DEEPVIEW

PRODUCT DATASHEET

DRIVE-BY-WIRE DEVELOPMENT KIT

X2000 ADAS DEVKIT

Full-stack ADAS development on NVIDIA Orin

275 TOPS	8× GMSL2	100 Hz	1 TB	20 FPS
AI PERFORMANCE	CAMERA INPUTS	CAN-FD CONTROL	ONBOARD SSD	CAMERA + INFERENCE

The **X2000 ADAS DevKit** is a comprehensive development platform designed to accelerate the creation and testing of advanced driver assistance systems. Built around the **NVIDIA Orin SoC with 275 TOPS** of performance, the X2000 delivers real-time AI-powered perception and decision-making, with a lane centering reference design based on OpenPilot. The system includes an AR0234-based forward camera, onboard SSD storage for logging, and full drive-by-wire integration initially for the Ford Mach-E — with a modular architecture supporting up to 8 GMSL2 cameras, ready for integration with other vehicle platforms.

NVIDIA Inception Program

Platform Overview

Features

- **AI Vehicle Computer**
  AI inference and drive-by-wire control, featuring the NVIDIA Orin SoC with 275 TOPS.
- **Data Logger**
  Forward camera + CAN, frame-aligned and recorded to the onboard 1 TB SSD in one-minute clips.
- **Drive-by-Wire API**
  CAN-FD control of steering, gas and brake.
- **Lane Centering**
  OpenPilot AI Model reference design running in real time on the Orin GPU.
- **20 FPS Perception Loop**
  Camera capture and AI inference at a locked 20 Hz, synchronized with vehicle CAN.
- **Forward Camera Module**
  AR0234 global-shutter sensor with GMSL2 serialization and Sunex M12 optics.
- **Pathfinder Software Suite**
  Browser-based console for driving, replaying, simulating and training — 100% of source code available for customer development.

> **All-in-one ADAS Development Kit.** Vehicle AI computer with realtime AI Inference and CAN-FD drive-by-wire, along with data logging and model training — everything needed to rapidly iterate new ADAS features.

System Specs

Processor	12-core Arm Cortex-A78AE CPU
GPU	NVIDIA Jetson AGX Orin SoM — 275 TOPS
Forward Camera	AR0234, 1920×1080 at 20 Hz, M12 lens
Camera Inputs	Up to 8× GMSL2 cameras with Sunex M12 lenses
Mass Storage	1 TB SSD
Connectivity	LTE, WiFi, USB, Ethernet, HDMI, CAN-FD
CAN-FD	2× interfaces, 100 Hz processing rate
Drive-by-Wire	Ford Mach-E supported; additional platforms on request

Physical

Size	234 mm × 193 mm × 56 mm
Construction	Fanless, rugged aluminum enclosure for in-vehicle mounting
Power	12 V vehicle supply

Software Stack

Languages / APIs	Python, Rust, OpenCV, TensorRT, CUDA, ONNX, PyTorch
AI Model	OpenPilot lane centering, TensorRT-accelerated
Console	Pathfinder web UI, served from the DevKit over WiFi / Ethernet

Up and Running in an Afternoon

01
Mount & wire
Four connections: 12 V, ground, CAN-FD pair, and the front camera feed (see wiring diagram).

02
Connect
Join the DevKit over WiFi or Ethernet and open the Pathfinder console in any browser.

03
Drive & record
The live cockpit verifies CAN health while every mile is logged in one-minute clips.

04
Train & validate
Retrain on-device with one click, review the run, then validate in closed-loop sim.

ADAS Software Modules

Accelerate time-to-market for NVIDIA automotive platforms

Leverage Deepview's 6+ years of experience designing and implementing production software for NVIDIA platforms. The X2000 ADAS DevKit accelerates ADAS development with a robust real-time architecture and native support for **Python, Rust, OpenCV, TensorRT, CUDA, ONNX, and PyTorch**.

The DevKit comes pre-configured with a flexible reference architecture that includes forward camera input via GMSL2 at 20 FPS, AI inference at 20 Hz, and 100 Hz CAN-FD drive-by-wire control[1]. All CAN data is aligned with 1080p video frames for frame-accurate playback and API-driven replay control. Data is easily pushed to your datacenter with a single API call.

