Tesla FSD Does Not Know What It Cannot Do

by Nick Clark | Published March 27, 2026 | PDF

Tesla's Full Self-Driving system uses a vision-based neural network to handle driving across diverse conditions. The ambition is genuine: a single system that drives everywhere without pre-mapped environments or predefined operational domains. But FSD does not maintain a structural capability envelope that formally defines what the system can and cannot reliably do under current conditions. It attempts every scenario and relies on the neural network's generalization. Capability awareness provides the structural primitive: a computed, persistent representation of what the system can do, updated in real time, governing what it attempts.

What Tesla built

FSD's approach is architecturally distinctive. Where other AV companies use lidar, pre-mapped routes, and defined operational design domains, Tesla uses camera-based perception and a neural network trained on billions of miles of driving data to handle the full range of driving scenarios. The system operates across highways, city streets, parking lots, and unstructured environments without requiring pre-mapping. The engineering bet is that a sufficiently capable neural network trained on enough data will generalize to all driving conditions.

When FSD encounters a situation it handles poorly, the failure mode is typically an intervention by the human driver. The system continues attempting to drive until the human takes over. There is no structural mechanism by which the system assesses its own capability for the current scenario and determines, before attempting it, that the scenario exceeds its reliable operating envelope.

The gap between attempting and knowing

The structural limitation is that FSD cannot distinguish between scenarios it handles well and scenarios it handles poorly until it attempts them. A human driver approaching an unfamiliar intersection in dense fog with unusual road markings recognizes that this specific combination of conditions exceeds their comfortable capability and adjusts behavior accordingly: slower speed, increased attention, possibly a different route. FSD approaches the same scenario with whatever the neural network produces.

The capability envelope formalizes what FSD implicitly lacks. It is a persistent, computed representation of the system's reliable operating domain that updates based on current conditions. Rain degrades the capability envelope for vision-dependent lane detection. Construction zones narrow the envelope for path planning. Unusual road geometries reduce the envelope for intersection handling. Each condition is evaluated against the system's demonstrated performance, and the resulting envelope defines what the system should attempt.

The temporal executability forecast extends this to time. A system that can reliably drive this highway now may not be able to do so in thirty minutes when weather conditions are predicted to worsen. Capability awareness that includes temporal forecasting allows the system to plan for capability changes before they occur.

Why disengagement data is not capability awareness

Tesla collects data from driver interventions to improve the neural network. This is retrospective capability data: learning from failures to reduce future failures. Capability awareness is prospective: knowing what you can do before you attempt it. The distinction matters because retrospective learning requires failures to generate training data, while prospective capability awareness prevents failures by not attempting scenarios outside the envelope.

What capability awareness enables

With capability awareness as a first-class cognitive primitive, FSD maintains a persistent capability envelope that updates based on current sensor conditions, environmental complexity, and the system's demonstrated reliability for similar scenario types. The joint condition, capability multiplied by time multiplied by uncertainty, determines whether the system should attempt a maneuver. When the envelope contracts due to degraded conditions, the system adjusts its behavior: reducing speed, increasing margins, or requesting human attention before the scenario exceeds capability rather than after.

Envelope negotiation enables the system to communicate its current capability state to the driver. Rather than a binary handoff between autonomous and manual driving, the system reports which capabilities are at full strength and which are degraded, allowing the driver to maintain appropriate situational awareness.

The structural requirement

Tesla's neural network approach to driving is bold and has produced real capability. The structural gap is self-knowledge: the system's ability to compute what it can reliably do under current conditions before it attempts it. Capability awareness provides the persistent envelope, temporal forecasting, and uncertainty-weighted execution gating that transform a system that tries everything into one that knows its limits and operates within them.

Capability Awareness All 21 steps →

Know what you can do before you try.

Primary Technical Disclosure
○ Capability-, Time-, and Uncertainty-Aware Execution in Autonomous Computational Networks
Secondary Technical
○ Capability as First-Class Computational State○ Capability Envelope for Substrates○ Temporal Executability Forecasting○ Uncertainty as First-Class Propagated Variable○ Capability Envelope Negotiation○ Capability Genealogy Tracking○ Biological Capability Extension○ Network-Level Capability Pressure○ Capability-Permission Distinction○ Capability-Native Computation○ Execution Synthesis and Non-Synthesis○ Agent Behavior Under Constraints○ Predictive Network Planning○ Multi-Agent Contention Resolution○ Capability Robustness Mechanisms○ Capability-Modulated Discovery Traversal○ Capability as Confidence Input○ Embodied Capability Envelopes○ Substrate Resource Negotiation
Applications (General)
○ Robotic Capability Assessment Before Commitment○ Edge Computing Resource Governance Through Capability Envelopes○ Capability Awareness for Surgical Robotics○ Capability Awareness for Agricultural Robotics○ Capability Awareness for Mining Operations○ Capability Awareness for Offshore Energy Operations○ Capability Awareness for Warehouse Logistics Robotics○ Capability Awareness for Construction Robotics
Applications (Specific)
● Tesla FSD Does Not Know What It Cannot Do○ John Deere's Autonomous Tractors Cannot Assess Their Own Limits○ KUKA Robots Execute Without Knowing Their Envelope○ FANUC Robots Have No Adaptive Capability Envelope○ Universal Robots Cobots Execute Without Knowing Their Limits○ ABB Robots Perform Without Self-Assessing Capability○ Yaskawa Motoman Robots Move Without Tracking Capability Drift○ Doosan Cobots Collaborate Without Capability Self-Knowledge○ Agility Robotics' Digit Walks Without Knowing What It Can Carry○ Figure's Humanoid Learns Tasks Without Knowing Its Envelope
Capability Awareness overview →
AQ

