COCE 2025 Keynote Speech:

Stochastic Spatiotemporal Multi-Agent Path Planning with Knowledge Graphs

Dr. Ted Holmberg | *October 2025*

Presenter:

Edward (Ted) Holmberg

Postdoctoral Research Associate

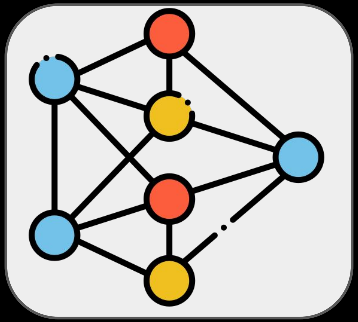
Gulf States Center for Environmental Informatics

@ University of New Orleans

Research interest:

Machine Learning

Graph Theory



Spatiotemporal Analysis

Data Visualization

Overview -- (Talk Breakdown)

1. **Big Picture** → Motivation & Problem Domain
    - Cycle: TED -- PREP -- TED -- WAITR

2. **ROBUST Networks (Overview)**
    - Math terms and Analysis

3. **PREP (Mapper):** per-window spatial pruning → NavGraph
    - Heatmap Abstraction
    - Proximal Recurrent

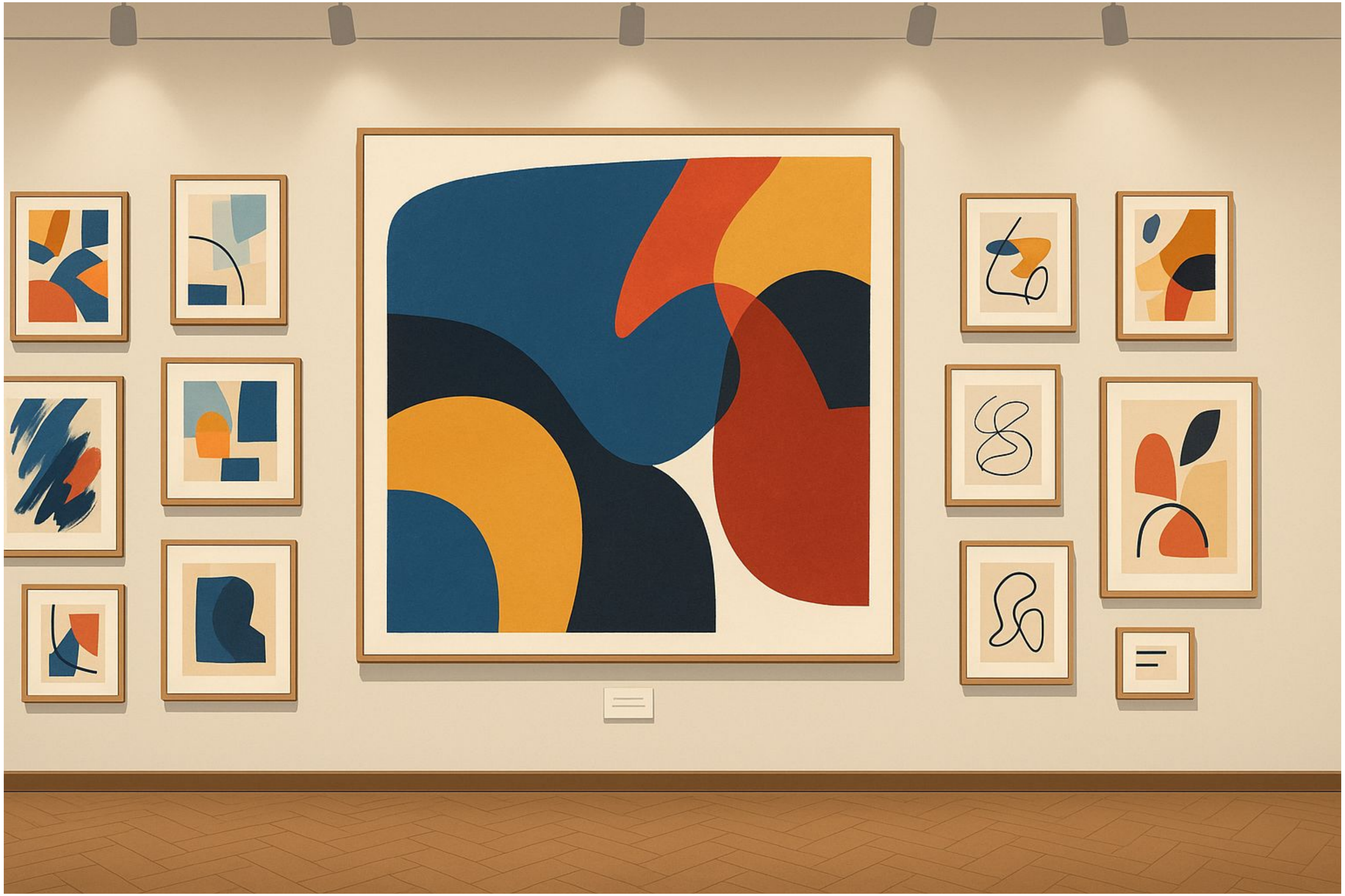
4. **WAITR (Planner):** spatiotemporal optimization → stitched long-horizon route (MPC repair)
    - Pathlets

5. **TED (Mission Compiler):** policy-to-tensor compilation & stochastic updates

    - **Contract (Interfaces):** The Mission Tensor Producer (TED) ↔ Consumers (PREP, WAITR)

6. **Q&A**

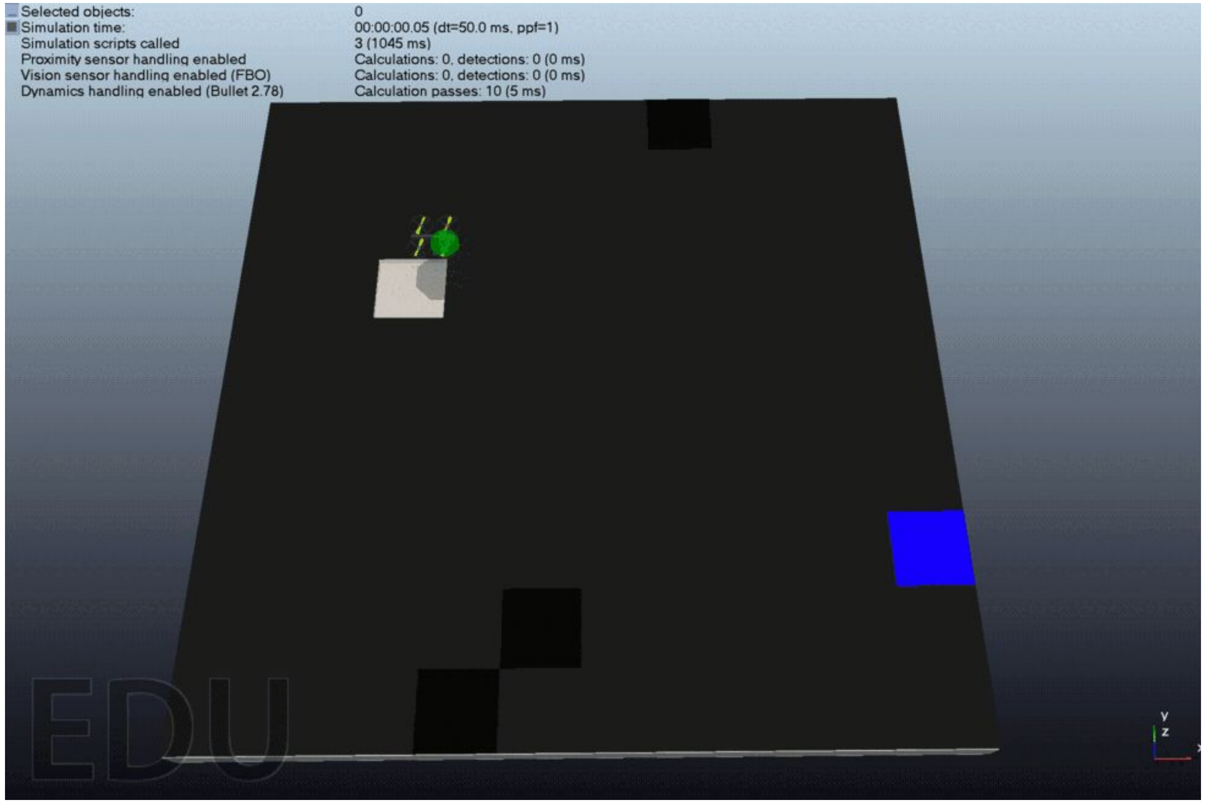
1 -- The Big Picture



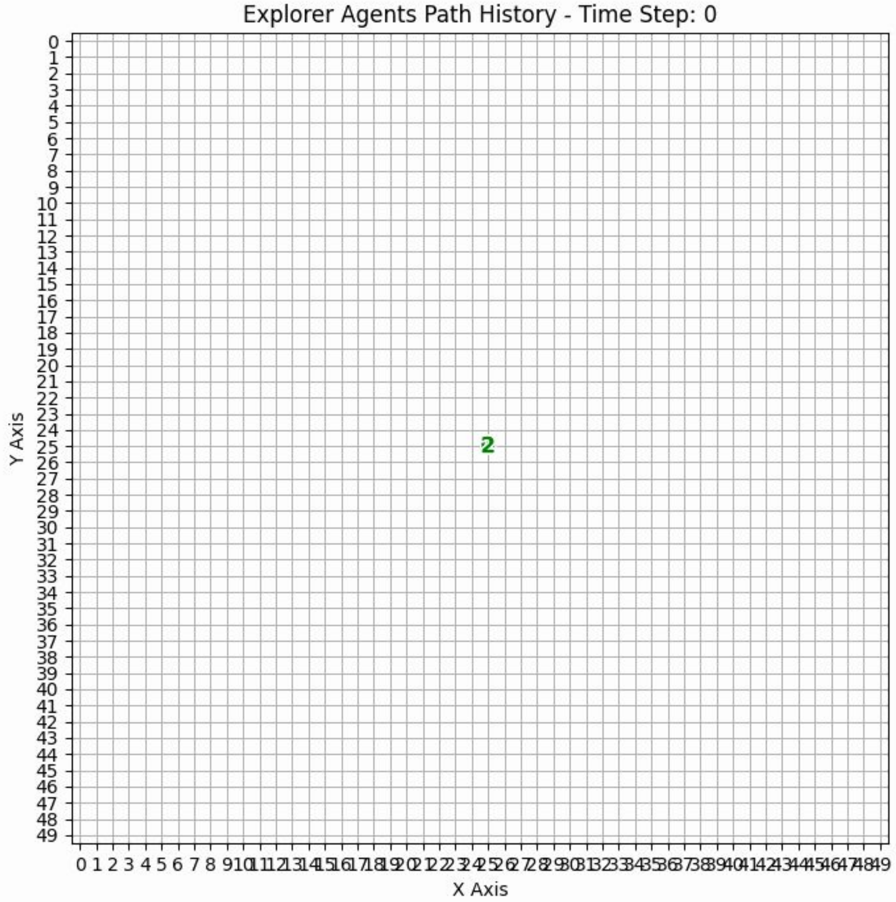
The Problem

Across many domains, teams of agents must coordinate & act while the world keeps changing.

- **Dynamics:** goals shift; risks/closures appear; conditions evolve.
- **Knowledge decays:** observations age; confidence drops over time.
- **Constraints:** access rules; safety policies; limited time/energy/budget.
- **Coordination:** multiple robotic/human agents must avoid conflict and share
- **Consequence:** static plans go stale; full replans are costly and brittle.



Where? - Imperfect Spatial knowledge



When? - Imperfect Temporal knowledge

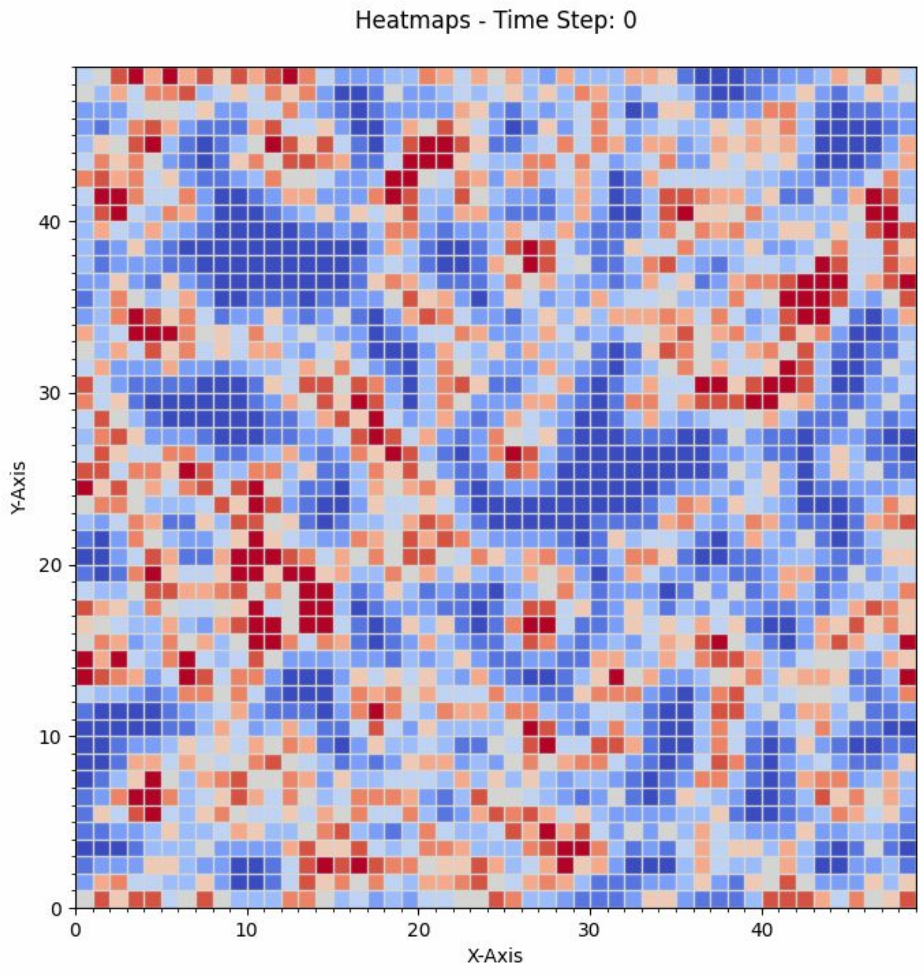
Introduction: Background

Spatiotemporal Data Challenges and ROBUST Networks

The Challenge:
Maximize Observational Coverage in Spatiotemporal Environments

The Need:
A framework to 'choreograph' data acquisition of dynamic temporal events.

This Research:
Focus on observer positioning that supports both scalability, and adaptability.



Introduction: Research Problem

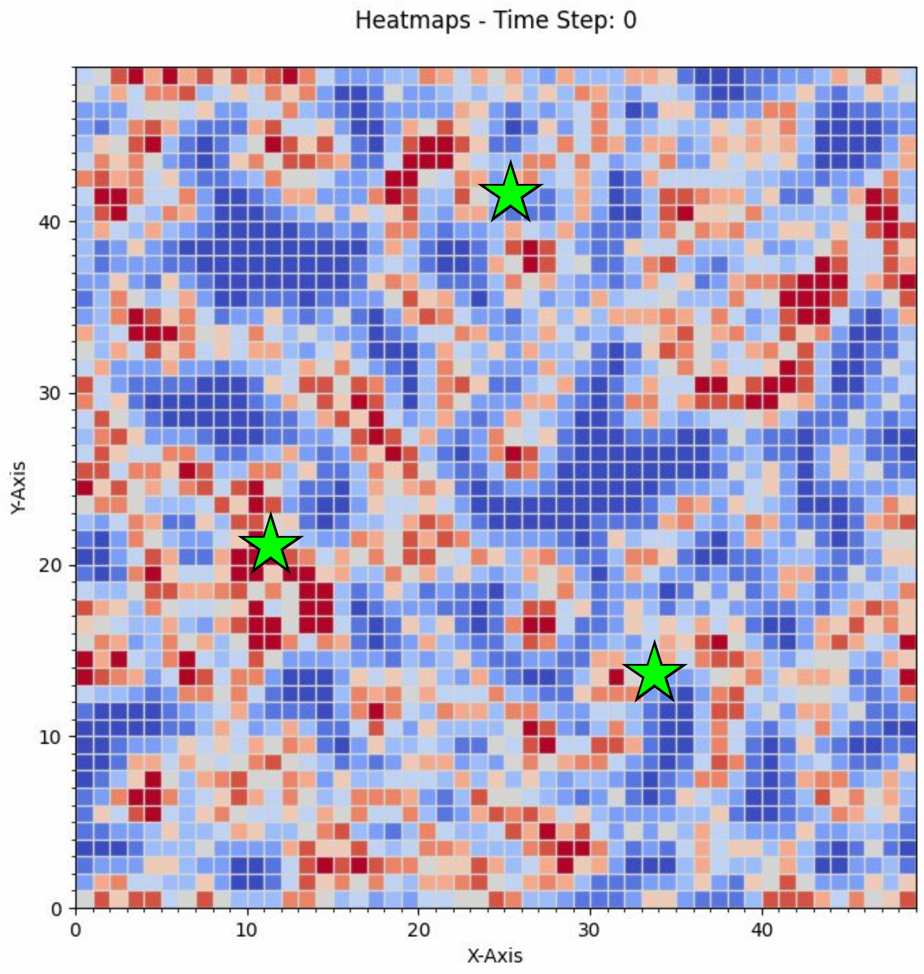
Main Problem Statement:

Bipartite Networks:
Networks comprising two distinct classes of nodes, with links only between nodes of different classes.

Spatiotemporal Dimensions:
These networks evolve over time and space, adding complexity to their structure and dynamics.

What Makes This Challenging?
- **Variability:** The dynamic nature of the networks, with nodes and links changing over time, stochastically.
- **Complexity:** Introducing both spatial & temporal dimensions means traditional methods may not be directly applicable.
- **Optimality:** Determining what "optimal" means in the context of these networks, given the numerous metrics & considerations.

Why It Matters:
Optimizing these networks can lead to more efficient resource use, faster response times, and better outcomes in applied scenarios.



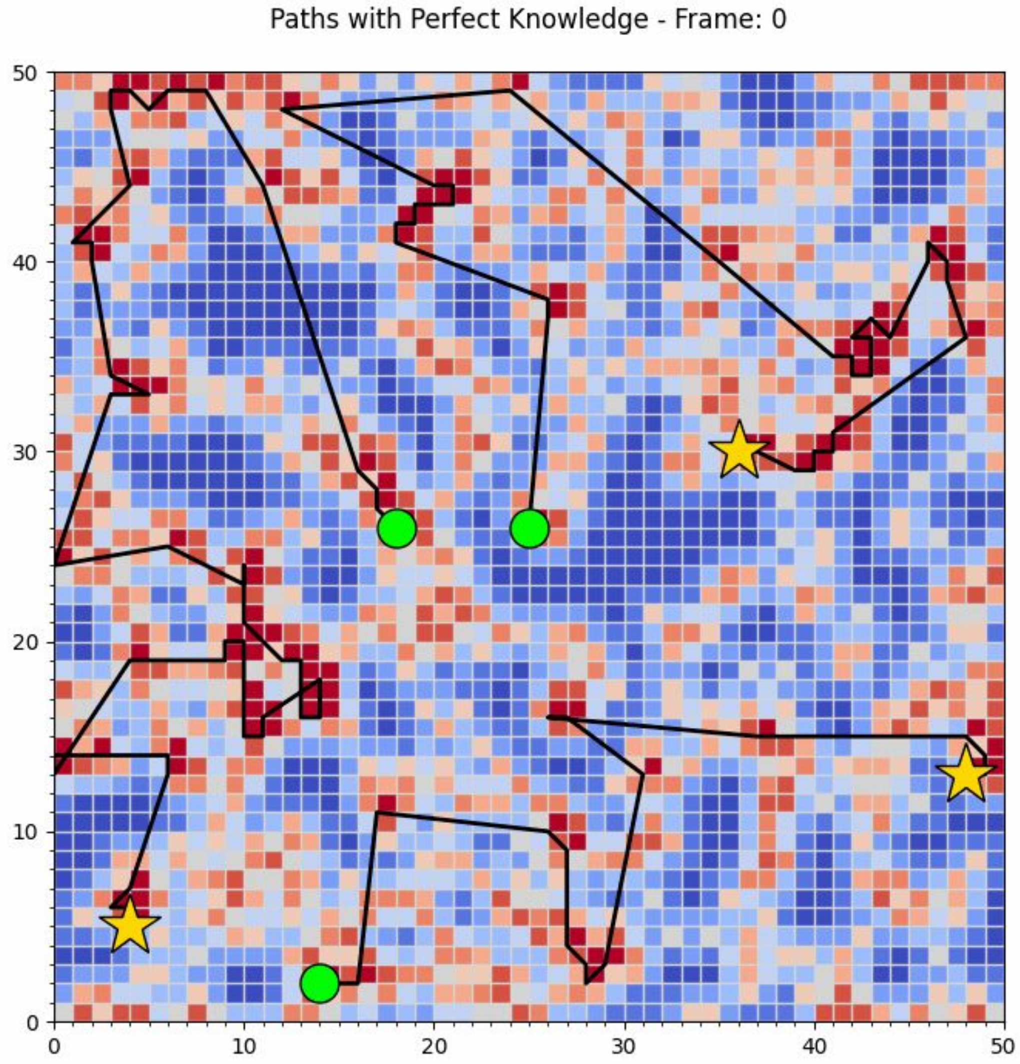
The Goal (what success looks like)

Produce feasible, high-value, explainable routes/assignments for one or more agents over a time horizon, continuously adapting as conditions and policies change.

- **Decision:**
  routes/assignments per agent across windows.

- **Objective:**
  maximize mission value subject to cost/feasibility
  - (time, energy, risk, access)

- **Outputs:**
  long-horizon routes
  + rationale (why-trace)
  + fast, incremental updates (low churn).



Introduction: Research Hypotheses

H1: Analysis & Insight of Observational Capabilities:

- ROBUST networks proposes a novel set of spatiotemporal graph analysis tools, with coverage, robustness measures, and centrality distribution analysis.

