

Kairos: Grounded Forecasting of Presence and Directional Flow in 4D Scene Graphs

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

Abstract— Long-term autonomy in human-populated environments requires anticipating whether and how people will move at times a robot has not yet observed. Existing representations of pedestrian motion face a tradeoff: they either forecast future activity, reducing each location to a scalar rate, or model the full directional distribution, holding it fixed in time. We present Kairos, a predictive directional-flow memory that extends a hierarchical 3D scene graph (3DSG) to a 4D scene graph (4DSG). Every observed voxel of the reconstructed geometry stores a directional mixture and a presence rate, and spectral predictors forecast, for any future query time, both the probability that people are present and the full directional distribution of their motion. Pairwise flow dependence between adjacent voxels supports conditional queries, and per-voxel predictive variances yield calibrated credible intervals that tighten as observations accumulate. We evaluate Kairos on three real pedestrian environments: a robot-collected campus dataset, a shopping mall, and a station concourse recorded continuously for eleven months. Its learned state remains consistent under loop-closure corrections, and its forecasts are competitive with dedicated occupancy and flow models trained on the full detection stream, although Kairos learns from only the small fraction available to a patrolling robot. Finally, we validate the representation on a downstream encounter-probability planning task, where plans computed over the Kairos forecasts encounter more people than plans computed over any time-invariant map at an equal success rate. We provide the code at <https://github.com/IacopomC/kairos>

I. INTRODUCTION

Environments shared with people are shaped by human motion. Long-term autonomy in such environments requires anticipating how people move through a space at a given time, a structure that cannot be inferred from observations of visible agents alone. Pedestrian flow is neither static nor random. For example, a corridor may carry predominantly one-way traffic during the morning commute and reverse later in the day, while a station platform may empty outward after a train arrives and fill again inward before the next departure. Such patterns are structured by daily and weekly periodicities that can be recovered from the sparse, irregular observations a robot gathers over long-term deployment [2], [3]. To support predictive planning, a flow model must be *directional* (a multi-modal distribution over heading, not just occupancy), *time-conditioned* (a distribution that shifts with the hour), and *grounded* (a distribution tied to the geometry the robot reconstructs and plans on).

Maps of Dynamics (MoDs) are models of motion learned as a persistent property of the environment [4]. Spectral

This work was partially supported by the Finnish Cultural Foundation, and by DGA_FSE T73_23R. I. Catalano is with the University of Turku, Finland. J. Peña-Queralta is with the Centre for Artificial Intelligence, Zürich University of Applied Sciences, Winterthur, Switzerland. J. A. Placed is with the Instituto Tecnológico de Aragón (ITA) and the University of Zaragoza, Spain. J. Civera is with the University of Zaragoza, Spain. Corresponding: imcata@utu.fi

