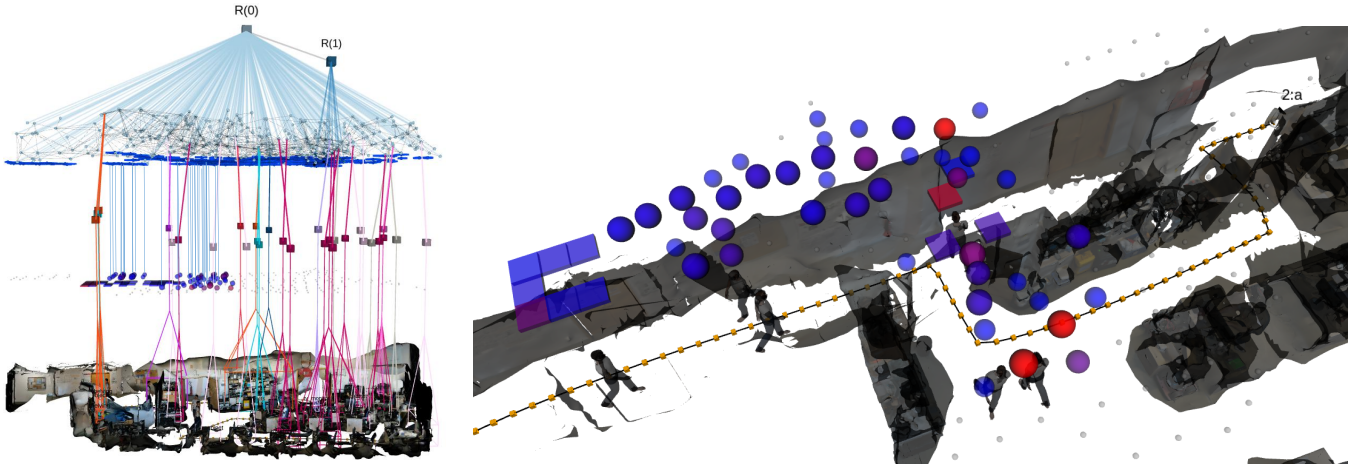
2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Aion: Towards Hierarchical 4D Scene Graphs with Temporal Flow Dynamics

Iacopo Catalano, Eduardo Montijano, Javier Civera, Julio A. Placed†, Jorge Peña-Queralta†



(a) Temporal 3D scene graph.

(b) Temporal flow dynamics layer.

Fig. 1: We propose Aion, a framework for modeling temporal flow dynamics within hierarchical spatial representations by integrating frequency-domain temporal flow modeling. The system transforms static 3D Scene Graphs (a) into 4D spatio-temporal representations, where navigational nodes are augmented with temporal flow predictions (b). Flow descriptors are visualized as spheres, where size encodes flow magnitude and color encodes directional entropy (blue: low entropy, red: high entropy). Flat square represent hash cells containing unbound dynamics.

***Abstract*— Autonomous navigation in dynamic environments requires spatial representations that capture both semantic structure and temporal evolution. 3D Scene Graphs (3DSGs) provide hierarchical multi-resolution abstractions that encode geometry and semantics, but existing extensions toward dynamics largely focus on individual objects or agents. In parallel, Maps of Dynamics (MoDs) model typical motion patterns and temporal regularities, yet are usually tied to grid-based discretizations that lack semantic awareness and do not scale well to large environments. In this paper we introduce Aion, a framework that embeds *temporal flow dynamics* directly within a hierarchical 3DSG, effectively incorporating the temporal dimension. Aion employs a graph-based sparse MoD representation to capture motion flows over arbitrary time intervals and attaches them to navigational nodes in the scene graph, yielding more interpretable and scalable predictions that improve planning and interaction in complex dynamic environments.**

I. Introduction

Autonomous robots operating in human-populated environments must navigate complex dynamic scenes where understanding spatial structure alone is insufficient for safe and efficient operation [1]. Anticipating human movements and environmental changes is critical in virtually any operating environment, not only to avoid collisions but also

† Equal Project Management.

This work was partially supported by the Kaute Foundation through the Tutkijat Maailmalle program, by DGA_FSE T73_23R and by project UNDERAIBOT (CPP2022-009792) funded by MICIU/AEI/10.13039/501100011033 and European Union (NextGenerationEU/PRTR).

Iacopo Catalano is with the University of Turku, 20014, Turku, Finland. (e-mail: imcata@utu.fi).

Jorge Peña-Queralta is with the Centre for Artificial Intelligence, Zürich University of Applied Sciences, Winterthur, Switzerland. (e-mail: penq@zhaw.ch).

Julio A. Placed is with the Instituto Tecnológico de Aragón (ITA) and the University of Zaragoza, María de Luna 3-7, Zaragoza, Spain (e-mail: jplaced@ita.es).

Javier Civera and Eduardo Montijano are with the University of Zaragoza, 50018, Zaragoza, Spain. (e-mail: jcivera@unizar.es, emonti@unizar.es).

to plan proactively accounting for past history or predicted motions [2].

Recently, 3D Scene Graphs (3DSG) have emerged as powerful abstractions for robotic spatial understanding, encoding hierarchical semantic structures that capture both geometric and semantic relationships in environments [3]. These layered graphs encode geometric, semantic, and topological information at multiple levels of abstraction (*e.g.,* low-level mesh geometry, high-level room semantics) providing a natural discretization of space into semantically meaningful locations. However, existing 3DSGs are fundamentally static, modeling spatial structure at a time instant and lacking the ability to capture and predict temporal flow dynamics, critical for autonomous systems operating in populated, changing environments. A number of works have introduced dynamics by modeling the temporal position of movement of either objects or agents (*e.g.,* human actors) [4], [5]. These approaches, while valuable in many environments, are not applicable to global dynamics.