Available Now

- Real-time **camera interfacing at 20 FPS**
- **Hardware-accelerated** video encoding
- Real-time **AI inference** using TensorRT
- **Lane centering** — OpenPilot AI Model
- **Drive-by-wire** at 100 Hz
- In-car display — **development UI**
- **Log CAN + video** to datacenter

In Development

- Multi camera — **8 cameras at 20 FPS**
- Multi camera — **8-camera video encode**
- **Depth processing** with radar and lidar interfaces
- **Emergency braking** reference design
- **Automated highway navigation**
- In-car display — **3D visualization**
- Functional safety — **ISO 26262**

LANE CENTERING REFERENCE DESIGN — OpenPilot AI Model · NVIDIA Orin

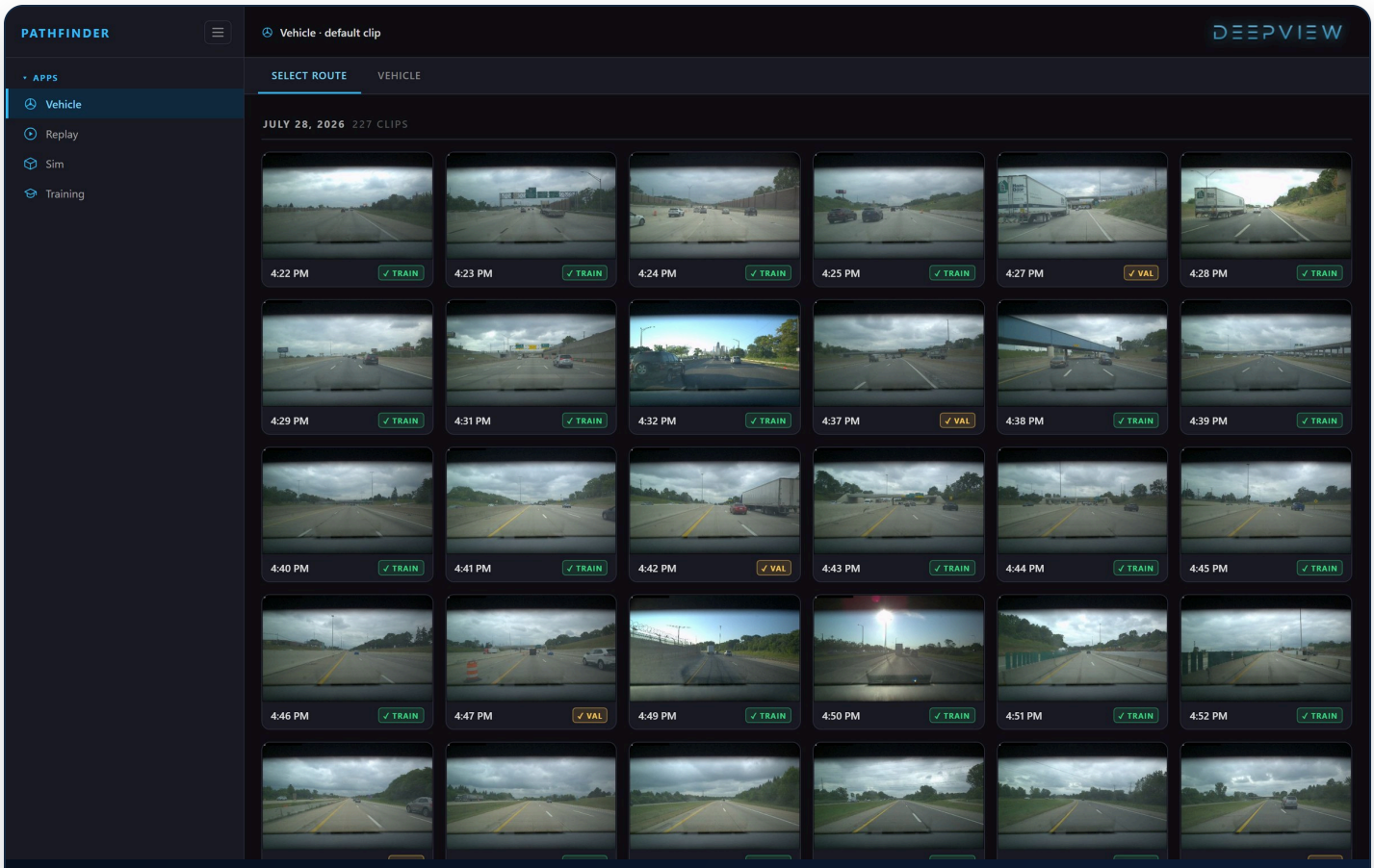
[1] Production automotive hardware is typically locked-down and difficult to iterate on. The DevKit offers a flexible environment for development and testing. Once complete, designs can be transitioned to production systems through your Tier 1 supplier.

Pathfinder Software Suite

A complete ADAS console, served straight from the DevKit to your browser

Every X2000 ships with **Pathfinder**, Deepview's browser-based console for the full development loop. Four integrated apps — **Vehicle**, **Replay**, **Sim**, and **Training** — cover live driving, frame-accurate playback, closed-loop simulation, and on-device model training. No IDE, no cross-compiling: connect over WiFi or Ethernet and drive.