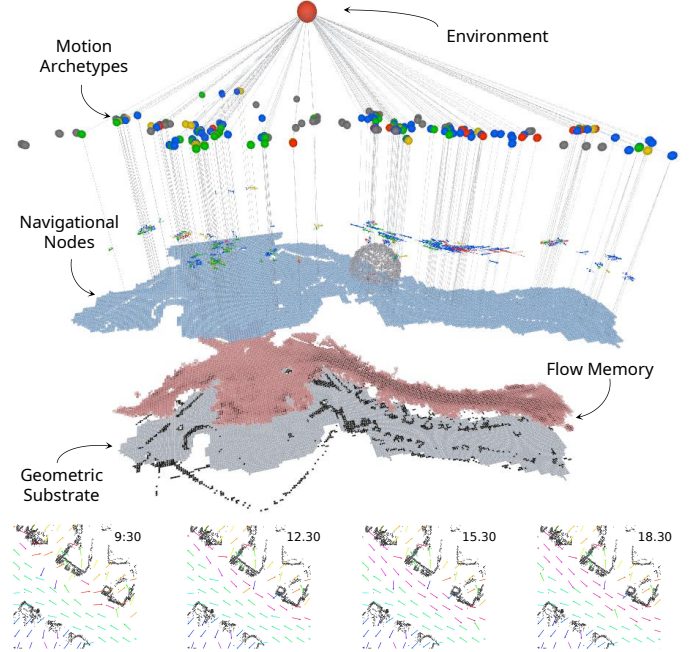


Fig. 1: **Top**: 4D scene graph after a one-day exploration in ATC [1]. The geometric gray substrate holds the occupancy map, with occupied cells in black, and flow voxels anchored to it in red, shaded by accumulated detections. The navigational layer on top of them carries the lifted flow: each arrow is a mixture component of the predicted heading distribution at that place, colored by its motion archetype (gray static, green unimodal, blue bimodal, red multimodal, yellow diffuse). Archetype nodes group adjacent navigational nodes of one archetype under the environment node. The translucent sphere marks the robot’s active sensing window. **Bottom**: Predicted flow over a portion of the east corridor at four times of the same day. Colors encode heading on a circular scale.

occupancy models [2], [5] forecast in time but predict only a scalar per cell, indicating whether a location will be busy but without describing how motion through it is organized. Directional models represent the full motion distribution but only as a long-run average, fixed once estimated and blind to how it shifts over the day [6], [7], [8].

Hierarchical 3D Scene Graphs (3DSG) provide a semantic structure of the environment grounded in geometry [9] but their temporal dimension tracks individual agents [10], [11], state changes [12], or the scene’s history for retrospective query [13], [14], capturing what happened rather than the recurring flow of a location. Recent work instead embeds a directional motion model as a layer of the hierarchy, fitting a per-node semi-wrapped Gaussian mixture [15], or forecasting directional flow over discrete orientation bins [16]. However, in both cases the estimate is bound to navigational nodes by

TABLE I: **Comparison of directional and spectral flow-dynamics methods.** The first five columns refer to properties that flow forecasting must satisfy. The last three are additional capabilities beyond the required set. — stands for *not provided*.

Family	Method	Required properties					Additional capabilities		
		Flow Representation	Forecast	Index	Warp	Online Fit	Flow Coupling	Presence	Forecast Uncertainty
Discrete MoD	Cond. Transition Maps [6]	transitions	—	grid	—	—	implicit	—	—
	CLiFF-map [7]	SW-GMM	—	grid	—	—	kernel imp.	—	—
	Dir. Grid Maps [8]	von Mises mixture	—	grid	—	—	—	—	—
	FreMEn [2], FrOctoMap [5]	—	occupancy	grid	—	✓	—	binary	—
	STeF-Map [17]	8-bin histogram	bin value	grid	—	✓	—	—	—
	Online CLiFF-map [18]	SW-GMM	—	grid	—	✓	—	—	—
Continuous MoD	NeMo-map [19]	SW-GMM	mixture params.	coordinates	—	—	—	—	—
	Variational GP activity [20]	—	activity rate	coordinates	—	—	—	GP rate	—
Scene-graph Dynamics	Aion [16]	8-bin histogram	bin value	nodes	transfer	✓	—	—	—
	Rheos [15]	SW-GMM	—	nodes	transfer	✓	—	—	—
	Kairos (<i>ours</i>)	SW-GMM, $K = 8$	mixing weights	voxels	re-key	✓	pairwise	Gamma-Poisson	posterior

proximity rather than to the reconstructed geometry.

Neither representation satisfies all three properties at once: a scalar state, a fixed average, a node-bound estimate, or per-agent tracks cannot, by construction, forecast how a location will flow at a future time. We present **Kairos**, a *predictive directional-flow memory* that extends a 3DSG to a 4D scene graph (4DSG). A spectral predictor on each mixing weight of a per-voxel directional mixture forecasts the entire distribution over time, anchored to the observed voxels of the robot’s TSDF reconstruction and kept consistent under map re-optimization.

At any traversable location and any future time, Kairos answers two questions a static map cannot: whether agents are likely to be present there, and if so, how they are likely to be moving, as a full directional distribution. A presence channel kept separate from flow distinguishes *no one passes here* from *no one has yet been observed*. Because the forecast covers the whole reconstructed environment and is exposed on the navigational graph, a robot can weigh entire routes before committing to one, comparing how each will flow at the time it would traverse it rather than planning against a snapshot.

