RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

SHOM
L'océan en référence

NAIADE

Designing ocean observing networks with machine learning

Jean-Marie Vient
Shom, Brest, France
jean-marie.vient@shom.fr

SynObs and Ocean Observing Co-Design
Mutsu, Japan, August 2026

github.com/Jvient/NAIADE

Designing a network is a choice under constraint

Rising logistics costs, uncertain funding, an explicit carbon target. What to keep, what to move and what to add is a design problem.

What does one sensor actually bring?

no routine way to measure the contribution of a single station

Which sensors are redundant?

and which gaps are left unmeasured

How many, where, at what cost?

under a fixed budget and a carbon ceiling

Three questions, three separate methods, three answers that cannot be compared. NAIADE is a testbed built to answer them on common ground.

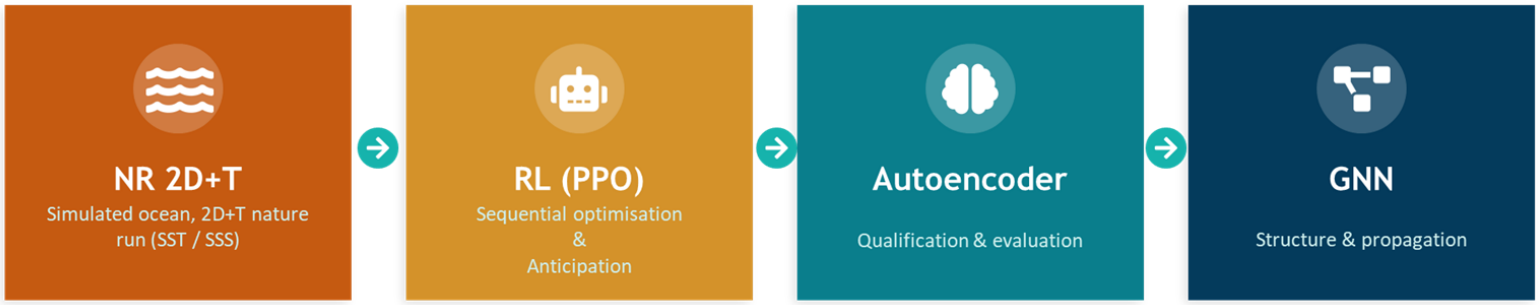
NAIADE in one slide

Network AI for Adaptive Design of Experiments

One synthetic ocean, used as ground truth by all three

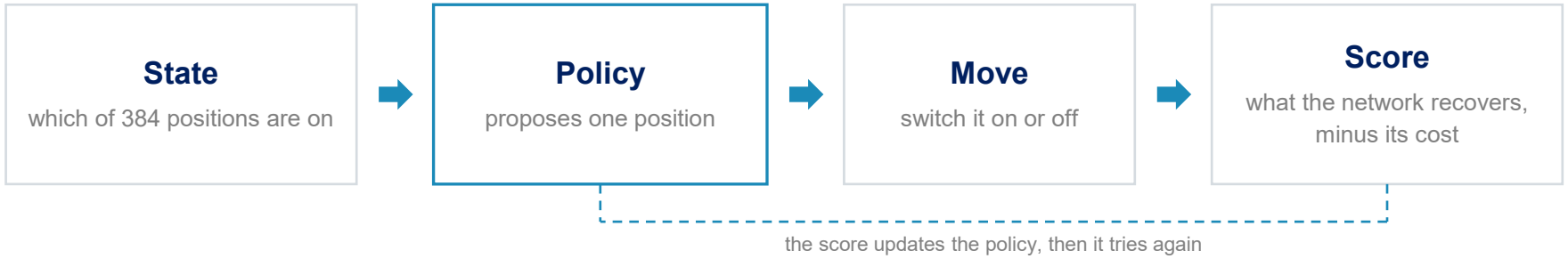
Reinforcement learning	AutoEncoder	Graph network
how many, where, at what cost	where the network is blind	which sensors are redundant

Each brick runs on its own, with its own entry point and its own report. Chained, the agent proposes a network, the graph diagnoses it, the autoencoder audits what is left unobserved. Same ocean and same scoring rule throughout, so the three outputs can be read against each other.



How the reinforcement learning works

Trial and error against a score



The score: how much of the field an optimal linear estimate recovers from those sensors alone. The criterion objective analysis has used for fifty years, not one invented for the occasion.

Minimum spacing is not encouraged through the reward, it is enforced: illegal positions are removed before the agent chooses.

