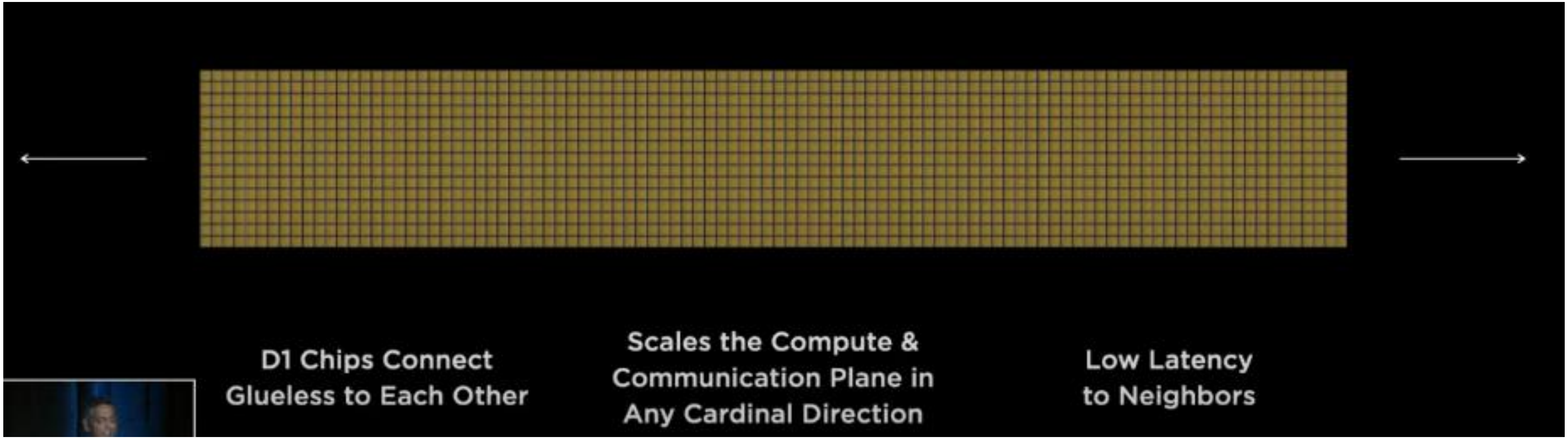


- 100% of the area is used for ML or bandwidth support.
- GPU-level compute with CPU-level flexibility.