We make the following contributions:

- **Full-distribution flow forecasting on the map:** Kairos is the first method to forecast the complete directional flow distribution, where prior flow layers predict a scalar summary or a static average, and continuous full-distribution forecasts are learned offline over abstract coordinates. Each voxel is updated online from a fixed-size summary, at a cost independent of how long the robot has been observing.
- **Grounded under map correction:** the forecast is *anchored to the geometry the robot reconstructs online*, coindexed with the scene graph’s own cells and equivariant under the corrections its backend applies, two properties that grid-based and proximity-attached flow maps do not satisfy.
- **Decoupled presence and flow:** a per-voxel presence channel distinguishes *no agents present* from *no flow*, so planners can reason independently about occupancy and motion.
- **Planner-facing integration:** per-voxel forecasts are aggregated up the hierarchy and annotated on the navigational graph, exposing time-conditioned flow through the standard scene-graph interface; we demonstrate it on a downstream encounter-probability planning task.

We further show that pairwise flow dependence over ad-

jacent voxels supports conditional queries and joint draws coherent with continuous motion, and that combining the posterior variance of the spectral coefficients with the measured scatter of the observations returns every forecast as a calibrated credible interval.

II. RELATED WORK

Kairos draws on two complementary lines of work: MoDs, which model motion as a persistent property of the environment [4], and 3DSGs, which provide semantic and topological structure to environment representations [9]. We review both against the three properties required for anticipatory planning: *directionality*, *time-conditioning*, and *grounding*. See Table I.

A. Maps of Dynamics

Directional models of flow. Conditional transition maps [6] model flow at each cell as a distribution over transitions to its neighbors, capturing dependencies between adjacent locations rather than estimating them independently. The resulting model is time-invariant. CLiFF-map [7] instead fits a per-cell Semi-Wrapped Gaussian Mixture Model (SW-GMM) over direction and speed, representing multimodal flow as a continuous density on the cylinder (the parametrization Kairos adopts). Directional grid maps [8] model angular uncertainty at each cell using a von Mises mixture, omitting speed. These approaches meet the directional requirement, but all estimate a long-run average at each cell, estimated once and fixed thereafter, with no temporal dependence.

Spectral models of periodic state represent the environment as a set of periodic processes recovered in the frequency domain from sparse, irregular observations [21]. FreMEn (Frequency Map Enhancement) [2] formalizes this for a binary state, retaining Fourier coefficients at candidate periods and evaluating them at any future time to obtain a probability. This representation is agnostic to what the state describes, which has indeed allowed its application to different domains. FrOctoMap [5] places FreMEn in a voxel grid for spatio-temporal occupancy; the Poisson-spectral model of [3] replaces the binary state with a counting process to forecast activity rate; STeF-Map [17] applies FreMEn to per-cell orientation histograms. Information-driven mapping

refines such models online through an incremental per-cell spectral update [22] (the form Kairos retains), and wrapped-time representations extend such spectral models to multi-day and non-sinusoidal periodicity [23]. These models meet the time-conditioned requirement, but each tracks a *scalar* per cell: an occupancy state, a rate, or a single histogram bin. STeF-Map comes closest, yet a histogram is a quantized, per-bin scalar rather than a continuous directional distribution. Kairos retains FreMEn spectral predictor and instantiates one per scalar channel of a distribution-valued state: on each mixing weight of a per-voxel SW-GMM, and on the occupancy rate.

Learned models. Recent work relaxes these limits along three axes. The first replaces the analytic spectral primitive with a learned temporal model. Neural sequence models predict how the motion field itself evolves [24]; a hybrid model pairs a periodic spectral term with an LSTM residual over orientation histograms [25]; and event-triggered formulations add adaptive cells and neural stochastic differential equations to track abrupt, non-stationary change [26], which recurrence alone cannot represent. The second relaxes the batch fit: online estimators update a per-cell mixture from streaming observations to follow drift without rebuilding [18]. The third uses a continuous formulation, removing the grid altogether. NeMo-map [19] learns an implicit neural field from spatio-temporal coordinates to the parameters of an SW-GMM, so it outputs a directional distribution at any location and future time without discretizing either. However, it is fit offline in batch and its readout is a function of a coordinate rather than of an element of the built map. A fourth direction of research focuses on estimating the field from scarce or absent local observations: a Gaussian process interpolates over sparse mobile-robot observations [20] and outputs a variance with each estimate, while transfer approaches infer the field from a floor plan [27] or regress it from egocentric views [28], providing a prior where it has not been initialized yet.