How the autoencoder works

Learn to fill in the blanks

skip connections keep the fine detail

What we see
the field at a few points, blank elsewhere

→

Encoder
compress

→

Decoder
expand

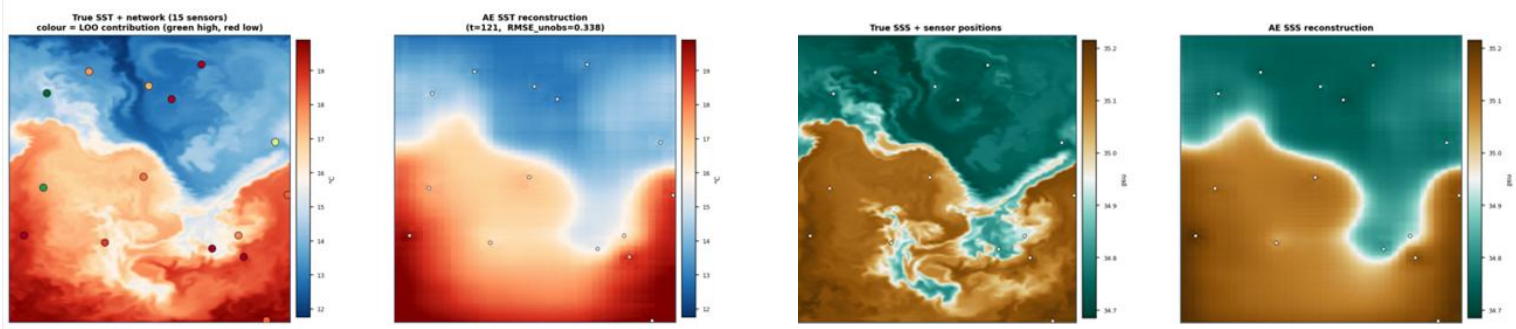
→

What we get
the field everywhere, filled in

Training: we take a field we know completely, hide all but a handful of pixels, and score the guess on the hidden pixels only, so it cannot win by copying its input.

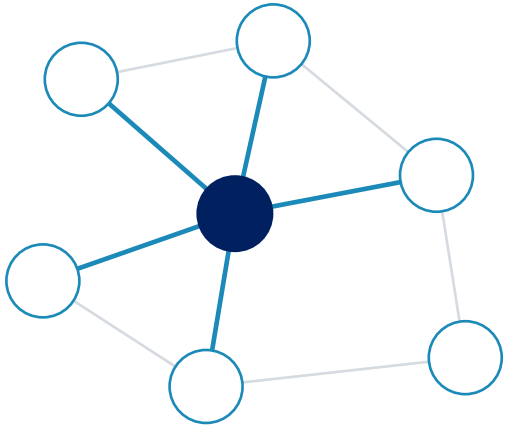
The mask changes at every step, so one trained model scores any network without retraining. It is fed in alongside the field, so the network knows where it is guessing.

Run it thirty times with random neurons switched off: the spread across runs is the uncertainty.



How the graph network works

Every sensor listens to its neighbours



a node per sensor, an edge where records move together

It listens

each node builds a summary of what its neighbours record, then updates itself with it

It weighs

attention learns which links carry information and which do not

It scores

a node whose neighbours already tell you what it tells you is redundant

Why it can score a sensor that does not exist yet: what it learns is a rule about neighbourhoods, not a value attached to a named sensor. Drop a new node anywhere, wire it to its neighbours, and the same rule scores it.