VEHICLE — live cockpit & recording
REPLAY — synchronized video + CAN playback
SIM — MetaDrive closed-loop scenarios
TRAINING — on-device model training & review

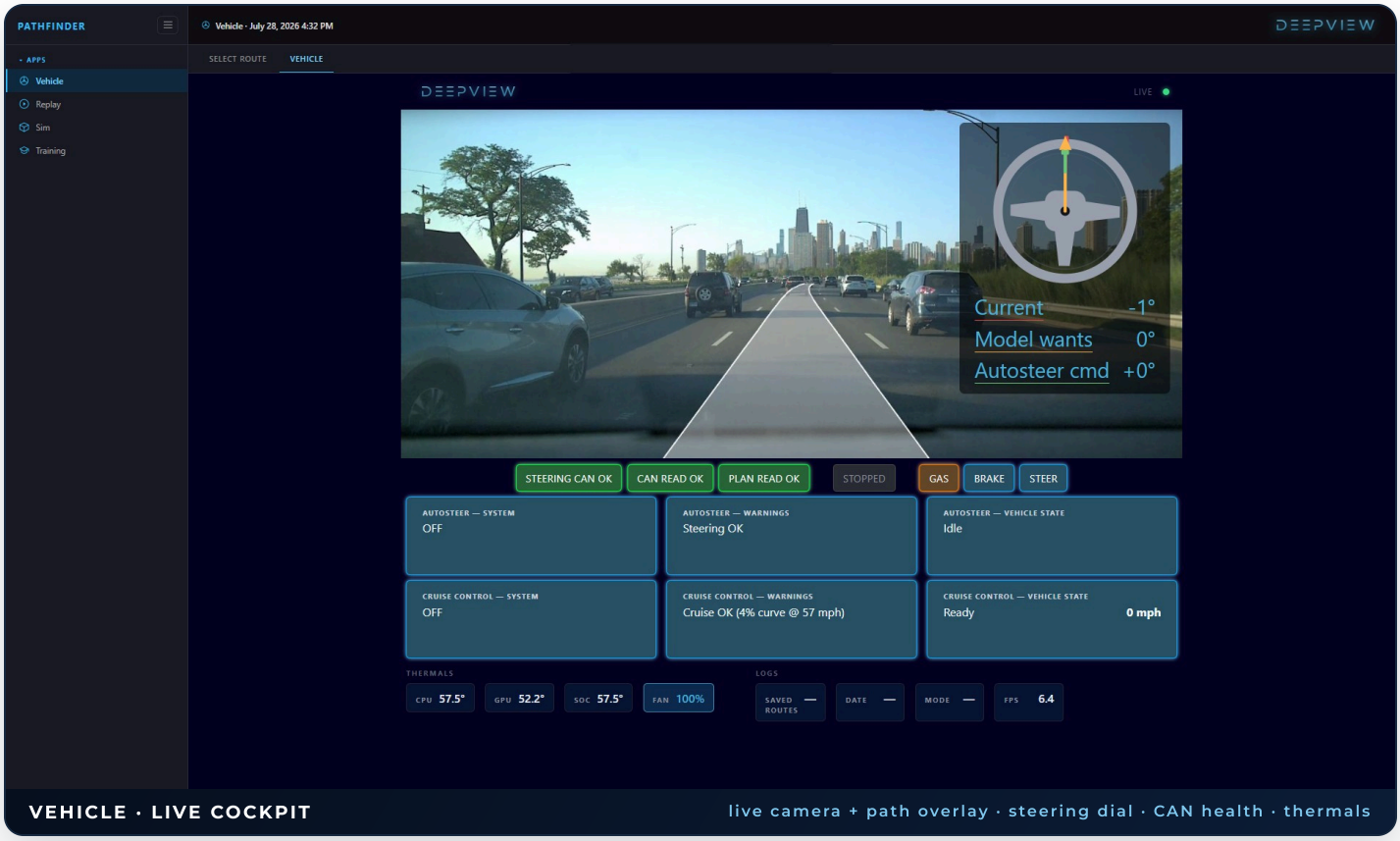


VEHICLE · ROUTE BROWSER — 227 one-minute clips · automatic train / val split

The **Vehicle** app opens on the route browser: a wall of one-minute driving clips, each thumbnail stamped with its capture time and a green ✓ **TRAIN** or amber ✓ **VAL** badge — Pathfinder assigns the 80/20 train/validation split automatically as clips are recorded. A single afternoon on the highway yields hundreds of labeled clips, browsable by date at a glance, and any thumbnail opens instantly into the clip inspector for frame-level review.