Individually the models above meet the directional and time-conditioned requirements, but none satisfies the grounding condition: each lives on an abstract grid or coordinate space, decoupled from the geometry a robot reconstructs and plans on, so the estimate cannot follow the map as it is corrected. In contrast, Kairos keys each channel to the active voxels of the robot’s reconstruction, and re-keys them after every correction the backend applies, so the forecast is defined on (and consistent with) the geometry the planner reads.

B. Dynamics in 3D Scene Graphs

3DSGs layer the world by increasing abstraction, grounding semantics and topology in the geometry the robot plans on, and thus satisfying by construction the one property the models above lack. Most 3DSGs are quasi-static; where dynamics are modeled, they take three forms: interaction-driven state change, per-entity temporal dynamics, and aggregate flow.

Interaction-driven state change. Action-aware scene graphs annotate nodes with affordances but treat them as time-invariant [29], [30]. Looper *et al.* [12] attach a learned variability score that estimates how likely an object is to change, without specifying when. This family captures whether and

how much a scene changes, but not when or how motion flows through it (*i.e.*, a property of objects rather than the motion through a place). Therefore, neither the directional nor the time-conditioned requirements apply.

Per-entity temporal dynamics. Kimera-DSG [10] models moving agents as dynamic nodes and CURB-SG [11] tracks vehicles for urban navigation. Khronos [13] and DAAAM [14] maintain geometrically grounded scene histories for retrospective queries, and [13] in particular maintains a consistent representation under backend optimization. However, their temporal models are defined independently for each entity: they track the history of individual agents rather than estimating recurring patterns of motion. Consequently, they cannot represent the flow associated with a location, nor do they support queries about its expected state at future times.

Aggregate flow as a graph layer. Closest to Kairos, two systems attach a MoD to the navigational layer. Aion [16] keeps a FreMEn model over each orientation histogram and forecasts three scalar quantities (magnitude, dominant direction, and entropy), discarding the distribution itself; Rheos [15] keeps the full distribution as a continuous SW-GMM but fits it once, with no dependence on time. Each therefore meets one of the two requirements the other misses. Both associate observations to navigational nodes by proximity, rather than the reconstructed geometry. Kairos closes both gaps by forecasting the full directional distribution, and keying the estimate to the active voxels of the reconstruction rather than to node proximity, thus holding under the backend corrections.

III. PROBLEM FORMULATION

We consider the long-term deployment of a robot in an environment populated with other mobile agents, *e.g.*, humans. The robot observes a stream of detections $\mathcal{Z}_{\leq T}^M = \{(t_k^j, \mathbf{x}_k^j, \mathbf{v}_k^j) : t_k^j \leq T\}_{j=1}^M$ for M agents up to time T , where, in the most general case, $\mathbf{x}_k^j \in \mathbb{R}^3$ denotes the 3D position of an agent j at time t_k^j , and $\mathbf{v}_k^j \in \mathbb{R}^3$ its corresponding velocity. Assuming that agents move on the ground, we characterize the flow by the horizontal component of their velocity. Let $\Pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ denote the projection onto the horizontal plane, such that $\Pi \mathbf{v}_k^j = (v_{k,x}^j, v_{k,y}^j)$. Each detection then defines a heading $\theta_k^j = \text{atan2}(v_{k,y}^j, v_{k,x}^j) \in [0, 2\pi)$ and a speed $\rho_k^j = \|\Pi \mathbf{v}_k^j\| \in \mathbb{R}_{>0}$, where $\|\cdot\|$ denotes the Euclidean norm. Since the detection times $\{t_k^j\}_{k=1}^{N_j}$ are sparse and irregularly spaced, the flow must be estimated from a limited number of unevenly distributed observations. We take a moving agent as the unit of observation: as stationary agents have no defined heading, they do not contribute to the flow distribution; and, similarly, in-place rotations are outside the scope of the model.

A. Dynamics Field

At any time t , each agent j occupies a position $\mathbf{x}^j(t)$ and moves with a velocity $\mathbf{v}^j(t)$. We model this as a point process on $\mathbb{R}^3$ whose points are marked by their heading and speed. The quantity to be estimated is the *intensity* of this process at an arbitrary position $\mathbf{x} \in \mathbb{R}^3$, heading θ , speed ρ , and time t ,

$$\Lambda(\mathbf{x}, \theta, \rho, t) \geq 0. \quad (1)$$

Marginalizing (1) over velocity yields the *occupancy intensity*

$$\lambda(\mathbf{x}, t) \triangleq \int_0^{2\pi} \int_0^\infty \Lambda(\mathbf{x}, \theta, \rho, t) d\rho d\theta, \quad (2)$$

whose integral over a region is the expected number of agents in that region at time t .