In the domain of temporal flow modeling, recent research on Maps of Dynamics (MoDs) [6] introduced models encoding spatiotemporal motion patterns, enabling robots to predict and reason about future environment states. Prior MoD approaches have predominantly focused on embedding dynamics in uniform grid-based occupancy maps, applying spectral and probabilistic methods to learn temporal patterns [7], [8], [9]. While effective, these grid-based models lack semantic understanding, enforce uniform spatial discretization regardless of place importance, and cannot leverage the hierarchical spatial abstractions that humans naturally use to navigate complex environments. Finally, these approaches do not scale well with the size of the environment, as hierarchical representations do.

This paper introduces **Aion**, a system that integrates temporal flow dynamics directly into hierarchical 3D scene graphs, bridging the gap between rich spatial-semantic, scalable representations and predictive temporal flow modeling. Unlike prior MoD approaches that focus on grid cells, Aion learns and embeds temporal patterns at the navigational level by leveraging the natural hierarchical discretization of 3DSGs. This enables robots to generate more interpretable and actionable predictions of how specific parts of the environment evolve over time, improving navigation and interaction in dynamic human environments. Aion represents the first 3D extension of MoDs within a semantically informed scene graph framework. Through the integration of temporal flow dynamics, we bridge the gap toward a Hierarchical 4D Scene Graph.

To deliver the above core contributions of Aion, this paper introduces:

- **Graph-based Maps of Dynamics:** We extend the concept of MoDs from uniform grid-based to semantically-informed hierarchical spatial representations, enabling temporal reasoning at meaningful navigation locations rather than arbitrary spatial discretizations.
- **Dynamic Topology Temporal Modeling**: We solve the fundamental challenge of maintaining temporal model consistency over dynamic scene graph topologies through position-invariant indexing, enabling seamless temporal learning as spatial understanding evolves during exploration (*e.g.,* in the event of loop closures).
- **3DSG Integration and Flow Prediction:** Our approach enables temporal flow prediction directly for navigational nodes (through integration in Hydra [10]) rather than arbitrary spatial grid cells, providing more interpretable and actionable temporal information for navigation planning.

II. Related Work

A. 3D Scene Graphs (3DSGs)

Hierarchical 3DSGs are structured representations that integrate geometric and semantic information to support spatial understanding in robotic systems [3]. By modeling environments as layered graphs 3DSGs enable reasoning and decision-making across multiple levels of abstraction (from low-level geometry to semantically meaningful entities such as objects, rooms, and buildings) enabling reasoning across multiple spatial and semantic scales [11], [12], [13]. This hierarchical structure allows robots to efficiently interpret, plan, and act in complex environments by leveraging compact semantic concepts instead of dense geometric data [12], [14]. Extensions to urban [5], [15] and agricultural [16] domains further demonstrate the adaptability of these models to domain-specific structures.

While prior work captures static or incrementally updated structure, it largely assumes a fixed spatial topology and lacks models of temporal evolution or dynamic state within the 3DSG structure itself. Action-aware graphs [17], [18], [19] introduce affordances into the hierarchy, but do not model how environments evolve over time. In contrast, learning-based 3DSGs [20], [21], [22] operate mainly on flat, object-centric graphs and focus on perception or language grounding, often without structural abstraction or dynamics.

Dynamic Scene Graphs: Specific works have extended 3DSGs to handle dynamic environments. Rosinol *et al.* [4], [23] introduce 3D Dynamic Scene Graphs to jointly represent geometry, objects and agents, capturing dynamic entities alongside static structure. Similarly, CURB-SG [5], [24] incorporates dynamic vehicles in its structure for urban mapping.

These methods demonstrate the value of dynamic scene graphs, however, they primarily focus on modeling agent trajectories, object motion, or scene evolution at the instance level. Such approaches do not scale well when capturing object flows (*e.g.,* how crowds navigate complex indoor environments such as airports or college campuses). In contrast, our work directly integrates *temporal flow dynamics* into the hierarchical structure of 3DSGs, transforming them for the first time into 4DSGs. This enables predictive reasoning about how people collectively move through environments over time, going beyond the dynamics of individual objects or agents.

B. Maps of Dynamics (MoDs) and Temporal Modeling

MoDs aim to capture typical patterns of motion in space, enabling robots to anticipate how agents or objects move through an environment [6]. Existing approaches fundamentally differ in what types of input data they require, and how they model spatial understanding. Early models [25], [26] use probabilistic frameworks to represent local transitions and directional flows between spatial regions, encoding movement patterns through structured dependencies. Subsequent extensions generalize these concepts by associating multimodal velocity distributions with discrete spatial locations, supporting richer motion representation based on long-term sensor data [8], [27]. While effective for short-term flow estimation and reactive planning, these approaches often rely on grid-based abstractions and are limited in modeling long-term temporal variations.

To address long-term dynamics, other frameworks model environments as time-varying probabilistic processes that capture periodicities in agent behaviors or environmental states, such as daily or seasonal cycles [28], [9], [29], [7]. These models facilitate predictive reasoning beyond static maps by learning temporal regularities. Recent efforts [30], [31] further extend this concept by learning how spatial motion patterns themselves evolve over time, or by incorporating static environmental geometry to inform dynamic predictions [32]. Despite their advantages, existing MoDs typically operate independently of higher-level semantic and structural information. In contrast, our approach integrates temporal flow dynamics directly within a hierarchical 3DSG, enabling structured reasoning and predictive planning over dynamic environments.

III. Method

Aion addresses the core challenge of integrating temporal flow dynamics within hierarchical 3D scene graphs, yielding a 4D spatio-temporal representation that augments 3DSGs with time-dependent behavioral information, enabling consistent temporal state representation over dynamic graph structures.

