ASME OMAE® 2026

45TH INTERNATIONAL OCEAN OFFSHORE AND ARCTIC ENGINEERING CONFERENCE®

GRAND NIKKO TOKYO DAIBA, GRAND TOKYO, JAPAN

ASME
OMAE® 2026
45TH INTERNATIONAL OCEAN OFFSHORE AND ARCTIC ENGINEERING CONFERENCE®

AquaScan

Graph-Based Learning for Distributed Marine Sensing

Abel Dantas

Faculty of Engineering, University of Porto
CTVC | coop.tech

OMAE 2026 | Paper #175455

Problem statement

Tracking marine entities that do not carry transponders —
fish, cetaceans, untagged or non-cooperative vessels —
in sparse, drifting sensor networks.

Operational context

Marine spatial use is intensifying: offshore wind, MPAs, fisheries, ports.

Existing monitoring infrastructure is built for *cooperative* targets (AIS, transponders).

Why this is hard

Non-cooperative targets do not broadcast, do not follow declared routes, and most of the area is unmonitored at any given moment.

Coverage gaps are precisely where the events of interest occur.

Position of this work

Author's research line:

Eventual convergence and Byzantine fault tolerance in distributed systems.

This paper is the marine-sensing instantiation of a broader question: *can collective relational reasoning across a network outperform per-entity estimators?*

Selected prior work

- Dantas, Baquero, Rodrigues. *CRDT-Based Game State Synchronization.* PaPoC 2025 (EuroSys workshop) — published.
- Dantas et al. *Bounding Byzantine Impact in Open CRDT Systems.* PaPoC 2026 (EuroSys workshop, Edinburgh) — published.
- Dantas. *Extending MadSim for Deterministic Byzantine Simulation.* SRDS 2026 Tool Paper — submitted.

Independent estimators: the Kalman family

1. Standard since the 1960s

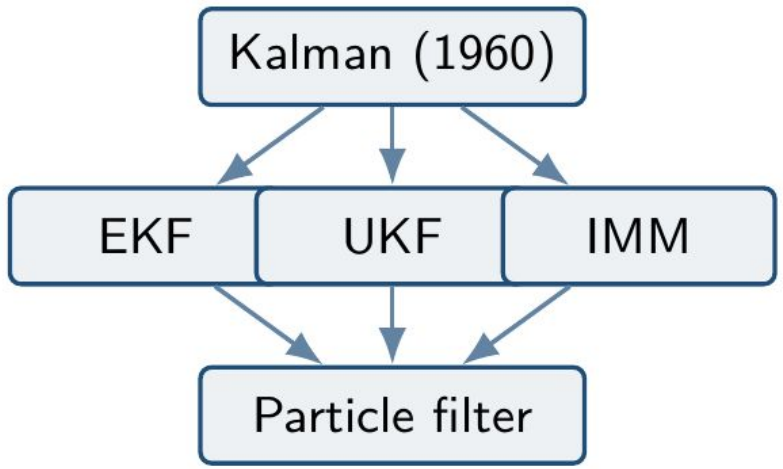
Kalman filtering and its descendants — EKF, UKF, IMM, particle filters — are the reference for marine trajectory estimation.

2. Where they work

Cooperative targets, known motion models, reliable observations.

3. Shared structural assumption

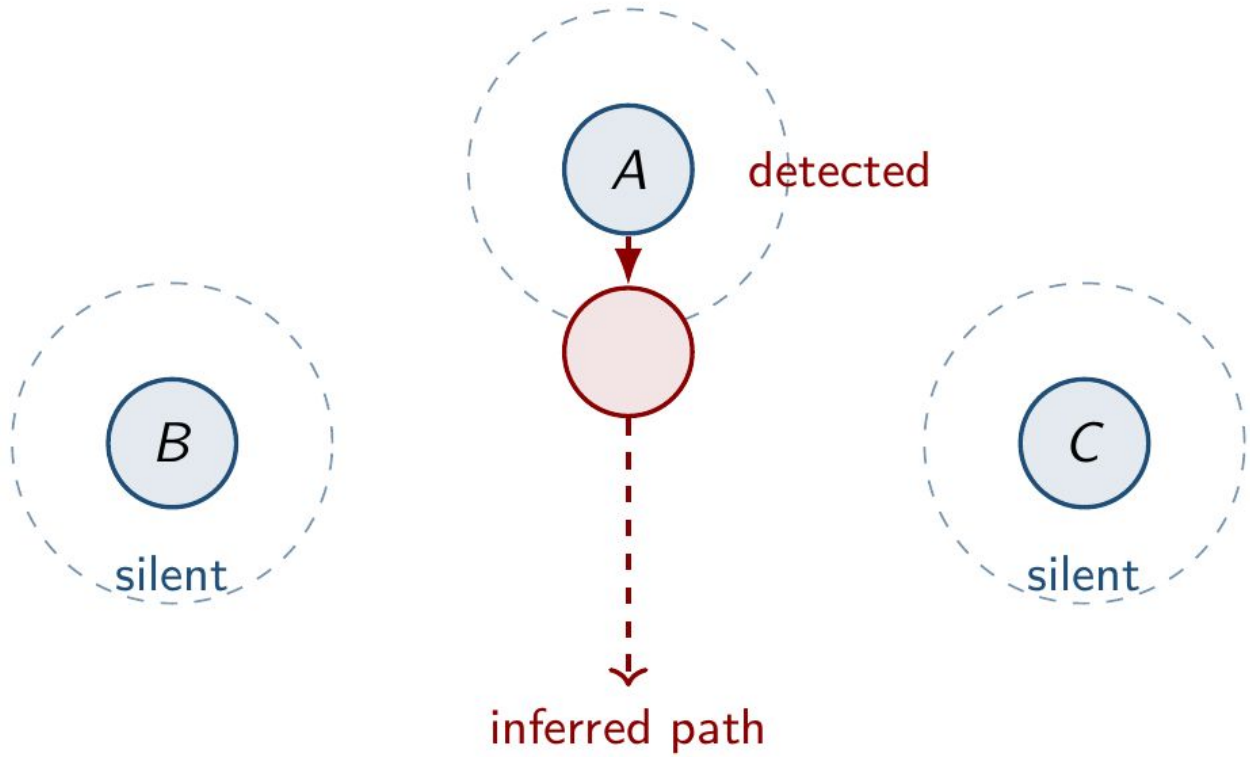
One estimator per target; targets tracked independently.