Normalizing (1) by (2) yields the *flow distribution*

$$p(\theta, \rho \mid \text{present}, \mathbf{x}, t) \triangleq \frac{\Lambda(\mathbf{x}, \theta, \rho, t)}{\lambda(\mathbf{x}, t)}, \quad (3)$$

where present denotes the event that an agent occupies $(\mathbf{x}, t)$ and (3) its velocity distribution at that position. It is a density on the cylinder $S^1 \times \mathbb{R}_{>0}$, normalized jointly over heading and speed by construction and it is only defined where $\lambda(\mathbf{x}, t) > 0$. Equations (2) and (3) are the intensity-mark factorization $\Lambda = \lambda p$ of the process, separating *how many* agents pass a location from *how* they move there.

Writing $\mathcal{D}(\mathbf{x}, t)$ for the flow distribution and the occupancy intensity at one position and time, the *dynamics field* $\mathcal{D}$ is distribution-valued with codomain $(\mathcal{P}(S^1 \times \mathbb{R}_{>0}) \cup \{\emptyset\}) \times \mathbb{R}_{\geq 0}$, where $\mathcal{P}(\cdot)$ denotes probability measures on a space and $\emptyset$ represents the positions at which (3) is undefined.

B. Estimation Problem

Given the stream $\mathcal{Z}_{\leq T}^M$, our goal is to forecast the dynamics field $\mathcal{D}(\mathbf{x}, t)$ at an arbitrary *future* query $(\mathbf{x}, t)$ with $t > T$. Note that forecasting beyond the observation horizon is well-posed only under recurrence. The assumption is stated on the joint intensity (1), a non-negative function of its arguments and therefore additively decomposable. We assume that the intensity Λ can be decomposed into a recurrent component $\hat{\Lambda} \geq 0$, associated with a set of periods fixed from the environment (e.g., daily cycles), and a residual random component ε without temporal structure:

$$\Lambda(\mathbf{x}, \theta, \rho, t) = \hat{\Lambda}(\mathbf{x}, \theta, \rho, t) + \varepsilon(\mathbf{x}, \theta, \rho, t). \quad (4)$$

$\hat{\Lambda}$ can be inferred from the observations $\mathcal{Z}_{\leq T}^M$ and extrapolated to future times, whereas ε is not forecast. The forecast field $\hat{\mathcal{D}}$ follows from $\hat{\Lambda}$ through (2) and (3), and carries $\emptyset$ in its flow component wherever the recurrent part places no occupancy.

Lifelong deployment precludes storing $\mathcal{Z}_{\leq T}^M$ in full, so any solution takes the form of an *online* estimator: a state $s \in \mathcal{S}$ whose size does not grow with the number of detections, updated recursively as detections $(t_k^j, \mathbf{x}_k^j, \mathbf{v}_k^j)$ arrive

$$s_k \leftarrow \mathcal{T}(s_{k-1}, (t_k^j, \mathbf{x}_k^j, \mathbf{v}_k^j)), \quad (5)$$

with $\mathcal{T} : \mathcal{S} \times (\mathbb{R} \times \mathbb{R}^3 \times \mathbb{R}^3) \rightarrow \mathcal{S}$, and a readout function $\mathcal{R}$ mapping the state s to the forecast at any future time,

$$\hat{\mathcal{D}}(\mathbf{x}, t) \triangleq \mathcal{R}(\mathbf{x}, t, s), \quad t > T, \quad (6)$$

with $\mathcal{T}$ and $\mathcal{R}$ to be determined. Memory therefore stays bounded however many detections arrive, and scales instead with the extent of the environment the robot has covered.

For the forecast to inform a robot's decisions, it must capture the direction in which agents move, how that direction changes through time, and where on the robot's own map the estimate applies. The first two are fixed by the field, and the third one

depends on the map built. The distribution-valued codomain $\mathcal{P}(S^1 \times \mathbb{R}_{>0})$ makes the solution *directional*: the forecast is the full, typically multi-modal distribution over heading and speed, and a scalar summary does not lie in that codomain. The future query $t > T$ makes it *time-conditioned*: the distribution is evaluated at t , and a long-run average is constant in t .

