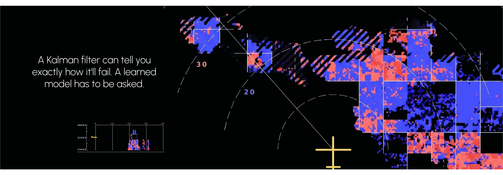
Case Study · Avionics

BUILDING A HYBRID NAVIGATION STACK WITH SQUINT VISION STUDIO

How embedding-space analysis can show where a learned model and a classical filter each earn their place.



An aircraft never relies on a single source to know where it is. Satellite positioning is accurate but slow and jammable. Inertial sensing is fast and self-contained but its error grows with time. Air-data sensors add airspeed and altitude, and the aircraft fuses all of it into one estimate of position, velocity and attitude.

That fusion has traditionally run through an error-state extended Kalman filter. Increasingly, teams are evaluating learned components to replace or supplement it, because a filter that must be hand-tuned for every error mode struggles with the conditions that cause the most trouble.

This case study is about the decision that follows, and about why the number most often used to make it is the wrong one.

THE CHOICE A NAVIGATION TEAM FACES

A learned fusion model is trained, evaluated, and posts a lower average error than the filter it might replace, which makes the engineering question look settled.

It is not, and the reason has nothing to do with the model being poor. Replacing a component whose failure modes are understood with one whose failure modes are unknown is a trade, not an upgrade. A filter is analytically tractable. Its response under a given error condition can be derived rather than measured, which is a substantial part of why it has held the role for decades.

A learned model gives up that property in exchange for handling conditions the filter cannot. Whether that trade is worth making depends entirely on where each component is strong, and an average error figure answers a different question.

THE LIMITS OF AN AGGREGATE SCORE

An aggregate score describes performance across a test distribution. It says nothing about how performance is distributed within it.

Squint's published research makes this point directly. Testing a classifier across a dataset, the team found that the likelihood of a mistake for any given prediction was not evenly spread. Across one class pair the baseline model carried a 10.9 percent top-one error rate overall. Separating the data by region produced a very different picture: 4.51 percent error across the region where the model had consistent experience, and 25.63 percent in the region where classes overlapped.

The same model. The same test set. A fivefold difference in error depending on which part of the data the prediction came from.

For a navigation team, the implication is that comparing two average error figures compares two summaries of unevenly distributed behavior. The learned model may be better in aggregate while being worse in the specific conditions the filter was designed around, or better only in conditions the aircraft rarely encounters.

COMPARING TWO MODELS BY REGION

Squint Vision Studio is built to make that distribution visible. It extracts the internal representations a network builds as it processes an input, then clusters them to show how the training data is organized inside the model, marking regions where accuracy falls below a chosen threshold as ambiguous.

Applied before a deployment decision rather than after one, this changes what a team is comparing. Instead of two numbers, they have a map of the sensor conditions the learned model resolves dependably and the conditions where it is extrapolating.

That map can be read against the filter's known behavior. The filter's performance envelope is derivable from its structure, so the comparison is between one component whose limits are analytically known and another whose limits are now empirically characterized. Where the learned model's dense regions cover conditions the filter handles poorly, the case for it is concrete. Where its sparse regions overlap conditions the filter handles well, the case is against.

Slice-level analysis supports the same comparison from another direction, reporting which parts of the data a model handles well and which it struggles with rather than collapsing everything into one figure.

DESIGNING THE HANDOVER

The comparison usually argues for neither component alone.

A learned model earns its place in the messy conditions that motivated it, such as multipath in difficult terrain, degraded satellite geometry, or interference patterns that resist closed-form modeling. The filter remains the better choice in the nominal conditions it was designed for and where its behavior can be predicted rather than sampled.

Vision Studio exports the region analysis as a watchdog, a lightweight decision layer that runs alongside the model rather than replacing it, deciding when the base model can be trusted and when further processing is worth the cost. In a hybrid architecture that becomes the handover mechanism. Sensor conditions inside the learned model's well-supported regions use its estimate. Conditions outside them fall back to the filter, which is not a degraded mode but the component better suited to that case.

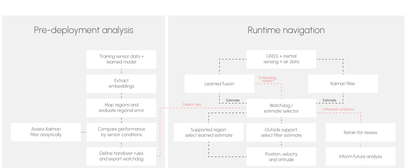
Conditions unlike anything in the training data are retained for review, because they mark a gap worth closing before the learned component is trusted more widely.
The result is an architecture where each component runs where the evidence supports it, and where the boundary between them is documented rather than assumed.

WHAT THIS EVIDENCE DOES NOT SETTLE

The regional comparison characterizes the learned model against its training distribution. It does not tell you how that model behaves in conditions absent from that distribution, which is the same limitation any empirical characterization carries.

It also says nothing about the classical filter. A Kalman filter produces no internal representations to cluster, so its side of the comparison still comes from analysis rather than from this method.

And the map is only as good as the data behind it. If the training set underrepresents a condition the aircraft will meet, the region covering it may appear better supported than it is. That risk is why the review path matters and why the boundary is worth revisiting as operational data accumulates.



A DECISION MADE ON STRUCTURE

Replacing a well-understood component with a learned one is a decision made under uncertainty, and the usual basis for it is a single comparison of averages between two systems whose behavior is unevenly distributed.

Examining where each component is strong replaces that comparison with something more useful. It tends to produce a different answer as well, because the honest conclusion is rarely that one component should replace the other. It is that each should run where it has been shown to work, with the handover between them defined in advance rather than discovered in service.

This case study describes a representative scenario rather than a specific engagement. The sensor fusion architecture, the tractability of classical filters, and the open questions around learned components in navigation are drawn from published research in the field. The regional error figures cited are from Squint Cognition's published work on image classification and are used to illustrate uneven error distribution rather than to describe navigation performance. Squint Vision Studio's capabilities, including embedding extraction, clustering-based region mapping, slice-level analysis and exportable watchdog modules, are described as documented by Squint Cognition rather than measured from this scenario.