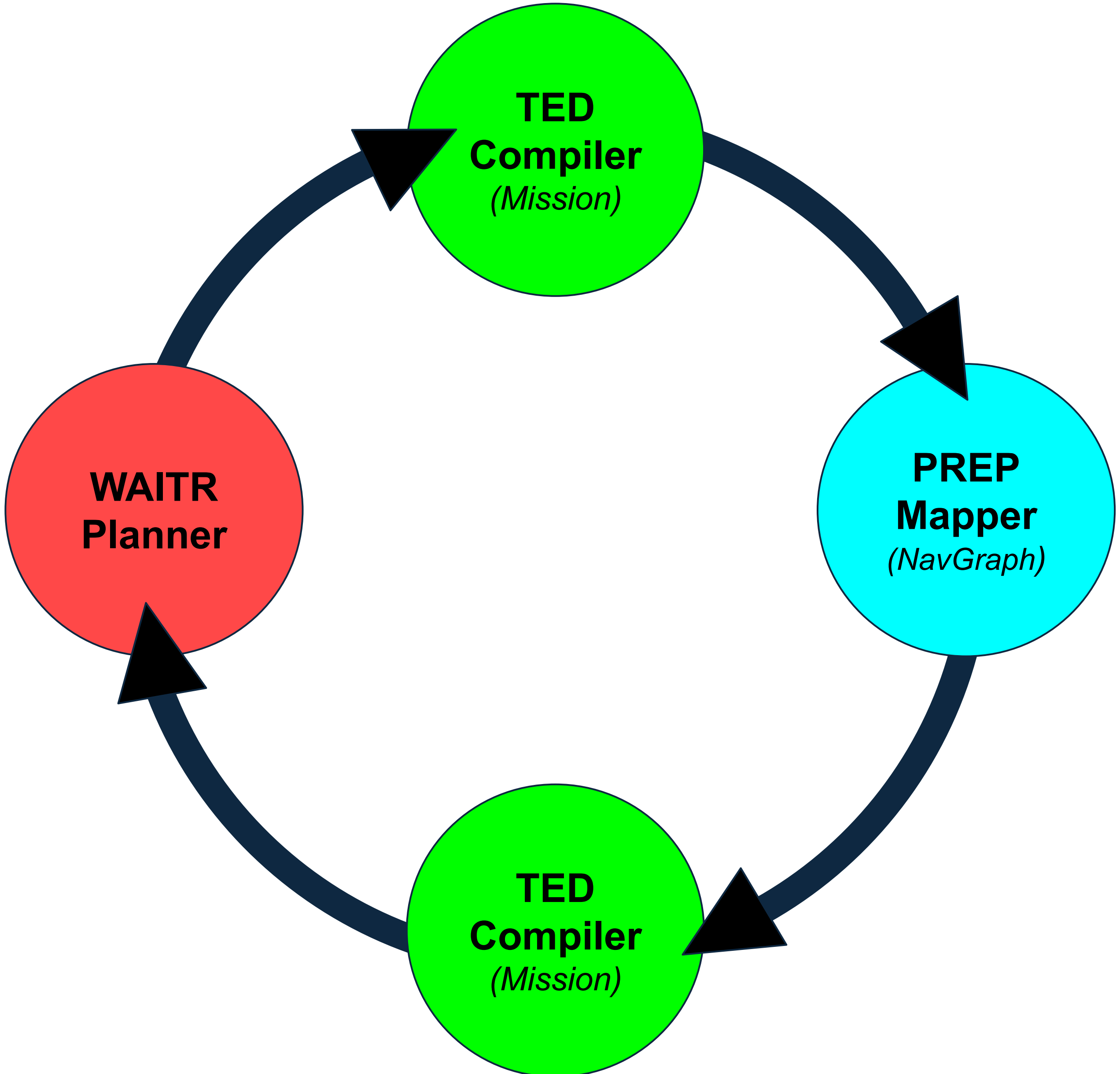
H2: Optimized Observer Node Placement:

- ROBUST networks will achieve an optimal balance between minimizing node insertions and maximizing event capture, demonstrating superior resource allocation that will outperform conventional models (k-means, dbscan, LP, etc.).

H3: Scalable & Rapid Execution in Support of Real-Time Decision Making:

- ROBUST networks are hypothesized to integrate the efficiency of vectorized computations and GPU-accelerated techniques, achieving the rapid processing speeds, while simultaneously delivering high accuracy as compared to much slower linear programming approaches.

The System at a Glance (closed loop)



2 — ROBUST Networks — *Motivation*

The Need for Spatiotemporal Bipartite Network Models

The Missing Link:

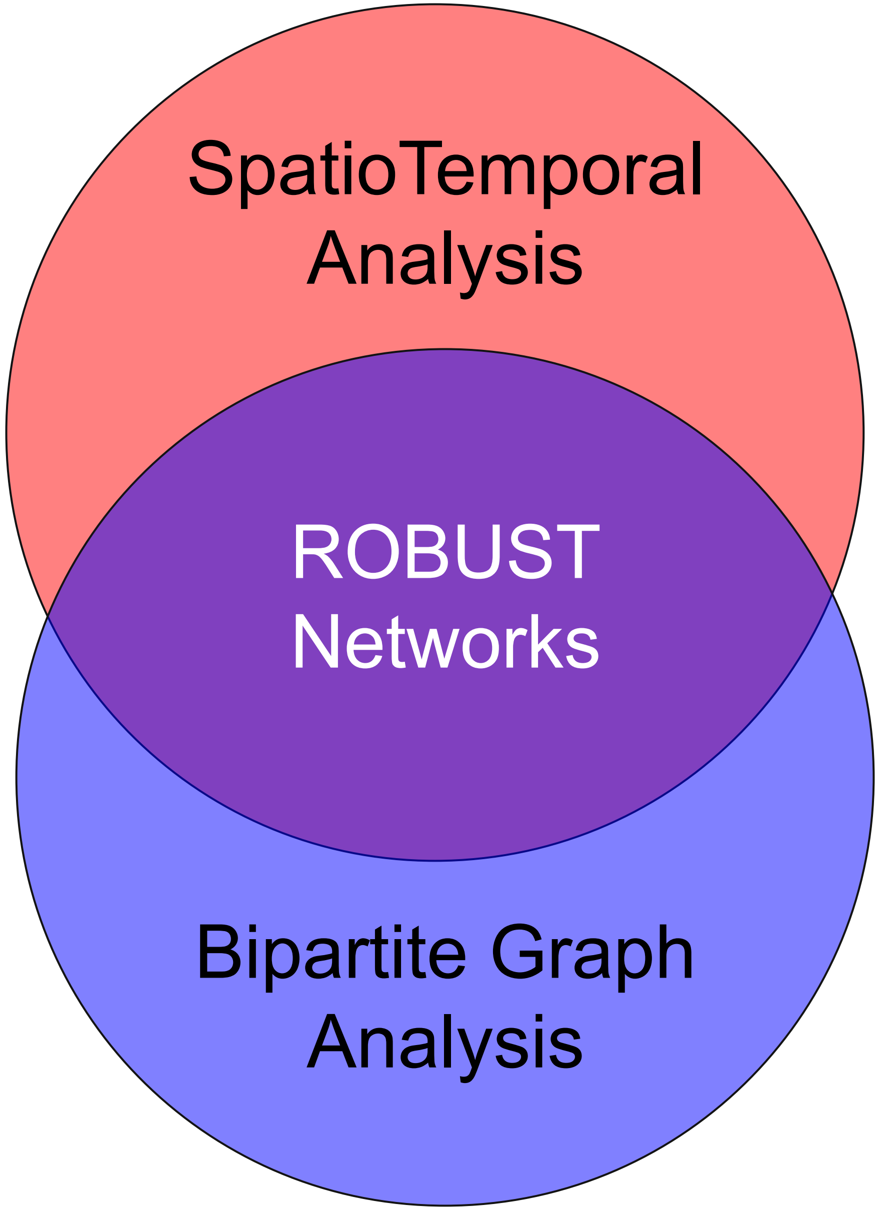
We lack models that integrate both spatiotemporal interactions AND the distinct relationships within bipartite networks.

Consequences:

- Missed insights into complex systems
- Inability to optimize for specific goals

The Opportunity:

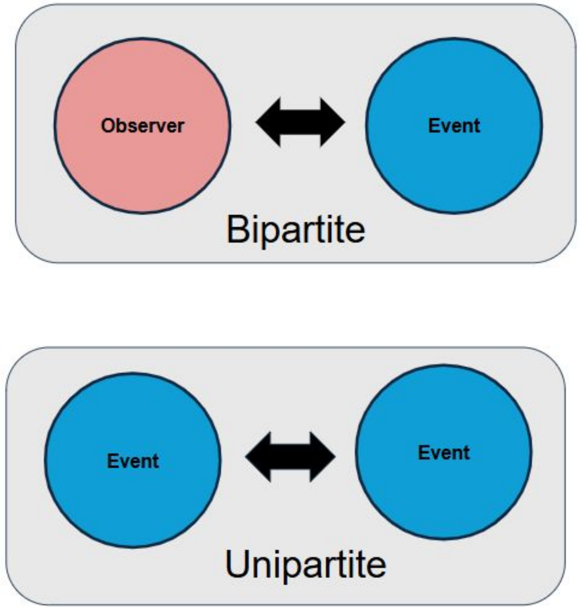
By extending network theory to merge these concepts, we can unlock greater accuracy and strategic control.

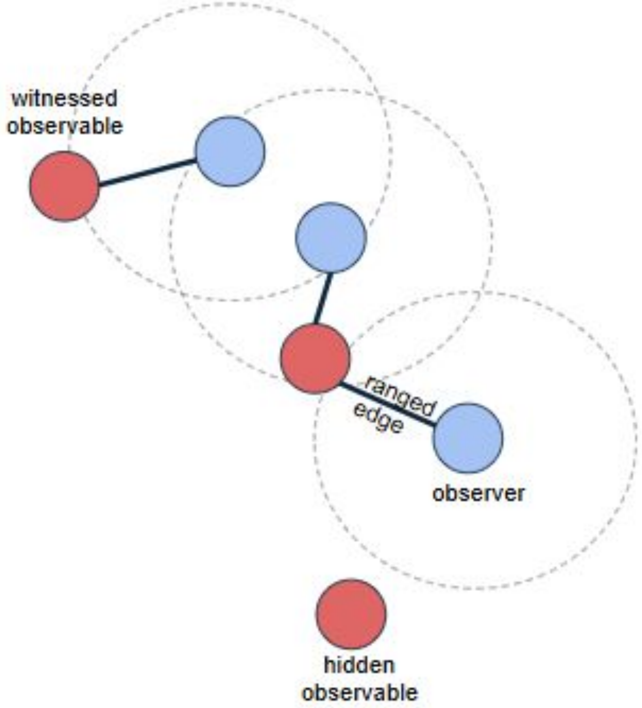


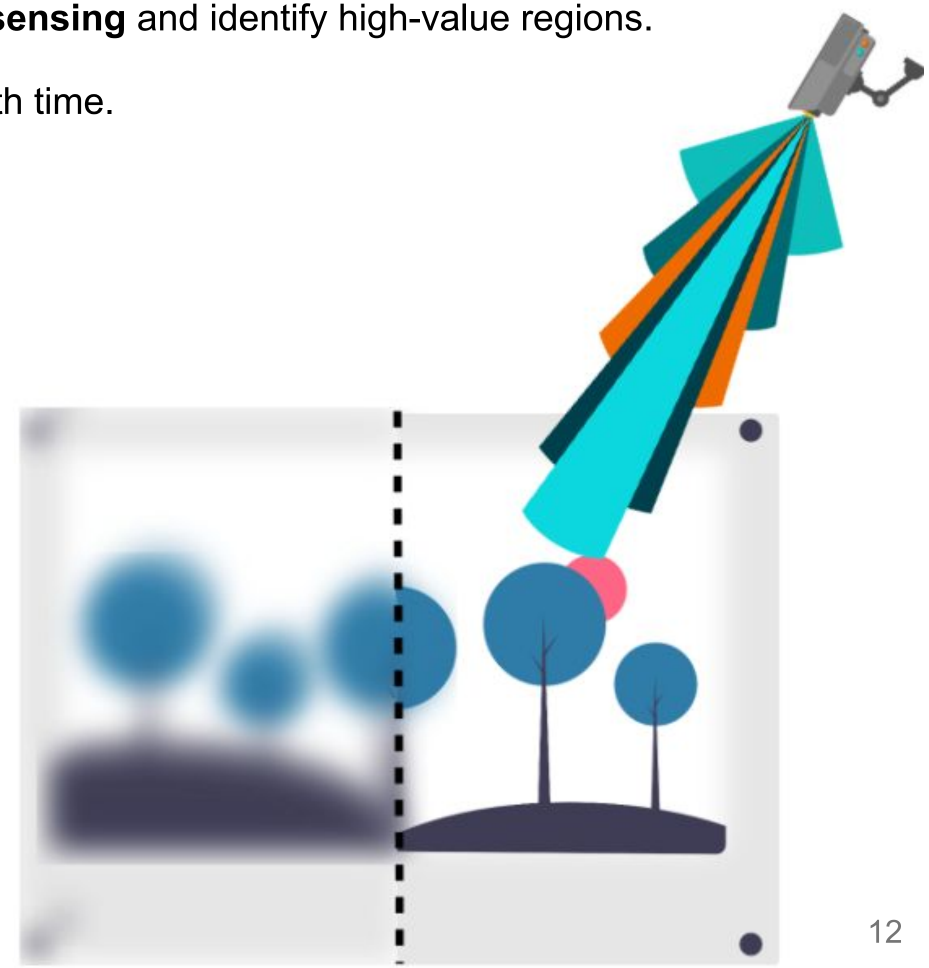
2 — ROBUST Networks — *Representation & Analytics*

Definition: A spatiotemporal, **bipartite** model of **dynamic observers** ↔ **dynamic observables** with metrics for **coverage, wiring, and resilience**.

- **Ranged observers (myopia):** View limits (range/FOV/LOS) are a defining property; they determine which observables are even feasible per window.
- **Bipartite core:** Cleanly separates **who senses** from **what is sensed** to reason about coverage and opportunity.
- **Unipartite projection:** Project to **observable–observable** links to **cluster what needs sensing** and identify high-value regions.
- **Spatiotemporal windows:** Evolve structure as $\{ G_t \}$; attributes and feasibility change with time.







2 — ROBUST Networks — Mathematical Formulation

Extending Spatiotemporal Networks to ROBUST Networks

Mathematical Formulation:

$$G_{robust} = (V, E, P, T, A_V , A_E)$$

- V: Set of *all nodes*, divided into:
  - V_O *Observer Nodes*, Entities capable of observing events.
  - V_E *Observable Nodes*, Events/ phenomena that is observable.
- E: Set of *edges* between observer and observable nodes.

Sets and Spatial Positioning & Temporal Behaviors

- ***Spatial Positioning (P):*** Maps node locations at timestep.
- ***Temporal Domain (T):*** Represents the time dimension.

Observed vs. Unobserved

- V_E^{obs} Observable nodes within the myopic range.
- V_E^{unobs} Observable nodes outside the myopic range.

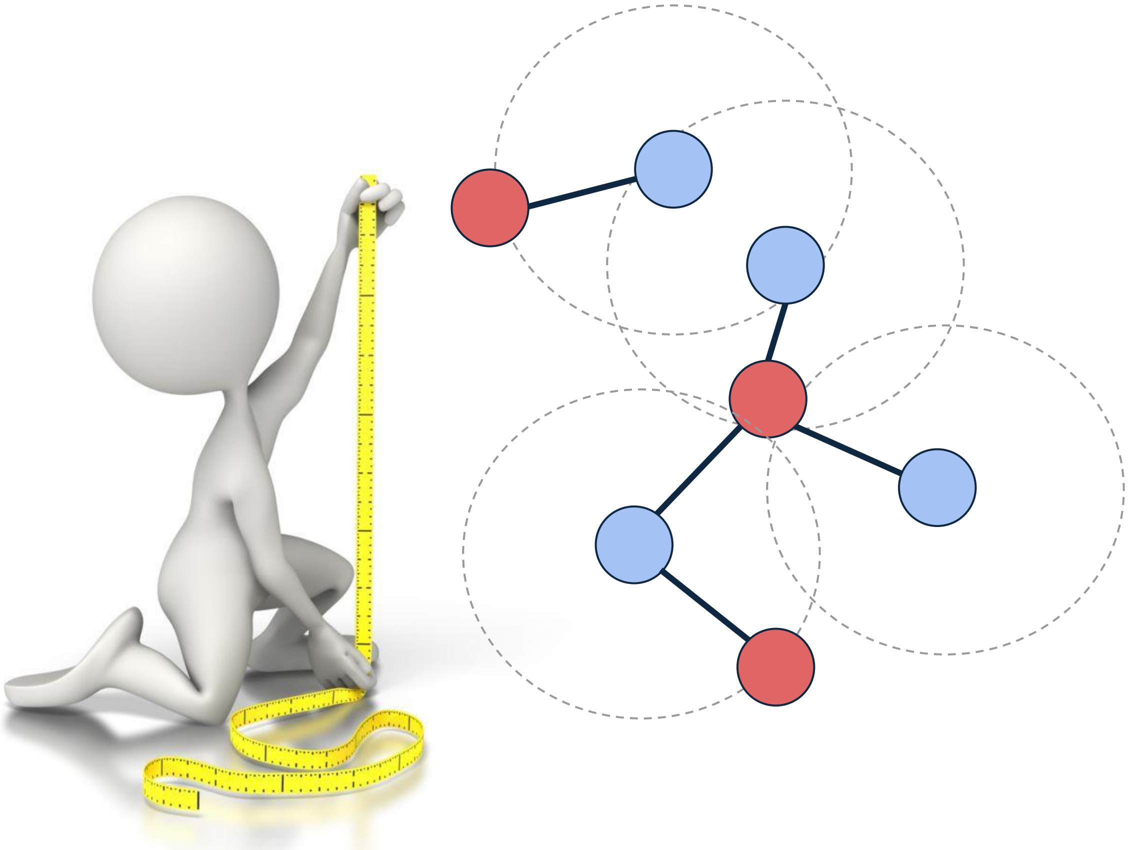
Node and Edge Attributes

- ***Node Attributes (A_V):*** Time-variant characteristics of nodes.
- ***Edge Attributes (A_E):*** Time-variant characteristics of edges.

2 — ROBUST Networks — Measures & Analysis

Novel Measures for ROBUST Network Analysis

- Six spatial metrics that turn structure into decisions (seed, add links, split spans, harden nodes).



Myopic Degree — Local Neighborhood Density

- **Definition:** Measures the connectivity of a node within a specific spatial range, focusing on the immediate neighborhood density.

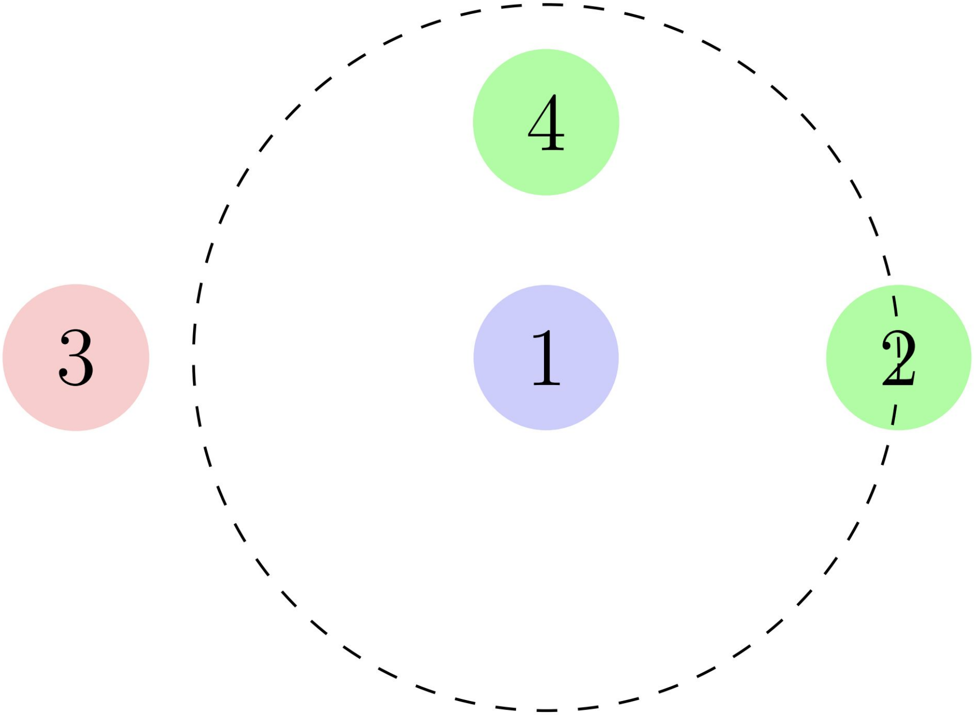
- **Purpose:** Quantifies a node's connections within a defined proximity, highlighting its interaction with nearby nodes.

Mathematical Representation

- The Myopic Degree or Spatial Degree of a node v_i is given by:

$$\boxed{\text{Spatial Degree}(v_i) = |\ \{v_j \in V : d(P(v_i), P(v_j)) \leq \boldsymbol{\theta}\ \}\ |}$$

where $\boldsymbol{\theta}$ represents a threshold distance for considering an edge to exist.



Why it matters

- Finds hotspots vs deserts at the current window.

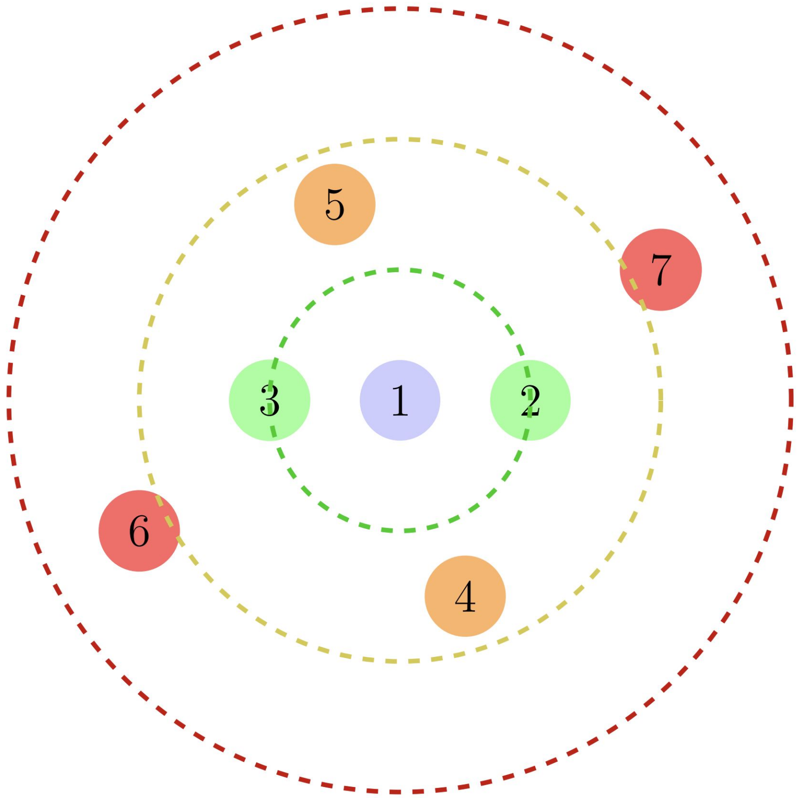
Use it to decide:

- Dense ⇒ seed medoids / anchor patrols.
- Sparse ⇒ insert/retask observers..

Spatial Closeness Centrality — Global Reach

- **Definition:** A measure assessing a node's centrality within a network, based on its spatial distance to all other nodes.

- **Purpose:** Provides a global perspective on a node's position and influence by considering its average spatial distance from the entire network.



Mathematical Formulation

- The Spatial Closeness Centrality of a node v_i is defined as:

$$C_{\text{closeness}}(v_i) = \left(\sum_{v_j \in V, v_j \neq v_i} d(P(v_i), P(v_j)) \right)^{-1}$$

where $d(P(v_i), P(v_j))$ represents the spatial distance between nodes v_i and v_j.

Why it matters

- Picks **globally near** sites that minimize average travel to everything.

Use it to decide:

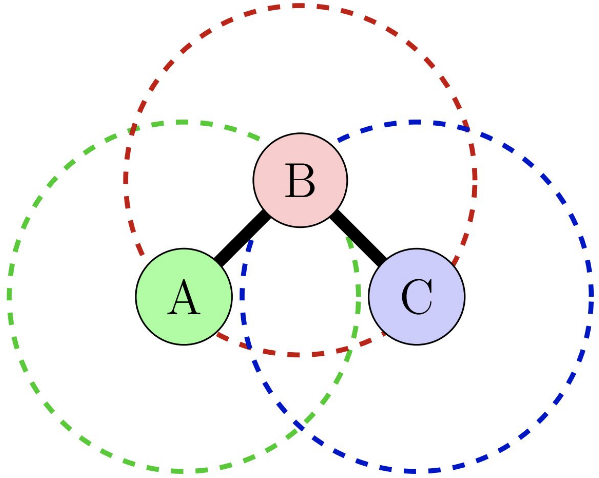
- Choose **cluster medoids/relays**; deprioritize peripheral anchors.