A synthetic ocean as controlled ground truth

python data/dataset.py --nt 1500 --seed 42

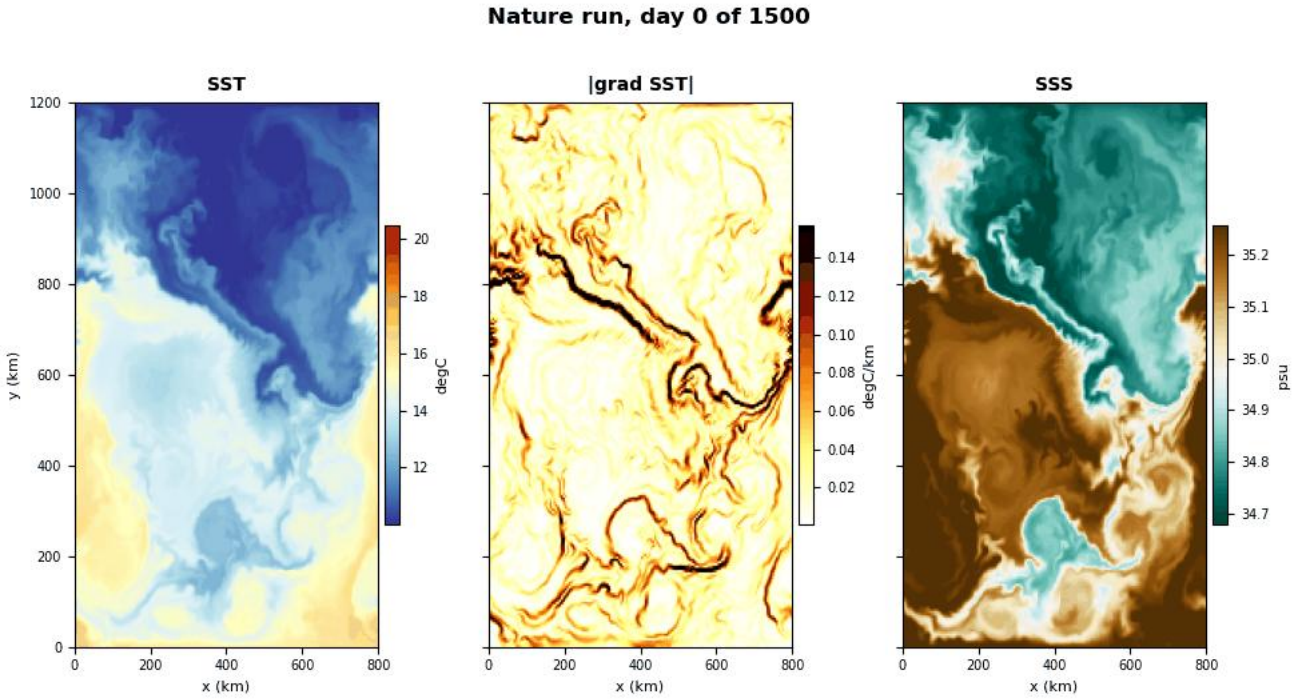
95 km
spatial decorrelation

12 days
decorrelation time

2.60 °C
SST variability

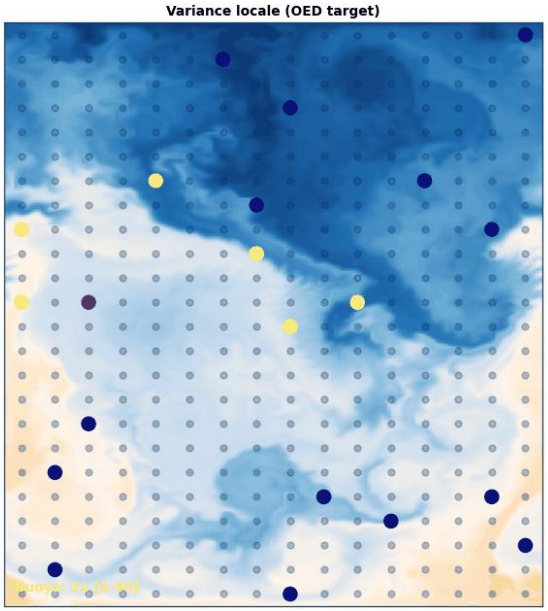
0.177 psu
SSS variability

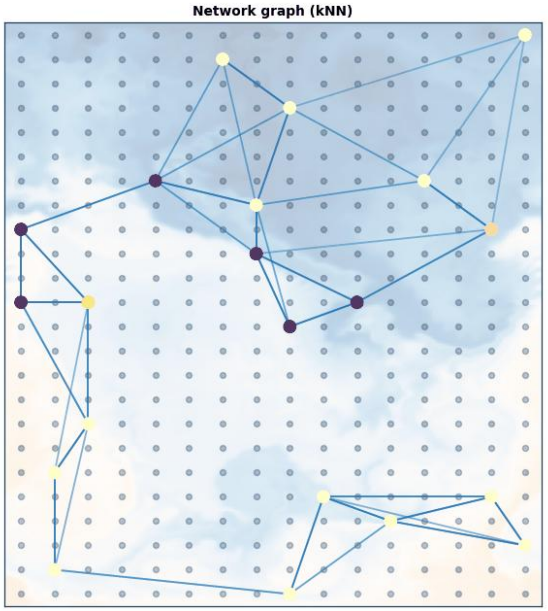
Fronts, filaments and eddies are not prescribed. They emerge from stirring by the flow competing with restoring towards climatology. A controlled ground truth: the scales are known, so an error can be measured without ambiguity.