Vehicle — Live Cockpit

The whole control chain on one screen, in real time



VEHICLE · LIVE COCKPIT — live camera + path overlay · steering dial · CAN health · thermals

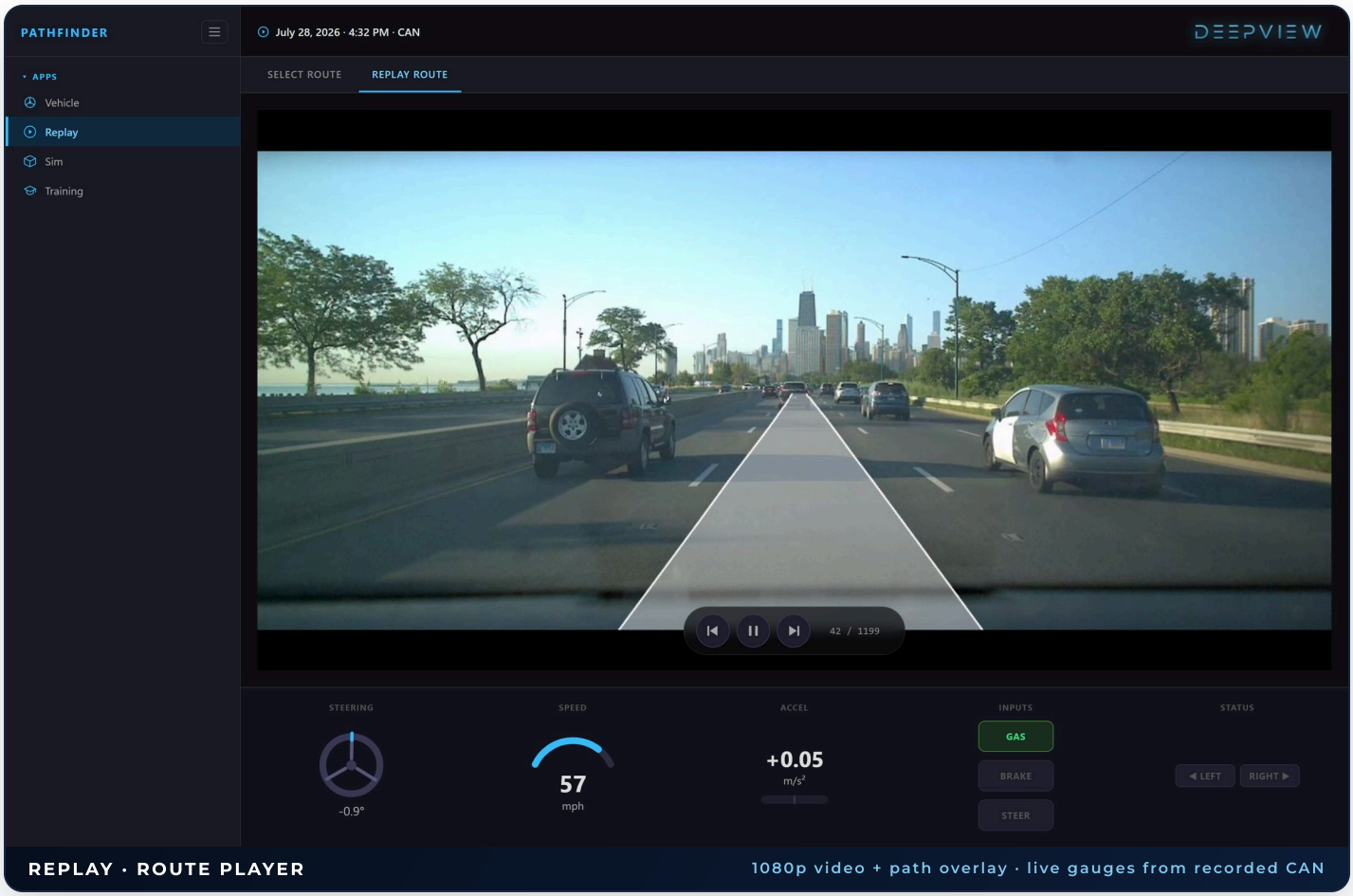
The live cockpit shows the vehicle's state in real time — captured here at highway speed, with the **LIVE** indicator lit and the predicted drive path drawn over the live camera feed. A steering dial overlays three needles — **Current**, **Model wants**, and **Autosteer cmd** — so you can watch the AI's intent against the wheel's actual angle, degree by degree. Bold status pills report the health of every link in the chain: **STEERING CAN OK**, **CAN READ OK** and **PLAN READ OK** glow green, faults flash red the moment they occur, and the **GAS / BRAKE / STEER** indicators light as the driver touches the controls.

Below, tiled panels split **Autosteer** and **Cruise Control** into system state, active warnings, and vehicle state ("Steering OK", "Cruise OK — 4% curve @ 57 mph", "Ready"), while the strip beneath keeps **CPU, GPU and SoC temperatures**, fan duty and UI frame rate in the corner of your eye — everything an engineer needs to trust the loop, on one screen.

> **Built for the passenger seat:** every safety-relevant state — override flags, CAN validity, plan health — is readable at arm's length, at highway speed, without touching a keyboard.