A. 3D Scene Graph Structure

A 3DSG is a structure used to represent entities in a 3D environment along with their spatial and semantic relationships. At a given time step t, the scene is modeled as the graph $\mathcal{G}^t \triangleq (\mathcal{V}^t, \mathcal{E}^t)$, where nodes $\mathcal{V}^t$ correspond to physical or conceptual elements, and edges $\mathcal{E}^t$ encode their relational structure. If organized hierarchically, the node set $\mathcal{V}^t$ is partitioned into disjoint subsets $\mathcal{V}^t = \bigsqcup_{\ell=1}^{L} \mathcal{V}_\ell^t$ corresponding to different abstraction levels $\ell \in \{1, \dots, L\}$, allowing the graph to encode both fine-grained geometry and high-level semantic groupings. Nodes are grouped in this case across levels of abstraction as $\boldsymbol{v}_{\ell,i}^t \in \mathcal{V}_\ell^t$ [3]. Each layer in this hierarchy serves a distinct purpose (see Fig. 2):

i) Low-Level geometry: Captures the physical layout of the environment, *e.g.,* through meshes or point clouds.

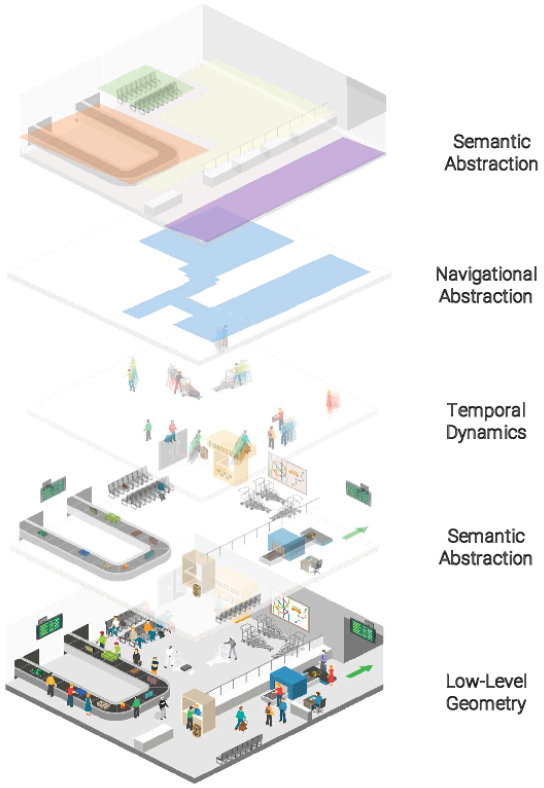


Fig. 2: Illustration of the hierarchical decomposition of a scene into different geometric, semantic and navigational layers, where temporal flow dynamics emerge from semantic structure and play a central role to enriching navigation.

ii) Motion graph/Spatial anchoring: Anchors observations in space and time, often encoding motion, agent trajectories, or dynamic entities.
iii) Navigational abstractions and action affordances: Represents traversable regions and their connectivity to support planning and action.
iv) Semantic abstractions: Groups entities into meaningful categories (*e.g.,* objects) for higher-level understanding.
v) Global structure: Integrates lower-level representations into a unified model of large-scale environments.

This layered structure allows 3DSGs to support both low-level geometric reasoning and high-level semantic interpretation, enabling robust scene understanding across a range of tasks in robotics and embodied AI.

Our approach leverages the inherent hierarchy of the scene graph to model how activity patterns evolve over time, extending this framework to incorporate *temporal flow dynamics* at the navigational level by adding an additional layer of abstraction to the conventional 3DSG scene decomposition (see Fig. 2).

This temporal layer augments navigational nodes with directional flow statistics and predictive models, enabling the graph to capture not only where motion occurs but also how it changes over time. It provides the foundation for the

methods introduced in the following sections, including our spatio-temporal modeling, sparse spatial hashing, and global temporal prediction mechanisms.

B. Spatio-Temporal Modeling

To model directional dynamics over time, Aion maintains sparse per-location orientation histograms, inspired by the grid-based motion modeling approach of [9] but extended to operate over navigational nodes in a sparse spatial hash. For notational simplicity, let us denote by $\boldsymbol{v}_{n,i}^t \in \mathcal{V}_n^t$ the i-th node that belongs to the navigational abstraction layer $\mathcal{V}_n^t$ at time t. Each node maintains a time-varying activity vector representing directional motion, discretized into B angular bins over $[0, 2\pi)$. Observed orientations are mapped to bins and incrementally accumulated to form a historical activity profile per node. Formally, each navigational node $v_{n,i}^t$ is associated with a temporal activity vector:

$$\boldsymbol{s}_i(t) \in \mathbb{R}^{B\times\lambda}, \tag{1}$$

where the b-th entry $\boldsymbol{s}_{i,b}(t) \in \mathbb{R}^{\lambda}$ denotes the observed activity level in direction bin b at time t, and λ is the dimension of the motion descriptor. The vector $\boldsymbol{s}_i(t)$ represents motion directions in the global coordinate frame, ensuring that the activity history remains spatially meaningful even if nodes are repositioned. The motion descriptors, such as flow magnitude, dominant direction, and directional entropy are computed from raw historical counts. These reflect empirical, observed motion patterns. Periodically, the histograms are normalized and passed to a global FreMEn model [28], which captures temporal periodicity and enables prediction of future motion trends at each node. Predicted vectors retain the same angular structure but represent model-inferred directional likelihoods rather than raw counts.