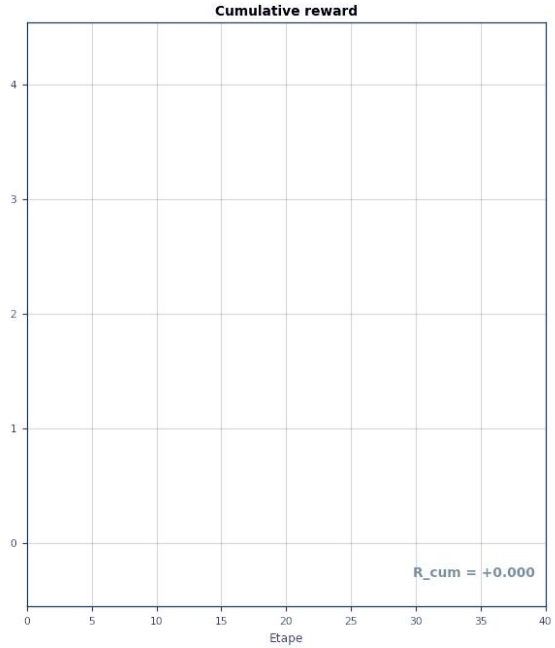


Brick 1: placement as a sequential decision

PPO on a candidate grid, scored by a classical criterion



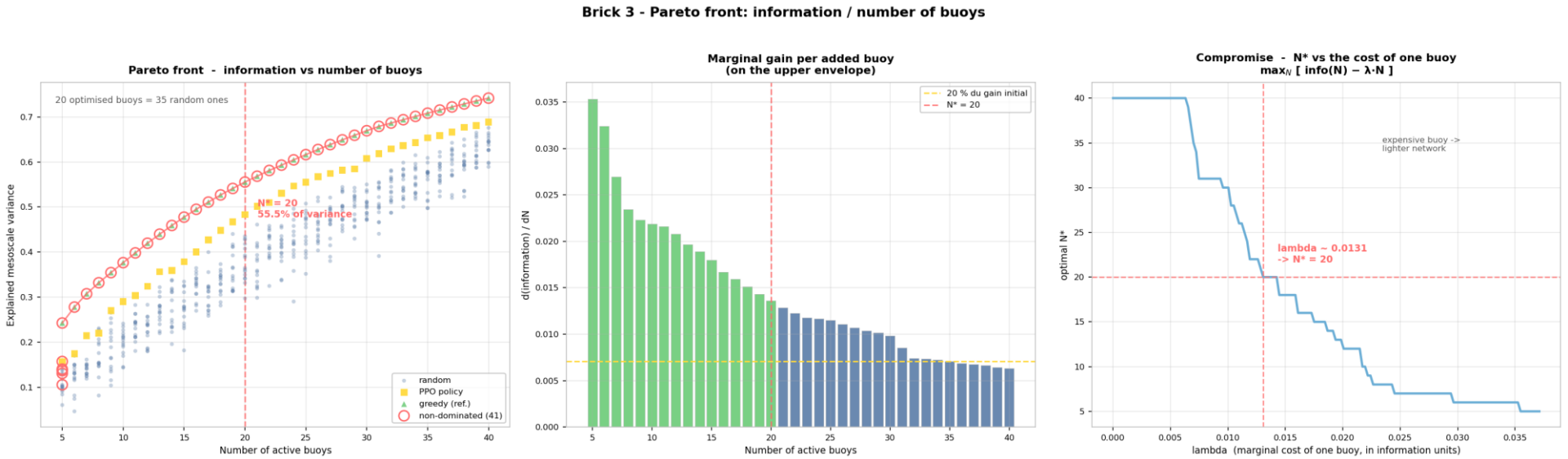




the optimised configuration over the local variance field

Overfitting is real: the raw sample covariance gives a negative out of sample score. NAIADE shrinks towards a parametric model, exactly as operational interpolation does.

Brick 1: information against number of buoys



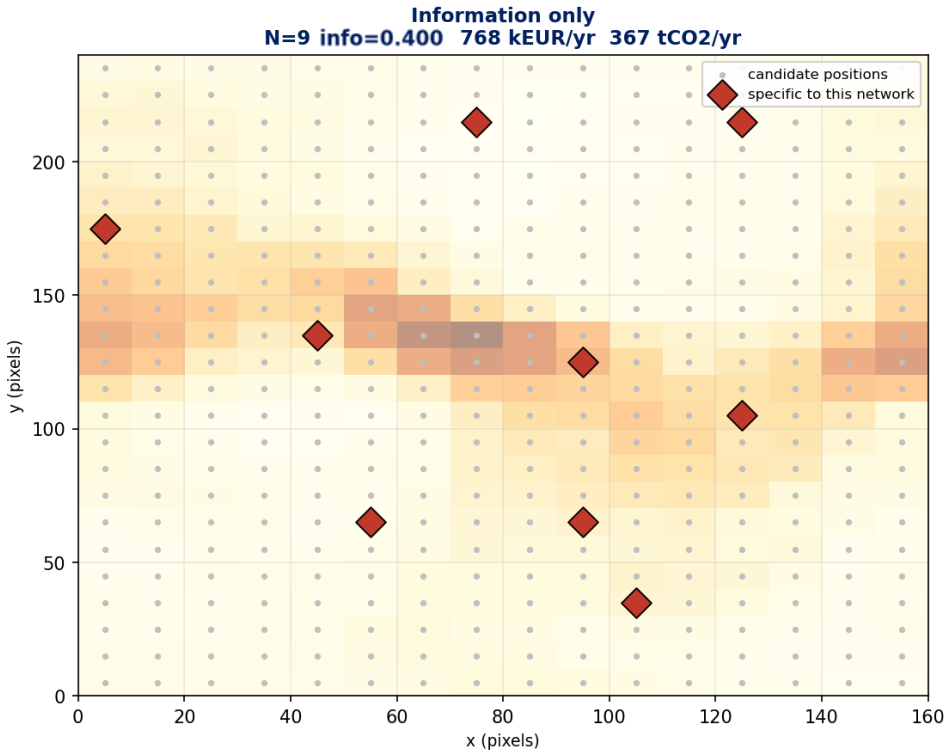
every configuration evaluated, the marginal gain per added buoy, and the best size against the cost of one buoy