D1 CHIP EXCELS

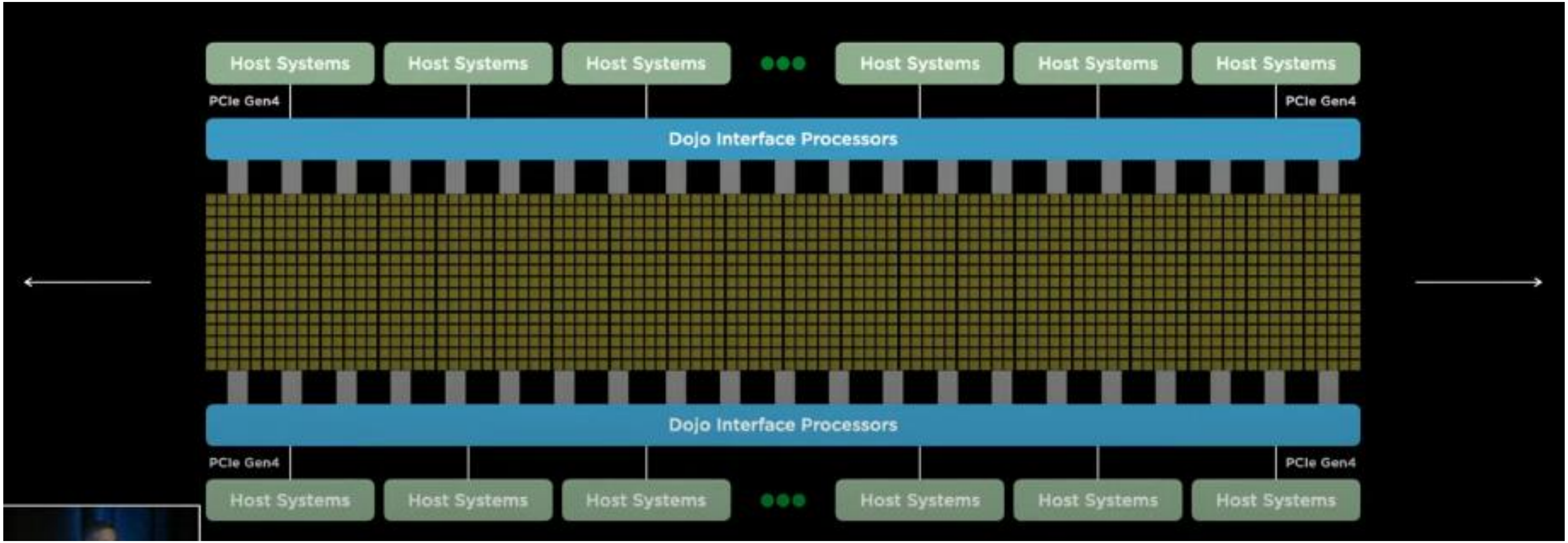


SUPER SCALE



- D1 chips connect to each other without additional hardware, Tesla put 500,000 training nodes together to form their compute plane.
- 1,500 D1 chips connected together.