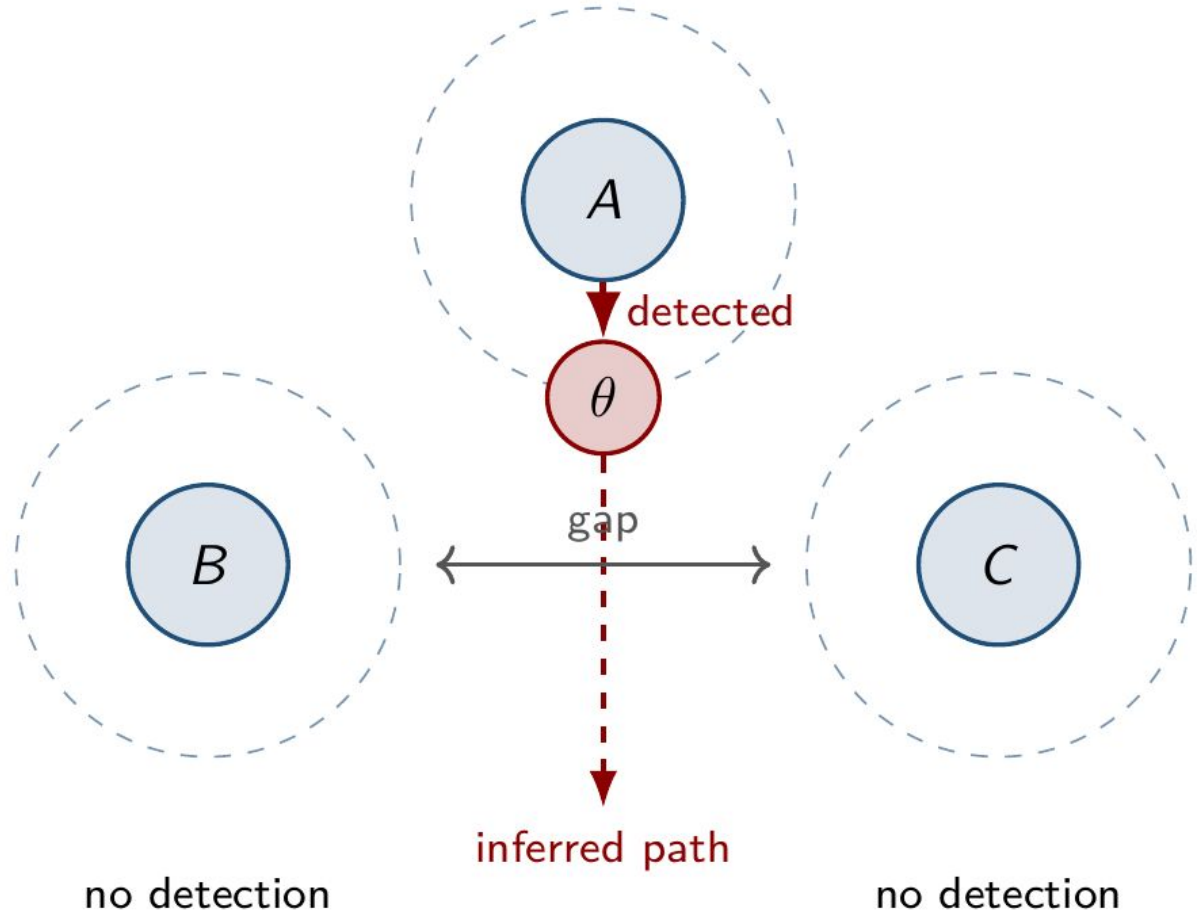
Common ancestry;
common independence assumption.

Non-detection as information

If sensor A detects an entity and neighbouring sensors B, C do *not*, the non-detections constrain the entity's possible trajectory.



Non-detection in the deployed geometry



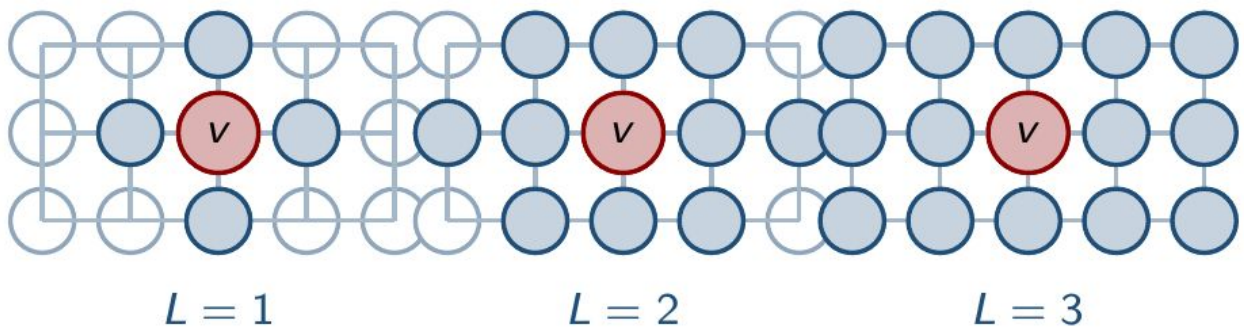
200 m sensing radius vs. 1 km grid spacing $\Rightarrow \approx 12.6\%$ of the area covered at any moment.

Graph Neural Networks

Message-passing update (one layer):

$$\mathbf{h}_v^{(l+1)} = \sigma\Big(\mathbf{W}_{\text{self}}^{(l)}\,\mathbf{h}_v^{(l)} + \sum_{u\in\mathcal{N}(v)} \mathbf{W}^{(l)}\,\mathbf{h}_u^{(l)}\Big)$$

After L layers, $\mathbf{h}_v$ encodes information from up to L hops away.



Each additional layer extends the receptive field by one hop.

Research question

RQ

Can heterogeneous Graph Neural Networks outperform constant-velocity Kalman filters for trajectory prediction in distributed marine sensing networks with intermittent observations?

Sub-questions.

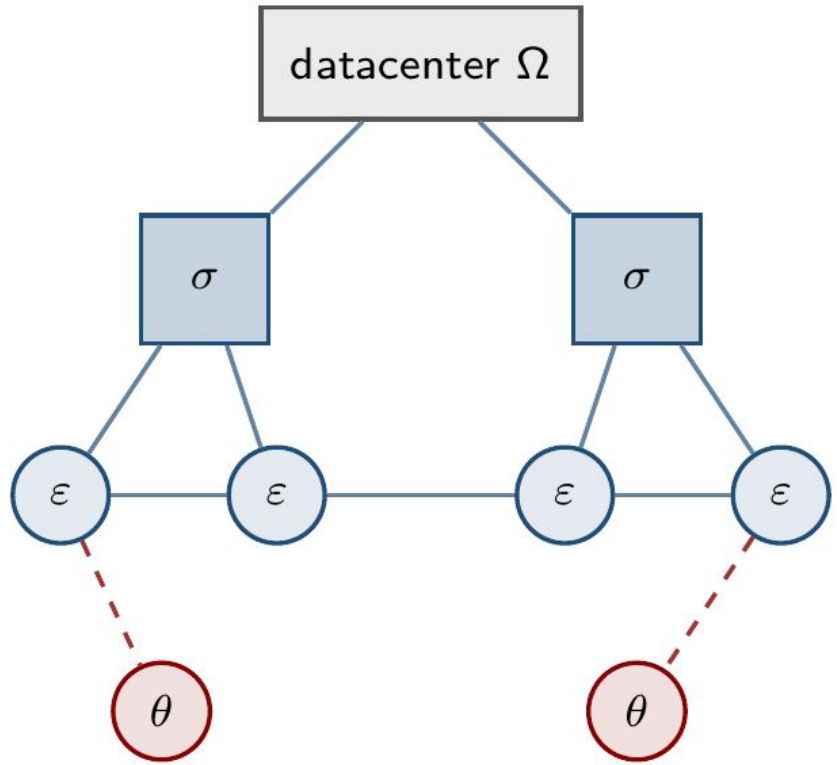
1. How does the comparison vary across prediction horizons (short, medium, long)?
2. To what extent does the GNN's advantage derive from network topology and multi-hop reasoning?