C. Sparse Spatial Hashing for Scalable Temporal Modeling

One of the key challenges is maintaining temporal model consistency as the set of navigational nodes evolves. Traditional grid-based temporal modeling approaches allocate memory for every spatial cell within predefined boundaries [9], leading to computational overhead when applied to realistic environments where activity is spatially sparse. To overcome this limitation, Aion adopts a sparse spatial hashing mechanism (see Fig. 3) that enables infinite spatial coverage while maintaining $\mathcal{O}(1)$ lookup performance and minimal memory footprint:

$$h(\boldsymbol{p}) = \text{hash}\left(\left\lfloor\frac{x}{\delta}\right\rfloor, \left\lfloor\frac{y}{\delta}\right\rfloor, \left\lfloor\frac{z}{\delta}\right\rfloor\right), \tag{2}$$

where $\lfloor\cdot\rfloor$ denotes the element-wise floor operation applied to each coordinate of the position vector $\boldsymbol{p} = [x, y, z] \in \mathbb{R}^3$, and δ is the spatial resolution parameter. The spatial grid is used as a pure mathematical coordinate system where memory allocation occurs only upon data storage. This enables:

- Spatial coverage: Any 3D position can be mapped to a cell without boundary constraints. The model retains historical data even when the state space expands without the risk of reinitialization of temporal flow dynamics.
- Memory efficiency: Storage requirements are proportional to the visited locations, rather than the map bounds.
- Dynamic scalability: The representation adapts seamlessly to environments of arbitrary size.

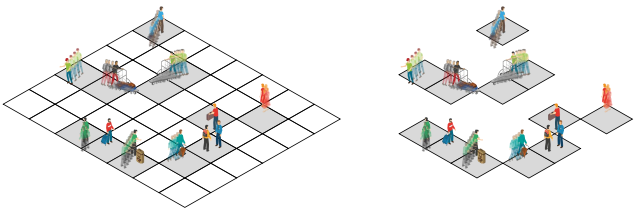
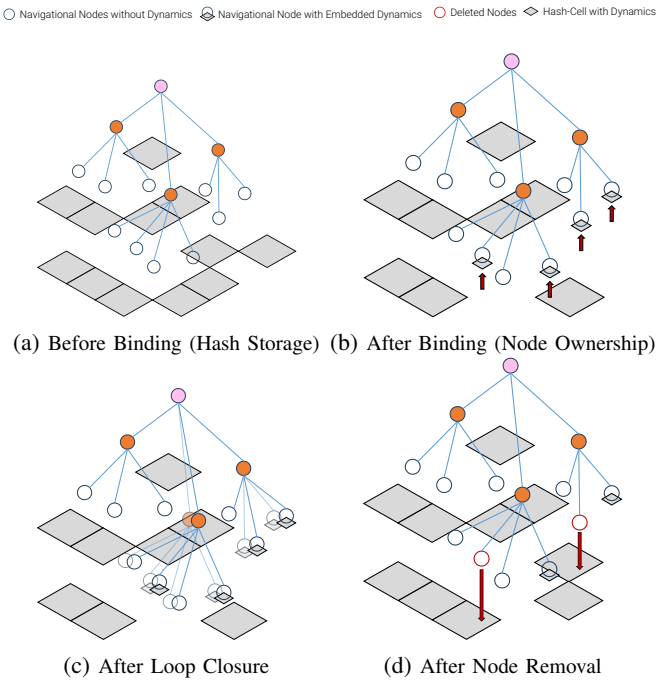


Fig. 3: Comparison between grid-based models (left) and the proposed sparse spatial hashing of Aion (right) to encode temporal flow dynamics. While the grid cells represent the same spatial locations, only occupied cells are stored in memory in the case of Aion.



(a) Before Binding (Hash Storage) (b) After Binding (Node Ownership)

(c) After Loop Closure (d) After Node Removal

Fig. 4: Illustration of temporal ownership transfer. (a) Temporal flow dynamics are first accumulated in spatial hash cells. (b) After a stability window, dynamics are bound to the nearest navigational node, which becomes the sole owner of the temporal history. (c) Following loop closure, nodes can be repositioned and with them the temporal flow dynamics. (d) If a node is removed during loop closure, the dynamics are restored into hash space and remain available for reassociation.

D. Global Temporal Model Architecture

To support temporal reasoning over dynamic environments, Aion represents spatiotemporal flow dynamics directly on top of a hierarchical 3DSG. Each node in the graph corresponds to a navigational node and becomes a unit of temporal prediction. This allows the system to forecast

human activity or motion trends in specific regions of the environment.

1) Global Dynamics Model: Rather than maintaining individual temporal models per node, Aion employs a single global temporal model that captures inter-node temporal dependencies while maintaining computational efficiency.

Given a dynamic entity located at position $\boldsymbol{p}_h$ with orientation θ_h, the system associates it to the nearest node $\boldsymbol{v}^* \in \mathcal{V}_n^t$ in the navigational abstraction layer, *i.e.,*:

$$\boldsymbol{v}^* = \underset{\boldsymbol{v}_{n,i}^t \in \mathcal{V}_n^t}{\arg\min} \|\boldsymbol{x}_{n,i}^t - \boldsymbol{p}_h\|_2 \,, \quad \text{s.t.} \quad \|\boldsymbol{x}_{n,i}^t - \boldsymbol{p}_h\|_2 \leq d_{\max} \,, \tag{3}$$

where $\boldsymbol{x}_{n,i}^t$ is the node 3D position and $d_{\max}$ is a distance threshold. The orientation discretization process maps continuous orientation to discrete bins:

$$\text{bin}(\theta_h) = \left\lfloor \frac{\theta_h + \pi}{2\pi/B} \right\rfloor \bmod B \,, \tag{4}$$

with B the number of orientation bins b (typically 8, providing 45° resolution).