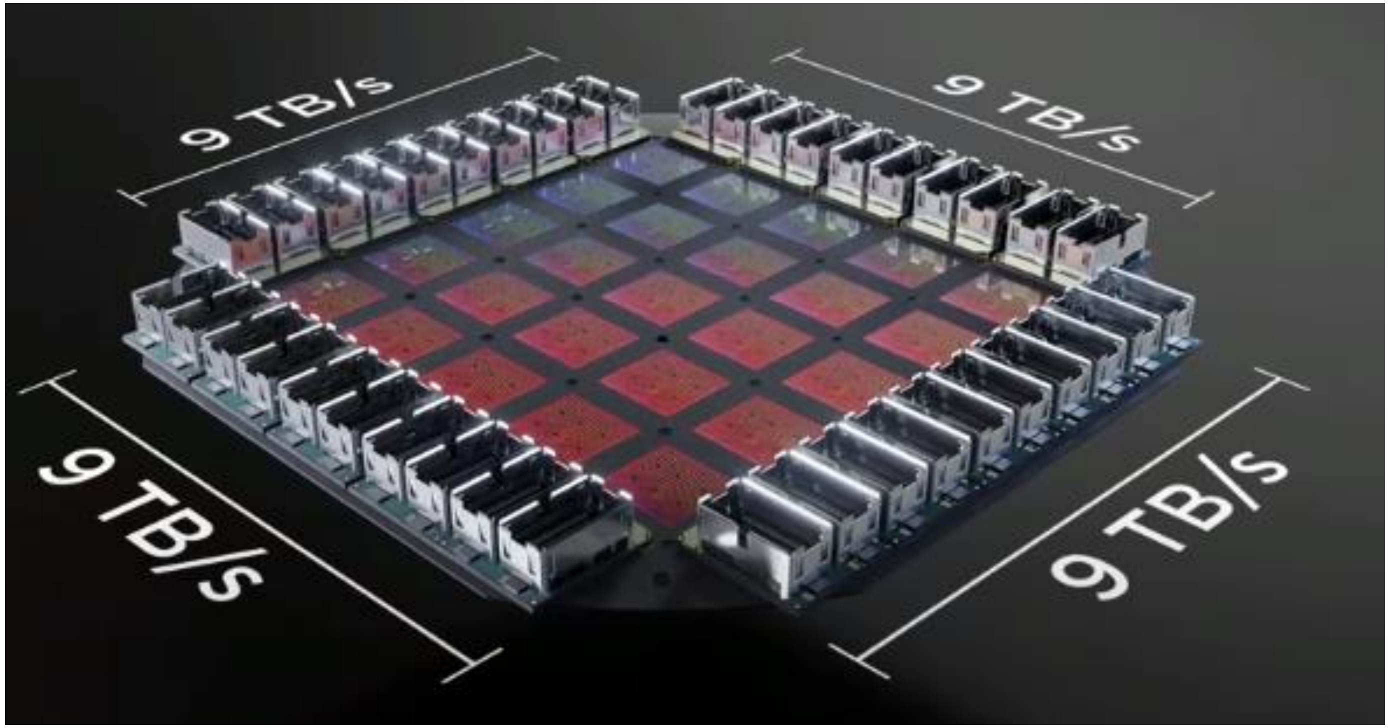
DOJO AT SCALE



- Then they added Dojo Interface Processors as the host bridge, connected PCI Gen4 interfaces.

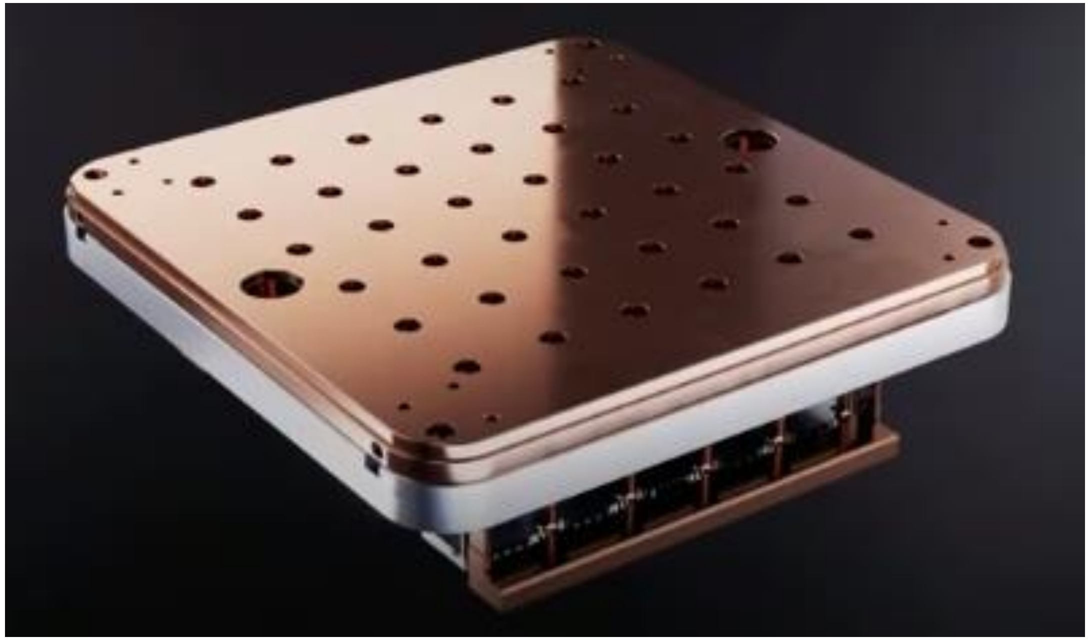
TRAINING TILE

- A training tile consists of 25 known 'good' Dojo processors.
- The maximum bandwidth is preserved.
- A high-density, high-bandwidth connection preserves the bandwidth coming out of the training tile.
- 9 PFLOPs BF16/CFP8
- Massive 36TB/s off-tile bandwidth
- Larges multi-chip-module in the industry