AquaScan architecture

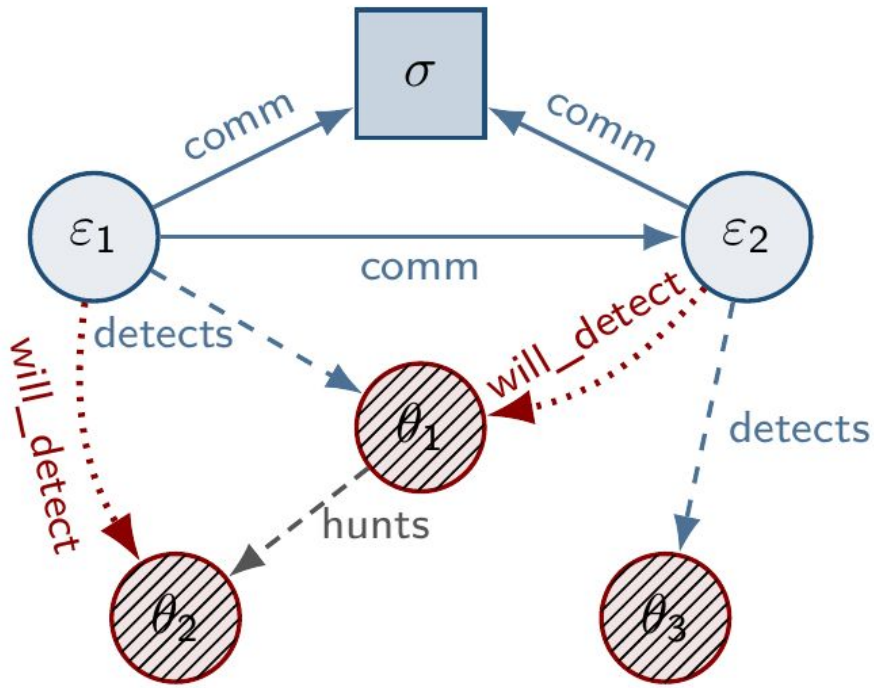
Three node types, organised hierarchically:

- **Sensor buoys** ε-nodes — mobile, low-cost; local sensing within a 200 m radius.
- **Hub buoys** σ-nodes — fixed; long-range aggregation and relay.
- **Entities** θ-nodes — marine animals or untagged vessels; observed only when within range of a sensor buoy.

Hierarchy is driven by an energy budget: dense, low-power edge devices supported by sparse, higher-capacity aggregators.



System as a heterogeneous graph

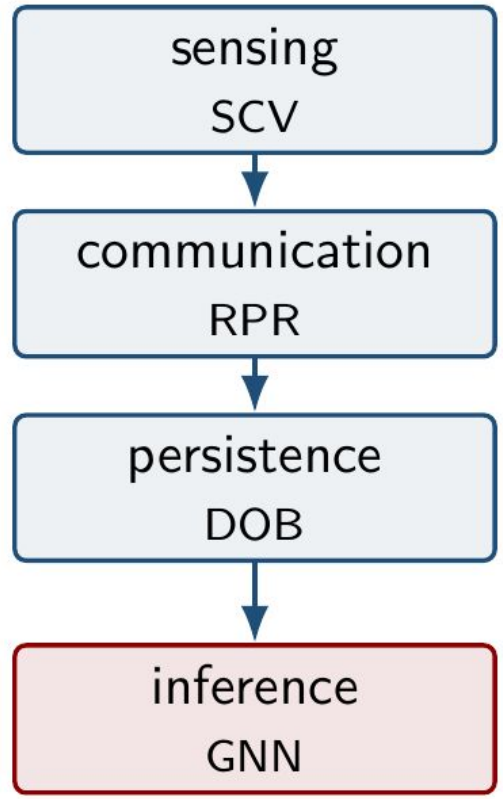


σ hub, ε sensors, θ entities. *comm* (solid), *detects* (dashed, historical), *will_detect* (dotted, prediction target).

Protocol abstractions

AquaScan defines three protocol abstractions that separate concerns across the network. They are presented in the paper; here, only the role each plays in the data pipeline.

Protocol	Role
SCV	Unified detection record across heterogeneous sensors.
RPR	Multi-hop communication with range-adaptive transmission.
DOB	Time-ordered local observation log.



The contribution is not the protocols themselves but the separation of concerns: the framework supports substitution of each layer independently.

Graph construction

At each time step, the network state is represented as a heterogeneous graph $G_t = (V, E_t)$.