In addition to per-node vectors, temporal activity across all spatial locations is tracked using a global vector defined over sparse hash keys:

$$\boldsymbol{s}(t) = [\boldsymbol{s}_{h_1}(t), \boldsymbol{s}_{h_2}(t), \ldots, \boldsymbol{s}_{h_N}(t)] \,, \tag{5}$$

where $h_1, h_2, \ldots, h_N$ are spatial hash keys ordered deterministically.

This representation offers several advantages over grid-based MoDs [6]. First, navigational nodes correspond to meaningful locations rather than arbitrary spatial cells and their density adapts to environmental structure rather than uniform discretization. Second, temporal patterns can be aggregated across the scene graph hierarchy with fewer spatial units compared to fine-grained grid representations.

2) Long-term Consistency through Temporal Ownership Transfer: Loop closure in graph-based mapping corrects accumulated drift by repositioning or removing navigational nodes. If temporal flow dynamics are stored only in spatial hash cells, such corrections can leave dynamics *stranded* at outdated coordinates or duplicated across nodes, breaking temporal consistency.

To avoid this issue, Aion introduces a mechanism of temporal ownership transfer, in which temporal history is moved from spatial hash cells to navigational nodes once the pose graph has stabilized. We define the binding function

$$\phi : h_i \mapsto \boldsymbol{v}_{n,i} \,, \tag{6}$$

which maps a spatial hash cell c_{h_i} at key h_i to its owning navigational node $\boldsymbol{v}_{n,i}$.

During exploration, temporal flow dynamics are first stored in hash space (Fig. 4a). After a stability window τ (a fixed time or an exploration horizon after which the graph structure is assumed to have converged, *e.g.,*no major pose graph updates or loop closures are expected), the temporal

Fig. 5: Virtual environment generated from real-world data and simulated agents.

state vector $\boldsymbol{s}_{h_i}(t)$ associated with cell c_{h_i} is transferred to the nearest node $v_{n,i}$ (Fig. 4b):

$$\boldsymbol{s}_i(t) \leftarrow \boldsymbol{s}_{h_i}(t), \quad \phi(h_i) = \boldsymbol{v}_{n,i} \,, \tag{7}$$

and the data associated with hash cell c_{h_i} is removed from the hash map. A lightweight redirect table maintains the mapping $\phi(c_{h_i})$, ensuring that any subsequent updates arriving at the coordinates of h_i are routed to node $v_{n,i}$. If the node is repositioned due to loop closure, its dynamics move with it, since $\boldsymbol{s}_i(t)$ is stored directly in the node (Fig. 4c). If the node is removed, ownership is released:

$$\boldsymbol{s}_{h_i}(t) \leftarrow \boldsymbol{s}_i(t), \quad \phi(h_i) = \varnothing \,, \tag{8}$$

so that the temporal history is rematerialized in hash space at the node's final pose, ready to be reassociated with a new node when one appears nearby (Fig. 4d).

This move-semantics approach prevents duplication of temporal histories, ensures memory efficiency, and preserves consistency under graph corrections. Hash cells act as provisional holders, while navigational nodes become the long-term owners of temporal flow dynamics once stable.

E. Real-time Integration Architecture

Aion integrates seamlessly with existing 3DSG systems, specifically Hydra [10], through an architecture designed to preserve real-time performance without altering core scene graph processing. This is achieved through:

- **Asynchronous Processing**: Scene graph updates and agent detections are handled asynchronously, ensuring that temporal modeling does not block real-time 3DSG construction. A dedicated temporal modeling thread accumulates observations and periodically updates both the global FreMEn and historical models.
- **Efficient Memory Management**: A sparse spatial hashing scheme bounds memory growth in proportion to environment coverage rather than overall map size, critical for long-term autonomous operation.
- **Service Interface**: Temporal predictions are exposed via ROS services, enabling integration with existing navigation and planning systems without requiring architectural changes to client systems. Aion extends

Hydra's capabilities by adding a parallel navigational layer that enriches the existing scene graph, providing additional temporal context.

IV. Experimental Evaluation

A. Experimental Setup

We evaluate the effectiveness of Aion in capturing and predicting motion dynamics in indoor environments. Owing to the lack of directly comparable methods using 3DSGs for spatio-temporal dynamics modeling, we define a structured comparison against a grid-based approach (based on [9]), which serves as the reference baseline in our analysis.

1) Dataset and Environment: We developed a synthetic dataset derived from a virtual reconstruction of our university office environment, comprising a central workspace and a connected hallway (Fig. 5).The dataset consists of 20 distinct scenes, each featuring 6 human agents navigating the environment along predefined routes. In each scenario, a mobile robot traverses the environment following specified paths. We use the simulator-provided detections to compute the spatio-temporal model discussed in Section III.

2) Comparison Framework: Our method employs a sparse graph-based hybrid representation, where spatial locations are encoded as nodes in a 3DSG and hash cells. Given the mismatch in representation granularity and topology, a direct comparison is non-trivial. To enable a fair evaluation, we propose a two-stage comparison procedure. First, we construct a reference grid-based temporal model, which aggregates movement patterns into fixed-size spatial cells. Second, we discretize the output of our method into the same grid resolution, allowing for a direct comparison of the modeled dynamics.

We define two categories of evaluation, focusing on both observed and predicted spatio-temporal dynamics:

- **Historical Dynamics**: This evaluation assesses how well the method captures aggregate movement patterns from past observations.
- **Temporal Predictions**: In this setting, we evaluate the ability of the method to capture cyclical dynamics pattern using a learned temporal model. Specifically, we use the Fremen temporal model [28] in both the grid-based and graph-based representations inspired by Molina *et al.* [9].