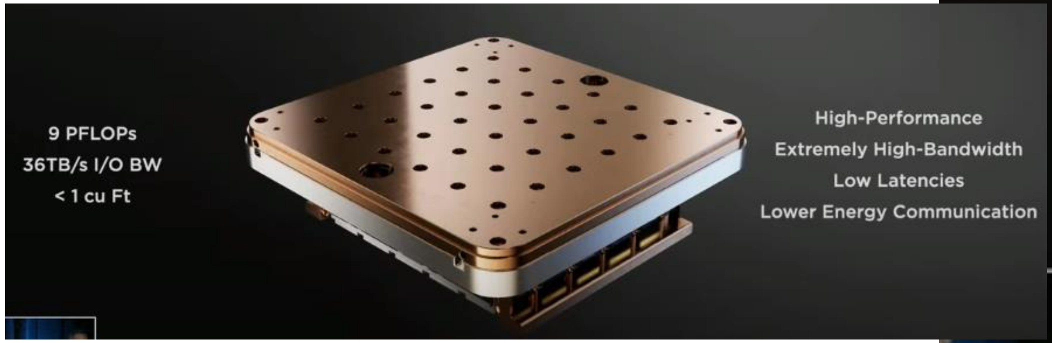
POWER DELIVERY

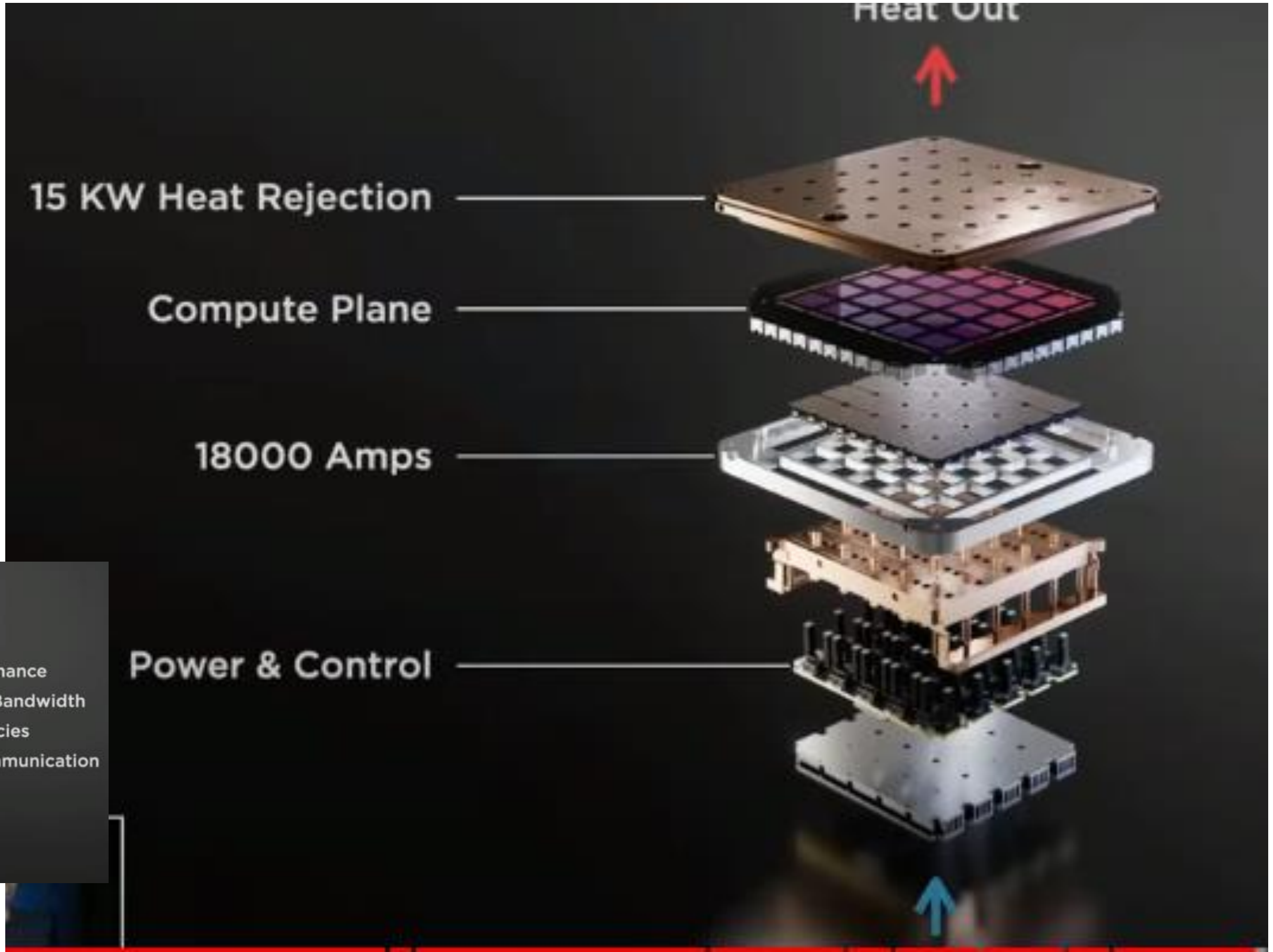
- To power the tile, Tesla created voltage regulators that re-flowed directly on top of each Dojo chip.
- Integrated the electrical, mechanical and thermal pieces with a 52VDC input.