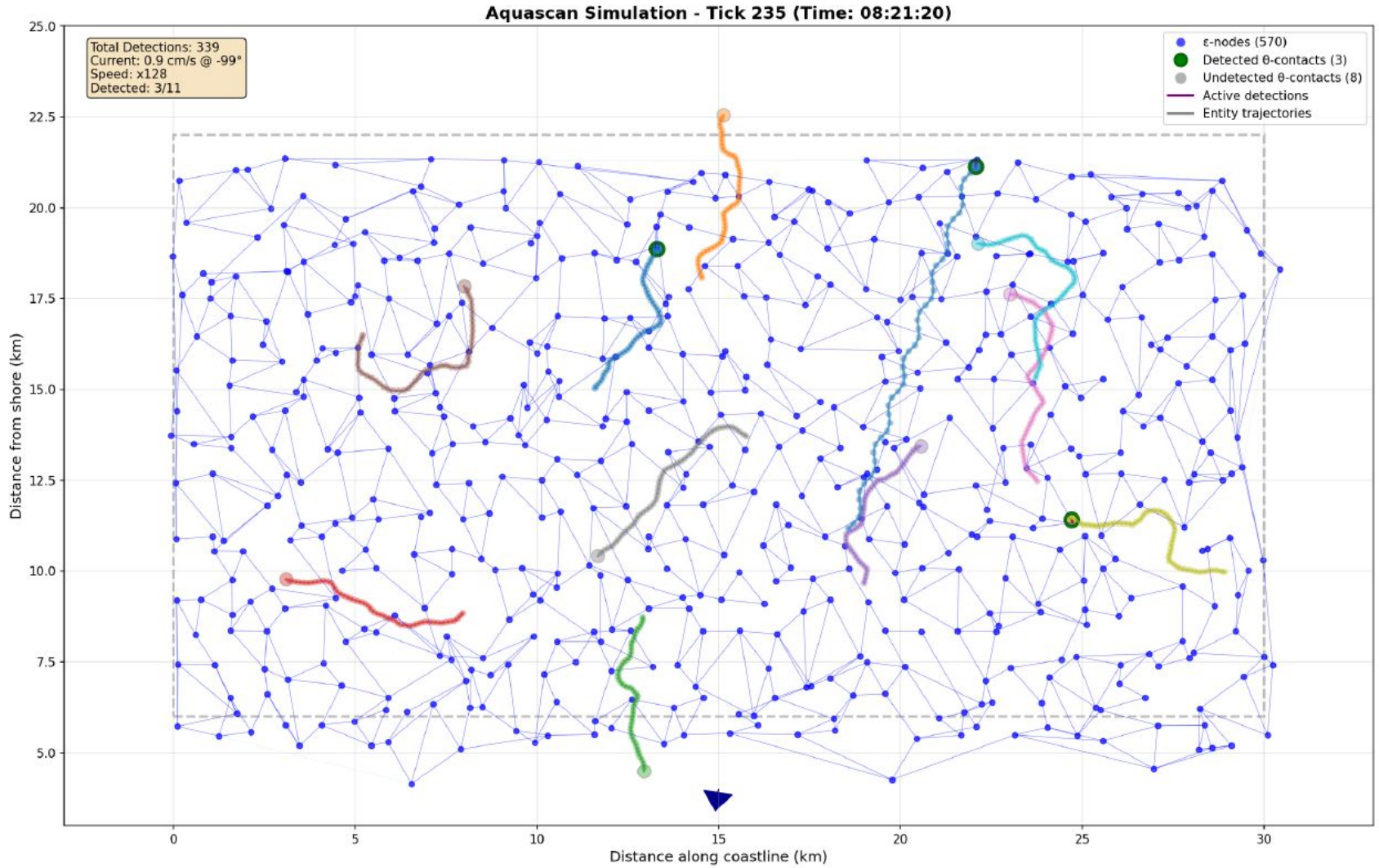
Nodes.

- V_ε: sensor buoys. V_θ: entities.
- Feature: $\mathbf{h}_v^{(0)} = [x, y, \Delta x, \Delta y]^\top$, averaged over a 60-tick context window.

Edges (typed):

- `communicates`: $\varepsilon \leftrightarrow \varepsilon$. Topology given by Delaunay–Voronoi mesh (each sensor connects to 3–5 neighbours; mesh recomputed locally as sensors drift).
- `detects`: $\varepsilon \to \theta$. Sensor observed entity within the last context window.
- `will_detect`: $\varepsilon \to \theta$ at $t + \Delta t$. Masked during training; this is the prediction target.

Simulation environment



Experimental setup

- Region: 30×16 km
- 570 sensor buoys, hexagonal grid, $\sim$1 km spacing
- 15–20 entities; two motion classes:
  - Drift-dominated random walk (*fish*)
  - Directed sinusoidal trajectory (*cetaceans*)
- Currents: Perlin-noise field; 6 h tidal period
- 100 independent 24-h simulations
- $\sim$25,000 graphs; 23.5 M prediction instances
- Temporal split: 70 / 15 / 15

The regime under test

Sensing radius 200 m + 1 km spacing

$\Longrightarrow$ 12.6% direct coverage

87.4% coverage gap

This is the regime in which independent estimators degrade and the relational signal becomes load-bearing.

Task formulation

Link prediction on a heterogeneous spatiotemporal graph.

Given G_t, predict the set of `will_detect` edges in $G_{t+\Delta t}$.

Δt **(ticks)**	**Wall-clock**	**Regime**
30	$\approx$ 5 min	short
100	$\approx$ 18 min	medium
150	$\approx$ 27 min	long

Class imbalance: positive edges constitute 0.065% of candidate pairs.
Loss: binary cross-entropy with positive-class weighting $w_+ = |E^-|/|E^+|$.

1 tick = 10.7 s of real time.

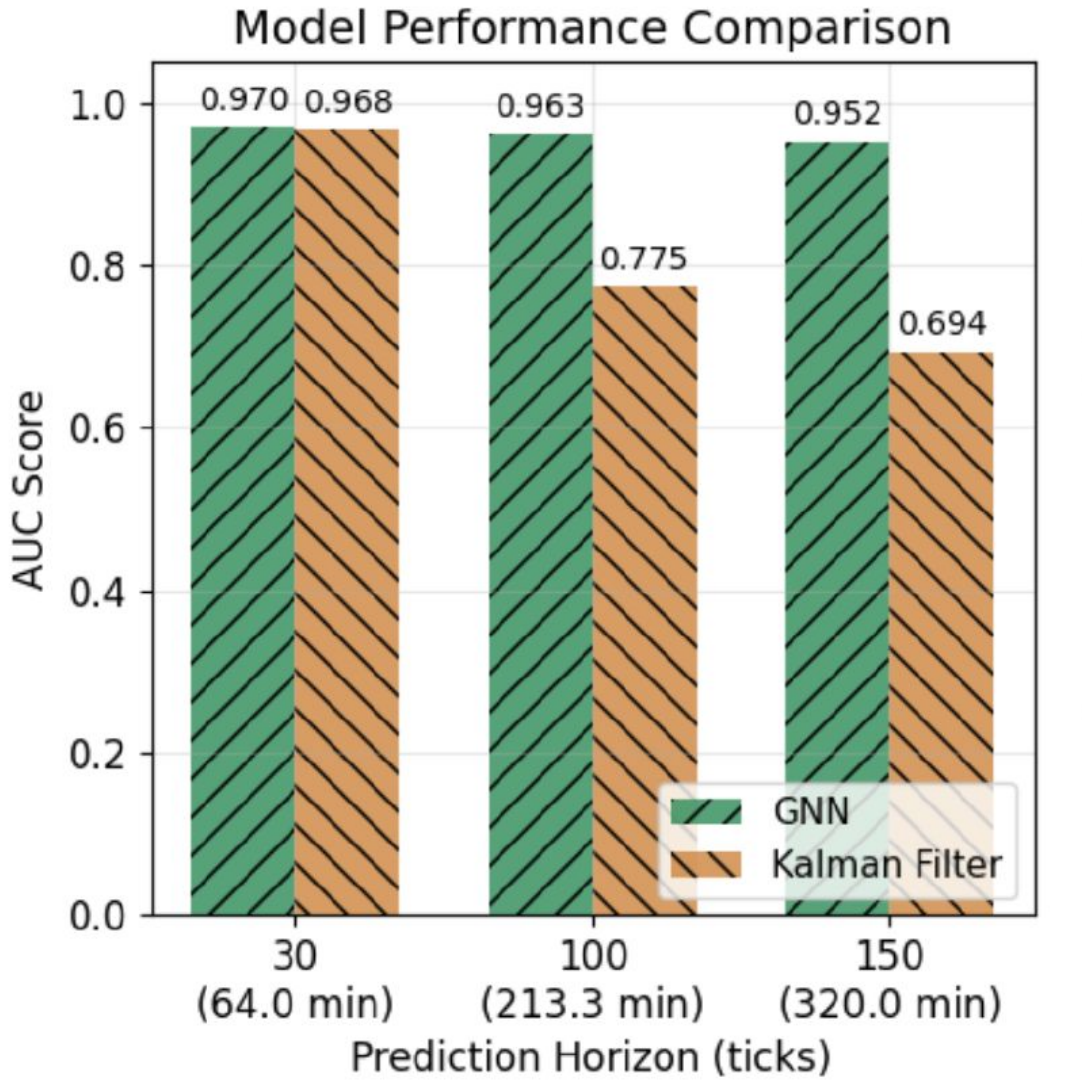
Results — AUC across prediction horizons