3) Evaluation Metrics: Given the probabilistic and directional nature of the data, we employ the following metrics to compare the resulting distributions:

- **Jensen-Shannon (JS) Divergence** $(0-1)$: Measures the similarity between two probability distributions, bounded between 0 (identical) and 1 (maximally different). Suitable for general-purpose distribution comparison.
- **Bhattacharyya Distance** $(0-\infty)$: Quantifies probability mass overlap between distributions. Higher values indicate less overlap in the probability densities.
- **Wasserstein Distance**: Also known as Earth Mover's Distance, this metric reflects the cost of transforming one distribution into another. For directional data, it corresponds to the average angular displacement required.
- **Circular Correlation**: Directional correlation between two vector fields.

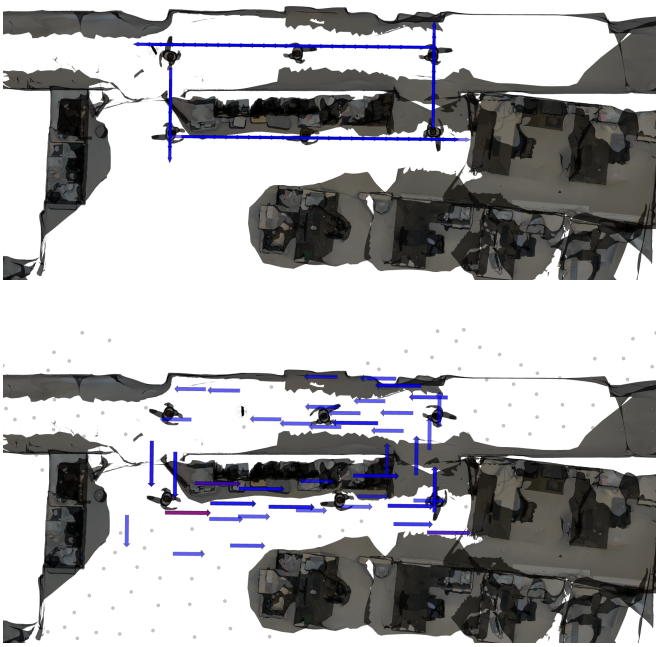


Fig. 6: Qualitative comparison of motion dynamics reconstructed for one scene with the grid-based system (top) and Aion (bottom). Arrow markers indicate the dominant direction of motion. Despite the different spatial discretization, Aion is capable of reproducing directional patterns.

B. Results and Analysis

To assess similarity between the graph-based and grid-based models, we compute metrics per scene and report the mean and standard deviation across all datasets (see table I). This allows us to capture both the central tendency and variability of each method's performance across different environments.

The results indicate that both methods capture related aspects of the underlying dynamics, though the degree of alignment varies across measures. For entropy, the models show consistent agreement in the historical case, with low Wasserstein distances (W = 0.12) suggesting that both representations identify comparable levels of variability in human activity. Predictions also show overlap (JS = 0.35), but with higher variance, suggesting that cyclical temporal models are more sensitive to how the spatial representation aggregates data. Flow magnitude distributions are less closely aligned: while historical activity levels show moderate similarity (JS = 0.53), predicted flows diverge more strongly (JS = 0.68), pointing to differences in how each representation scales activity over time.

Directional flow analysis produces the largest quantitative discrepancies, with low circular correlations (r = 0.12) and

TABLE I: Comparison between Aion and grid-based structure for different data types. The results in the table show that comparable accuracies are obtainable with Aion, with the additional benefits of scale through hierarchy as well as data structure optimizations.

Map Type	Data Type	JS Divergence	Bhattacharyya Distance	Wasserstein Distance	Circular Correlation
Entropy	Historical	0.49 ± 0.13	1.3 ± 0.68	0.12 ± 0.05	–
Entropy	Predicted	0.35 ± 0.32	3.83 ± 4.69	0.18 ± 0.14	–
Flow	Historical	0.53 ± 0.11	2.02 ± 2.06	40.17 ± 21.82	–
Flow	Predicted	0.68 ± 0.02	6.42 ± 3.76	76.16 ± 16.93	–
Direction	Historical	–	–	15 ± 11.23	0.12 ± 0.07
Direction	Predicted	–	–	103.03 ± 14.15	0.1 ± 0.05

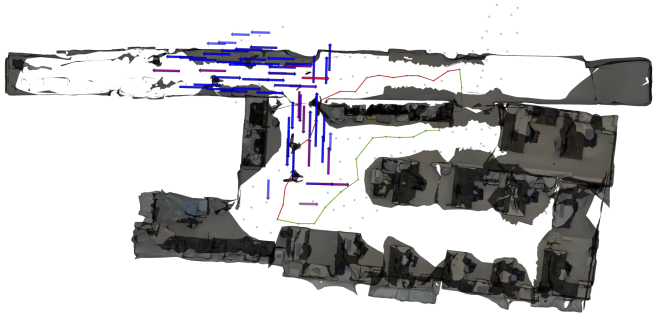


Fig. 7: Qualitative comparison of A^* planning results on the navigational layer of the 3DSG. In red: the baseline planning without temporal dynamics. In green: the computed path obtained with our proposed method, which integrates temporal motion patterns as edge costs. The resulting path demonstrates improved efficiency avoiding crowded areas.

high Wasserstein distances. However, qualitative comparisons (Fig. 6) demonstrate that directional patterns are in fact well reproduced. The gap arises primarily from differences in spatial discretization: the grid-based system evaluates orientation over fixed cells, while Aion employs an adaptive structure of nodes and hash-cells whose placement varies across runs and datasets. This flexibility allows Aion to capture meaningful motion at navigationally relevant locations, but it also introduces misalignments when distributions are forced into a grid-based evaluation framework.