2 — ROBUST Networks — Edge-Based Spatial Metrics

Spatial Edge Density — Constrained Wiring

- **Definition:** Evaluates the network's closeness to maximal connectivity within spatial constraints, focusing on the overall edge concentration.

- **Key Aspect:** Takes into account the spatial arrangement and limitations such as physical distance and geographical barriers, offering a global connectivity perspective.

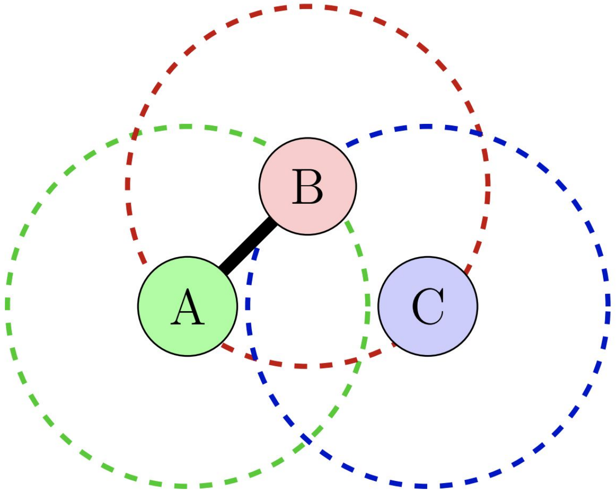


Maximally Connected Network

Mathematical Formulation

- This measure adapts traditional edge density by considering the maximum feasible edges within spatial limitations, rather than the theoretical maximum in a complete graph.

$$\text{Spatial Edge Density} = \frac{\text{Number of Actual Edges}}{\text{Maximum Feasible Edges under Spatial Constraints}}$$



Less Connected Network

Why it matters

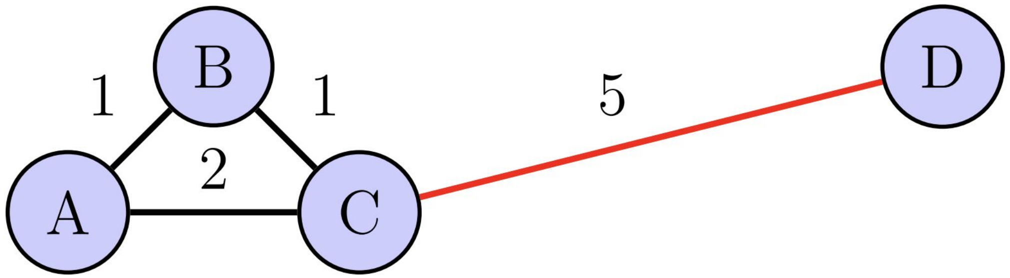
- Reveals if we're **under-wired given feasibility** (range/LOS/barriers)

Use it to decide:

- **Raise candidate K** or add links where density is low

Edge Length Proportion — Long-Edge Burden

- **Definition:** A spatial metric quantifying the proportion of an individual edge's length relative to the total length of all edges in a network.
- **Purpose:** Provides insights into the significance of a specific edge within the overall network structure, useful for infrastructure planning and spatial resource optimization.



Mathematical Representation

- The Edge Length Proportion of edge (e) is calculated as:

$$\text{Edge Length Proportion (of edge e)} = \frac{\text{Length of edge e}}{\sum_{\text{all edges } e' \in E} \text{Length of edge } e'}$$

This ratio evaluates an edge's scale of contribution to the network's total length, highlighting its relative importance.

Why it matters

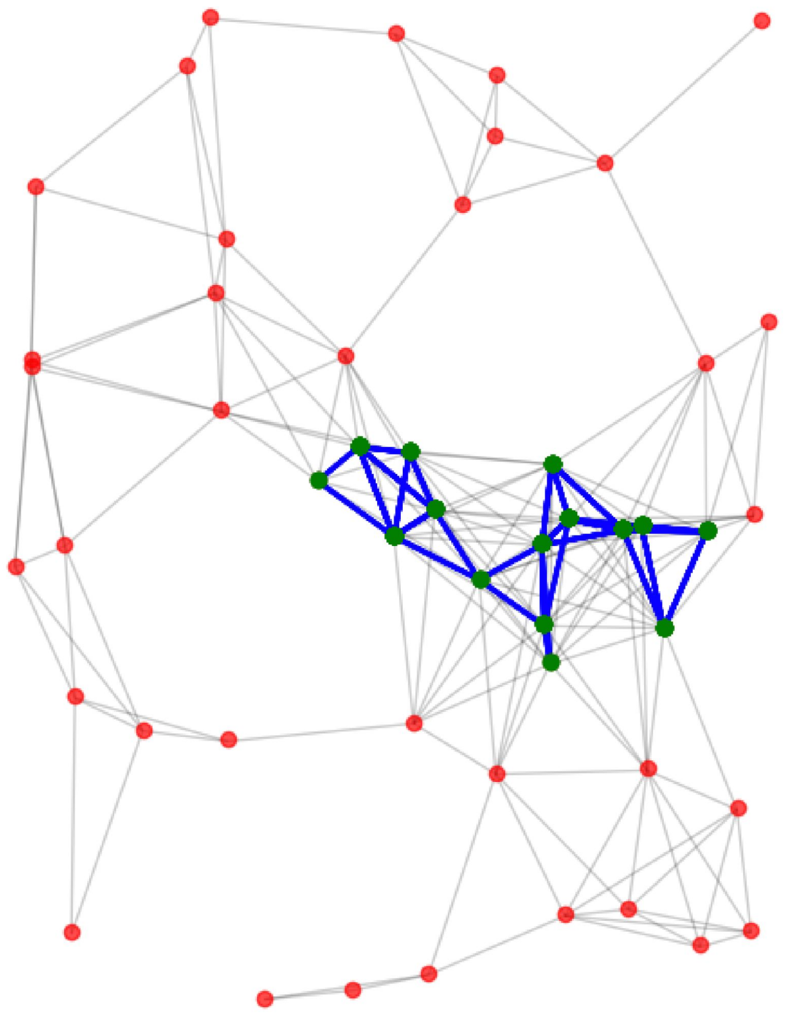
- Flags fragile, costly spans dominating movement.

Use it to decide:

- Cap edge length & insert intermediates to split long hops.

Spatial Clustering Coefficient — Compactness

- **Purpose:** Adapts the traditional clustering coefficient for spatial networks, emphasizing the importance of both the number and compactness of closed triplets.
- **Insight:** Offers a measure of the tendency for nodes to form tightly-knit, geographically proximal communities, enhancing our understanding of spatial network dynamics.



Mathematical Representation

$$\text{Spatial Clustering Coefficient}(v) = \frac{\text{Number of closed triplets involving } v}{\text{Number of all possible spatially close triplets involving } v}$$

Why it matters

- Quantifies tight, efficient neighborhoods (closed triplets within a radius)

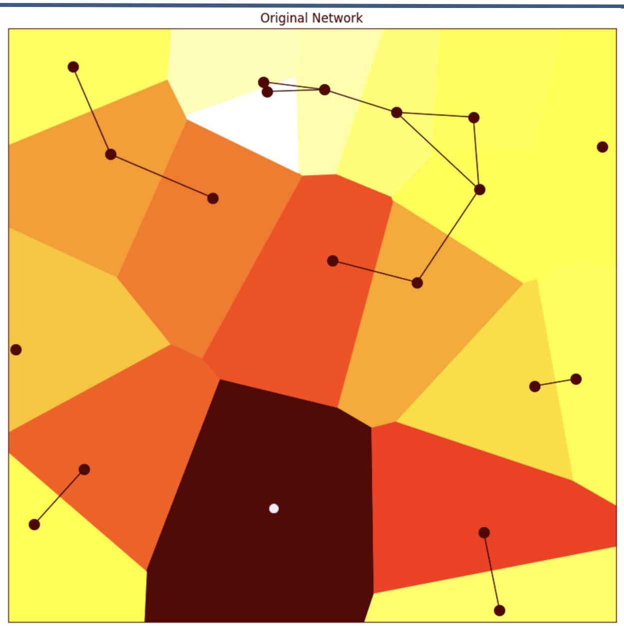
Use it to decide:

- Prefer compact clusters for stable local ops; avoid diffuse ones as anchors.

2 — ROBUST Networks — Graph-Based Spatial Metrics

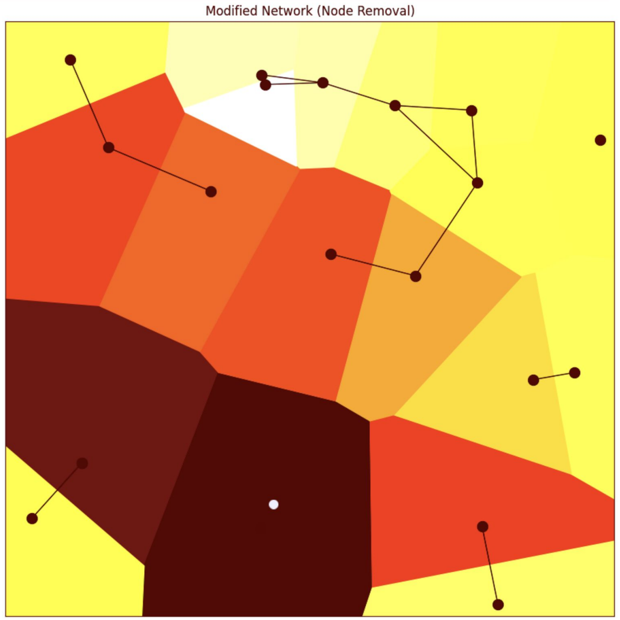
Spatial Resilience — Failure Impact

- **Core Concept:** Goes beyond edge connectivity, focusing on preserving the integrity of spatial segments under the control or influence of nodes.
- **Purpose:** Measures network robustness in terms of spatial coverage and influence.



Mathematical Representation

$$R_s = \frac{1}{|V|} \sum_{v_i \in V} \nVdash_{\text{intact}}(S(v_i))$$



Why it matters

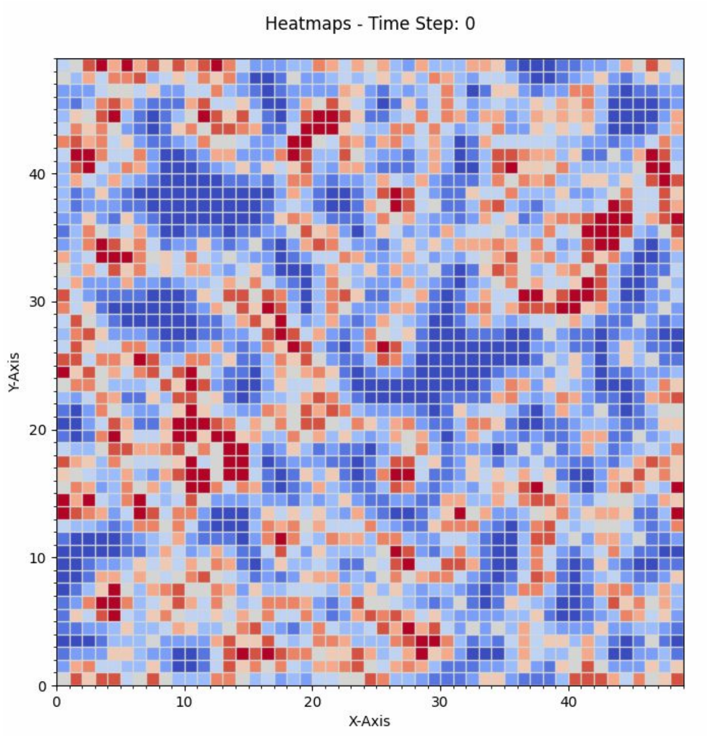
- Measures coverage lost when a node fails (segment integrity)

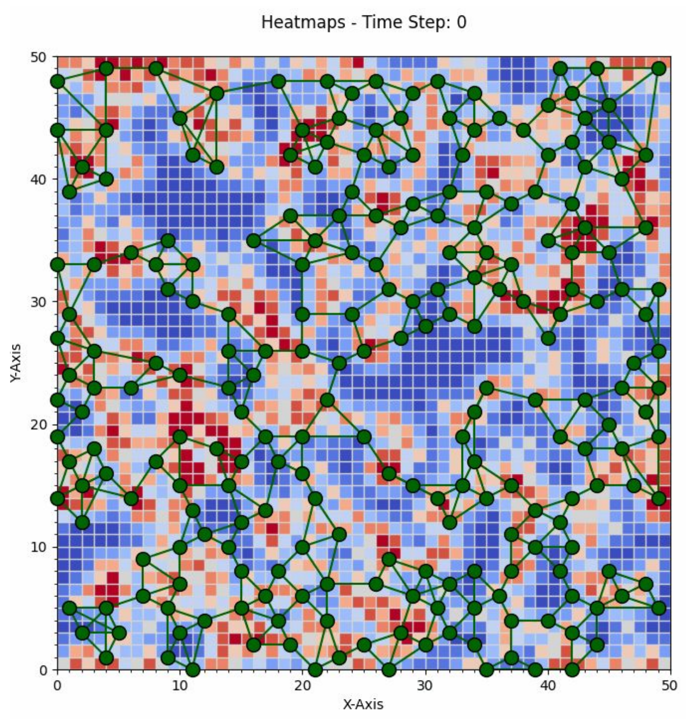
Use it to decide:

- Harden/duplicate high-criticality nodes; pre-plan alternates.

3 — PREP Mapper — Proximal Recurrence Extract & Project

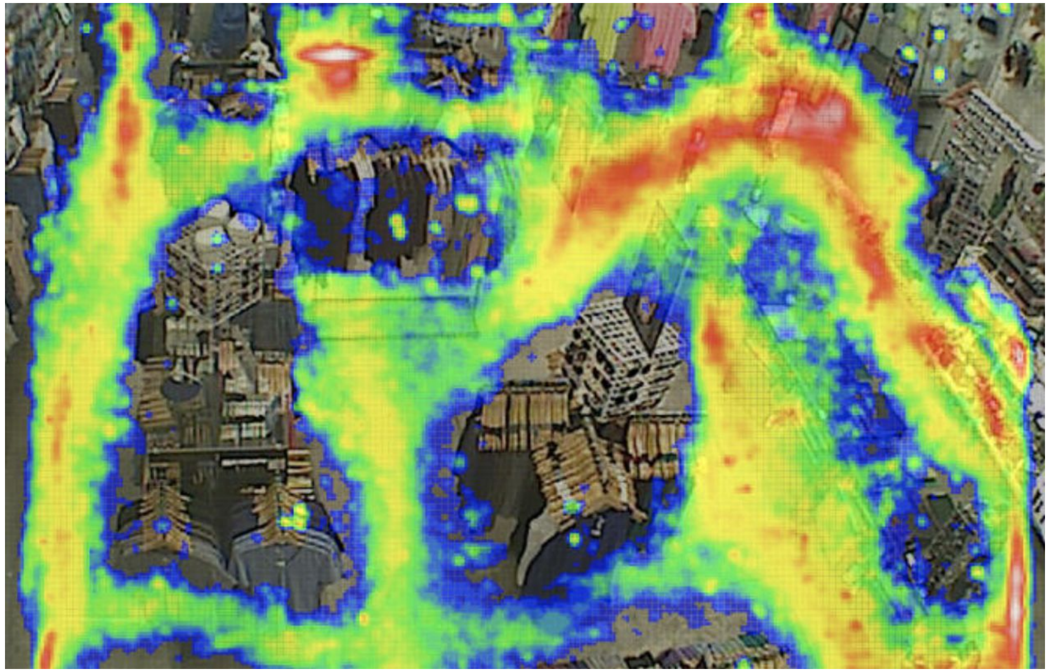
- **Reality:** Missions live in **continuous space**; value, risk, and feasibility **change over time**.
- **Without PREP:** Pixels/raw POIs → **huge search**, **noisy picks**, **slow replans**.
- **PREP's job: Extract** the top-K candidate locations per window using (PR/WPR).
- **Then:**
  - **Stationary case:** place observers at those K locations.
  - **Mobile case:** use those locations to build a **sparse, feasible NavGraph** for planning.

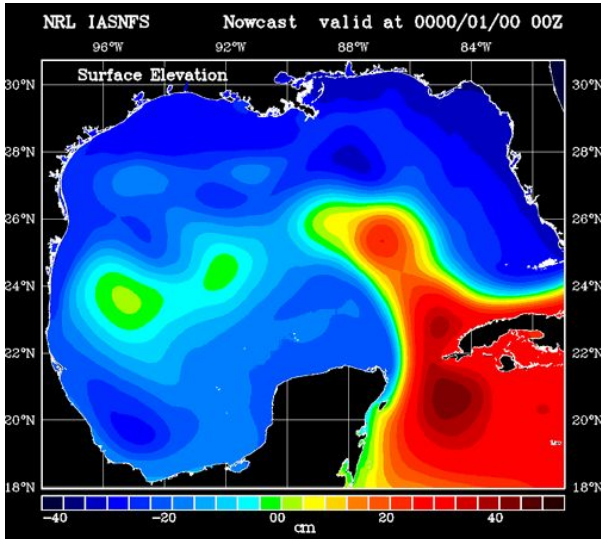


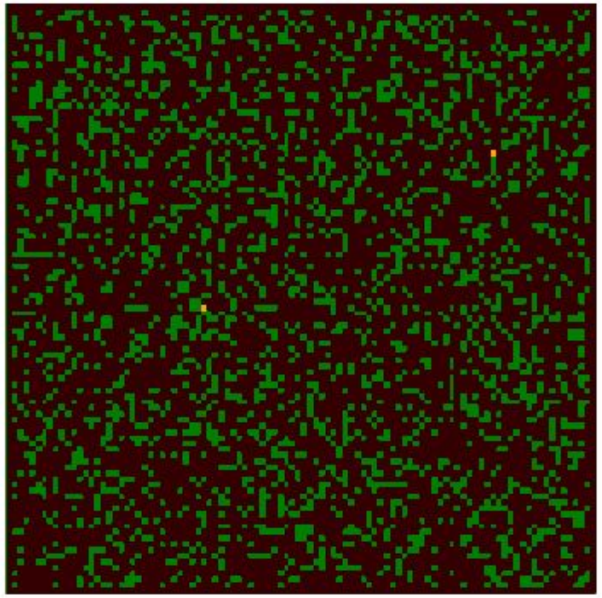


3 — PREP Mapper — Heatmap Abstractions

- We use a **field/tensor** to encode **Priority** (not just "heat").
- The field **prioritizes the spatiotemporal operational space** for a given context.
- The **context is swappable** (policy/data); the **planner contract stays the same**.
- Same PREP logic works for **any ROI** (crisis, utility, uncertainty, …).
- A **dynamic field** simply means **cell values change over time**.
- A dynamic field means that a cell may account for multiple events over time.







3 — PREP Mapper — Bipartite Dynamics

Analyzing Efficacy & Coverage of Observers

Algorithm Overview

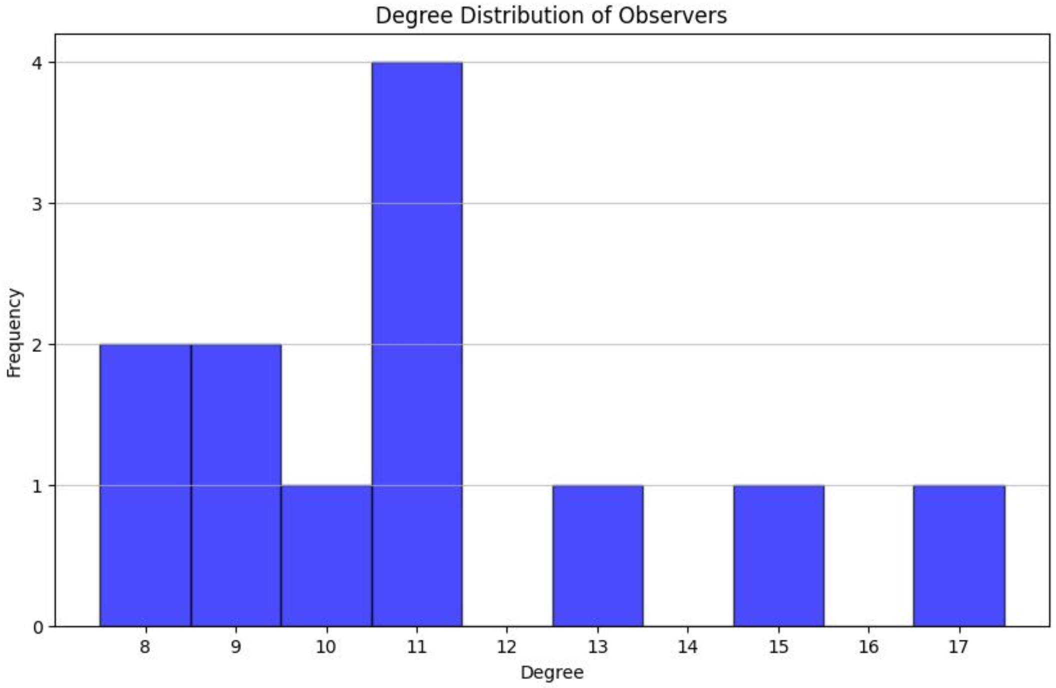
Input
- Observer Nodes
- Event Nodes

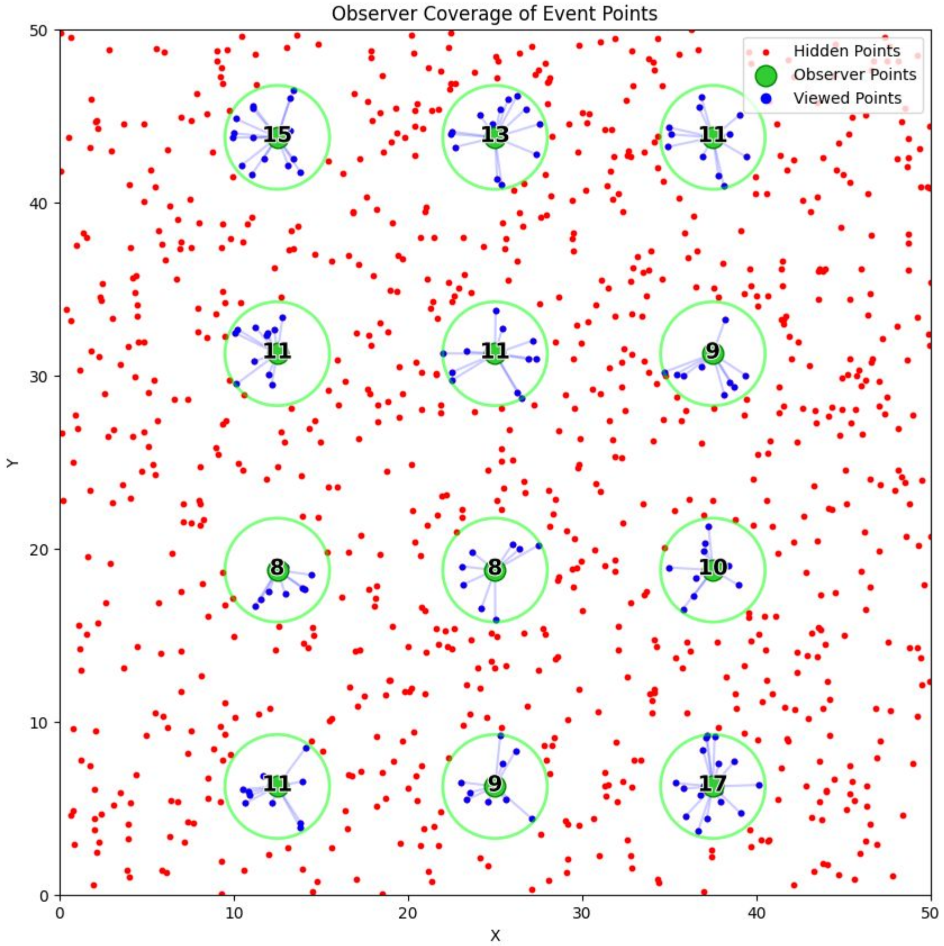
→ Myopic Filtering → Link Generation → Centrality Ranking → Event Classifier →

Output
- ranked observers
- observed points

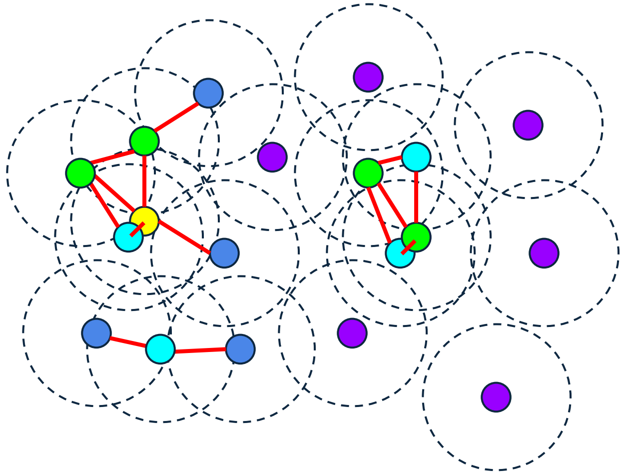
Goal:

To identify coverage areas, detect gaps, & pinpoint observer placement weaknesses.