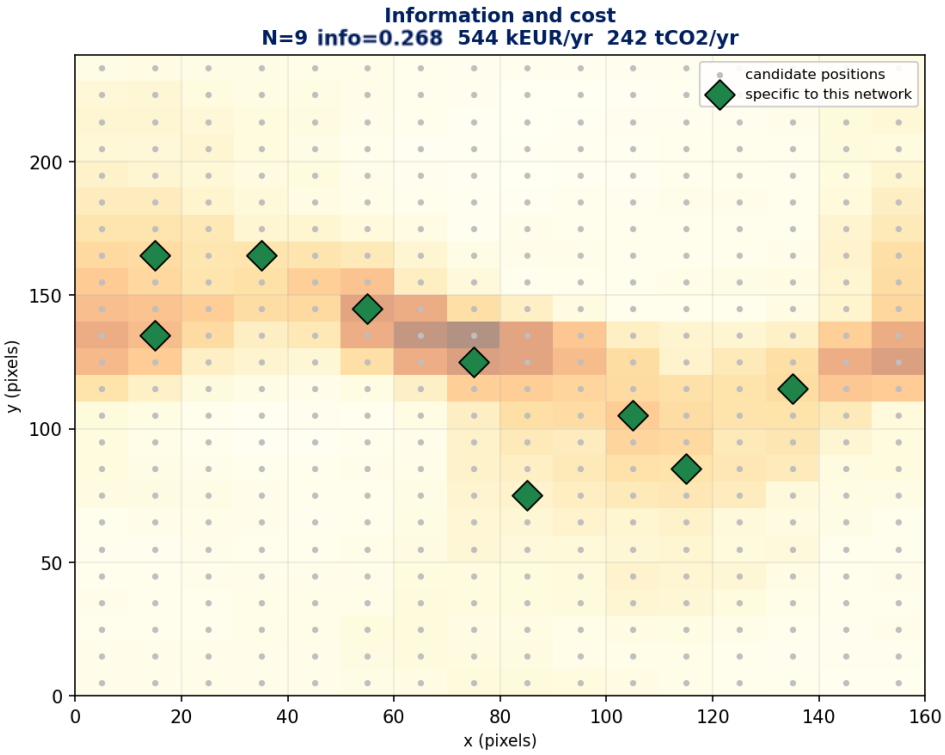
20 buoys explain **55,5 %** of the mesoscale variance

and are worth 35 placed at random, on this synthetic ocean

Brick 1 bonus: information against cost and carbon

Same size, different answer: adding cost and carbon moves 9 of the 9 buoys





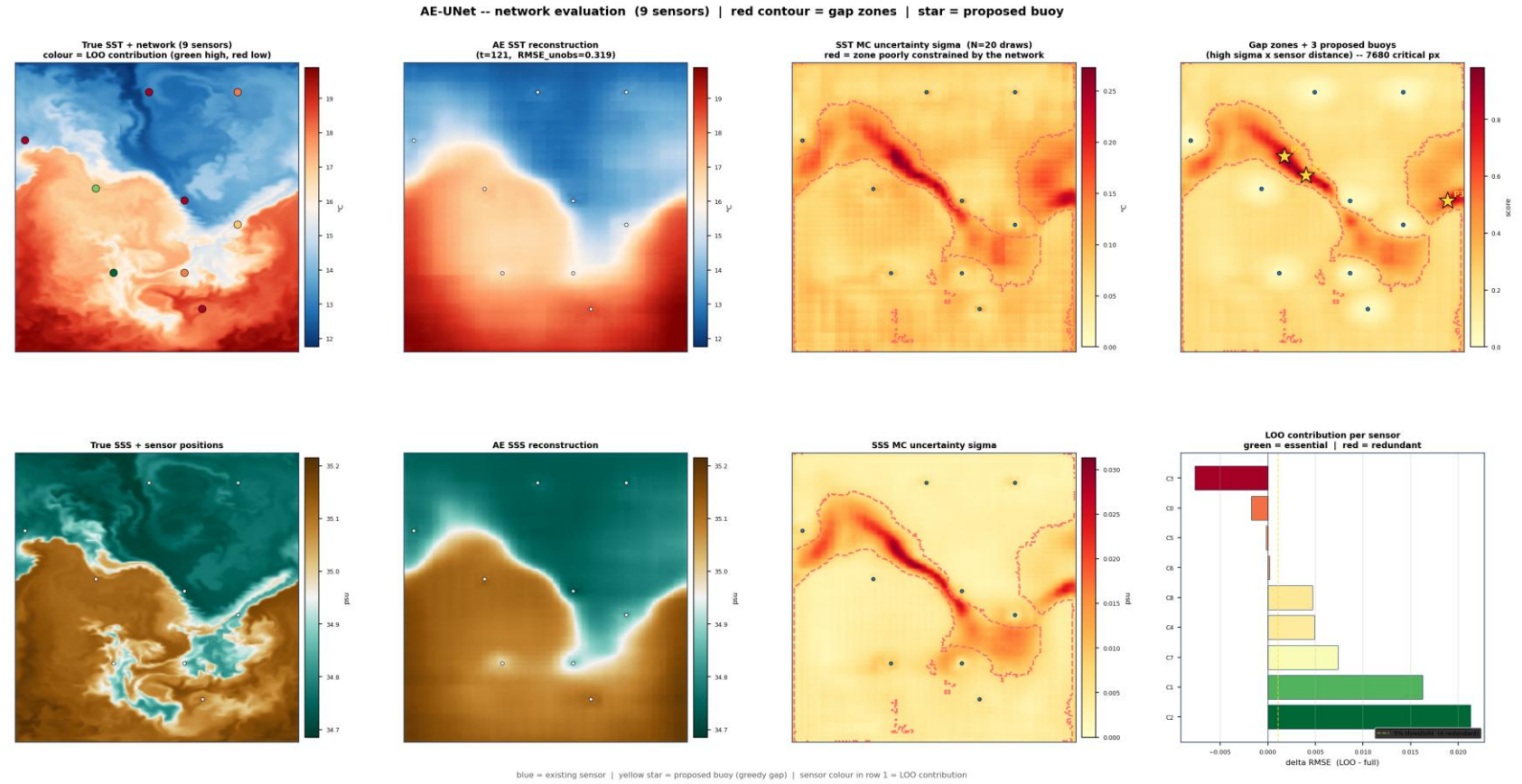
operating cost -29%, CO2 -34%, information +49% | maintenance tour 3667 km against 2424 km