C. Grounding

Let $\mathcal{G}^T$ denote the metric-semantic representation estimated by the robot up to time T , $\mathcal{M}^T \subset \mathbb{R}^3$ the region of the environment it has covered, and $\mathcal{A}^T$ the index set of the metric portion of $\mathcal{G}^T$. The mapping $\Xi^T : \mathcal{M}^T \rightarrow \mathcal{A}^T$ carries a covered position to the element of that portion containing it. An estimate is *grounded* when it is coindexed with $\mathcal{G}^T$ and remains so as the map is updated (e.g., after a loop closure). *Coindexing* requires the readout (6) to factor as

$$\mathcal{R}(\mathbf{x}, t, s) = \tilde{\mathcal{R}}(\Xi^T(\mathbf{x}), t, s). \quad (7)$$

The forecast can be then addressed by an element of $\mathcal{G}^T$ rather than by a coordinate external to it. As Ξ^T is a component of $\mathcal{G}^T$, it is itself an estimate, which will be revised as the reconstruction is. Correction of accumulated pose drift displaces geometry already built. We model it as a warp $\mathcal{W} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, a diffeomorphism acting on $\mathcal{Z}_{\leq T}^M$ by

$$\mathcal{W} \cdot \mathcal{Z}_{\leq T}^M \triangleq \{(t_k, \mathcal{W}(\mathbf{x}_k), \mathbf{J}_{\mathcal{W}}(\mathbf{x}_k) \mathbf{v}_k)\}_k, \quad (8)$$

displacing positions and transporting velocities by the Jacobian $\mathbf{J}_{\mathcal{W}}$, so a correction that rotates a region of the map rotates the observed headings by the same amount. Let $\mathcal{W}_*$ denote the pushforward of (8) onto the codomain of $\mathcal{D}$, the warp acting at the level of densities. The estimate is *warp-consistent* when

$$\mathcal{F}(\mathcal{W}(\mathbf{x}), t; \mathcal{W} \cdot \mathcal{Z}_{\leq T}^M) = \mathcal{W}_* \mathcal{F}(\mathbf{x}, t; \mathcal{Z}_{\leq T}^M), \quad \forall \mathbf{x} \in \mathcal{M}^T, \quad (9)$$

where $\mathcal{F}$ is the composite of the update (5) and the readout (6). Under (9) a correction relocates where a forecast applies and rotates the directions it predicts, leaving the accumulated statistics otherwise unchanged. A correction may carry two elements of $\mathcal{A}^T$ onto one (e.g., after a loop closure that aligns two observations of the same region), so $\Xi^T \circ \mathcal{W}$ need not be injective even where $\mathcal{W}$ is. Equality in (9) then holds at every corrected index reached from exactly one element; where several are carried onto one, their accumulated statistics are pooled and (9) holds for the pooled estimate.

IV. SCENE GRAPH SUBSTRATE

Section III-C requires an estimate coindexed with $\mathcal{G}^T$ and warp-consistent under corrections applied to it. We take $\mathcal{G}^T$ to be a hierarchical 3DSG built online from the robot's sensor stream [31], which supplies an index set, a backend that emits the corrections $\mathcal{W}$, and a navigational layer.

A. Hierarchical 3D Scene Graph

A 3DSG $\mathcal{G}^t \triangleq (\mathcal{V}^t, \mathcal{E}^t)$ organizes the environment into nodes $\mathcal{V}^t = \bigsqcup_\ell \mathcal{V}_\ell^t$ at increasing levels of abstraction. Inter-layer edges connect each node to its parent one level above; intra-layer edges relate nodes within a level. The choice of

layers is task dependent but can be generalized as a metric substrate, a motion graph, a navigational and action affordance abstraction layer, a semantic abstraction and a global structure [9]. The graph is built incrementally from the sensor stream, so the nodes, the edges, and the geometry beneath them all change as the robot operates. The dynamics field of Section III is defined over four of its elements.