Replay — Frame-Accurate Playback

Every frame of video, every CAN message, perfectly aligned



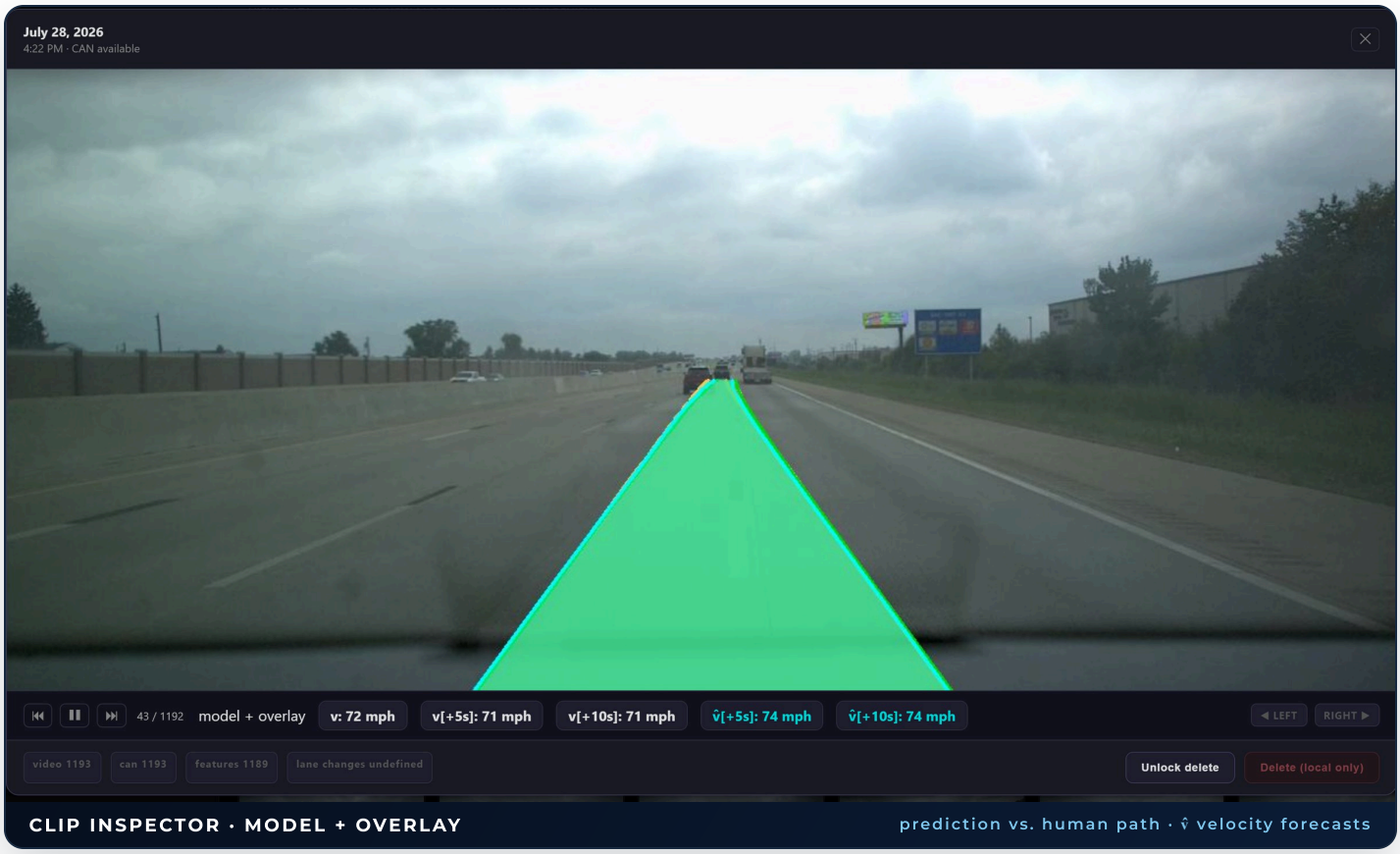
Select any clip and **Replay** plays it back exactly as the vehicle saw it: the predicted drive path is drawn over the road as a bright ribbon, while the gauge deck below re-animates the recorded CAN — steering angle turning the on-screen wheel (**−0.9°**), an arc speedometer sweeping to **57 mph**, longitudinal acceleration in m/s^2, and the driver's **GAS / BRAKE / STEER** inputs lighting up green the instant they were touched.

A frame counter (**42 / 1199**) with transport controls steps single frames forward or back for precise event analysis, and **LEFT / RIGHT** status chips flag recorded lane-change intent. Because every CAN message is aligned to its video frame at capture time, what you scrub is exactly what happened — no interpolation, no drift, mile after mile.

> **API-driven too:** the same replay engine is scriptable, so regression suites can sweep thousands of recorded frames through a new model build and diff the predictions automatically.

Clip Inspector — Model vs. Ground Truth

The loaded network re-drives every recorded mile



Opening any clip launches the **clip inspector** — the loaded network runs over the recorded video and paints its predicted path as a translucent green carpet, edged cyan where the model's prediction is compared against the human-driven ground truth. The telemetry bar reads actual speed **v: 72 mph** alongside the recorded future velocities (**v[+5s]: 71 mph, v[+10s]: 71 mph**) and the model's own forecasts (**v̂[+5s]: 74 mph, v̂[+10s]: 74 mph**) — an instant, visual answer to "what would the network have done here?"