Horizon	Kalman (CV)	GNN	p-value
30 t (≈5 min)	0.968 ± 0.011	0.970 ± 0.012	0.23
100 t (≈18 min)	0.775 ± 0.018	0.963 ± 0.013	<0.001
150 t (≈27 min)	0.694 ± 0.021	$\mathbf{0.952 \pm 0.014}$	<0.001

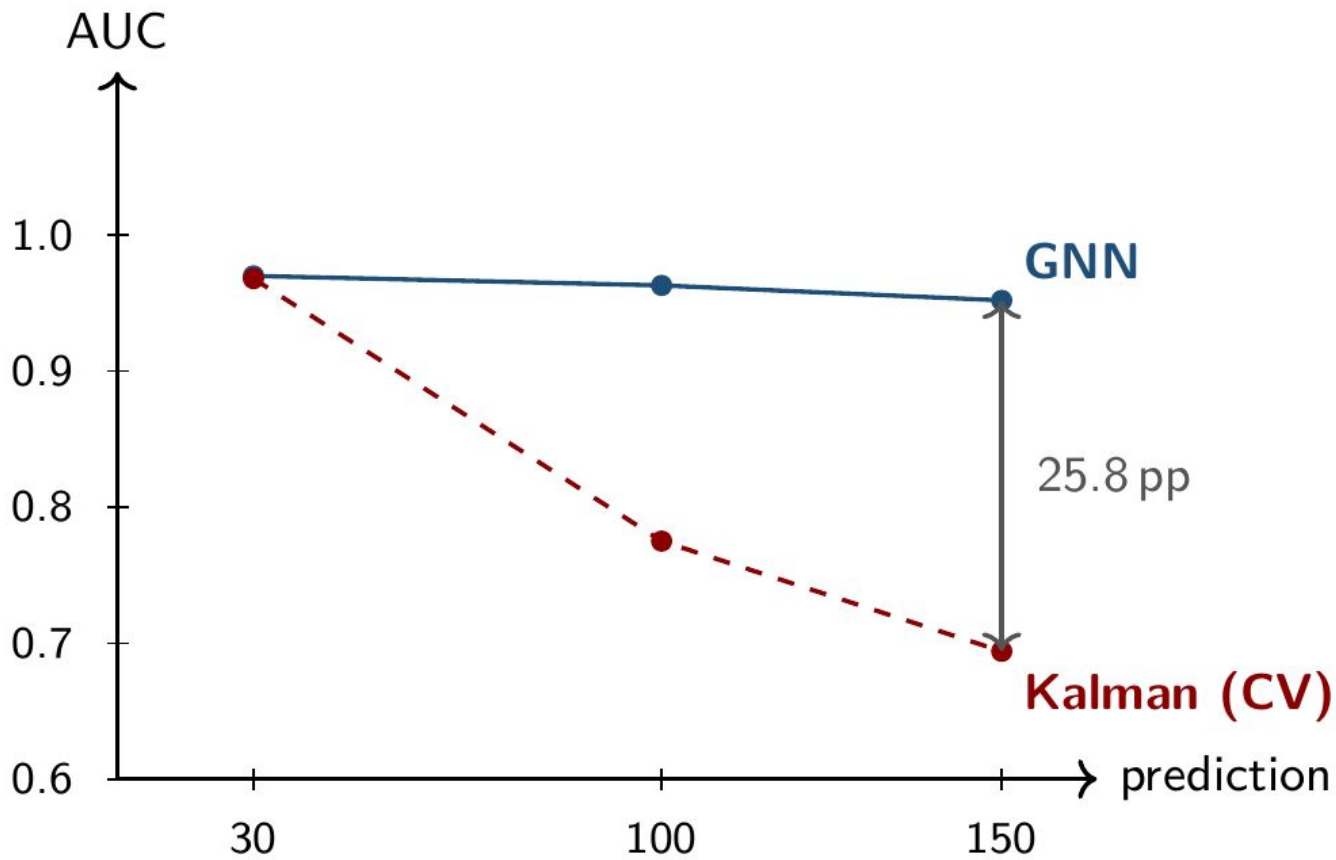
- **Short horizon:** no statistically significant difference. The two methods are equivalent when observations are dense and the prediction window is short.
- **Medium horizon:** 18.8 pp gap.
- **Long horizon:** 25.8 pp gap.

Means and standard deviations over 100 independent 24-h simulations.

Results — AUC across horizons



Results — degradation profile



Kalman: 28% relative drop in AUC from 30 to 150 ticks.

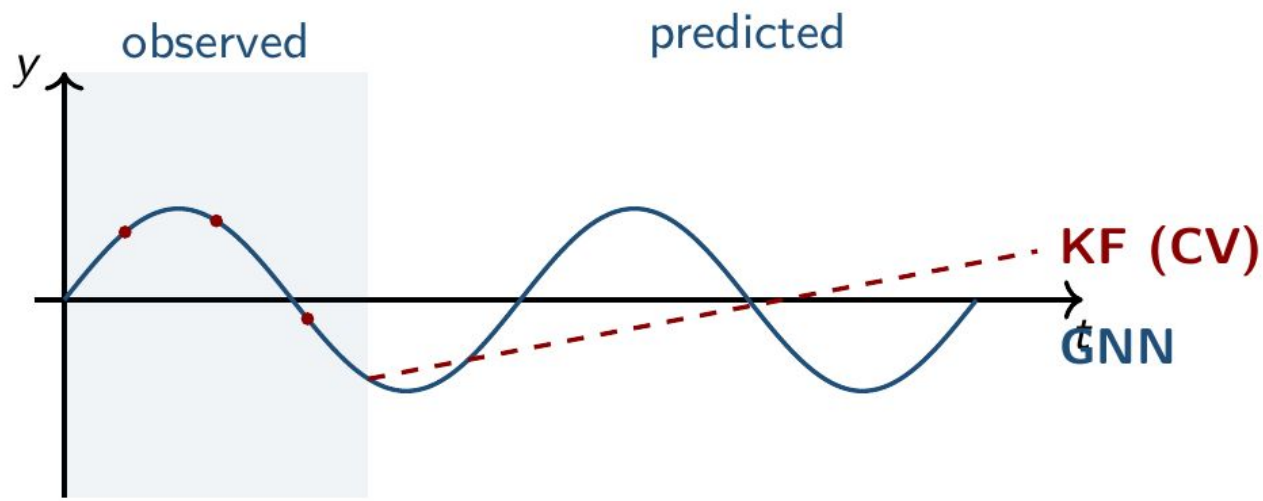
GNN: 2% relative drop.