TRAINING TILE

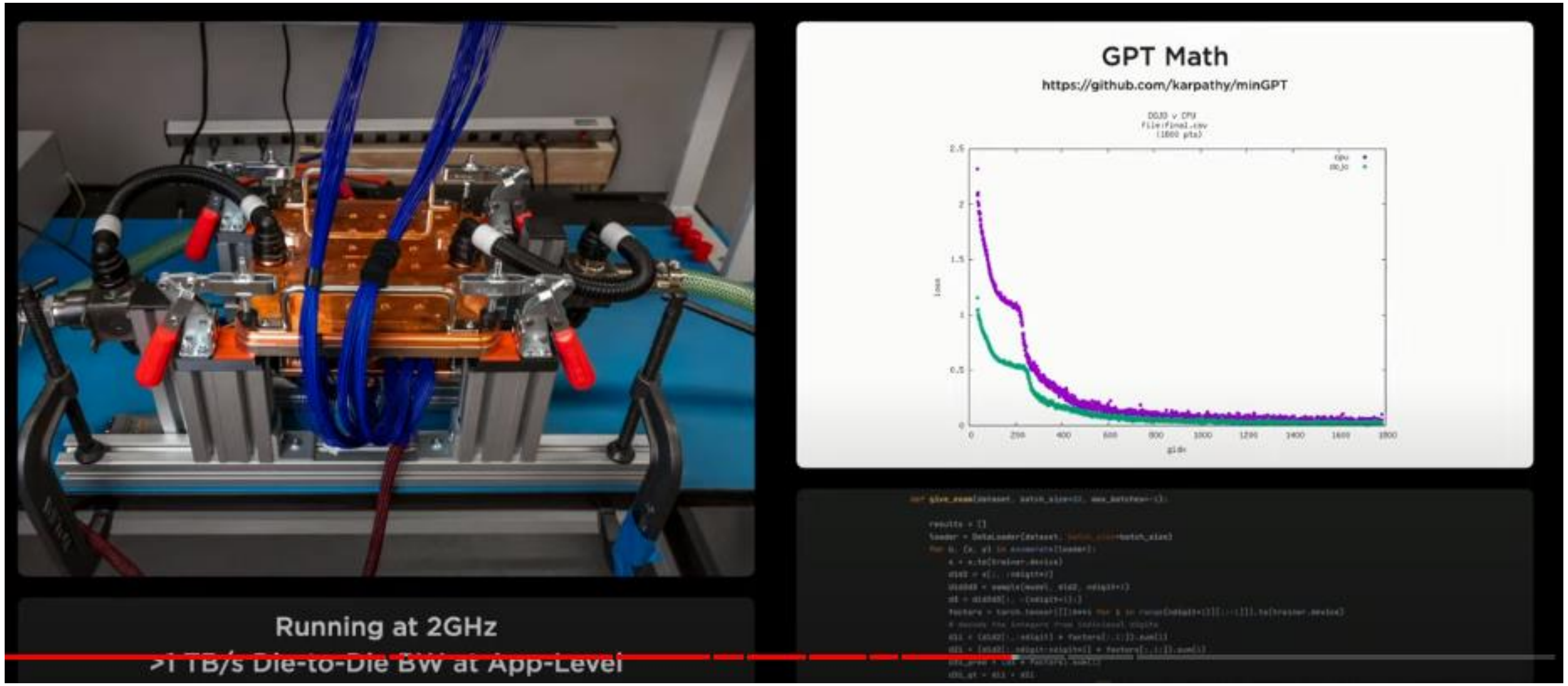
- Unprescedented integration.