ROBUST Graphs and Proximal Recurrence



Objective:

- **Network Analysis**: Employ graph theory to analyze the distribution of resources and to identify key points for optimal observation within the network.

Approach (ROBUST):

Conceptualize the placement problem within the ROBUST framework, focusing on network characteristics derived from graph theory.

Step-by-Step Approach:

1. **Event Point Extraction:** Isolate unobserved event points from the ROBUST network, which represent potential areas requiring coverage. *expected:* ($O(n)$)

$$P_x = \{ x_1, x_2, \ldots, x_m \},$$
$$P_y = \{ y_1, y_2, \ldots, y_m \}$$

2. **Link Generation:** Calculate the connections between unobserved points within the observer's range, constructing a graph where points are nodes and links indicate potential coverage. *expected:* ($O(m^2)$)

$$C = \{(e_i, e_j) : e_i, e_j \in UE, i \neq j\}$$

3. **Degree-Based Sorting:** Organize event points in descending order of their degree—i.e., the number of links to other points—prioritizing points with the highest connectivity for resource placement. *expected:* ($O(m \log m)$)

$$D_c = \{ C_i \in C_{\text{sorted}} : C_i \text{ is maximal dense}\}$$

3 — PREP Mapper — Unipartite Dynamics

Proximal Recurrence Clustering

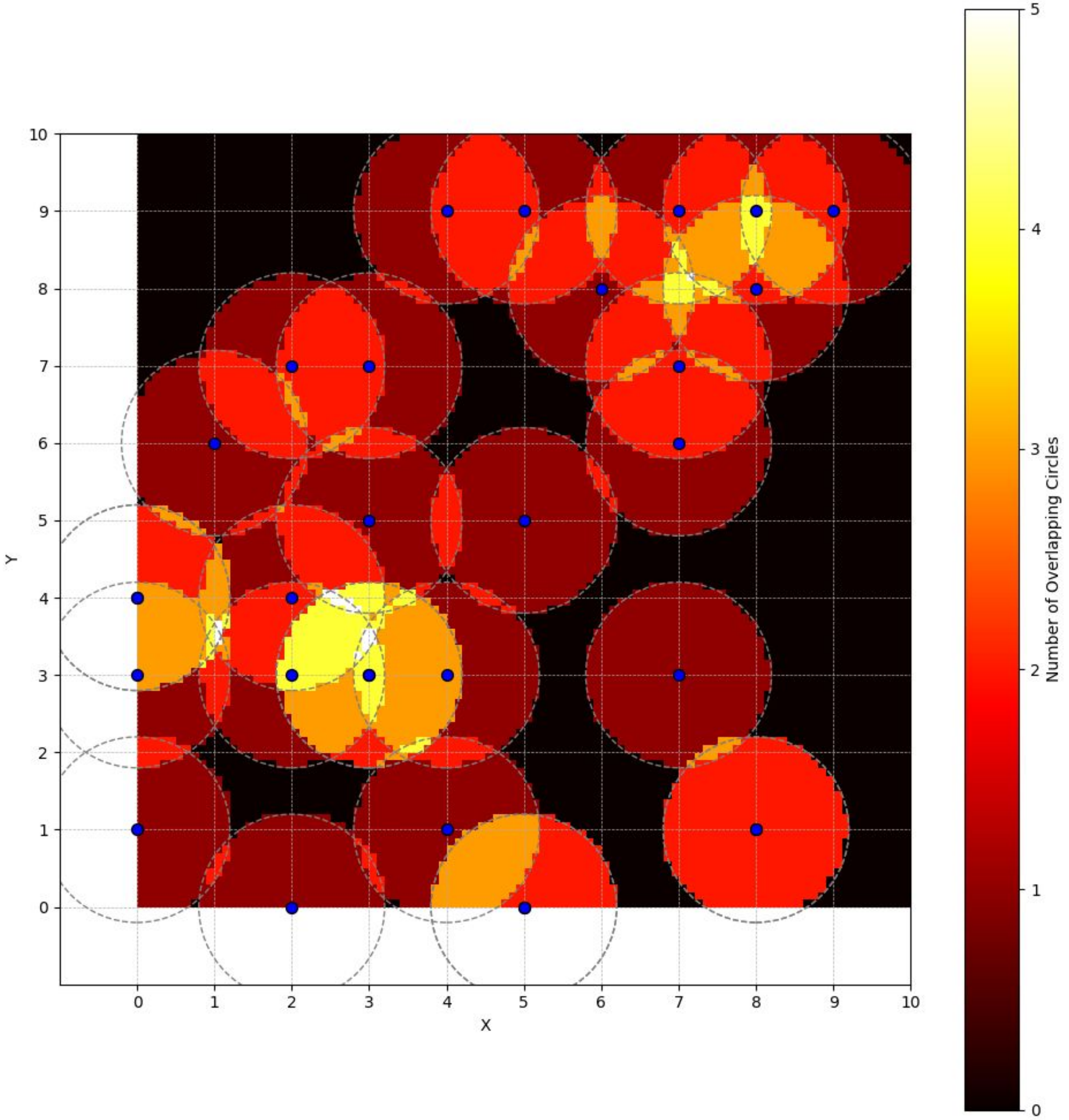
Proximal Recurrence (PR)

- **Objective:** Maximize monitoring efficiency by identifying areas with a high concentration of unobserved events.

- **Steps:**
  1. *Count Events Within Range:* Assess each unobserved event to count nearby events within sensor range, including both existing and predicted future events.

  2. *Identify Densest Cluster:* Find the area with the highest density of events, using the counts from step 1 as a guide.

  3. *Select for Node Insertion:* Choose the identified densest cluster as the priority location for deploying a new sensor node.



Unobserved Points: Blue dots are locations & circle ts view range.

Interpretation of Color Intensities

- **Black**: Not monitored.
- **Dark Red:** Low overlap and coverage
- **Red:** Moderate overlap and coverage
- **Yellow**: High overlap and coverage.
- **White**: Optimal due to highest overlap.

3 — PREP Mapper — Unipartite Dynamics

Maximizing Coverage with Optimal Placements

Problem:

- Multiagent Temporal Pathing of Ranged Observational Units

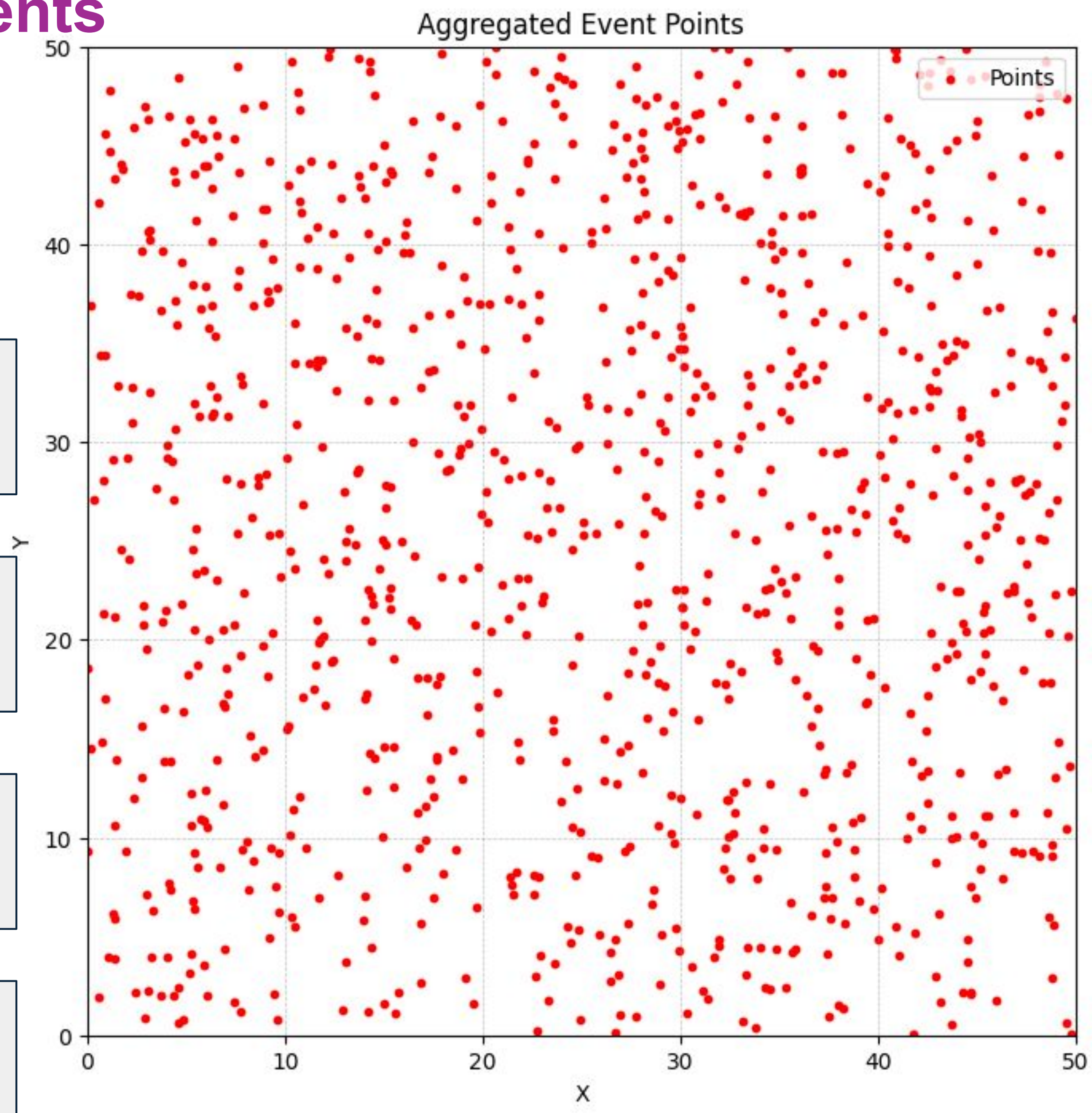
Algorithm Overview

Heatmap Generation:
Create a matrix representation of point density to simplify the problem space.

Kernel Convolution:
Apply a circular kernel to the heatmap to map out potential coverage zones.

Centroid Identification:
Employ an iterative process to select optimal resource locations, enhancing coverage efficiency.

Iterative Optimization:
Remove covered points from consideration, preventing redundancy.



3 — PREP Mapper — Unipartite Dynamics

Heatmap Creation for Point Density Matrix

Point Density Heatmap

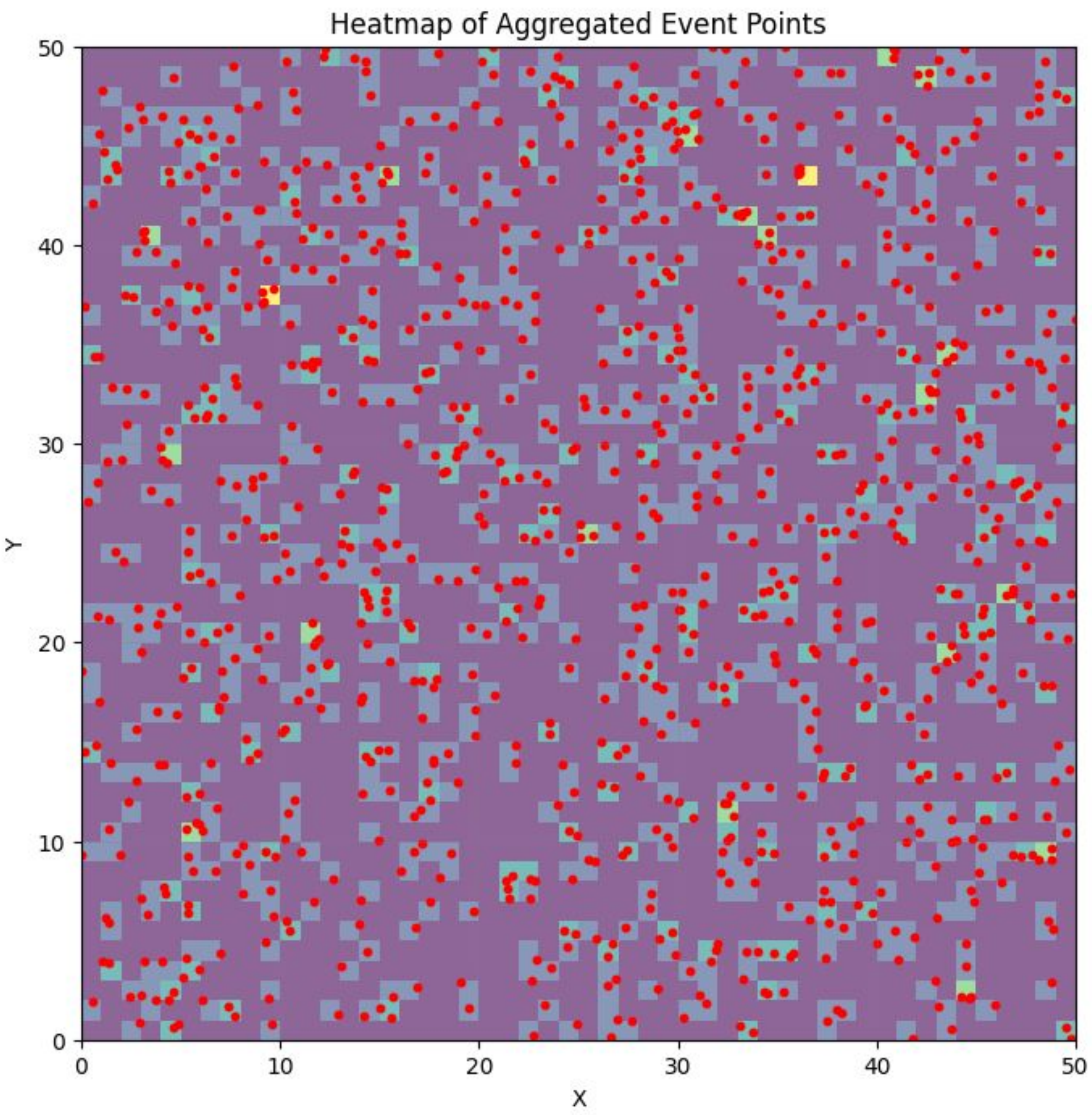
Heatmaps, represented as matrices of spatial point densities, dramatically reduce the $O(n^2)$ complexity typically associated with point-to-point evaluation.

Binning as Spatial Partitioning

Through binning, points are partitioned into discrete spatial indices based on their coordinates, creating a higher-level density matrix. Each cell within this matrix represents the aggregate density of points.

Towards Practical Application

The implementation leverages computational accelerations like CUDA to handle large-scale datasets effectively. This ensures the algorithm's scalability and enhances its efficiency.



3 — PREP Mapper — Unipartite Dynamics

Kernel Convolution for Coverage Mapping

Kernel Convolution:

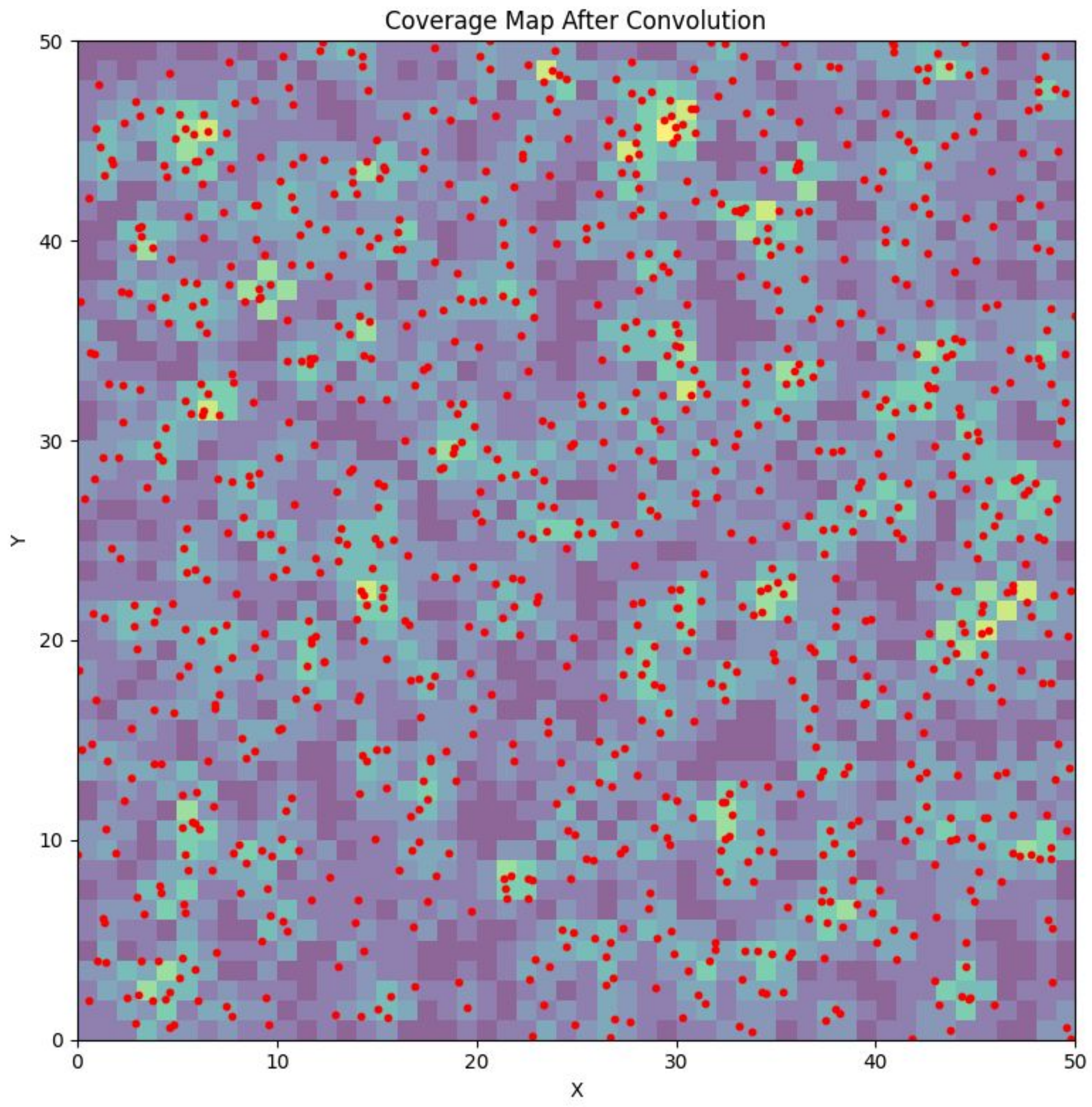
Apply a predefined shape (e.g., a circle) that represents the area each point covers. The radius of this shape correlates to the coverage area.

Convolution Process:

The algorithm overlays this kernel shape onto the heatmap matrix. Convolution identifies regions with high densities of points under the kernel area, indicating high potential for coverage.

Identifying Coverage Areas:

By scanning across the heatmap, convolution highlights areas where the cumulative density—under the kernel's footprint—reaches a maximum. These areas signify optimal locations for resource placement to maximize coverage.



Identifying Optimal Placements

Process Overview:

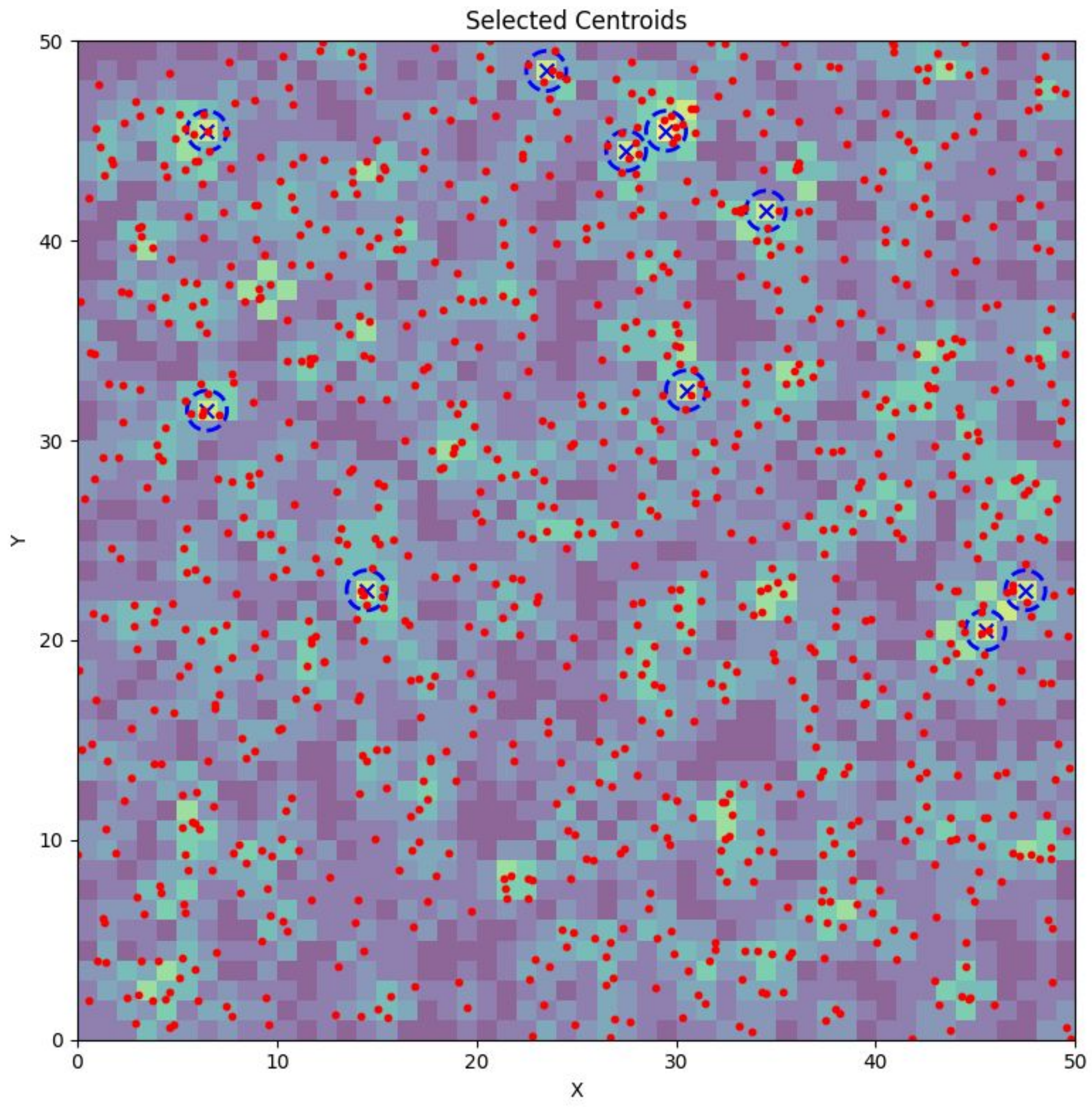
The algorithm seeks out the maximum values in the coverage map, which result from the convolution process, as potential centroids.