The shapes differ qualitatively: the GNN is approximately flat across horizons; the KF degrades approximately linearly.

Mechanism — multi-hop reach and non-detection

Two effects, both unavailable to per-entity estimators by construction:

- **Multi-hop reach.** With $L = 3$ layers on a $\sim$1 km grid, each sensor incorporates information from up to $\sim$3 km away.
- **Non-detection signal.** Silent sensors also participate in message passing; their silence constrains the trajectory.



Schematic: trajectory shape under each model.

Limitations and scope

Synthetic data

Hand-crafted motion models, not validated against marine biology. Chosen to be distinct and learnable, not realistic.

Baseline scope

Constant-velocity Kalman only. EKF, UKF, IMM, particle filters, and modern sequence models are not evaluated here.
Prior literature suggests they reach $\sim$0.75–0.82 AUC at 150 ticks — below the GNN, but the gap narrows.

Idealised sensing

Perfect positioning ($\delta_{\text{pos}} = 0$); noise-free communication. We hypothesise the GNN degrades more gracefully than per-entity estimators under realistic noise; this is not yet demonstrated.

ASME OMAE® 2026
45TH INTERNATIONAL OCEAN OFFSHORE AND ARCTIC ENGINEERING CONFERENCE®

AquaScan: Graph-Based Learning for Distributed Marine Sensing

OMAE2026-175455

Thank you.

Implications and motivations

- At deployable scales, the binding constraint is rarely sensor density — it is the cost of physical infrastructure relative to compute.
- Halving sensor spacing increases deployment and maintenance cost *quadratically* in the monitored area.
- The relational-reasoning advantage (25.8 pp at 150 ticks) corresponds to coverage that would otherwise require $\sim 4\times$ denser sensor placement under independent tracking.
- Complementary direction: drive sensor-unit cost down. Cheaper, more numerous edge devices amplify the value of network-aware inference — sparse coverage stops being a hardware problem.
- OMAE-relevant deployments where this trade matters: wind-farm operational monitoring, MPA enforcement, dark-vessel detection in restricted areas.

AquaScan is released as an open framework so this trade-off can be evaluated against alternative architectures.

On the choice of baseline

Anticipating an objection raised in review:

> *"The constant-velocity Kalman filter is the simplest variant in the family."*

Three reasons we keep it as the reference.

1. It is the documented, reproducible baseline in marine tracking literature.
2. Stronger *independent* estimators (EKF, UKF, IMM, particle filter) reduce the per-entity error but still cannot consume non-detections or multi-hop signals — by construction.
3. Reported lifts from non-linear variants in comparable settings are on the order of 10–15 pp at long horizons; the relational gap remains material.

The KF is tuned, not strawmanned. The aim of the comparison is to isolate the contribution of the *graph*, not to claim the strongest possible per-entity estimator.

Model: heterogeneous GraphSAGE

Three message-passing layers with type-specific weights:

$$\mathbf{h}_v^{(l+1)} = \sigma\Big(\mathbf{W}_{\text{self}}^{(l)}\mathbf{h}_v^{(l)} + \sum_{r\in\mathcal{R}}\sum_{u\in\mathcal{N}_r(v)} \mathbf{W}_r^{(l)}\,\text{MEAN}(\mathbf{h}_u^{(l)})\Big)$$

with $\mathcal{R} = \{\texttt{communicates}, \texttt{detects}, \texttt{rev\_detects}\}$.

Decoder. Dot-product on the final embeddings of each candidate (ε, θ) pair:

$$p_{\varepsilon\theta} = \sigma(\mathbf{h}_\varepsilon^{(3)} \cdot \mathbf{h}_\theta^{(3)}).$$

Implementation. PyTorch Geometric. Hidden dimension 64. ReLU + batch normalisation. Adam, $\eta = 10^{-3}$. Early stopping (patience 10) on validation BCE.

Code, configurations and trained checkpoints are available in the repository (Zenodo-archived).

Baseline: constant-velocity Kalman filter

One independent filter per entity. State $\mathbf{x} = [x, y, v_x, v_y]^\top$. Linear motion model:

$$\mathbf{x}_{k+1} = F\mathbf{x}_k + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(0, Q).$$

Process noise q selected by grid search over $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}\,\mathrm{km}^2/\mathrm{tick}^2$ on validation: $q = 10^{-2}$.
Measurement noise $r = 0.01\,\mathrm{km}^2$, corresponding to $\sigma = 100\,\mathrm{m}$ (half the sensing radius).

Prediction beyond the last observation: pure motion-model extrapolation under the constant-velocity assumption.

Backup — dataset

	30 t	**100 t**	**150 t**
Total graphs	14,700	7,700	2,700
Training	10,290	5,390	1,889
Validation	2,205	1,155	405
Test	2,205	1,155	406
Total predictions	13.8 M	7.2 M	2.5 M
Dataset size	5.1 GB	2.7 GB	0.96 GB
Positive-class weight	1,534 : 1	712 : 1	364 : 1

Hardware: NVIDIA T4 (16 GB) + 32 GB system memory.
Training: $\approx$4 h per configuration. Inference latency: $<$50 ms per prediction.

Backup — hyperparameters

GraphSAGE.

- Layers: 3 heterogeneous SAGEConv.
- Hidden dimension: 64.
- Activation: ReLU + BatchNorm.
- Optimiser: Adam, $\eta = 10^{-3}$.
- Batch size: 32 graphs.
- Early stopping: patience 10, max 100 epochs.
- Aggregator: mean.

Kalman.

- State: $[x, y, v_x, v_y]^\top$, constant velocity.
- Process noise q: grid search on validation, $q = 10^{-2}$.
- Measurement noise $r = 0.01\,\mathrm{km}^2$, $\sigma = 100\,\mathrm{m}$.
- One filter per entity.

Backup — extended discussion of the baseline

Anticipated objection: *CV-KF is the weakest member of the Kalman family; the reported gap overstates the GNN's advantage.*

Response.

1. The CV-KF here is grid-search tuned and selected on validation. It is not strawmanned within its model class.
2. Non-linear and multi-model variants (EKF, UKF, IMM, PF) typically lift AUC by 10–15 pp at long horizons in comparable tracking scenarios. At 150 t, this would place the strongest such estimator at $\sim$0.75–0.82.
3. The remaining gap to the GNN ($\sim$13–20 pp at 150 t) is the portion attributable to relational reasoning specifically: multi-hop reach and non-detection.
4. No per-entity estimator can consume the non-detection signal by construction, regardless of motion-model sophistication.