FIRST FUNCTIONAL TILE

- On a limited cooling bed, Tesla was able to run GPT2 on the compute tile.



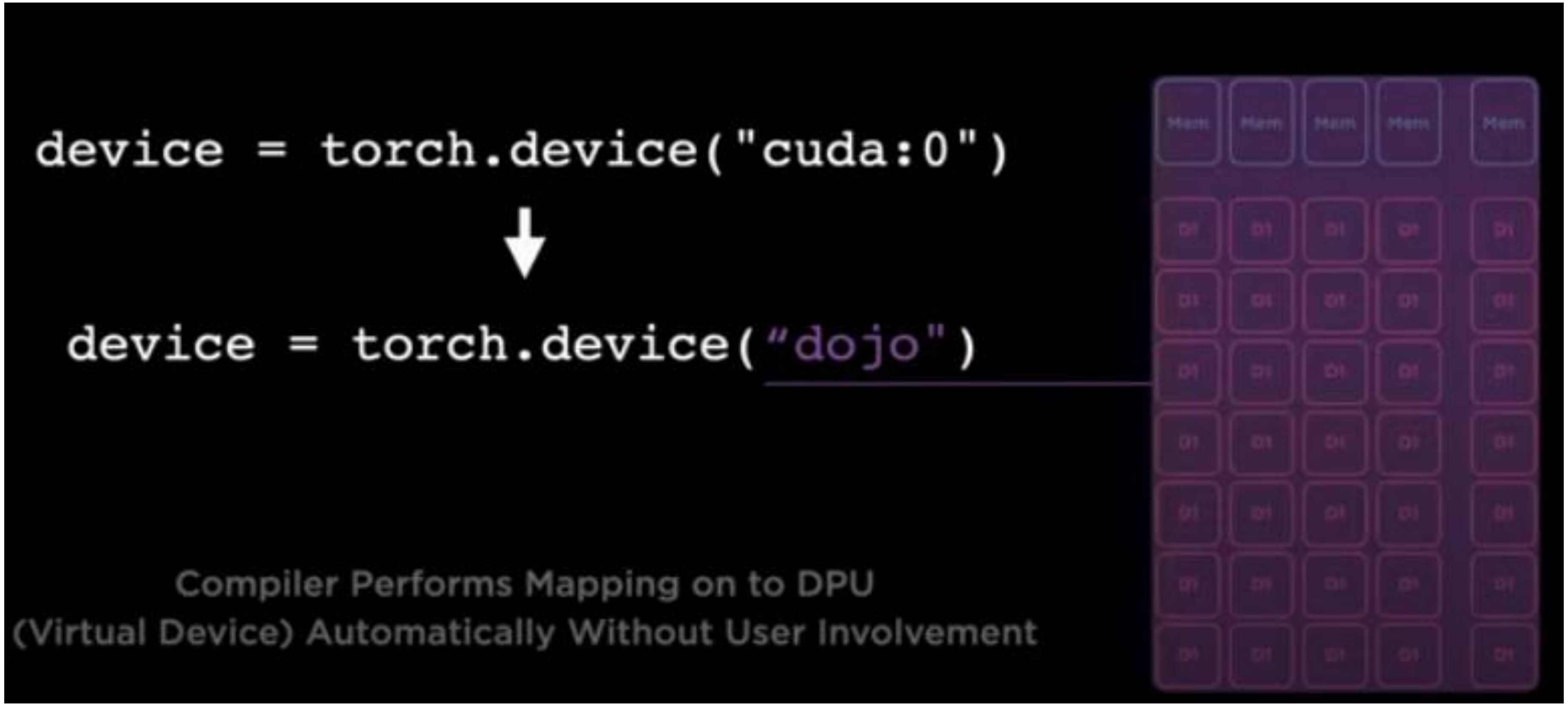
SAMPLE TRAINING MATRIX

- A 2x3 tile tray is a sample training matrix, with two trays in a cabinet.
- Tesla assembled 10 cabinets with 1.1 ExaFlop capability.