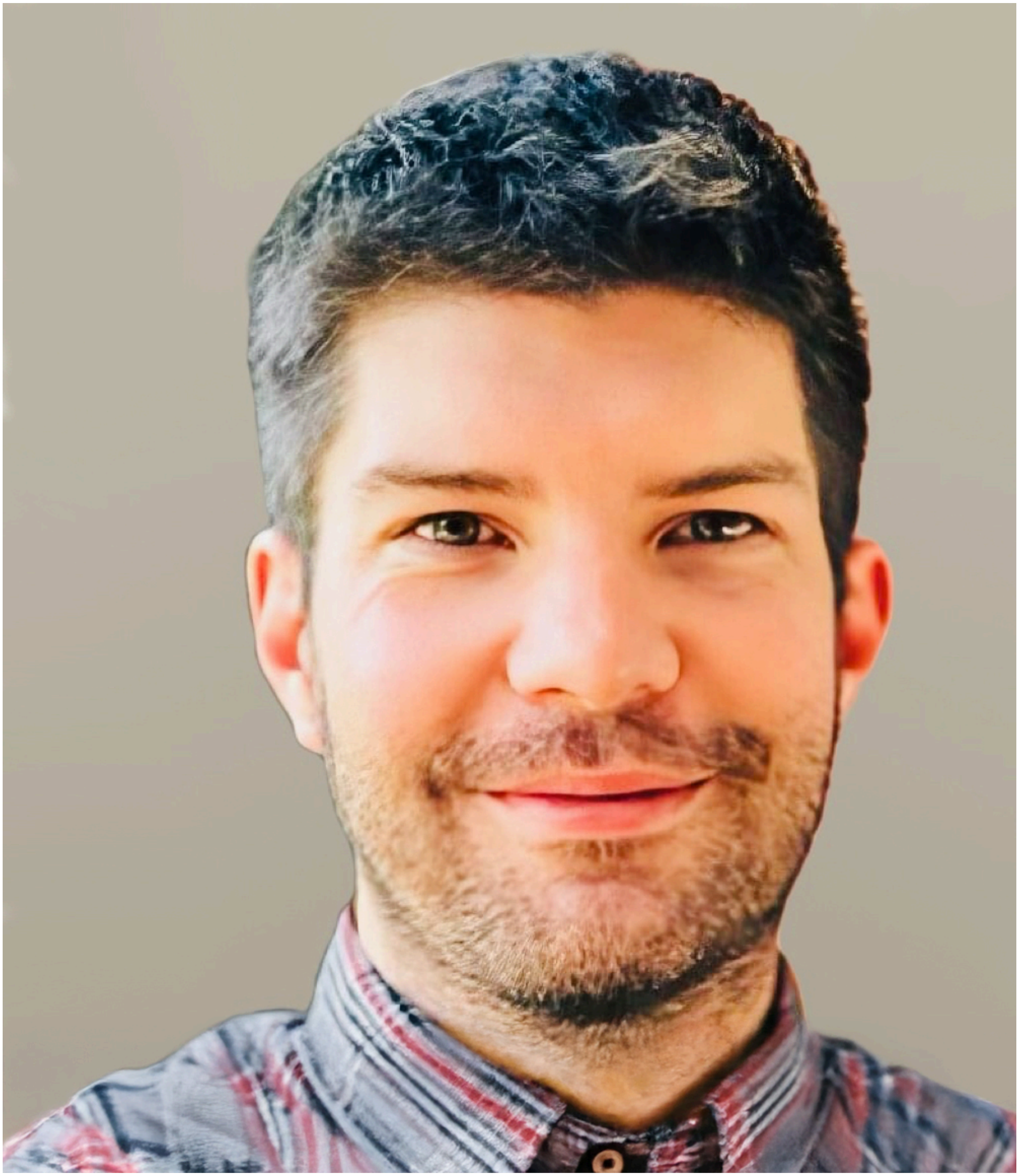
deterministic
autonomy

Legal

Last updated: 2026-03-03

- [Opportunities]
- [Sitemap]

-
- nick@qu3ry.net
- 72 28 14 36 01


Invented by Nick Clark | Founding Investors: Devin Wilkie