The claim is therefore not "GNN beats KF", but: graph-structured inference recovers information that any independent estimator leaves on the table.

Backup — ablations not yet run

Hyperparameters selected from preliminary experiments rather than systematic search:

- Hidden dimension (64).
- Number of message-passing layers (3) — governs the spatial reach of multi-hop reasoning.
- Aggregator (mean) — alternatives: max, attention, LSTM aggregator.
- Convolution family: SAGEConv vs. GCN, GAT, GIN, GGNN.
- Context window (60 ticks).

The GNN does not provide calibrated uncertainty estimates; this is a prerequisite for operational deployment and remains open.

Backup — robustness to real-world conditions

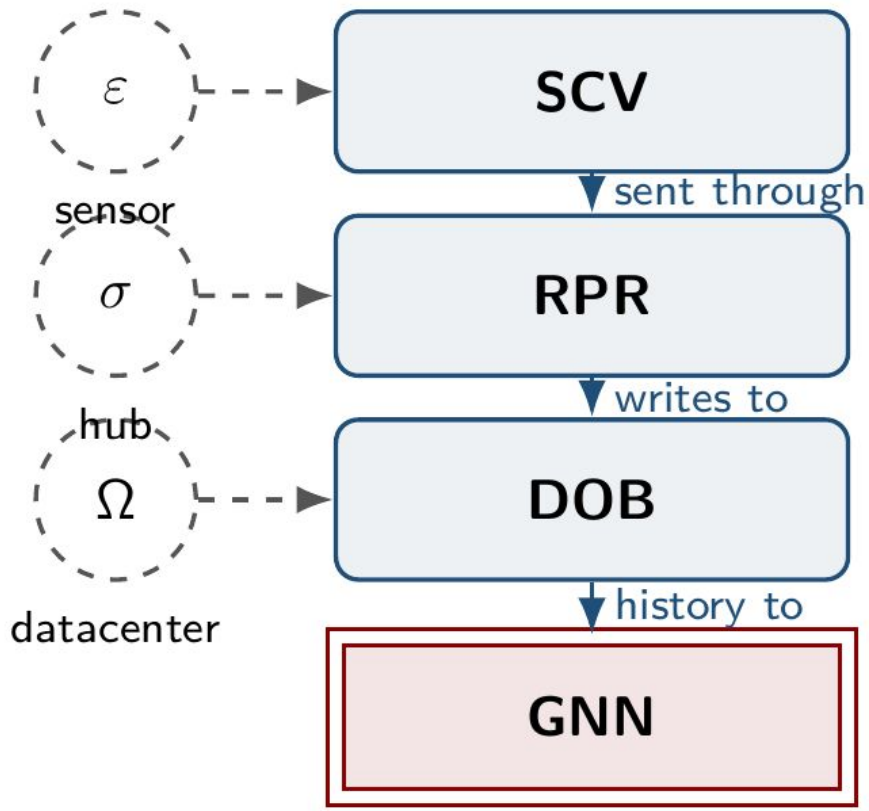
Idealised assumption	Effect under relaxation
Perfect positioning, $\delta_{\text{pos}} = 0$	Noisy positions $\Rightarrow$ noisy features, mis-formed mesh. GNN can learn to discount unreliable sensors; KF requires explicit covariance retuning.
Reliable communication	Packet loss / partitions $\Rightarrow$ dropped edges. GNN handles varying graph structure via message passing.
Noise-free detection within 200 m	Acoustic noise, multipath, biology $\Rightarrow$ false detections. GNN can downweight noisy neighbourhoods.
Fixed sensor positions	Drift, irregular spacing $\Rightarrow$ KF requires coordinate transforms; GNN re-tessellates locally.

These relaxations remain to be evaluated empirically.

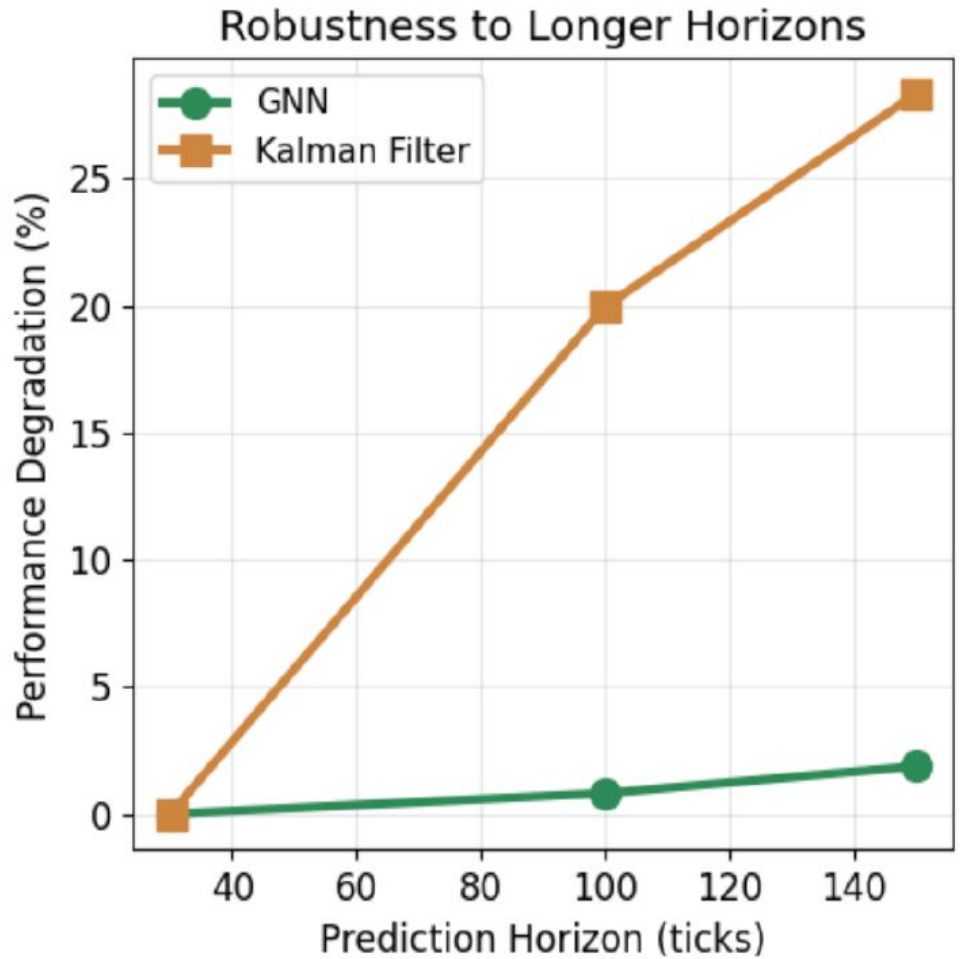
Backup — future work

1. **Empirical grounding.** Integrate observed marine tracking data; calibrate motion models against measured trajectories.
2. **Stronger baselines.**
   - Classical: EKF, UKF, IMM, particle filter.
   - Sequence models: LSTM, temporal CNN, transformer.
   - Alternative GNNs: GCN, GAT, GIN, GGNN.
3. **Ablations.** Depth, hidden width, aggregator, edge types, feature subsets, context window.
4. **Uncertainty quantification.** Calibrated confidence intervals on GNN predictions.
5. **Byzantine extension.** MadSim-based simulation harness with adversarial nodes (false detections, suppression, position spoofing). Connects to the author's ongoing BFT line.