Iterative Selection:

- **Initial Identification:** Locate the highest density area in the coverage map as the first centroid.
- **Coverage Optimization:** After selecting a centroid, the algorithm "zeros out" the points within its coverage radius on the heatmap. This step prevents double counting of covered points in subsequent iterations.
- **Repeat Until Completion:** Continue this process, iteratively identifying and zeroing out coverage areas, until the maximum predetermined number of centroids are selected or no significant points remain uncovered.

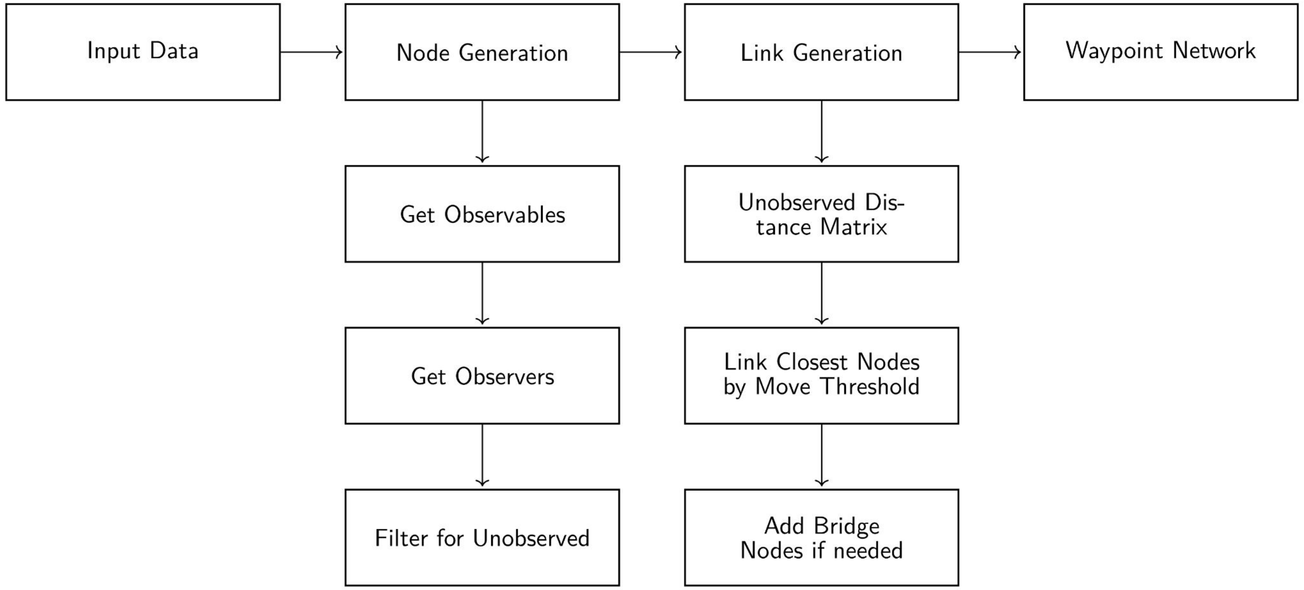
Maximizing Coverage:

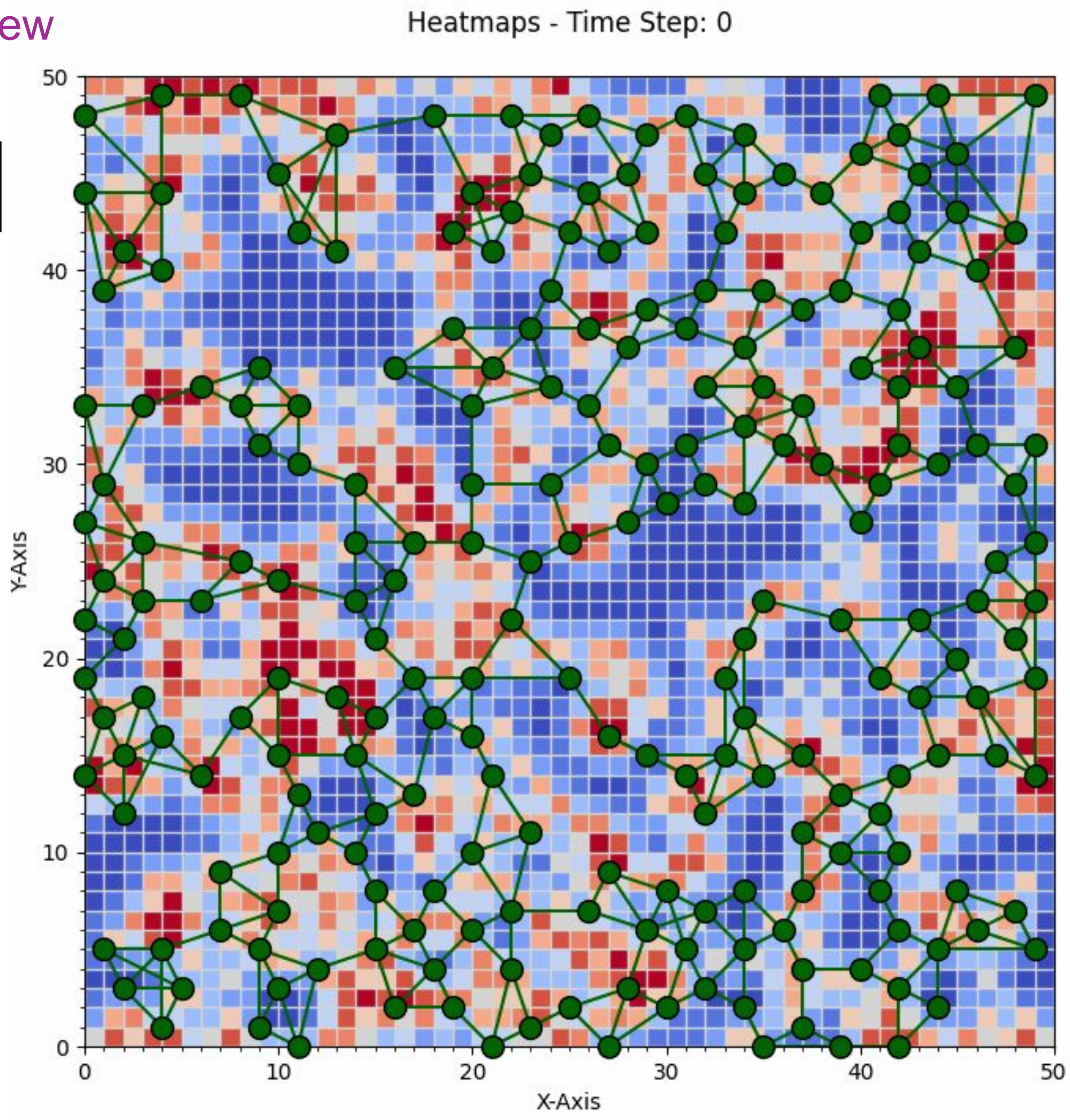
Through this iterative approach, the algorithm efficiently distributes centroids to areas of highest point density, ensuring optimal coverage across the entire dataset.



3 — PREP Mapper — Output - Nav Graph

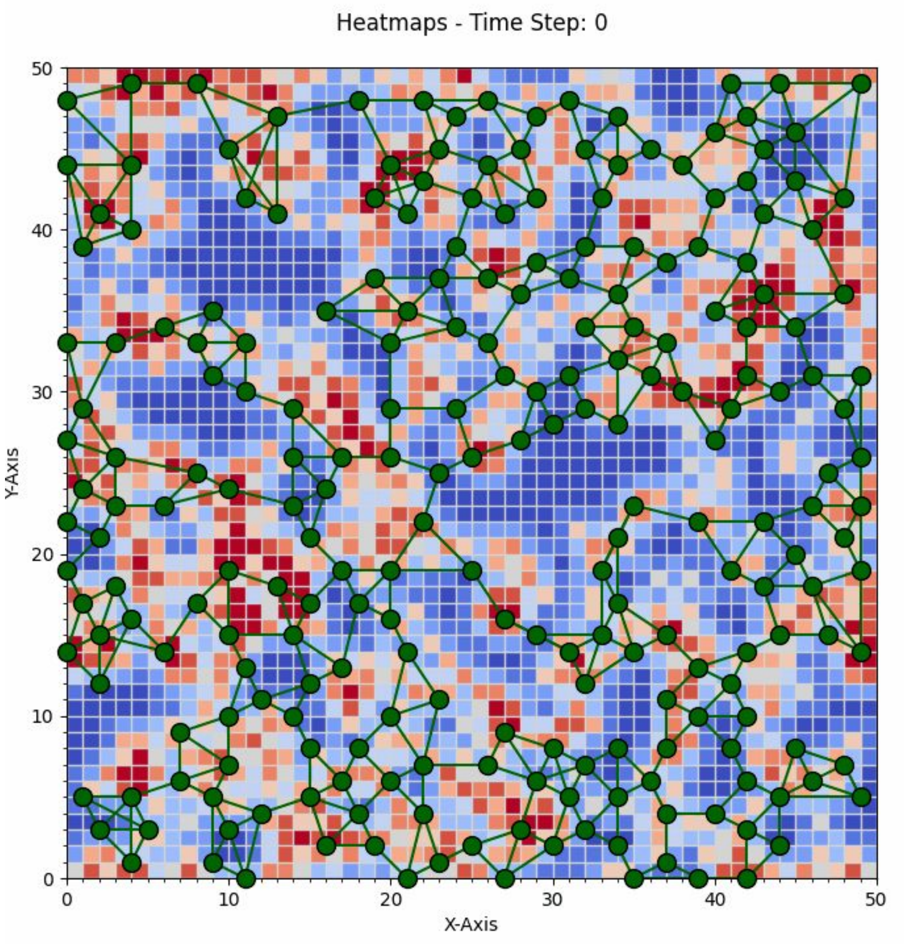
Proximal Recurrent Event Partition (PREP) Mapper - Overview

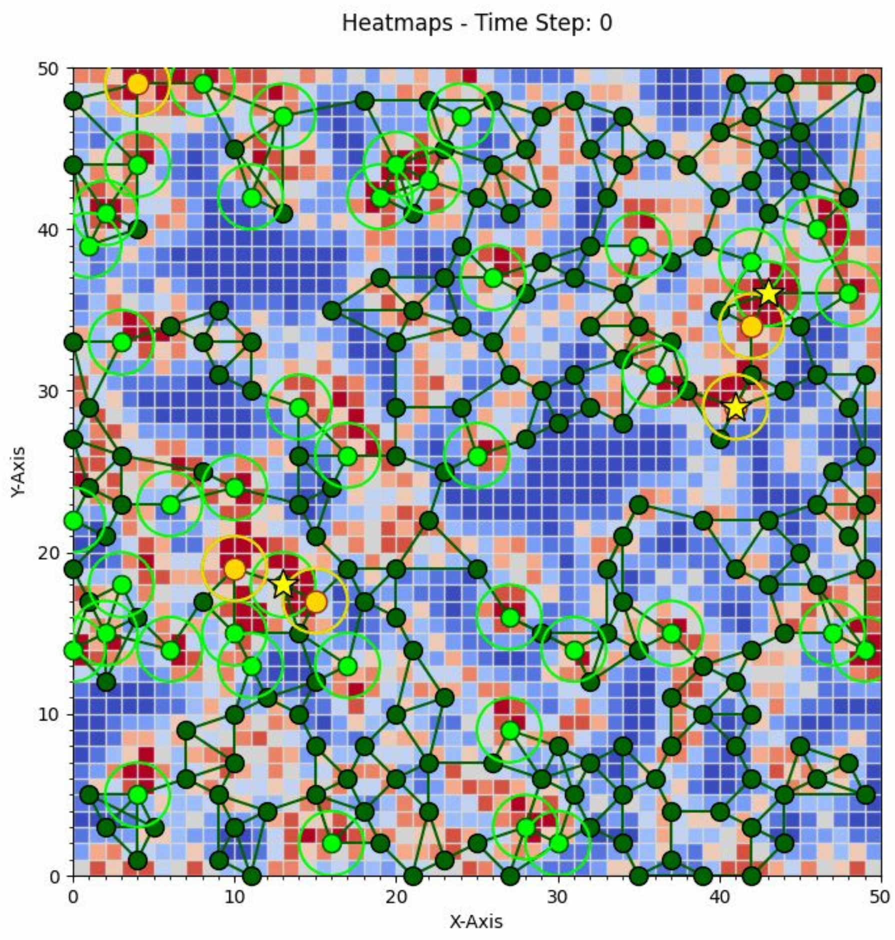




4 — WAITR Planner — Motivation

- **Goal:** optimize **long-horizon** routes over **time windows** with **no-overlap** between selected paths.
- **Inputs:** per-window **Nav Graphs** from PREP + **seam costs**.
- **Output:** stitched, **deconflicted** multi-agent plans over (t=1..T).





4 — WAITR Planner — Observer Nodes - Sampling Behaviors

Observer Movement: Adapting to a Dynamic World

Types of Observers

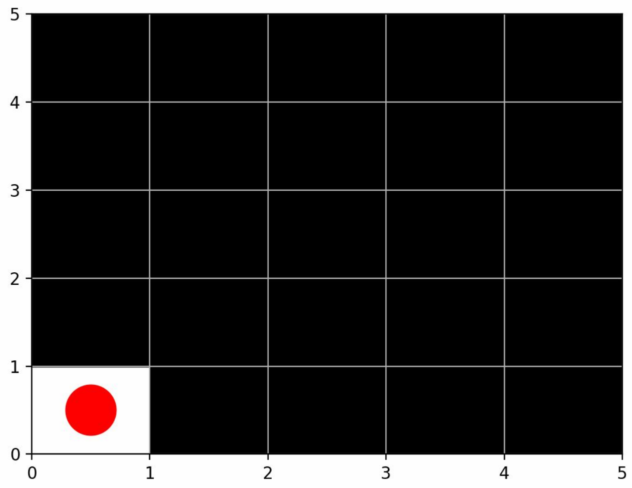
- **Static:** Fixed location, ideal for consistent monitoring of critical areas.
- **Dynamic:** Capable of movement, enhancing adaptability.
    - **Discrete:** Move at intervals or in response to triggers.
    - **Continuous:** Can move in real-time for tracking and rapid adjustment.

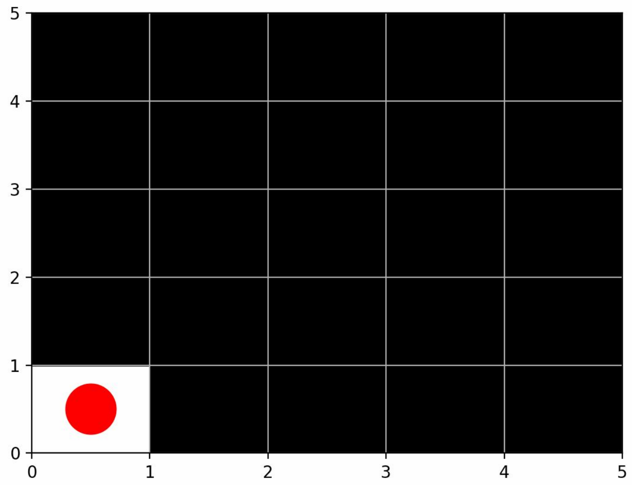
Environment Matters

- **Obstacle Avoidance:** Navigating through physical barriers.
- **Terrain Adaptation:** Compatibility with various environments.
- **Energy Management:** Efficient use of power.

Asymmetric Movement:

ROBUST may leverage a mix of observer types for a balanced and effective approach.





4 — WAITR Planner — Pathlets & the Lookup Table

Build pathlets

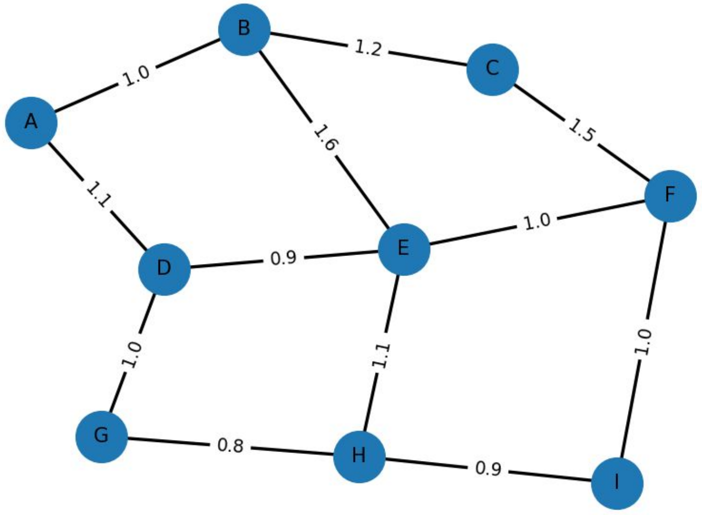
Precompute **shortest simple paths** with **≤ HmaxH_{\max}Hmax** hops between candidate nodes in GtG_tGt. Each path = a **pathlet**.

Store in the Lookup Table (LUT).
For each ordered pair keep **one best** entry: **id, start→end, hops, cost/length, nodes**.

Why this step.

Converts into a **cache of reusable local routes**, making temporal stitching **fast** and **stable**.

Window t — NavGraph G_t



start	end	hops	cost	nodes
A	C	2	2.2	A→B→C
A	E	2	2.0	A→D→E
A	G	2	2.1	A→D→G
B	D	2	2.1	B→A→D
B	F	2	2.6	B→E→F
B	H	2	2.7	B→E→H
C	A	2	2.2	C→B→A
C	E	2	2.5	C→F→E
C	I	2	2.5	C→F→I
D	B	2	2.1	D→A→B
D	F	2	1.9	D→E→F
D	H	2	1.8	D→G→H
E	A	2	2.0	E→D→A
E	C	2	2.5	E→F→C
E	G	2	1.9	E→D→G
E	I	2	2.0	E→F→I
F	B	2	2.6	F→E→B
F	D	2	1.9	F→E→D
F	H	2	1.9	F→I→H
G	A	2	2.1	G→D→A
G	E	2	1.9	G→D→E
G	I	2	1.7	G→H→I
H	B	2	2.7	H→E→B
H	D	2	1.8	H→G→D
H	F	2	1.9	H→I→F
I	C	2	2.5	I→F→C
I	E	2	2.0	I→H→E
I	G	2	1.7	I→H→G

4 — WAITR Planner — Seam Costs between windows

What a seam is.

A **seam** prices the hand-off from a pathlet in window t to a pathlet in t+1.

Matrix:

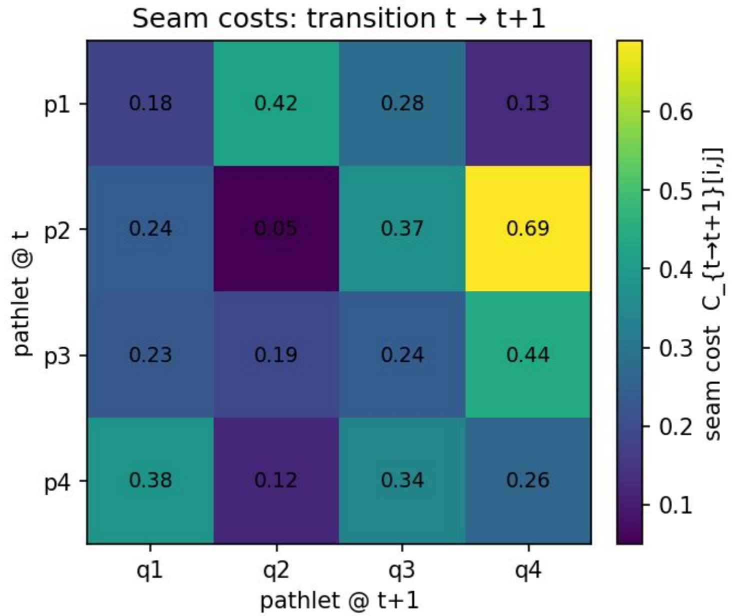
cost of transitioning **pathlet i @ t → pathlet j @ t+1**

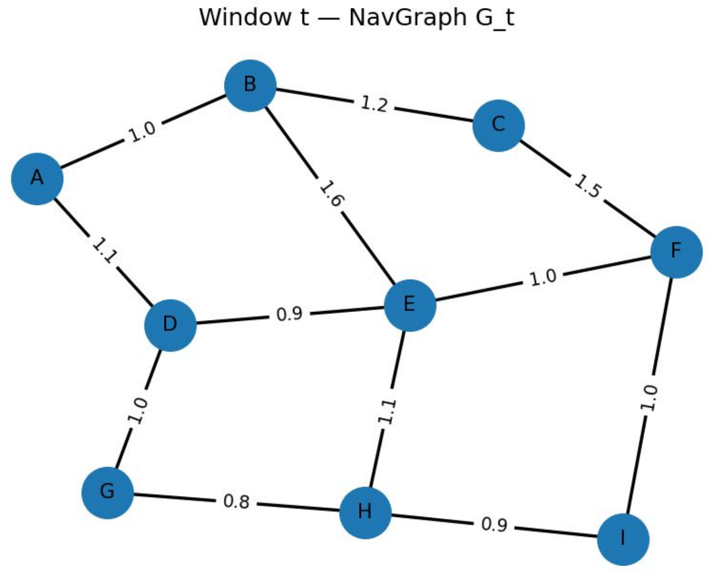
What goes into the cost.

Distance/motion to the next start, feasibility (range/LOS/terrain), policy risk, turn/hand-off penalties, timing/energy/battery alignment, and any “stay-on-node” discount.

Why it matters.

Seams turn per-window routes into an **inter-window fabric**; getting them right makes the DP stitching **both fast and stable**.





4 — WAITR Planner — Temporal Append via DP

State.
For each time window and each candidate pathlet, keep the **best score so far** if the plan ends with that pathlet.