C. Planning

This section illustrates how the temporal flow dynamics encoded in Aion can be used to inform navigation decisions. The goal is not to propose a new planning algorithm, but to demonstrate that the additional information provided by Aion can be integrated into standard methods such as A^*.

Each node in the navigational graph is annotated with dynamic attributes (*entropy*, *flow magnitude*, and *flow direction*) combined into a temporal cost reflecting both the variability of activity in a region and whether a planned movement aligns with the dominant flow. Edges that pass through uncertain or opposing-flow regions therefore incur a higher cost than those through stable or aligned areas. The resulting traversal cost between nodes i and j is defined as:

$$\text{cost}(i,j) = d(i,j) + (\bar{c}_t + c_d(i,j)) \cdot d(i,j)\,, \qquad (9)$$

where $d(i,j)$ is the Euclidean distance between nodes i and j, $\bar{c}_t$ represents the average temporal cost of the two nodes, and $c_d(i,j)$ is a directional penalty. This formulation extends conventional distance-based planning with dynamic penalties that encourage safer and more context-aware paths.

Results. As shown in Fig. 7, paths planned with Aion's temporal dynamics tend to avoid regions of high entropy and flow dynamics. In contrast, a purely distance-based A^* planner often selects shorter but less robust routes that cut directly through uncertain areas. This qualitative comparison highlights that Aion's representation can provide meaningful guidance for navigation in dynamic environments, even when applied within standard planning frameworks.

V. Conclusion

We introduce Aion, a system that augments hierarchical 3D scene graphs with temporal flow dynamics, providing a step toward semantically informed 4D scene representations. Our approach embeds dynamics into 3DSGs, representing the first unification of Maps of Dynamics and Scene Graphs. Aion enables temporal reasoning at navigationally meaningful locations, and maintains consistency under evolving graph topologies through a mechanism of temporal ownership transfer, ensuring that motion histories remain valid under structural updates of the graph (e.g., during optimization steps such as loop closure).

Our experimental results demonstrate that Aion captures and reproduces spatio-temporal motion patterns with good agreement to grid-based baselines, while offering a more interpretable and semantically structured representation. With this initial version, we provide a complete implementation and integration into Hydra's 3DSG framework, laying the groundwork for future research on applying 4D scene graphs in human-aware navigation, advancing long-term autonomy in highly dynamic settings.

Overall, this work lays the foundation for future research on applying 4D scene graphs in real-world environments. Future work will be directed towards larger-scale deployments, and human-aware navigation, to demonstrate the potential of Aion for long-term autonomy in highly dynamic settings.

References

[1] Fabien Grzeskowiak, David Gonon, Daniel Dugas, Diego Paez-Granados, Jen Jen Chung, Juan Nieto, Roland Siegwart, Aude Billard, Marie Babel, and Julien Pettré. Crowd against the machine: A simulation-based benchmark tool to evaluate and compare robot capabilities to navigate a human crowd. In *IEEE Int. Conf. on Robot. Autom.*, pages 3879–3885. IEEE, 2021.

[2] Diego Paez-Granados, Yujie He, David Gonon, Dan Jia, Bastian Leibe, Kenji Suzuki, and Aude Billard. Pedestrian-robot interactions on autonomous crowd navigation: Reactive control methods and evaluation metrics. In *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, pages 149–156. IEEE, 2022.

[3] Iacopo Catalano, Carlos Cueto Zumaya, Julio A Placed, Javier Civera, Wallace Moreira Bessa, and Jorge Peña-Queralta. 3d scene graphs in robotics: A unified representation bridging geometry, semantics, and action. *Authorea Preprints*, 2025.

[4] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. 3d dynamic scene graphs: Actionable spatial perception with places, objects, and humans. *arXiv:2002.06289*, 2020.

[5] Elias Greve, Martin Büchner, Niclas Vödisch, Wolfram Burgard, and Abhinav Valada. Collaborative Dynamic 3D Scene Graphs for Automated Driving. In *IEEE Int. Conf. on Robot. Autom.*, pages 11118–11124, 2024.

[6] Tomasz Piotr Kucner, Martin Magnusson, Sariah Mghames, Luigi Palmieri, Francesco Verdoja, Chittaranjan Srinivas Swaminathan, Tomáš Krajník, Erik Schaffernicht, Nicola Bellotto, Marc Hanheide, et al. Survey of maps of dynamics for mobile robots. *The Int. J. of Robotics Research*, 42(11):977–1006, 2023.

[7] Tomas Krajnik, Jaime Pulido Fentanes, Grzegorz Cielniak, Christian Dondrup, and Tom Duckett. Spectral analysis for long-term robotic mapping. In *IEEE Int. Conf. on Robot. Autom.*, pages 3706–3711, 2014.

[8] Tomasz Piotr Kucner, Martin Magnusson, Erik Schaffernicht, Victor Hernandez Bennetts, and Achim J Lilienthal. Enabling flow awareness for mobile robots in partially observable environments. *IEEE Robotics & Automation L.*, 2(2):1093–1100, 2017.

[9] Sergi Molina, Grzegorz Cielniak, Tomáš Krajník, and Tom Duckett. Modelling and predicting rhythmic flow patterns in dynamic environments. In *Towards Autonomous Robotic Systems*, pages 135–146, 2018.

[10] Nathan Hughes, Yun Chang, and Luca Carlone. Hydra: A Real-time Spatial Perception System for 3D Scene Graph Construction and Optimization. In *Robotics: Science and Systems*, 2022.

[11] Yun Chang, Nathan Hughes, Aaron Ray, and Luca Carlone. Hydra-Multi: Collaborative Online Construction of 3D Scene Graphs with Multi-Robot Teams. In *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, pages 10995–11002, 2023.