Paper Fig. 2 — Protocol abstraction hierarchy



Paper Fig. 7 — Performance degradation (paper figure)



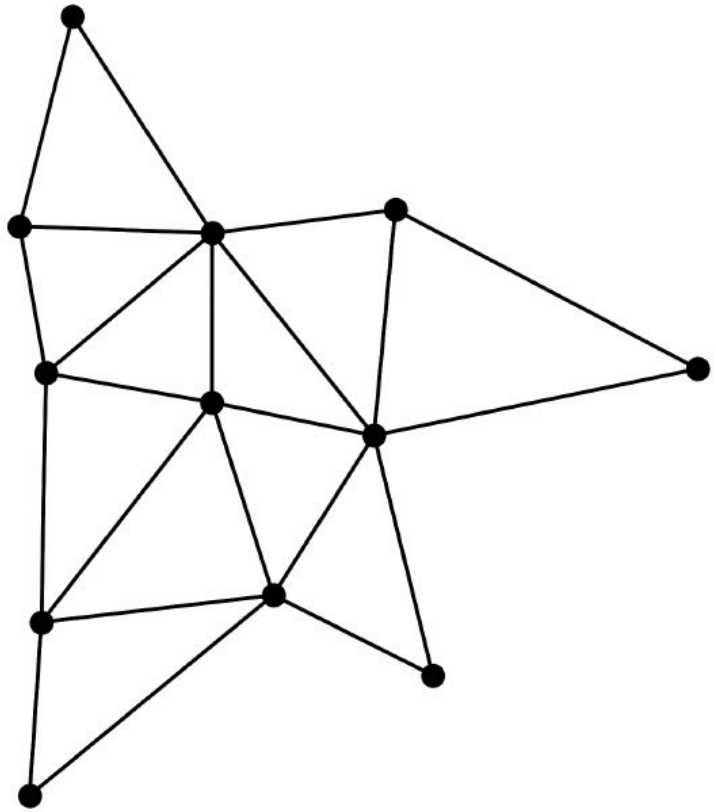
Using 30-tick horizon as baseline (100%). GNN: −1.9% at 150 t. Kalman: −28.3% at 150 t.

Paper Table 2 — Sensor density configurations

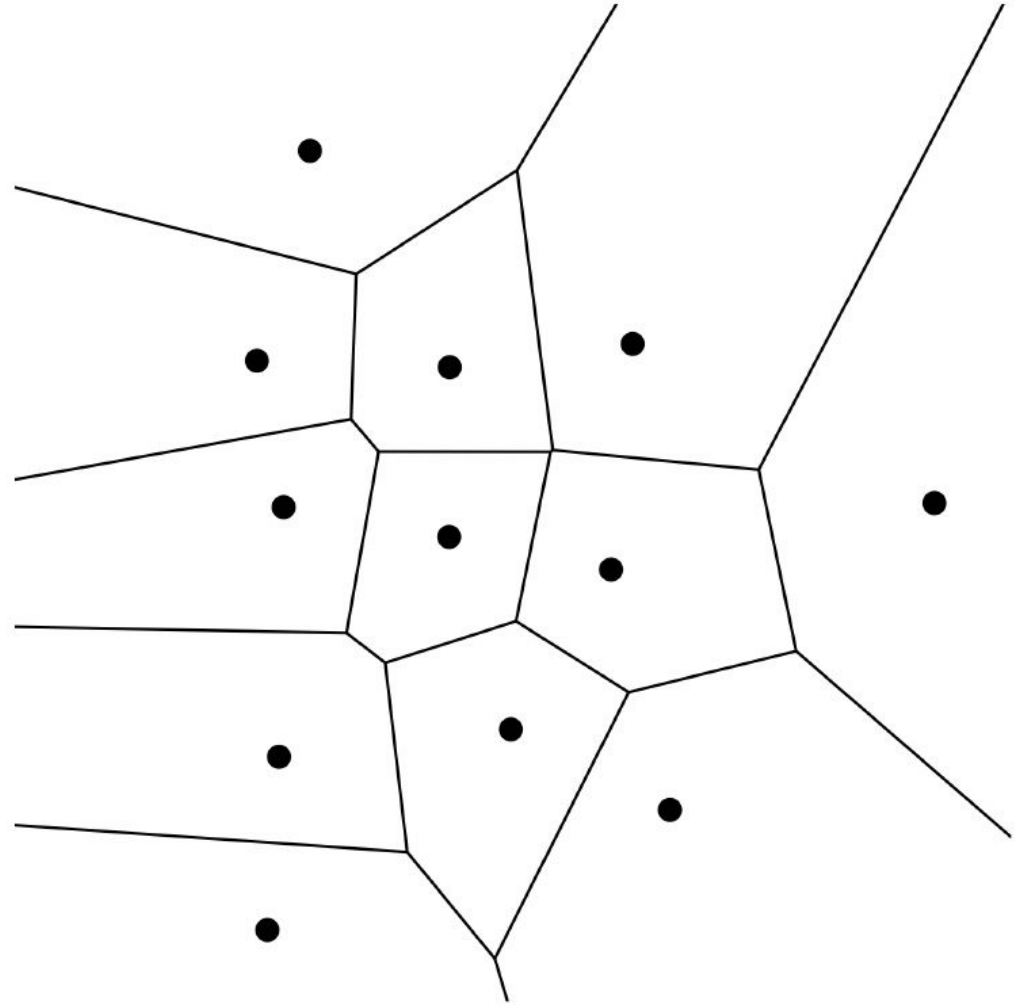
Grid resolution	Horizontal spacing	Vertical spacing	Total sensors
5 km	5.0 km	4.33 km	24
1 km	**1.0 km**	**0.87 km**	**570**
0.5 km	0.5 km	0.43 km	2,220
0.1 km	0.1 km	0.09 km	55,500

Hexagonal layout: vertical spacing $= \frac{\sqrt{3}}{2} \times$ horizontal. Hex layout is 13.4% more efficient than rectangular grid. **1 km** (bold row) used in this paper: $\sim$12.6% direct coverage, 87.4% gap.

Communication topology — Delaunay–Voronoi mesh



Delaunay triangulation at deployment.



Voronoi partition for local updates.