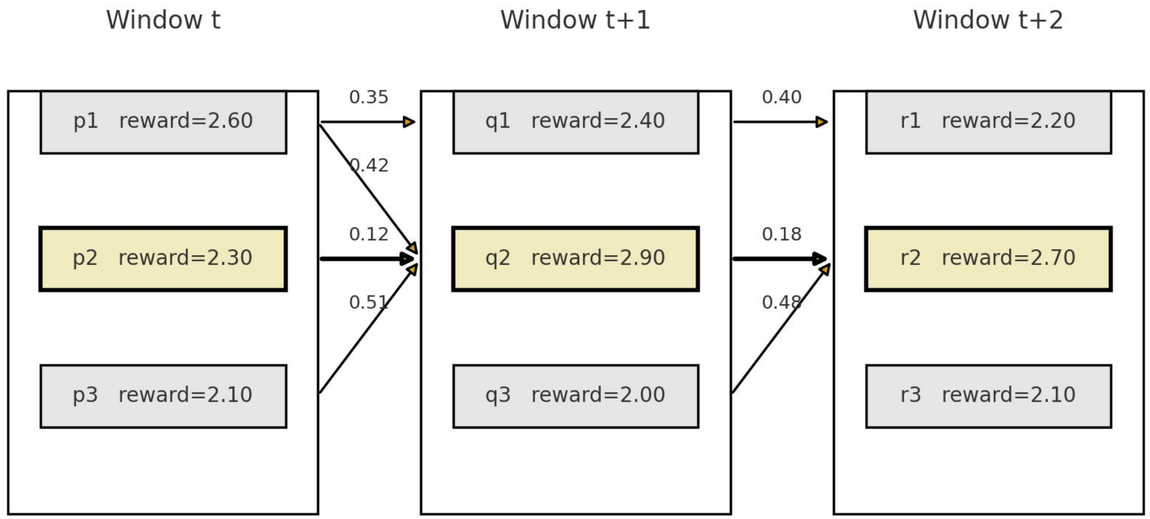
How we update.
To score a pathlet in the next window, we:

1. start from the **best score** of some pathlet in the current window,
2. **subtract the seam cost** to move from that current pathlet to the next one,
3. **add the reward** of the next pathlet.
 We try all predecessors and **pick the best combination**.

What we remember.
Along with the score, we store **which predecessor** gave that best result, so we can **reconstruct the full route** at the end.

Why this works.

We **extend the best partial plan one window at a time**, which makes planning **fast** and the final routes **stable** across windows.



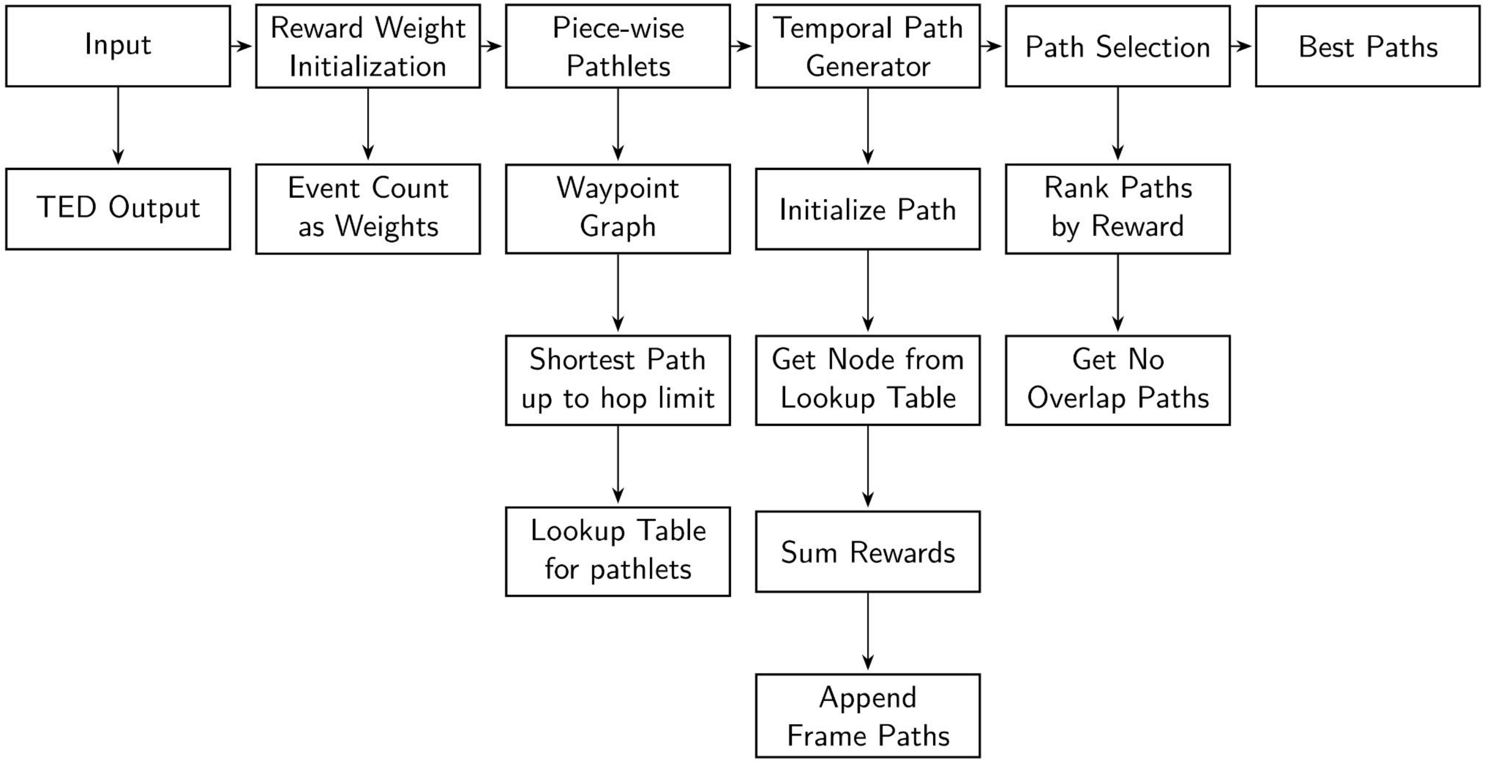
We choose the chain with high total reward and low total seam cost.
Colored boxes = chosen per-window pathlets; bold arrows = chosen seams.

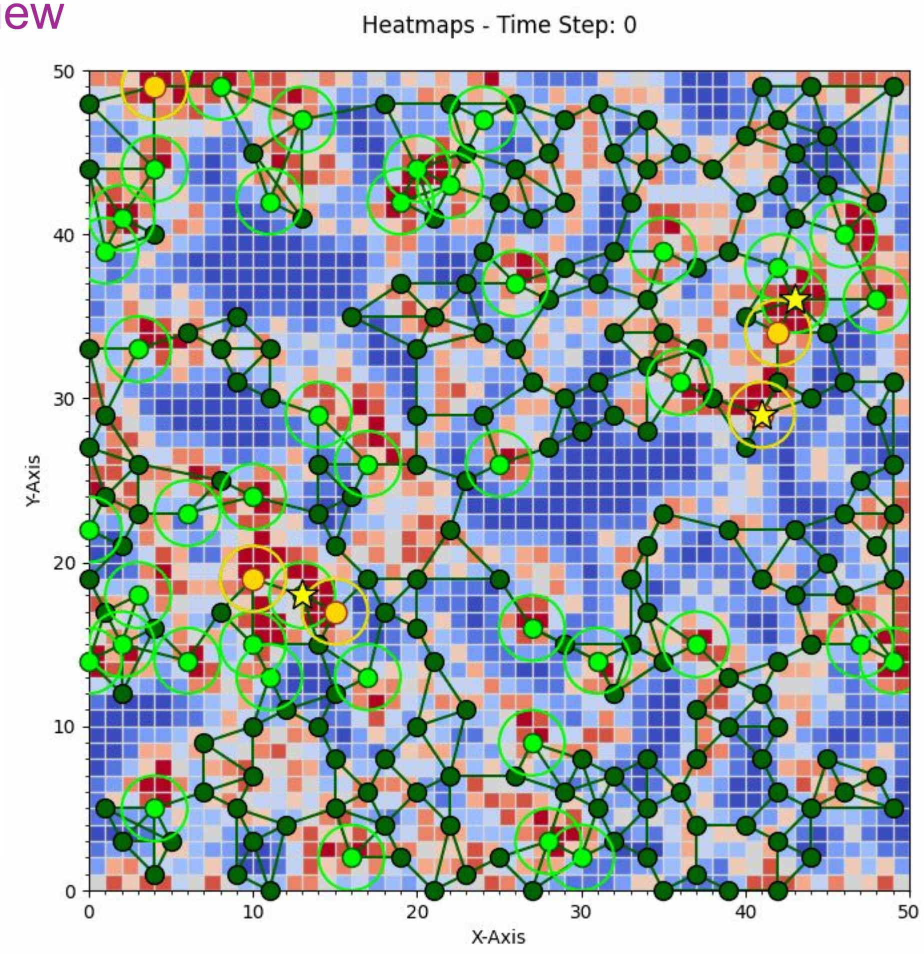
4 — WAITR Planner — No-Overlap Selection (deconfliction)

- **Constraint (within a window).** Chosen pathlets **cannot share nodes** (and optionally, edges).
- **Selection rule (fast default). Flatten → sort by score → accept if disjoint.** Skip any candidate that touches already-selected nodes.
- **Exact alternative (when needed).** Solve a **small ILP / maximum-weight set-packing** over pathlets vs. their node sets.

4 — WAITR Planner — *Full Cycle:* Phase 3 - WAITR Planner

Weighted Aggregate Inter-Temporal Reward (WAITR) Planner - Overview





Operational Dynamics:

- **Initial Condition:** All nodes initially start scoreless, symbolizing a state with no detected events.
- **Activation and Scoring:** Nodes activate upon detecting events, receiving scores that become part of the pathway calculations.
- **Score Retention & Update:** Active nodes maintain their scores, with potential updates from new events or better paths via neighbors.
- **Efficient Computation by Pruning:** Inactive nodes—those without new events —stay dormant in subsequent processes, focusing efforts on change-prone areas.

4 — WAITR Planner — *Full Cycle:* Overview

Introduction to Dynamic Sensor Placement

- This builds on the ROBUST Network, introducing a process divided into three phases to determine optimal multi-agent sensor paths.

Phase 1:
Proximal Recurrent Event Partition (PREP) Mapper

↓

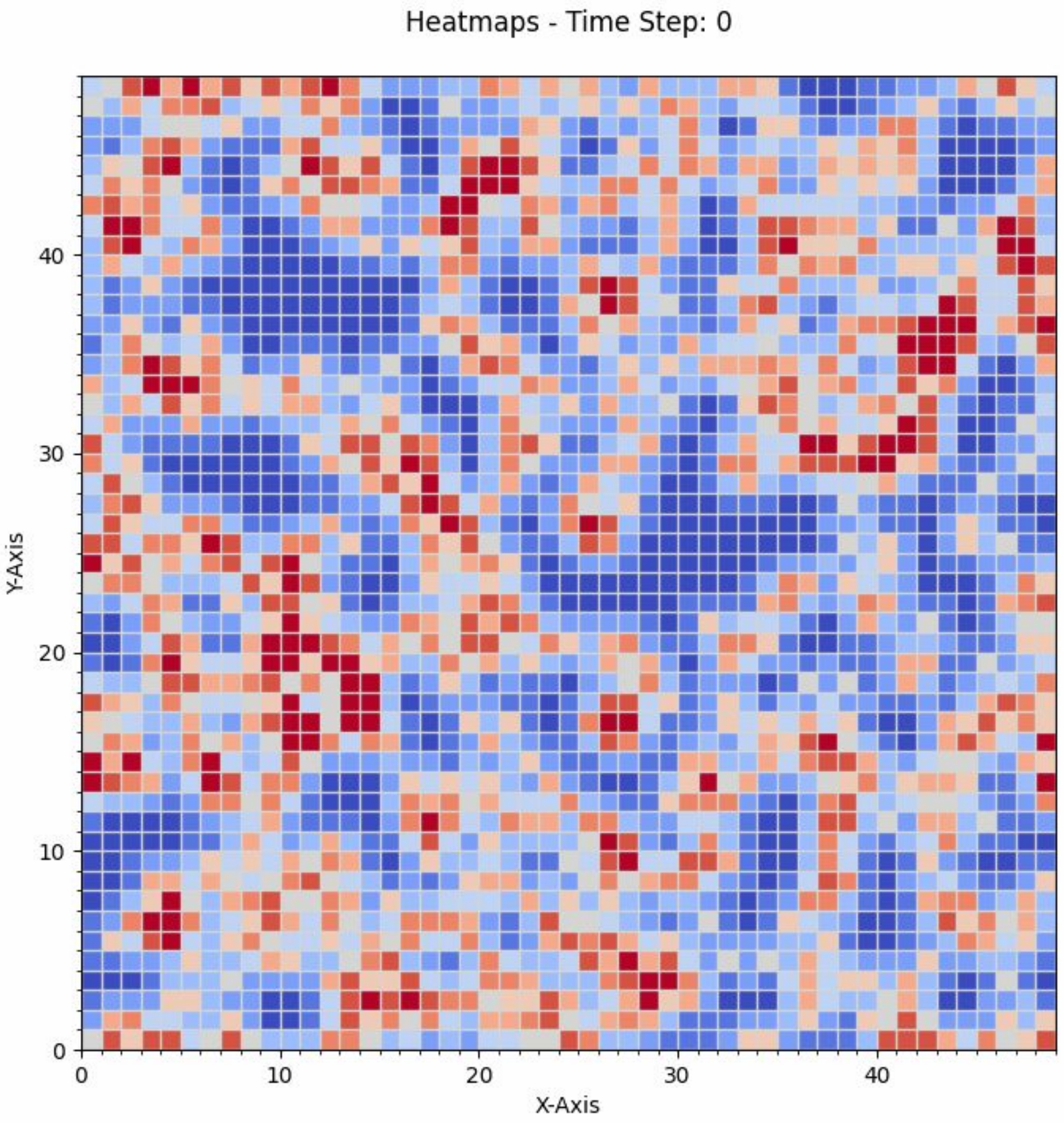
Phase 2:
Tensor-Encoded Dynamics (TED) Compiler

↓

Phase 3:
Weighted Aggregate Intra-Temporal Reward (WAITR) Planner

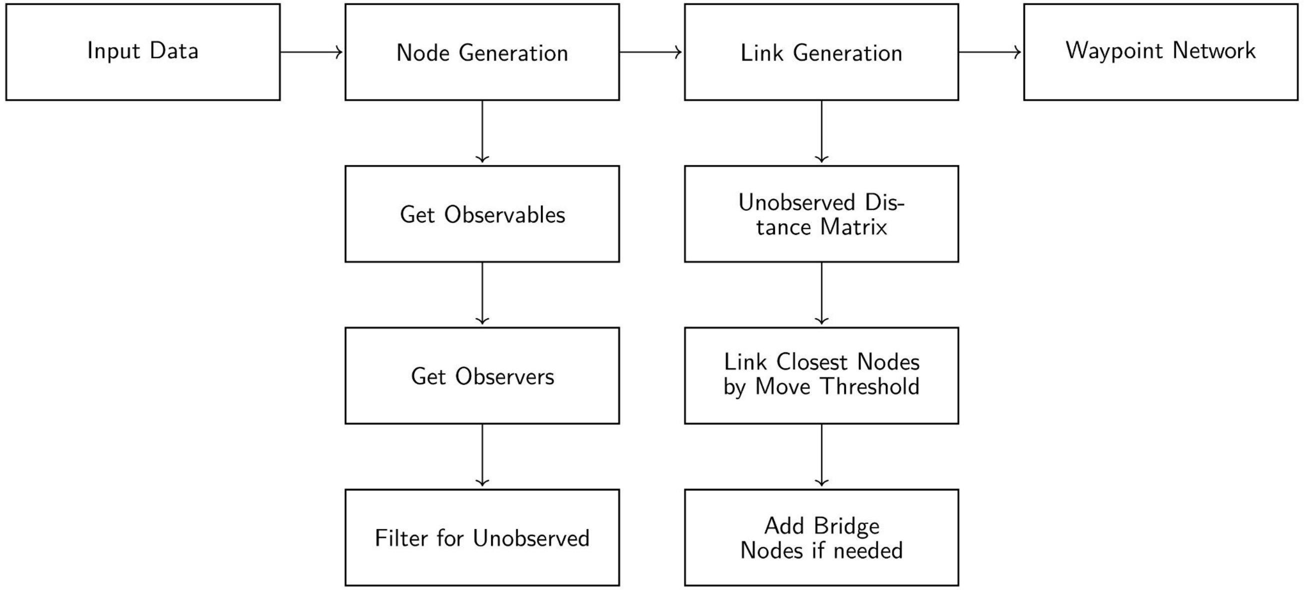
↓

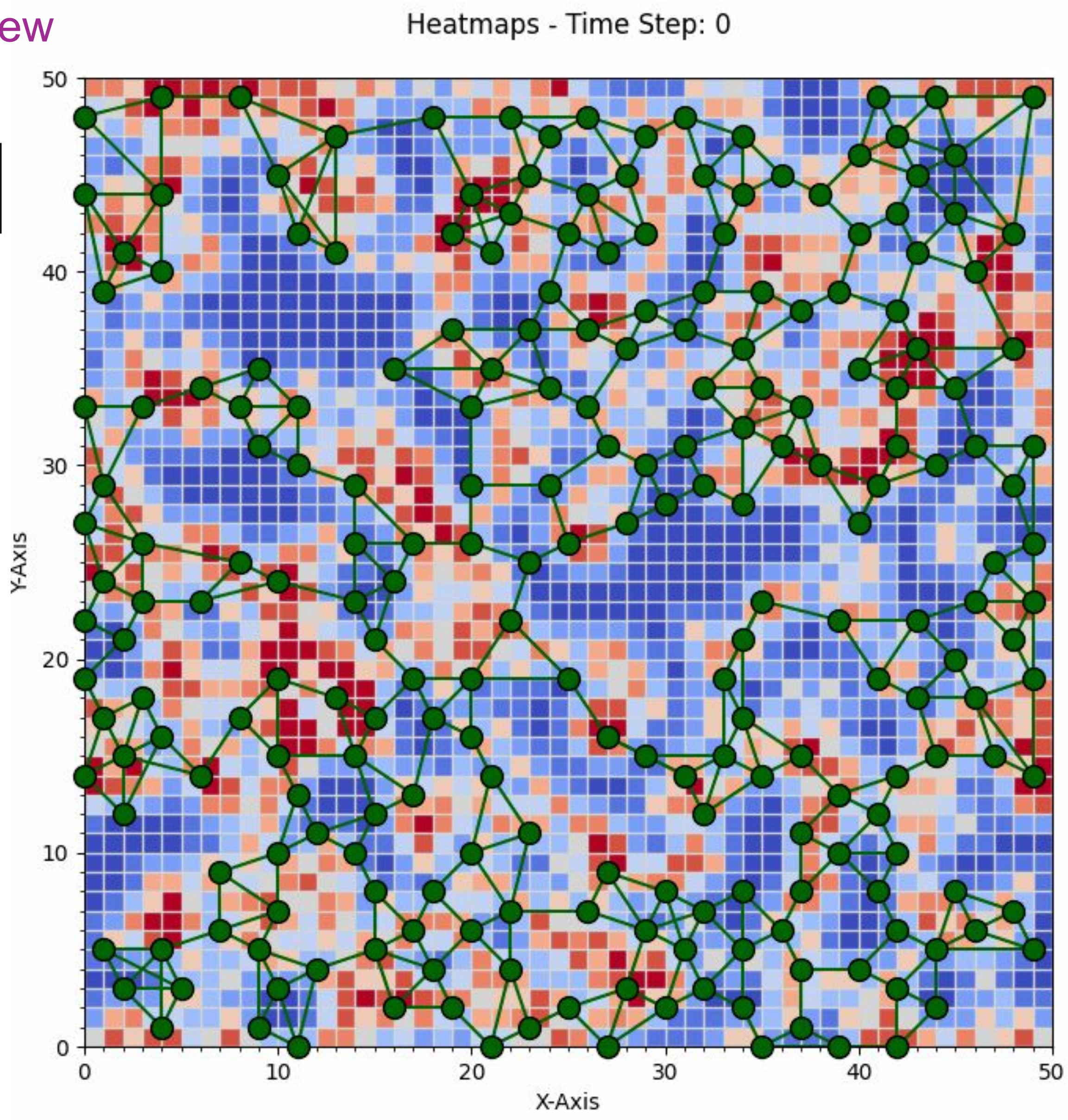
Output:
Long-Horizon Temporal Path



4 — WAITR Planner — *Full Cycle:* Phase 1 - PREP Mapper

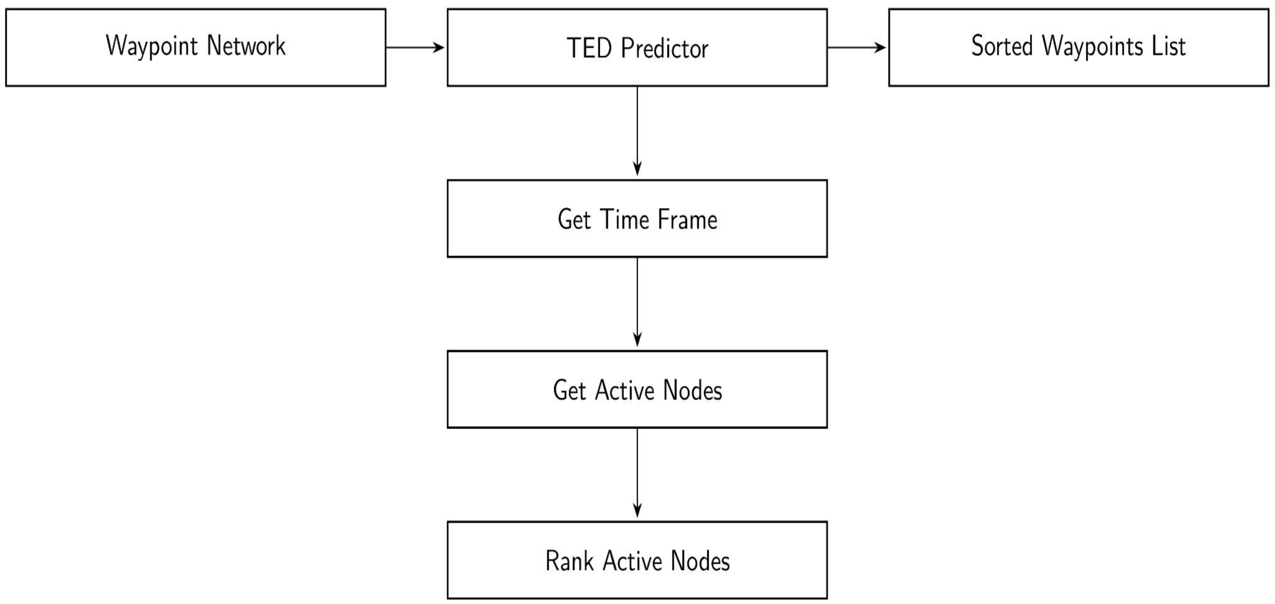
Proximal Recurrent Event Partition (PREP) Mapper - Overview