cost is not proportional to the number of buoys. At a fixed number it varies by a factor 1.3 to 1.6 with how spread out the network is, because of the maintenance tour. That is what makes the two objectives genuinely antagonistic.

Cost and carbon figures are indicative orders of magnitude, isolated in one file so they can be calibrated on a real network.

Brick 2: reconstruction and uncertainty

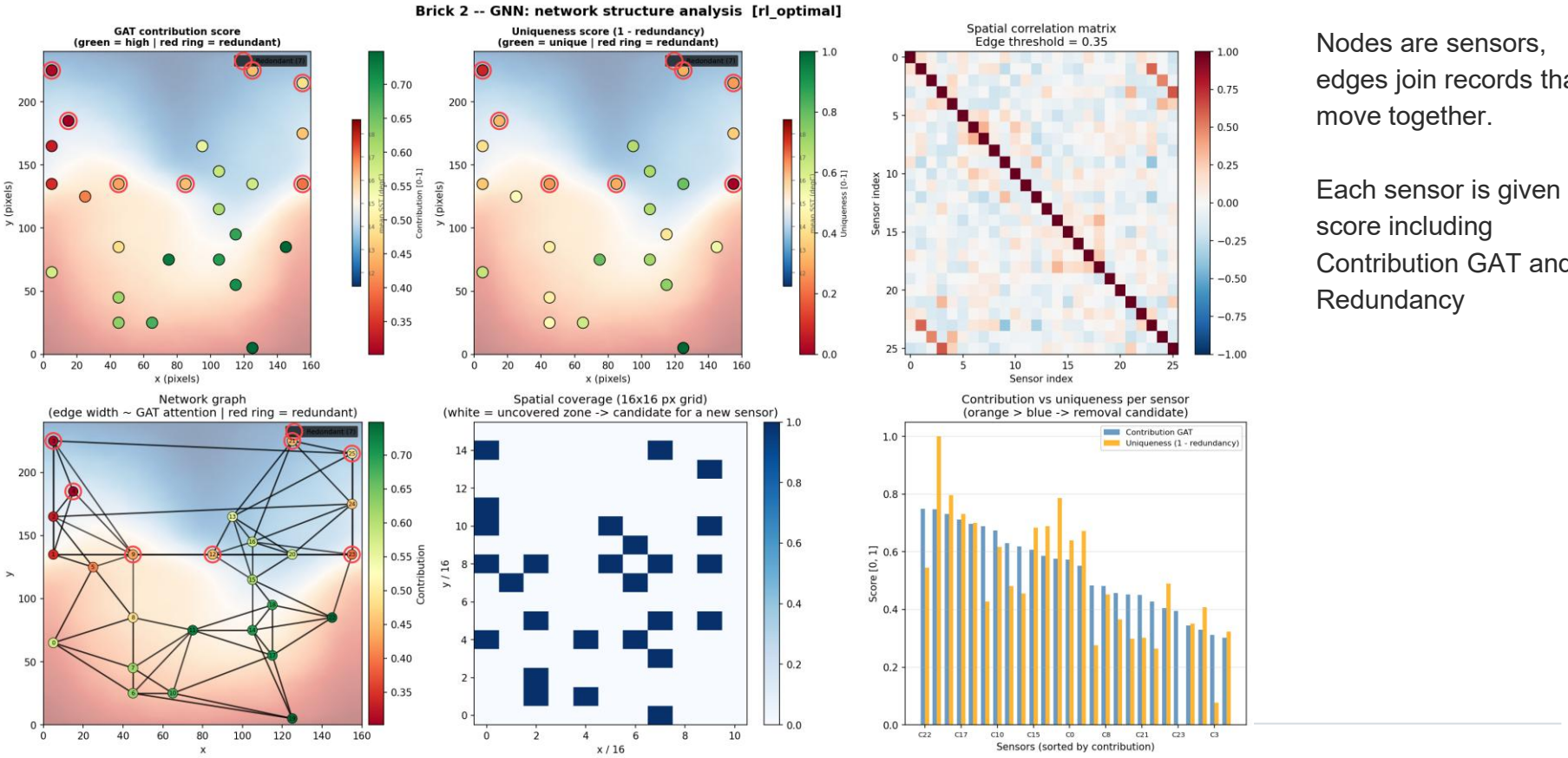
U-Net conditioned on the observation mask, with MC-Dropout