[12] Nathan Hughes, Yun Chang, Siyi Hu, Rajat Talak, Rumaia Abdulhai, Jared Strader, and Luca Carlone. Foundations of spatial perception for robotics: Hierarchical representations and real-time systems. *The Int. J. of Robotics Research*, page 02783649241229725, 2024.

[13] Hriday Bavle, Jose Luis Sanchez-Lopez, Muhammad Shaheer, Javier Civera, and Holger Voos. S-Graphs+: Real-time Localization and Mapping leveraging Hierarchical Representations. *IEEE Robotics & Automation L.*, 8(8):4927–4934, 2023.

[14] Saad Ejaz, Marco Giberna, Muhammad Shaheer, Jose Andres Millan-Romera, Ali Tourani, Paul Kremer, Holger Voos, and Jose Luis Sanchez-Lopez. Situationally-aware path planning exploiting 3d scene graphs. *arXiv:2508.06283*, 2025.

[15] Yinan Deng, Jiahui Wang, Jingyu Zhao, Xinyu Tian, Guangyan Chen, Yi Yang, and Yufeng Yue. OpenGraph: Open-Vocabulary Hierarchical 3D Graph Representation in Large-Scale Outdoor Environments. *IEEE Robotics & Automation L.*, 2024.

[16] Adam Mukuddem and Paul Amayo. Osiris: Building Hierarchical Representations for Agricultural Environments. In *IEEE Int. Conf. on Robot. Autom.*, pages 15797–15803, 2024.

[17] Zachary Ravichandran, Lisa Peng, Nathan Hughes, J Daniel Griffith, and Luca Carlone. Hierarchical representations and explicit memory: Learning effective navigation policies on 3d scene graphs using graph neural networks. In *IEEE Int. Conf. on Robot. Autom.*, pages 9272–9279, 2022.

[18] Christopher Agia, Krishna Murthy Jatavallabhula, Mohamed Khodeir, Ondrej Miksik, Vibhav Vineet, Mustafa Mukadam, Liam Paull, and Florian Shkurti. Taskography: Evaluating robot task planning over large 3d scene graphs. In *Conf. on Robot Learn.*, pages 46–58, 2022.

[19] Samuel Looper, Javier Rodriguez-Puigvert, Roland Siegwart, Cesar Cadena, and Lukas Schmid. 3D VSG: Long-term Semantic Scene Change Prediction through 3D Variable Scene Graphs. In *IEEE Int. Conf. on Robot. Autom.*, pages 8179–8186, 2023.

[20] Lianggangxu Chen, Xuejiao Wang, Jiale Lu, Shaohui Lin, Changbo Wang, and Gaoqi He. CLIP-Driven Open-Vocabulary 3D Scene Graph Generation via Cross-Modality Contrastive Learning. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pages 27863–27873, 2024.

[21] Ziqin Wang, Bowen Cheng, Lichen Zhao, Dong Xu, Yang Tang, and Lu Sheng. VL-SAT: Visual-Linguistic Semantics Assisted Training for 3D Semantic Scene Graph Prediction in Point Cloud. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pages 21560–21569, 2023.

[22] Changsheng Lv, Mengshi Qi, Xia Li, Zhengyuan Yang, and Huadong Ma. SGFormer: Semantic Graph Transformer for Point Cloud-Based 3D Scene Graph Generation. In *AAAI Conf. on Artificial Intell.*, volume 38, pages 4035–4043, 2024.

[23] Antoni Rosinol, Andrew Violette, Marcus Abate, Nathan Hughes, Yun Chang, Jingnan Shi, Arjun Gupta, and Luca Carlone. Kimera: From slam to spatial perception with 3d dynamic scene graphs. *The Int. J. of Robotics Research*, 40(12-14):1510–1546, 2021.

[24] Tim Steinke, Martin Büchner, Niclas Vödisch, and Abhinav Valada. Collaborative Dynamic 3D Scene Graphs for Open-Vocabulary Urban Scene Understanding. *arXiv:2503.08474*, 2025.

[25] Tomasz Kucner, Jari Saarinen, Martin Magnusson, and Achim J Lilienthal. Conditional transition maps: Learning motion patterns in dynamic environments. In *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, pages 1196–1201, 2013.

[26] Zhan Wang, Rares Ambrus, Patric Jensfelt, and John Folkesson. Modeling motion patterns of dynamic objects by iohmm. In *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, pages 1832–1838, 2014.

[27] Ransalu Senanayake and Fabio Ramos. Bayesian hilbert maps for dynamic continuous occupancy mapping. In *Conf. on Robot Learn.*, pages 458–471, 2017.

[28] Tomáš Krajník, Jaime P Fentanes, Joao M Santos, and Tom Duckett. Fremen: Frequency map enhancement for long-term mobile robot autonomy in changing environments. *IEEE Trans. on Robotics*, 33(4):964–977, 2017.

[29] Tomáš Vintr, Sergi Molina, Ransalu Senanayake, George Broughton, Zhi Yan, Jiří Ulrich, Tomasz Piotr Kucner, Chittaranjan Srinivas Swaminathan, Filip Majer, Mária Stachová, et al. Time-varying pedestrian flow models for service robots. In *European Conf. on Mobile Robots*, pages 1–7, 2019.

[30] Junyi Shi and Tomasz Piotr Kucner. Learning state-space models for mapping spatial motion patterns. In *European Conf. on Mobile Robots*, pages 1–6, 2023.

[31] Junyi Shi and Tomasz Piotr Kucner. Learning temporal maps of dynamics for mobile robots. *Robotics & Autonomous Syst.*, 184:104853, 2025.

[32] Francesco Verdoja, Tomasz Piotr Kucner, and Ville Kyrki. Bayesian floor field: Transferring people flow predictions across environments. In *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, pages 12801–12807, 2024.