Integrity chips at the bottom confirm the clip's data quality at a glance — **video 1193** frames, **can 1193** messages, **features 1189** extracted — and lane-change annotations travel with the clip. Deletion is deliberately two-step (**Unlock delete**) so a stray click never costs you training data.

Sim — Closed-Loop Simulation

Test the same code that drives the car, without leaving your desk

SIM · METADRIVE HIGHWAY LOOP — closed-loop autosteer · manual override panel · live gauges

The **Sim** app drops the full X2000 software stack into a **MetaDrive** highway scenario. The rendered forward view streams into the same perception pipeline used on the road, and the model steers the simulated vehicle in closed loop — engage **AUTOSTEER**, set a cruise speed, and dial in curvature bias from the transport bar. The **Manual Control** panel mirrors the on-road cockpit: wheel angle, **Model wants**, and **Autosteer cmd** shown side-by-side with steering health, and brake/gas overrides one click away.

> **Why it matters:** new models can be validated on the Highway Loop before they ever touch a CAN bus — regression-test a retrained network in minutes, then take the identical binary to the vehicle.

Scenario control

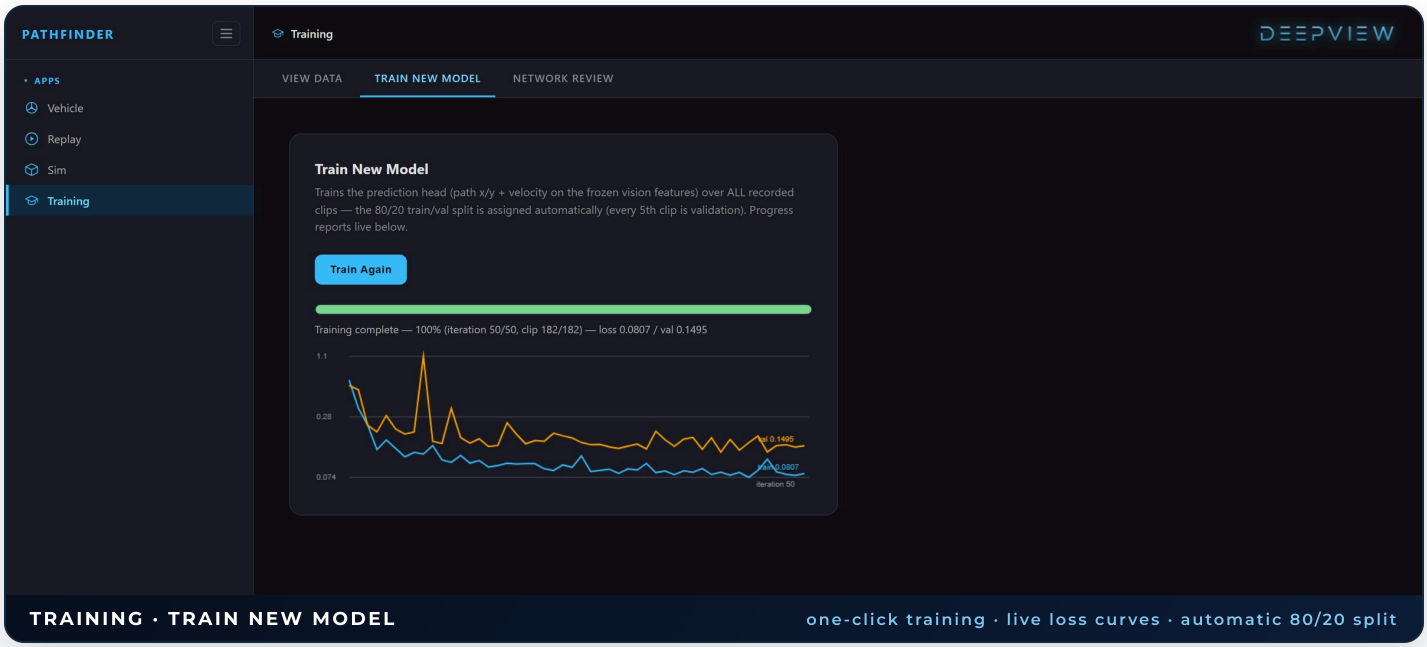
- **MetaDrive Highway Loop** scenario included
- Pause / reset, cruise set-speed, curve bias
- **LONG / LAT** control toggles for split testing

Identical stack

- Same TensorRT engine as on-road inference
- Same 20 Hz perception loop and control cadence
- Same UI, gauges and override semantics