The gap map: high uncertainty combined with distance to the nearest sensor, so it returns candidate positions rather than a picture to interpret.

Brick 3: the network as a graph

Attention for redundancy



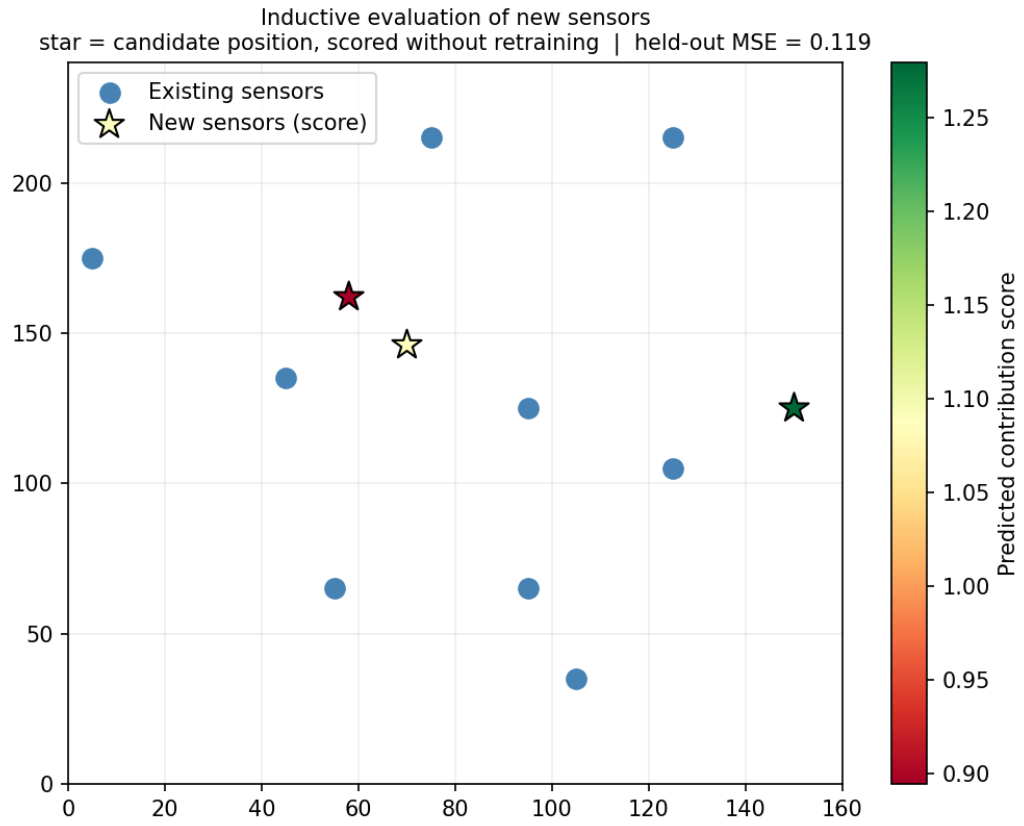
Nodes are sensors, edges join records that move together.

Each sensor is given a score including Contribution GAT and Redundancy

on the reference run, 8 of the 30 sensors come out redundant.

Brick 3: scoring a sensor that does not exist yet

Inductive evaluation. no retraining



The model generalises to positions it has never seen, so it answers what a new mooring or a new line would bring before anything is deployed.

What it learns is the correlation structure, not the geography, which is what makes reuse on another network plausible.

the stars are the positions the autoencoder proposed: one brick feeding the next, scored without retraining.