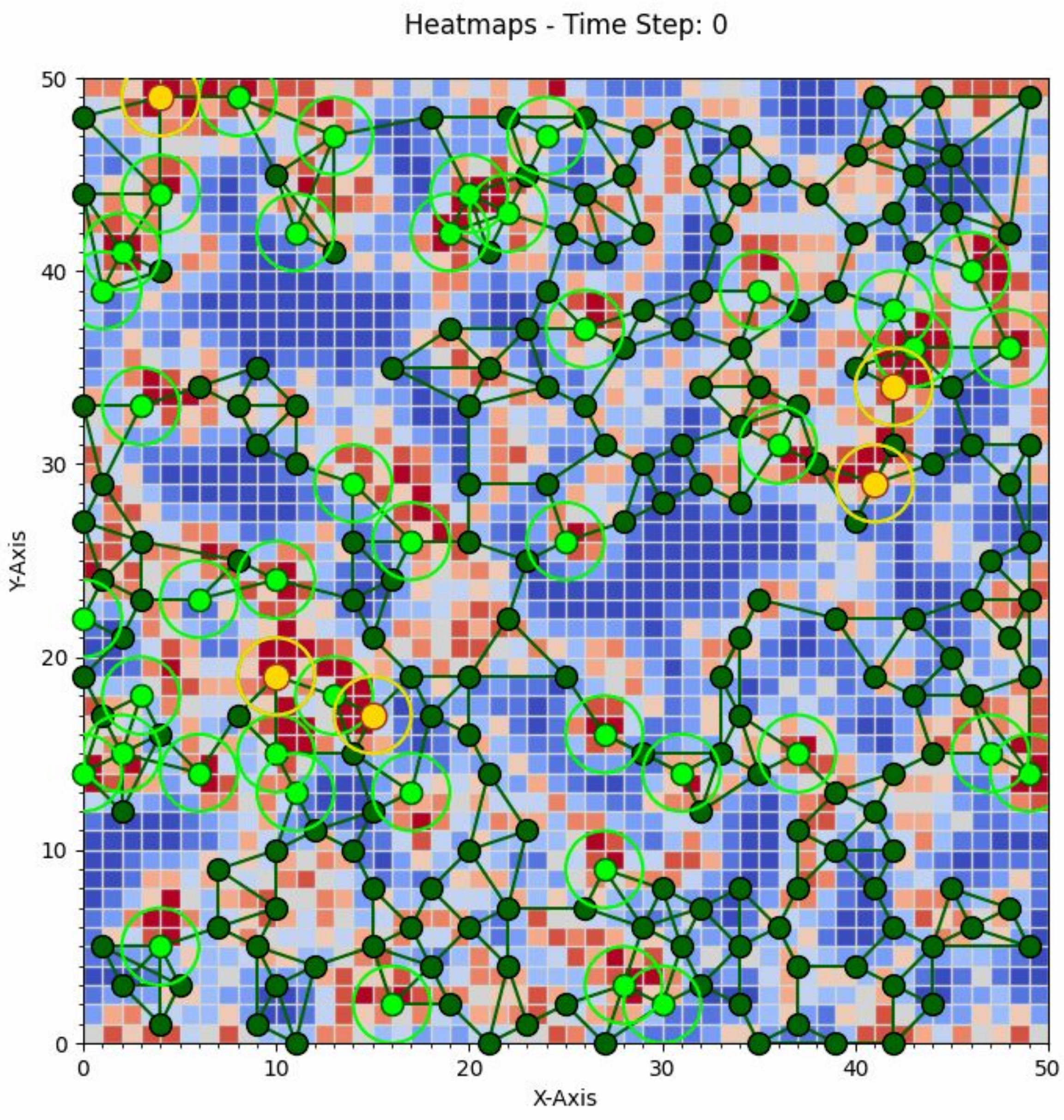




4 — WAITR Planner — *Full Cycle:* Phase 2 - TED Compiler

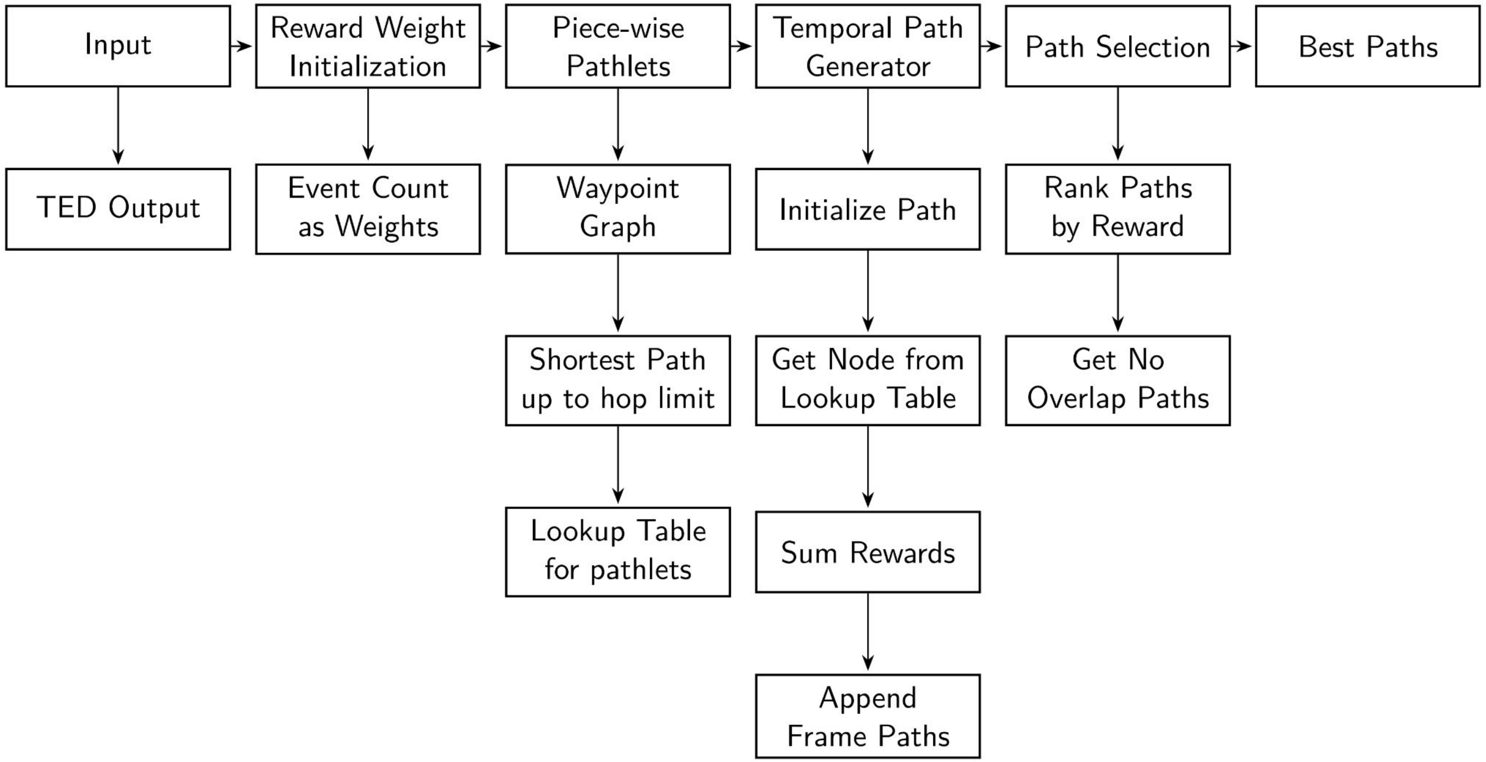
Temporal Event Dynamics (TED) Predictor - Overview

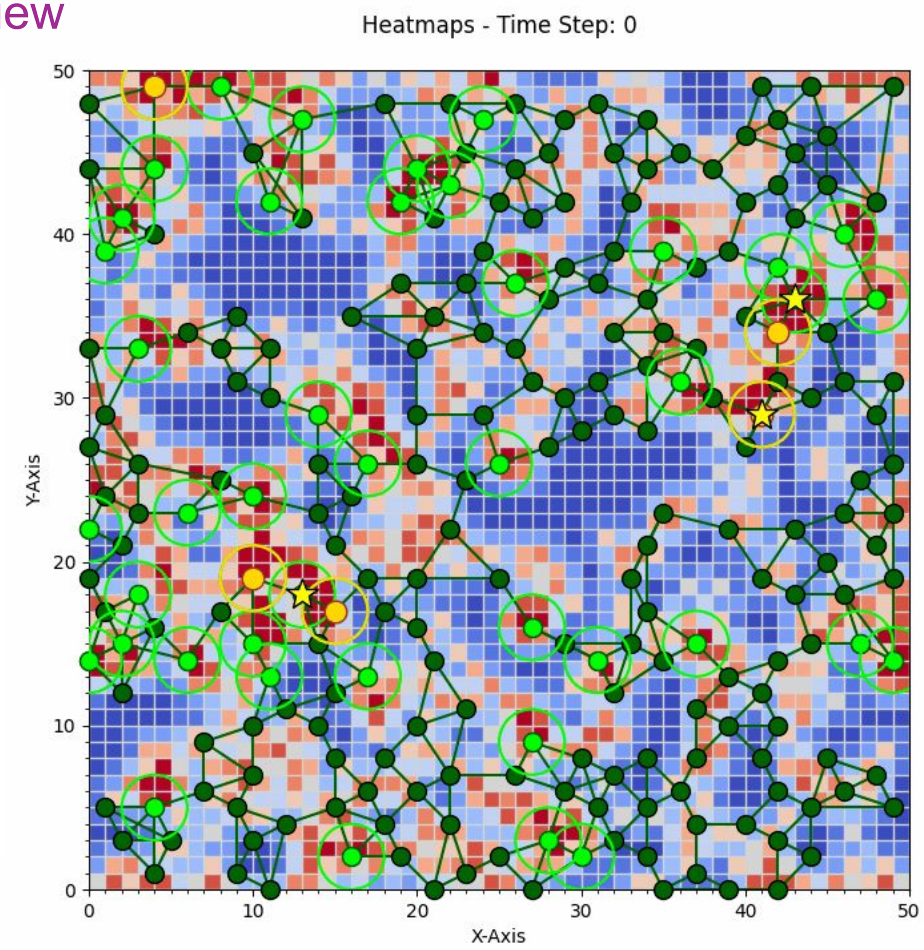




4 — WAITR Planner — *Full Cycle:* Phase 3 - WAITR Planner

Weighted Aggregate Inter-Temporal Reward (WAITR) Planner - Overview



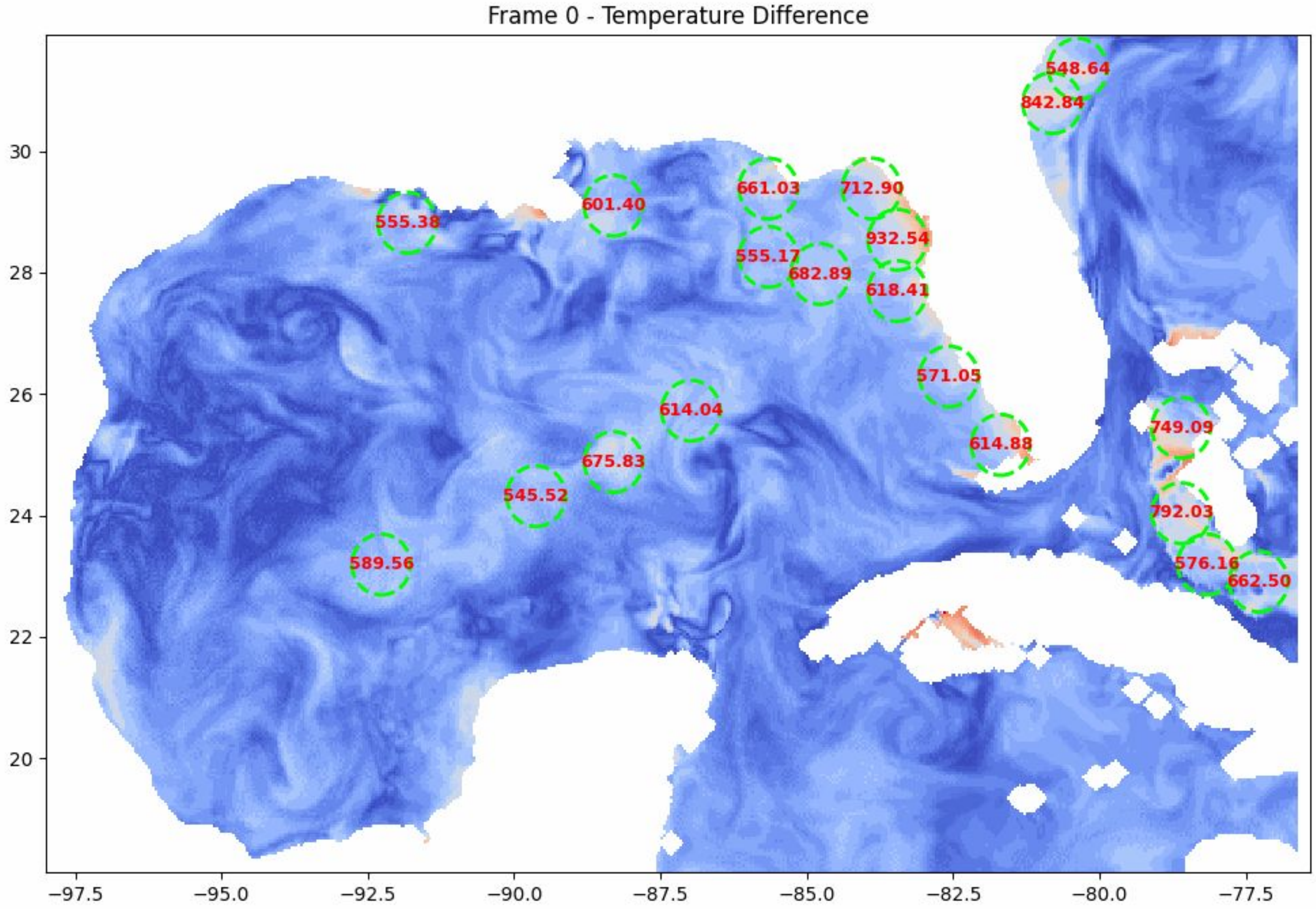


Operational Dynamics:

- **Initial Condition:** All nodes initially start scoreless, symbolizing a state with no detected events.
- **Activation and Scoring:** Nodes activate upon detecting events, receiving scores that become part of the pathway calculations.
- **Score Retention & Update:** Active nodes maintain their scores, with potential updates from new events or better paths via neighbors.
- **Efficient Computation by Pruning:** Inactive nodes—those without new events —stay dormant in subsequent processes, focusing efforts on change-prone areas.

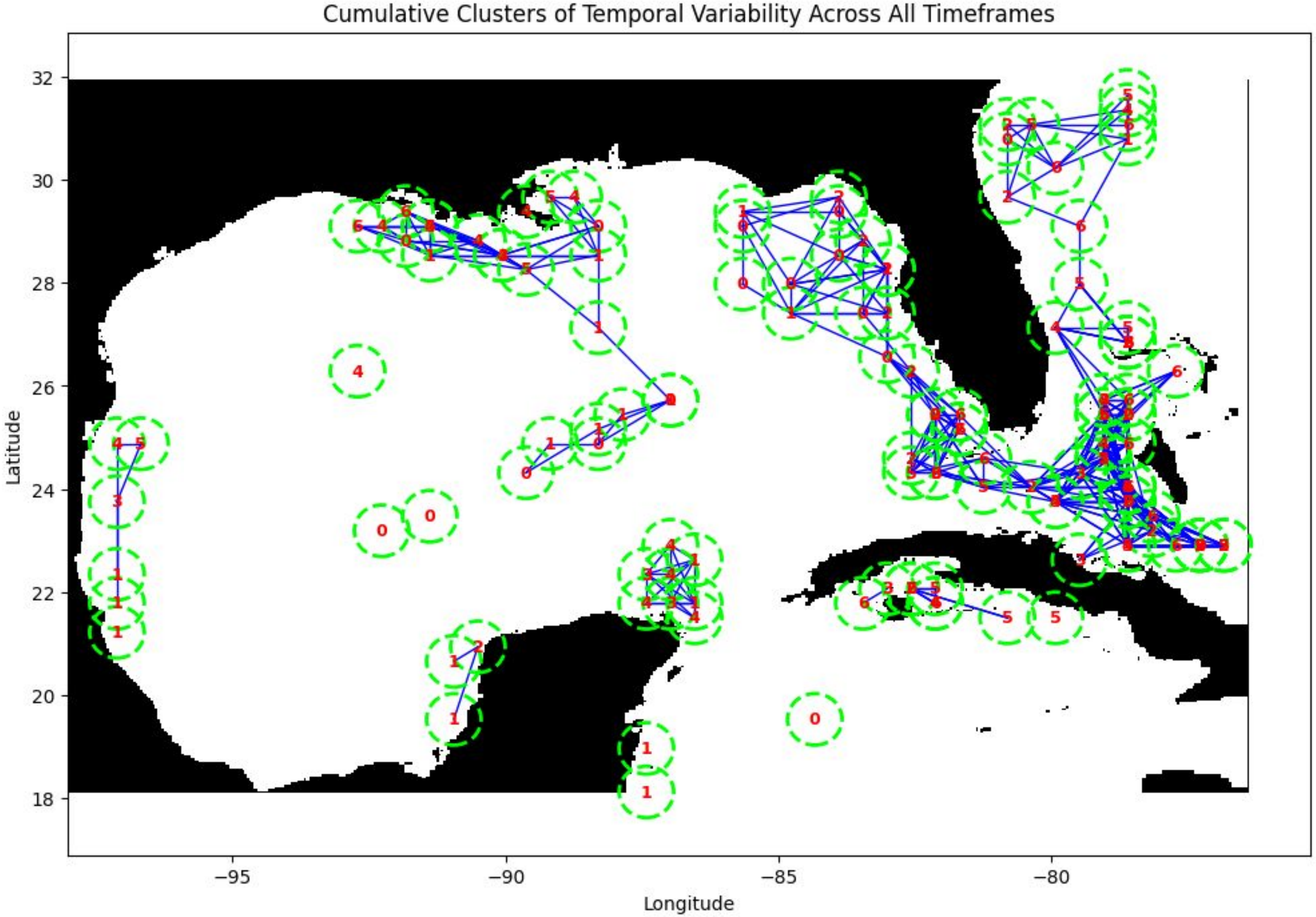
Case Study: Multiagent Planning

Weighted Proximal Recurrence Clustering



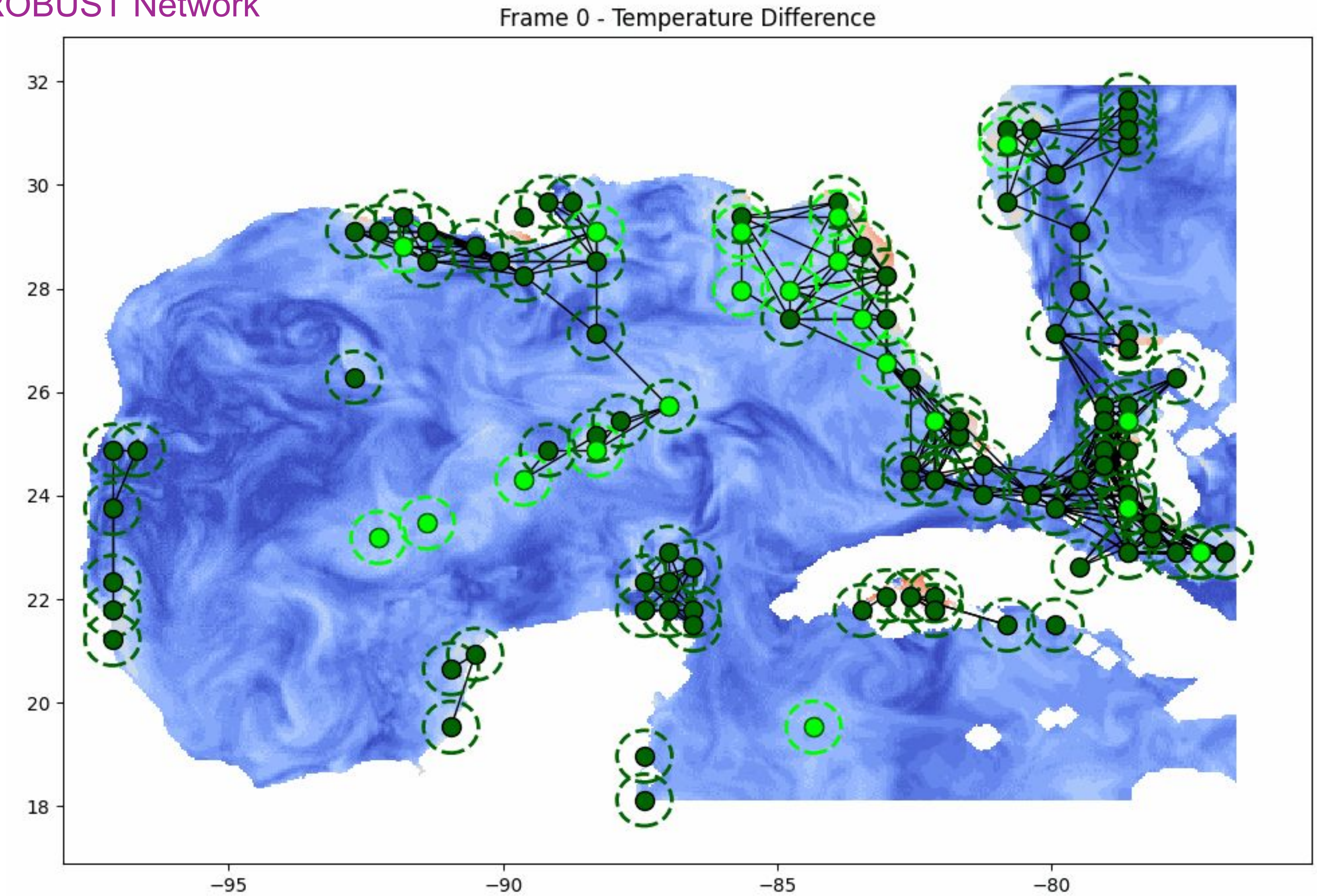
Case Study: Multiagent Planning - Methods

ROBUST Network



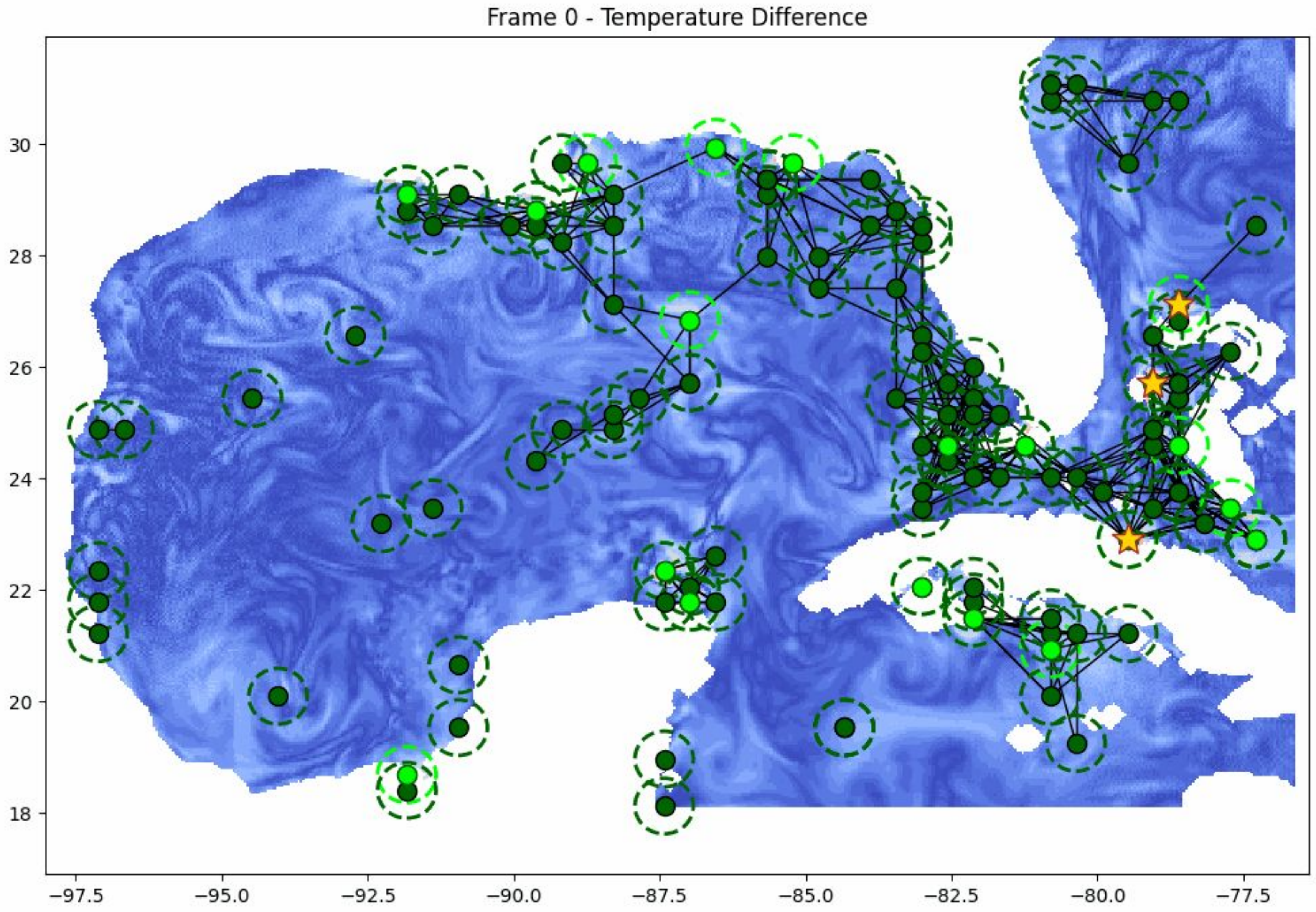
Case Study: Multiagent Planning - Methods

TED Predictor in ROBUST Network



Case Study: Multiagent Planning - Methods

WAITR Planner: Optimizing Sensor Paths



Case Study: Multiagent Planning - Results

Coverage of WPR Clustering Techniques

Overview:

- The Weighted Proximal Recurrence (WPR) method's capacity to cover significant events across the spatiotemporal domain is critical.
- This defines the upper bound limit we can achieve in planning strategies due to waypoint placement while limiting spatial space complexity

Key Coverage Insights:

- **Top 1 Waypoint** provides 8.8% coverage - a single sensor captures nearly a tenth of all significant events.
- **Top 5 Waypoints** enhance coverage to 33.11% - a moderate network of sensors significantly increases event detection.
- **Top 20 Waypoints** achieve a notable 72.7% coverage - showing that a well-planned network can access the vast majority of activities.