- 120 training tiles, 3,000 Dojo chips, >1M traing nodes

LOGICAL VIEW OF THE SYSTEM

- Not every job requires a huge cluster. The compute plan can be subdivided into sections that can be used for different workloads.

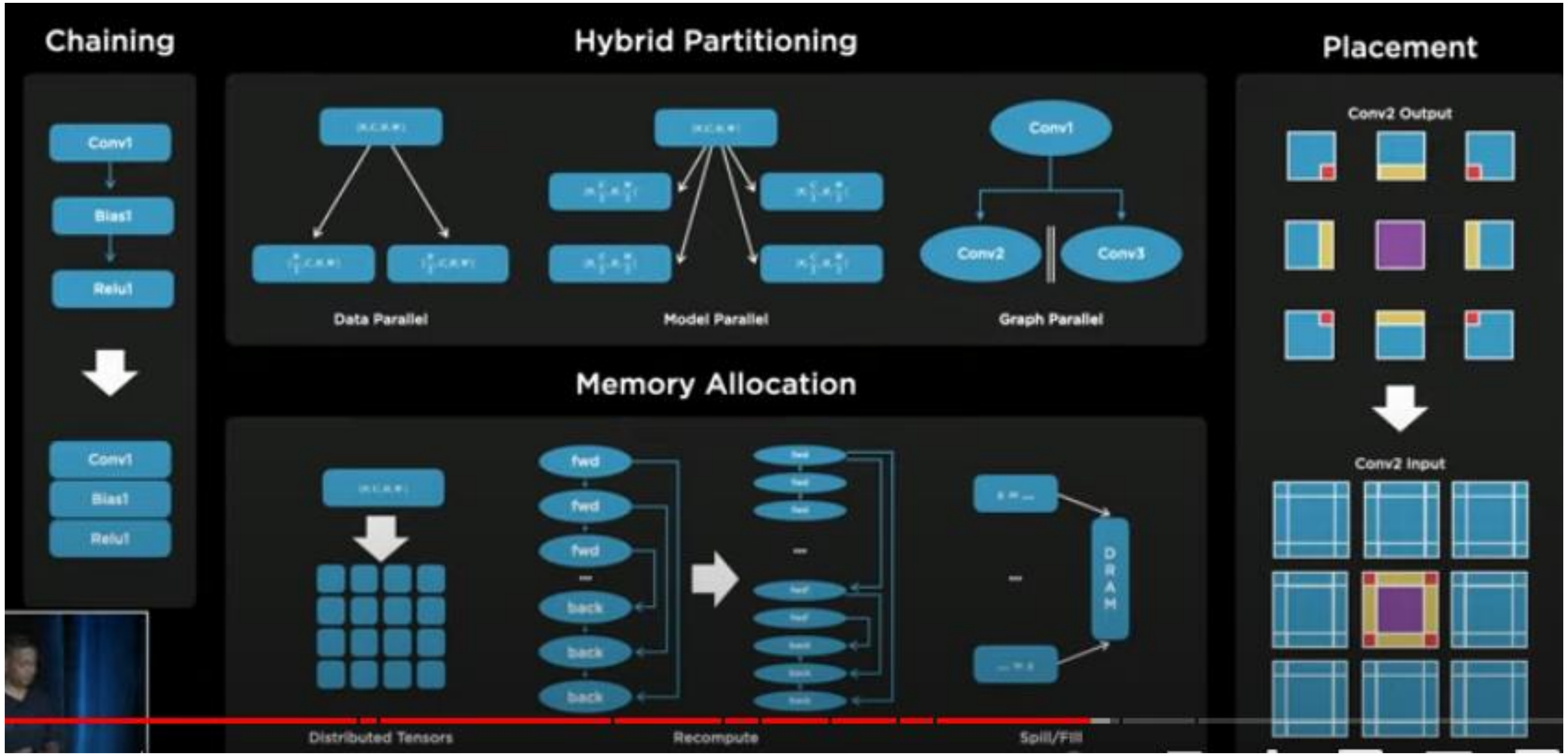
HOW DOES A USER LEVERAGE THIS SYSTEM?