Modular by construction

Three bricks, each usable alone

- each has its own entry point, its own figures and its own report
- the nature run is the only input; masks, graphs and RL environments follow automatically
- a model output can replace the synthetic ocean through the same interface

Use case	Method	Output
Value of an existing sensor	AE + Shapley	Contribution score per sensor
Network qualification	GNN + AE	Coverage and gap map
Anticipating network evolution	Multi-objective RL	Optimal network under constraints
Value of adding a glider or float	AE + inductive GNN	Marginal information gain against cost
Reducing the carbon footprint	RL + Pareto and specific cost	Trade-off between information and number of buoys

One shared protocol

- same ocean, same reference network, same scoring rule
- two seeds fix the ocean and the starting array
- every run writes a timestamped report with its parameters
- cost and carbon live in one file, made to be recalibrated

What the framework is, and what it is not yet

What it is

- a modular, reproducible testbed for network design
- three approaches on one ocean and one protocol, so their answers compare
- a criterion taken from classical objective analysis
- open source, runs on a laptop

What it is not yet

- a single idealised domain, other regimes need recalibrating
- surface only, no vertical structure and no biogeochemistry
- no assimilation in the loop, so no forecast impact
- **reproducible on the ocean and the network, not yet on the training**

What the framework is, and what it is not yet

What it is

- a modular, reproducible testbed for network design
- three approaches on one ocean and one protocol, so their answers compare
- a criterion taken from classical objective analysis
- open source, runs on a laptop

What it is not yet

- a single idealised domain, other regimes need recalibrating
- surface only, no vertical structure and no biogeochemistry
- no assimilation in the loop, so no forecast impact
- **reproducible on the ocean and the network, not yet on the training**

Next

- a realistic model output in place of the synthetic ocean
- the vertical, and mixed platforms
- **assimilation in the loop, with forecast error as the reward**
- **cost and carbon calibrated on real logistics, with regional constraints**

To conclude the NAIADE proof of concept

A testbed, not a result

three approaches on one ocean and one protocol, so their answers can be compared

Modular by design

each brick runs alone or chained, and the synthetic ocean can be swapped for a model output through the same interface

Cost and carbon belong in the objective

not in a filter applied afterwards

Statistical care matters more than the method

de-seasonalising, shrinking the covariance and validating out of sample moved the answers more than the choice of network

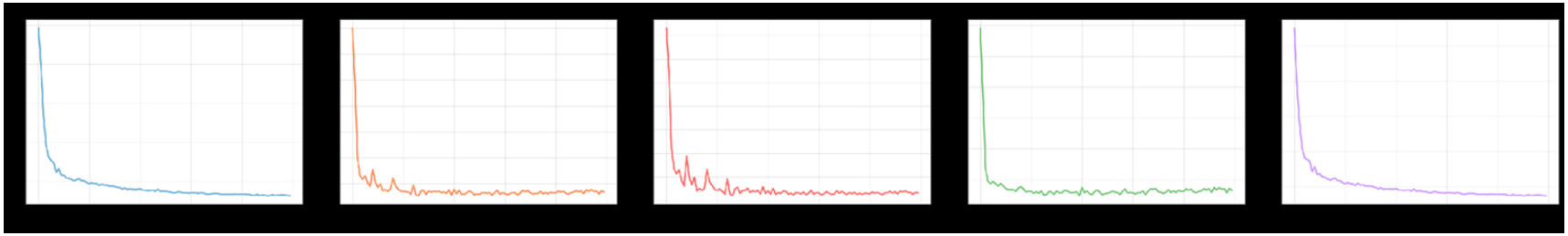
Thank you

jean-marie.vient@shom.fr

github.com/Jvient/NAIADE

Backup

Backup: autoencoder training



loss, validation error on unobserved pixels, then error in physical units

The two channels are normalised differently, so a single aggregate error converts to neither. They are reported separately for that reason.

Backup: PPO training



384 candidate positions, feasible maximum 96 buoys under the separation constraint.

Backup: reproducing the figures

One command

```
python run_demo.py --mode pipeline \
       --seed_ocean 42 --seed_buoys 7 --nt 1500 \
       --rl_grid_x 16 --rl_grid_y 24 \
       --rl_n_max 40 --rl_steps 50000 \
       --ae_epochs 200 --ae_base_ch 32 \
       --gnn_epochs 500 --ocean_gif
```

Each brick also runs on its own.

Two seeds control everything

One fixes the ocean, the other the starting array. Same seeds, same ocean, same network.

Every run writes a timestamped report next to its figures.

Backup: selected references

Bretherton, Davis and Fandry (1976) Objective analysis and design of oceanographic experiments applied to MODE-73. Deep-Sea Research.

Krause, Singh and Guestrin (2008) Near optimal sensor placements in Gaussian processes. JMLR 9.

Gasparin et al. (2019) Requirements for an integrated in situ Atlantic observing system. Frontiers in Marine Science 6, 83.

Velickovic et al. (2018) Graph attention networks. ICLR.

Hamilton, Ying and Leskovec (2017) Inductive representation learning on large graphs. NeurIPS 30.

Schulman et al. (2017) Proximal policy optimization algorithms. arXiv 1707.06347.

Gal and Ghahramani (2016) Dropout as a Bayesian approximation. ICML.

Perez et al. (2018) FiLM: a general conditioning layer. AAAI.

Racault et al. (2023) Carbon footprint benchmarking for ocean observation. Frontiers in Marine Science 10.