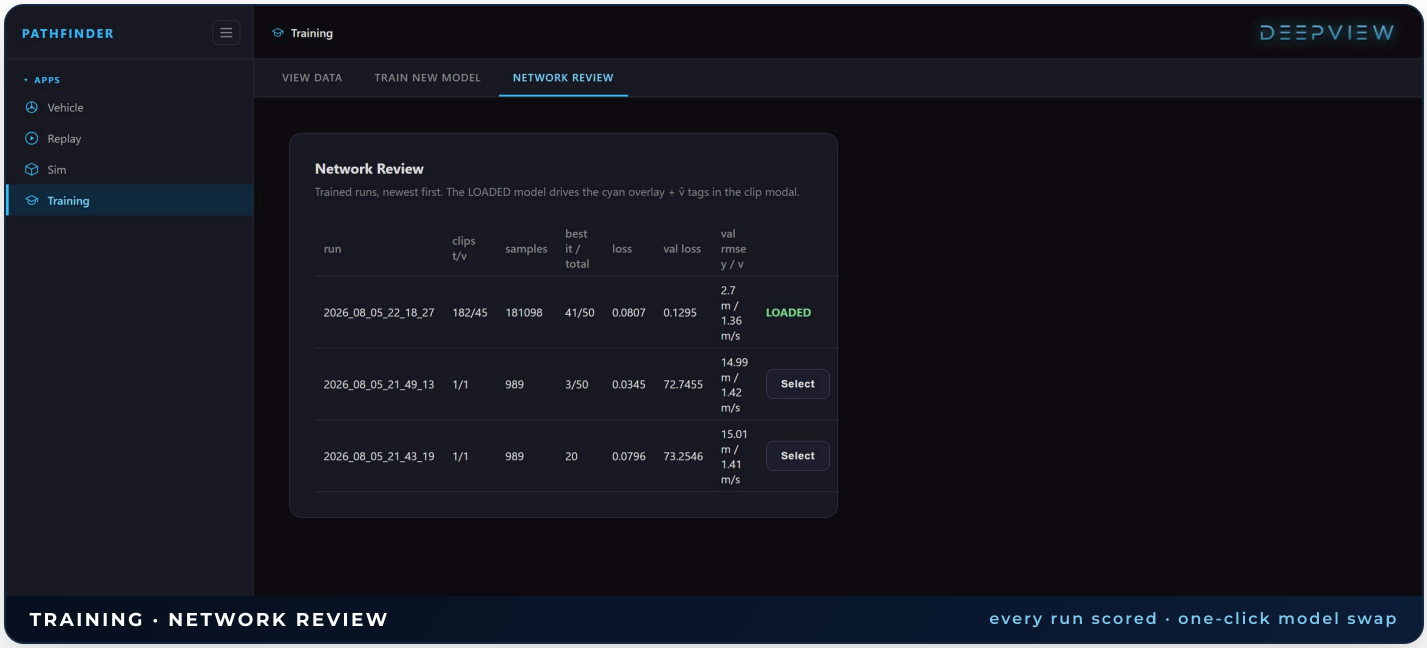
Training — Transfer Learning On-Device

Record in the morning, retrain over lunch, validate by afternoon



TRAINING · TRAIN NEW MODEL — one-click training · live loss curves · automatic 80/20 split

One button — **Train Again** — retrains the prediction head (path x/y + velocity on the frozen vision features) over every recorded clip. The 80/20 train/validation split is assigned automatically (every 5th clip is validation), and progress streams live: a green completion bar, iteration and clip counters, and paired loss curves converging in real time — here reaching **train loss 0.0807 / val 0.1495** across 50 iterations on 182 clips.



TRAINING · NETWORK REVIEW — every run scored · one-click model swap

Network Review keeps every trained run, newest first, with the numbers that matter: clip counts, sample totals (**181,098** here), best iteration, train and validation loss, and validation RMSE for path and velocity (**2.7 m / 1.36 m/s**). The **LOADED** model drives the cyan overlay and v̂ tags in the clip inspector — press **Select** on any other run to hot-swap it and compare generations of your network against the same clips.

In-Vehicle Development UI

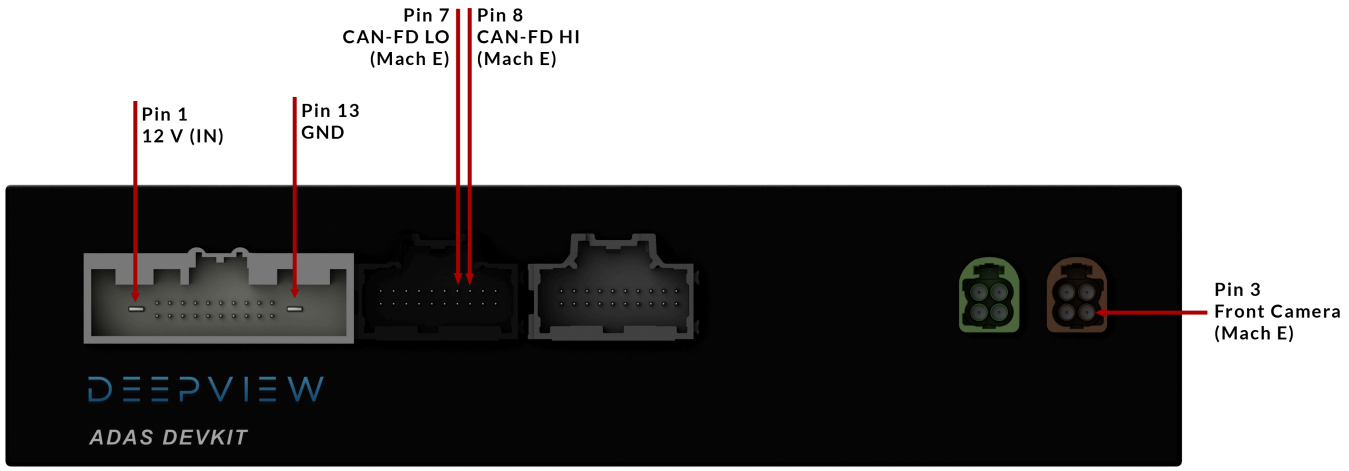
Live camera, live CAN, live model — on the vehicle display or any browser