Cluster Approach	Top 1	Top 5	Top 10	Top 20
Aggregated WPR Counts	1256	4726	7308	10378
WPR% (total=14273)	8.8%	33.11%	51.2%	72.7%

Case Study: Multiagent Planning - Results

Efficiency Comparison of Multiagent Path Planning Strategies

Timestep	WAITR Planner (%)	Greedy Planner (%)
Frame 0	476	1463
Frame 1	18	116
Frame 2	1236	674
Frame 3	14	5
Frame 4	305	0
Frame 5	430	0
Frame 6	332	187
Total (10378)	**2811 (27.1%)**	**2445 (23.56%)**

5 — TED Compiler — Knowledge Graph - Mission Encoder

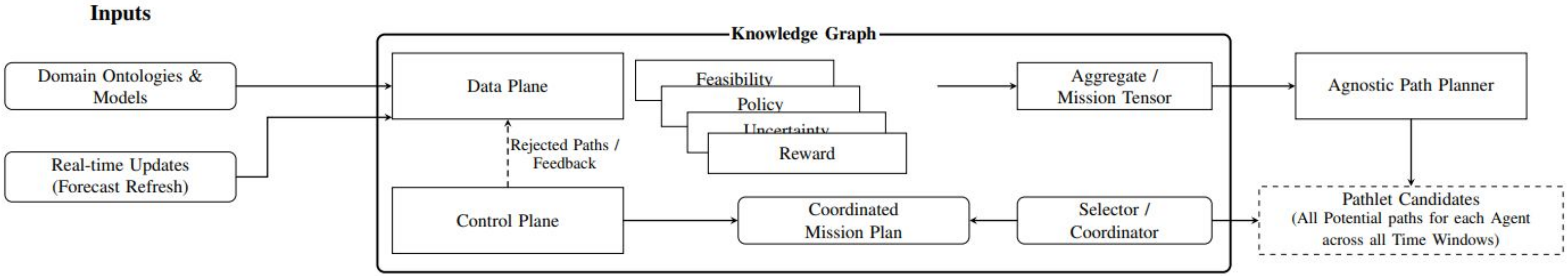


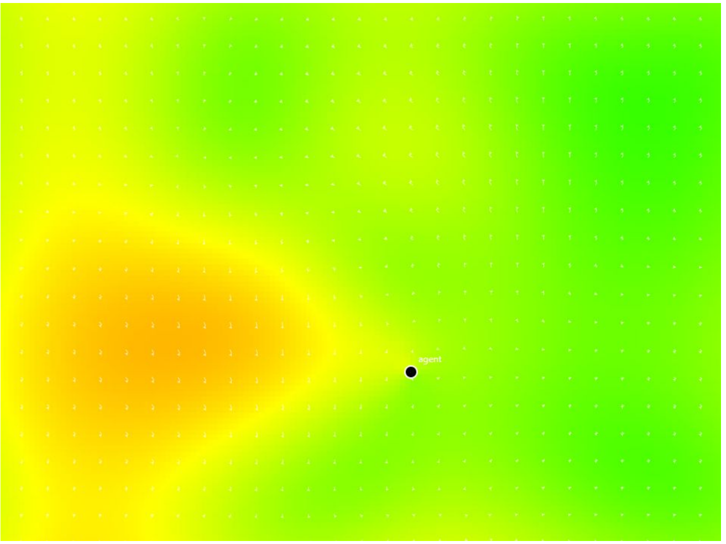
TABLE I: Core KG Schema: Representative Classes (Nouns)

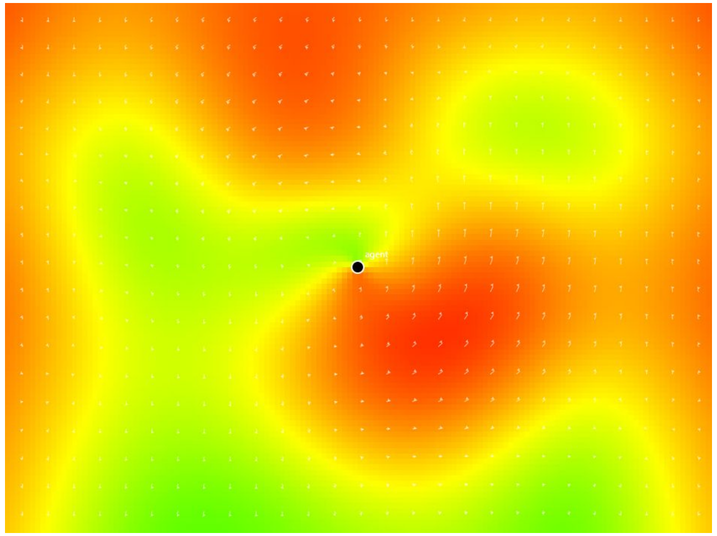
Class	Description
ex:TimeWindow	A temporal interval representing a single planning step, annotated with forecast confidence.
ex:ValueLayer	A raw or derived scientific data raster (e.g., SST frontness) for a specific time window [27], [28].
ex:Constraint	A spatial restriction, such as a no-go zone or a soft-penalty area, with a defined geometry.
ex:Policy	A named set of weights and rules that declaratively define an agent's behavior and objectives.
ex:Agent	An autonomous entity with a designated policy and a set of physical capabilities.
ex:Event	A discrete, high-value mission objective, such as a point of interest to be sampled.
ex:TensorArtifact	A compiled, mission-aware tensor; an output of the Data Plane (Φ_2).
ex:NavGraph	A traversable waypoint-and-edge graph for a time window, generated from a heatmap.

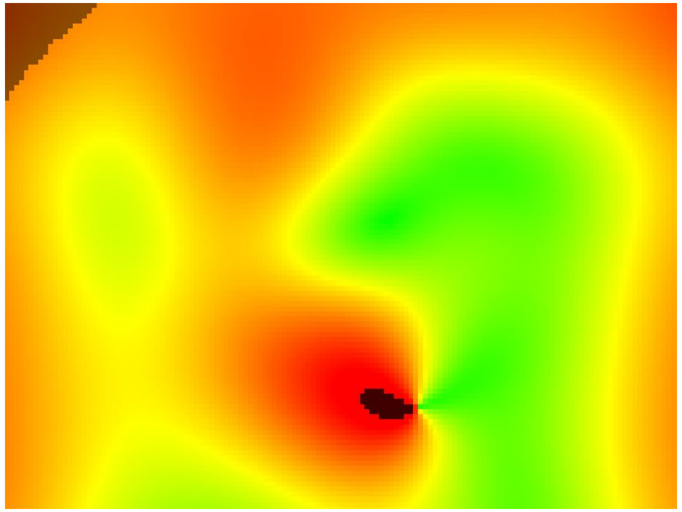
```
owl:Thing
|-- ex:MissionEntity   (mission elements)
|  |-- ex:Agent
|  |-- ex:Policy
|  |-- ex:Mission
|  +-- ex:Event
|-- ex:SpatiotemporalEntity   (entities with space/time)
|  |-- time:TemporalEntity → ex:TimeWindow
|  |-- ex:GridSpec
|  |-- ex:WorkField
|  |-- ex:ValueLayer
|  +-- ex:Constraint
|-- ex:Artifact   (compiled outputs of the KG)
|  |-- ex:TensorArtifact
|  |-- ex:NavGraph
|  |-- ex:Waypoint
|  |-- ex:TraverseEdge
|  +-- ex:EdgeCost
+-- prov:Activity   (actions: compilation/planning)
   |-- ex:CompilePhi2
   +-- ex:PlanRun
```

5 — TED Compiler — Knowledge Graph - Mission Encoder

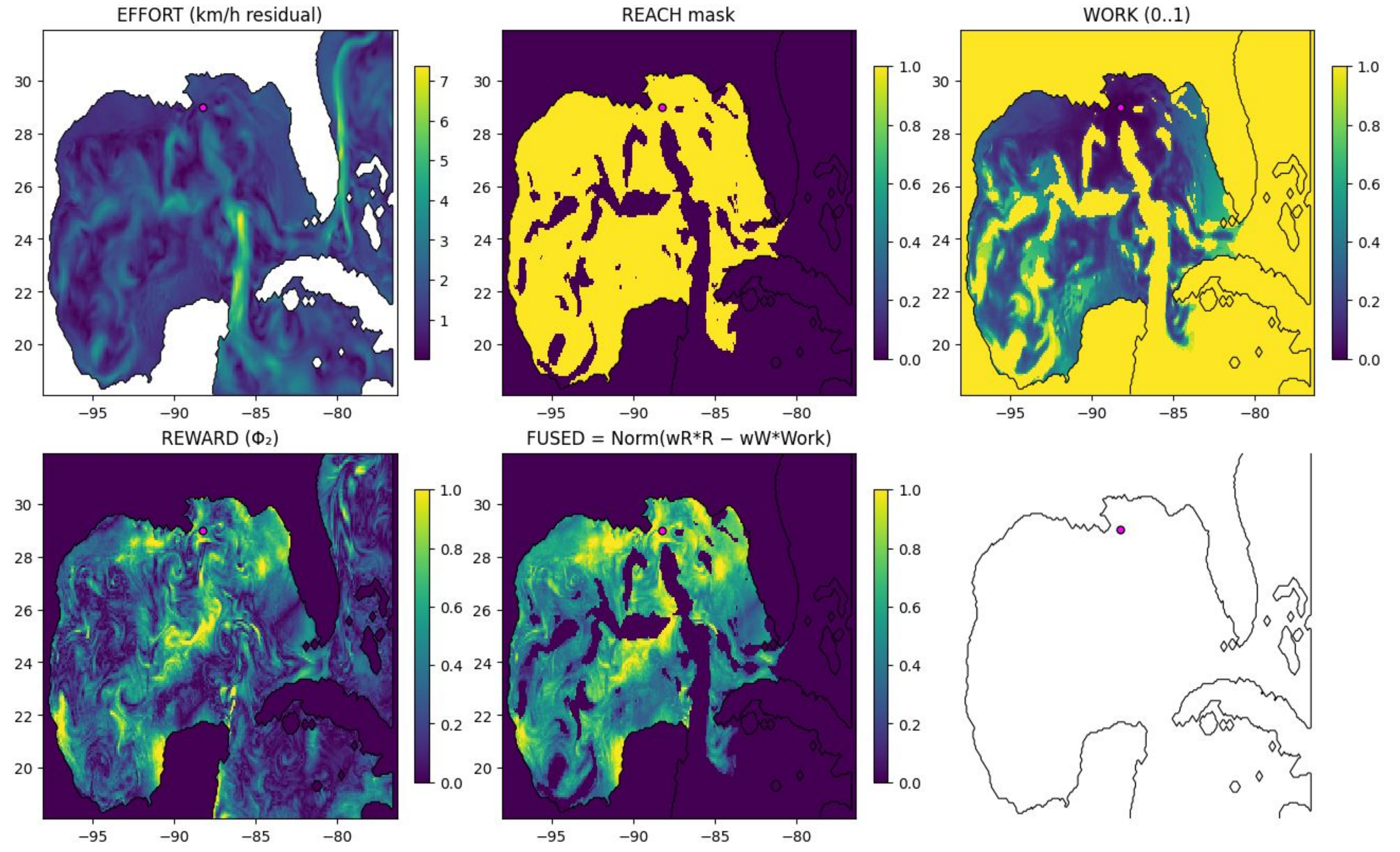
WorkTensor -- The cost of traversing the spatial domain, derived from agent location and environmental cost such as vector field



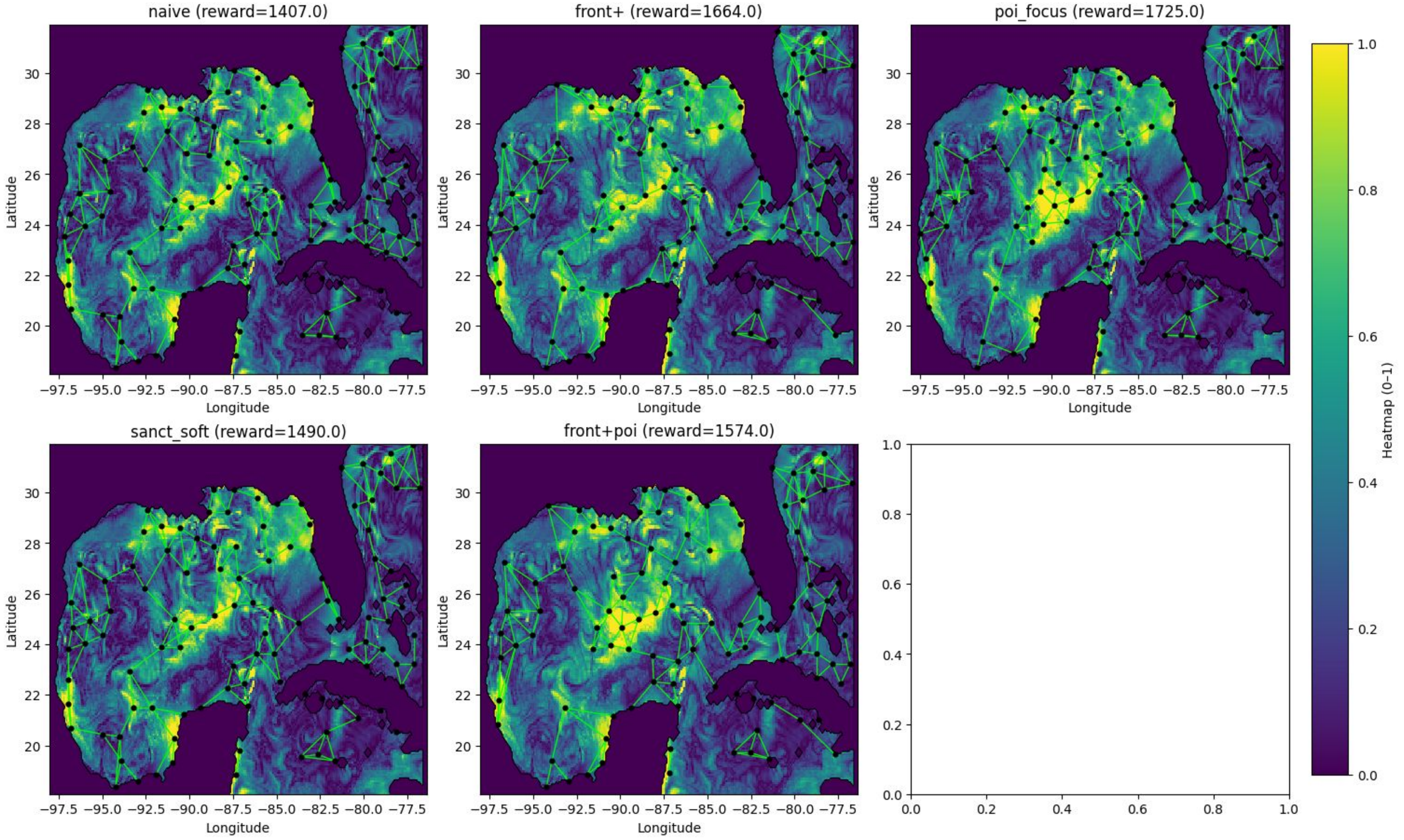




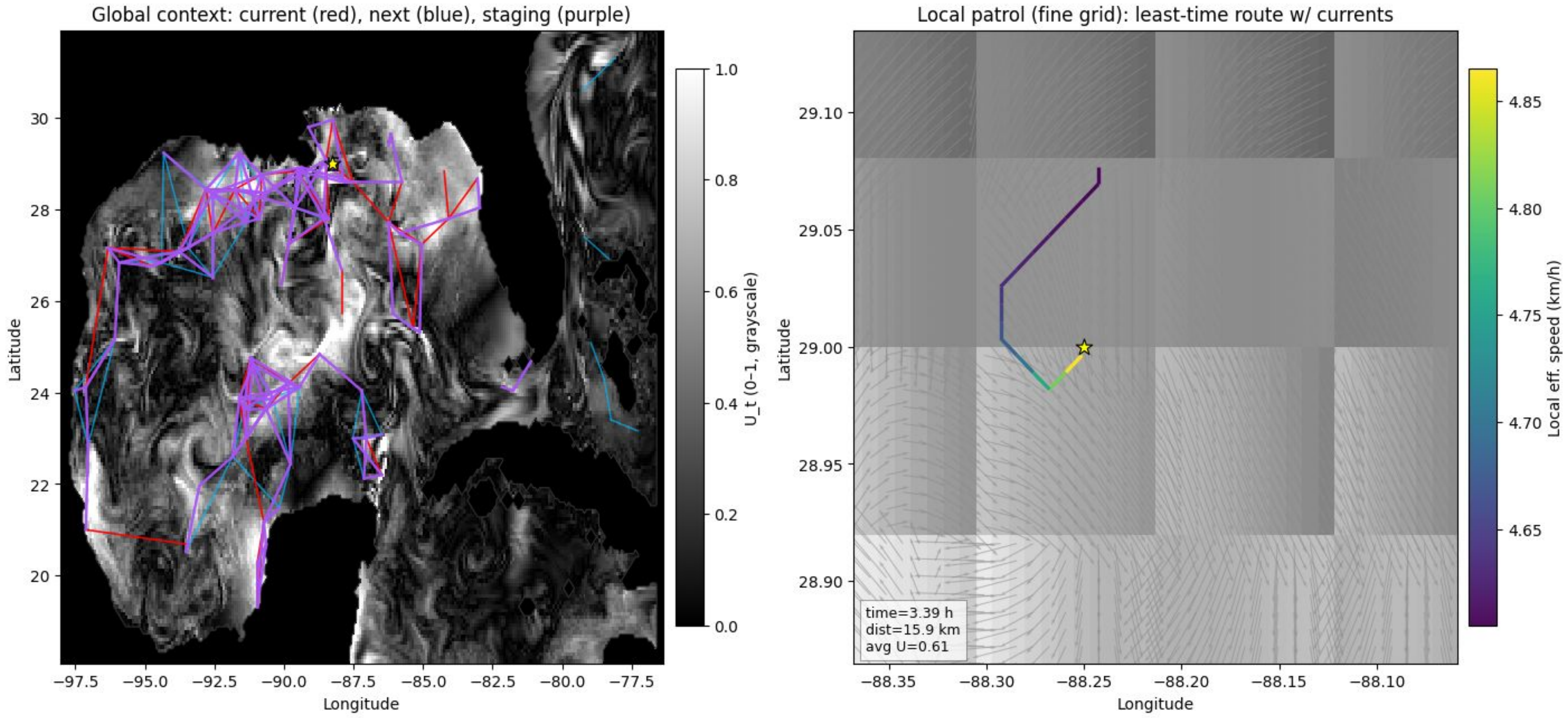
5 — TED Compiler — Knowledge Graph - Mission Encoder



5 — TED Compiler — Knowledge Graph - Mission Encoder



5 — TED Compiler — Knowledge Graph - Mission Encoder



Conclusions — What we built & why it matters

- **Separation of concerns → closed loop.**
  A planner-agnostic pipeline: **TED (compile facts) → PREP (extract candidates) → WAITR (stitch paths) → LOOP**, so we only fix what changed.

- **Representation that diagnoses before it plans.**
  ROBUST provides spatiotemporal, bipartite analytics (coverage, wiring, resilience) to decide *where/why* to add capacity—independent of fields or planners.

- **Search-space cut without losing value.**
  PREP reduces pixels/POIs to **K** high-yield candidates per window, giving a feasible **NavGraph** for mobile or placements for stationary cases.

- **Fast, stable routes across time.**
  WAITR caches local routes (pathlets), prices hand-offs with **seams**, and uses a DP sweep + no-overlap to produce **long-horizon, low-churn** plans

Limitations & scope (be explicit)

- **Forecast sensitivity:** Route quality depends on reward/effort estimates; we mitigate with risk-aware seams but full uncertainty propagation is future work.

Conclusions: Related Publications

"A Stochastic Geo-spatiotemporal Bipartite Network to Optimize GCOOS Sensor Placement Strategies"
2nd Workshop on Knowledge Graphs and Big Data, In Conjunction with IEEE Big Data 2022

"STROOBnet Optimization via GPU-Accelerated Proximal Recurrence Strategies"
3rd Workshop on Knowledge Graphs and Big Data, In Conjunction with IEEE Big Data 2023

"Knowledge Graph-Based Multi-Agent Path Planning in Dynamic Environments using WAITR"
4th Workshop on Knowledge Graphs and Big Data, In Conjunction with IEEE Big Data 2024