- Minimal code change to scripts.

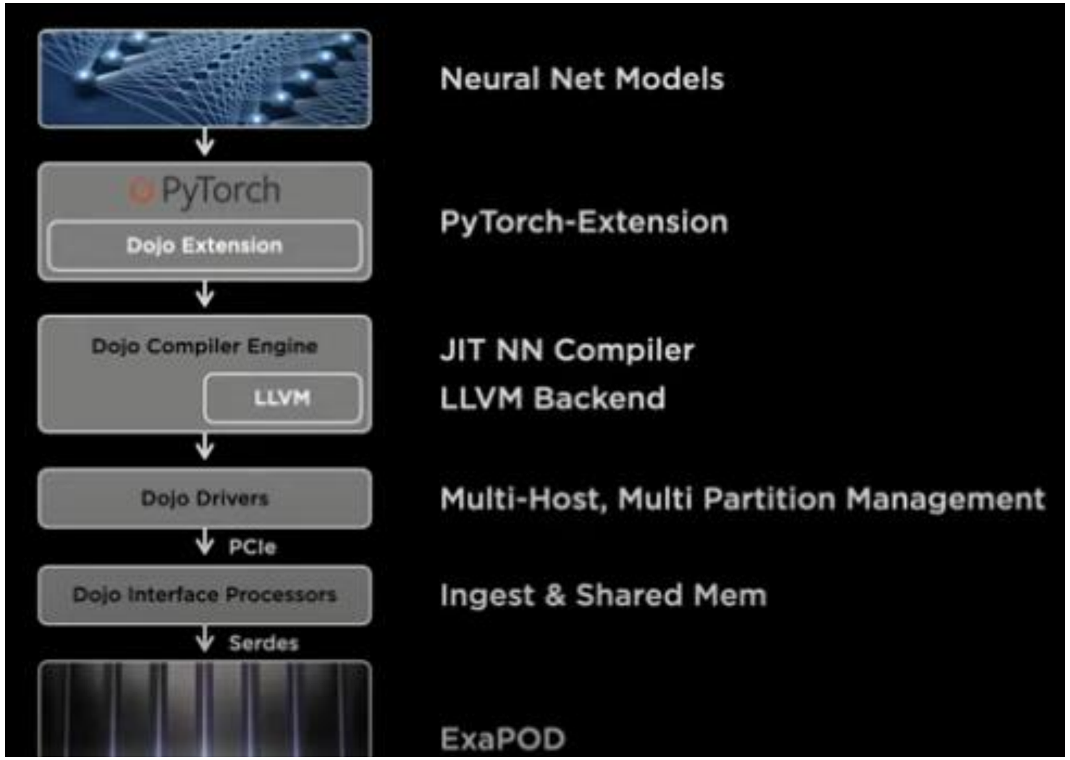
DOJO COMPILER ENGINE

- The compiler uses multiple techniques to extract the most performance from the compute configuration.
- The compiler can handle highly dynamic control flows, such as loops and if/then/else branches.



SOFTWARE STACK

- Extensions to PyTorch.
- Custom profilers and debuggers that work with the new stack.



FASTEST AI TRAINING COMPUTER

- 4X performance (at the same cost)
- 1.3X Better Performance per Watt
- 5X smaller footprint

THANK YOU