The X2000's development UI runs on the vehicle's own screen via CarPlay or Android Auto, or in any browser — captured here **LIVE** on the highway. The camera feed carries the predicted drive path drawn over the lane in real time, while the steering dial reports **Current**, **Model wants**, and **Autosteer cmd** angles side by side.

Status pills verify the whole control chain at a glance — **STEERING CAN OK**, **CAN READ OK**, **PLAN READ OK** — with pedal indicators lighting as the driver touches gas, brake or wheel. Tiled panels track autosteer and cruise-control system state and warnings ("Cruise OK — 4% curve @ 57 mph"), and the strip below keeps CPU, GPU and SoC temperatures, fan duty and UI frame rate in view. Engineers verify system behavior on the road with zero external tools — if the car has a display, it has a debugger.

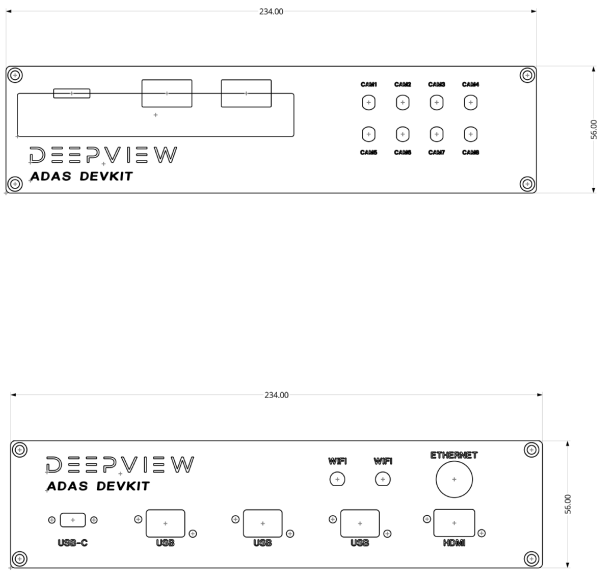
> **Everything in the loop is observable:** manual override flags, gas/brake inputs, CAN/MPC validity, steering angle and cruise status — synchronized with the live camera feed and the model's predictions.

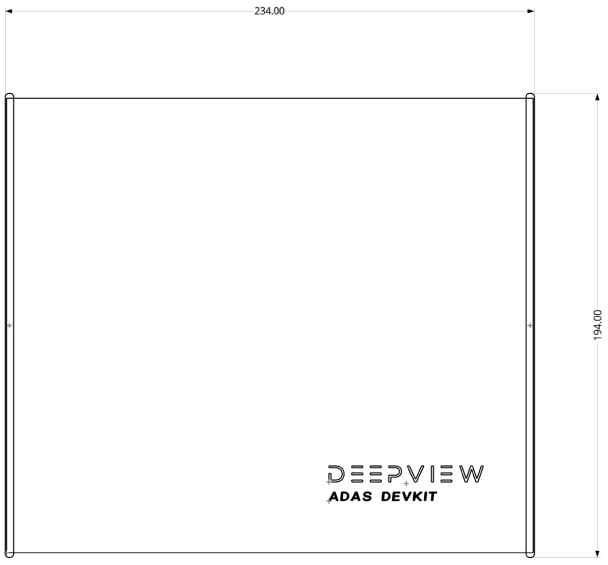
Wiring Diagram



Vehicle-side hookup uses four connections: **12 V power** (pin 1) and **ground** (pin 13) on the main harness, the **CAN-FD pair** (pins 7/8) into the Mach-E gateway, and the **front camera** FAKRA feed (pin 3).

Technical Drawings





Dimensions in mm. Front panel: 8× FAKRA GMSL2 camera ports (CAM1–CAM8) and vehicle harness connectors. Rear panel: USB-C, 3× USB, HDMI, Ethernet, and dual WiFi antennas.

Enclosure 234.00 × 194.00 × 56.00 mm — fanless, rugged aluminum construction, engineered for in-vehicle mounting.