1) *Voxels*: At the base of the graph is a Truncated Signed Distance Field (TSDF), which provides the 3D scene geometry from which the higher layers are derived. It partitions the environment into disjoint cells of side s_v , indexed by $\mathbf{i} \triangleq (i_1, i_2, i_3) \in \mathbb{Z}^3$, of which the *active voxels* $\mathcal{A}^t$ are those the TSDF has observed up to time t . The set therefore realizes the index set of Section III-C, the covered region is the union of its cells $\mathcal{M}^t = \bigcup_{\mathbf{i} \in \mathcal{A}^t} \text{cell}(\mathbf{i})$, and the assignment of a position to the cell containing it realizes the index map Ξ^t . Both sets grow as the robot explores the environment.

2) *Navigational Nodes*: Each node of the navigational layer $\mathcal{V}_{\text{nav}}^t$ is a collision-free location extracted from the Generalized Voronoi Diagram of the reconstructed free space [31]. It carries a *clearance*, the distance to the nearest reconstructed surface, and owns a *support set* of the active voxels nearest to it.

3) *Navigational Edges*: An edge denotes straight-line traversability between two navigational nodes [31]. Support sets and edges together induce a *traversability relation* over the active voxels: two voxels are related when both fall in one node's support set, or in those of two nodes joined by an edge.

4) *Room Nodes*: A room node groups the navigational nodes enclosed by one region of the environment, giving a level of the hierarchy above the navigational layer.

B. Grounding on the Graph

This choice of $\mathcal{G}^T$ satisfies the two conditions of Section III-C. Coindexing holds as the readout is evaluated at $\mathbf{i} = \Xi^T(\mathbf{x}) \in \mathcal{A}^T$ and because Ξ^T is the scene graph's own cell assignment. The forecast is addressed by a cell from which the higher-abstraction cells are themselves derived, so lifting an estimate to higher-abstraction nodes is a lookup over the set. Warp-consistency holds because each re-optimization of the pose graph warps the reconstructed geometry and, with it, the active voxels, by the action (8). Re-keying the cells through every such correction relocates where a forecast applies, leaving the accumulated statistics otherwise unchanged.

No previous map of dynamics satisfies both: grid-based models [17] keep their indices fixed while the corrected geometry moves away from them, and node-proximity models [16], [15] re-aggregate observations under a changed navigational layer rather than transporting the estimate (Section II, Table I).

V. MODELING SPATIO-TEMPORAL FLOW DYNAMICS

Evaluating the readout (6) at an arbitrary future query is intractable in general: the dynamics field is distribution-valued at every position and every future time, and its joint posterior from sparse, irregular observations is high-dimensional and coupled across space. Kairos makes it estimable through four reductions. First, the field is piecewise constant on the cells of the reconstruction and estimated independently at each one.

Second, at each cell the occupancy intensity (2) separates from the flow distribution (3), each carried by a *separate* channel. Third, the flow distribution factorizes over a fixed set of cardinal directions, reducing (3) to one time-varying mixing weight per direction, while the speed along each direction is summarized by a running mean and held time-invariant. Fourth, every resulting quantity, the occupancy rate and each directional weight alike, is a scalar trajectory in time, forecast by a single spectral primitive.

This reduces future-time estimation in (6) to a constant-memory online update that is *directional*, as the per-direction channels retain the full distribution; *time-conditioned*, as the spectral primitive forecasts each channel forward; and *grounded*, as every channel is keyed to the map's cells. Two further terms act across cells: evidence sharing, which shrinks a sparsely observed cell's weights toward its neighborhood's, and a pairwise dependence relating the flow at adjacent cells.

A. Piecewise-Constant Field

The state factorizes into one fixed-size per-cell state $s_{\mathbf{i}}$, realizing the abstract s of (5). A query at $(\mathbf{i}, t)$ retrieves that cell's state and evaluates it at t .

The occupancy intensity (2) is thus realized per voxel. Piecewise constancy holds $\lambda(\mathbf{x}, t) = \lambda_{\mathbf{i}}(t)$ for all $\mathbf{x}$ in the cell $C_{\mathbf{i}}$, so its integral over a cell collapses to $\int_{C_{\mathbf{i}}} \lambda(\mathbf{x}, t) d\mathbf{x} = \lambda_{\mathbf{i}}(t) s_v^3$, the expected number of agents in a voxel of side s_v at time t . A count over the cell and the intensity are then the same quantity up to the known cell volume, and the intensity, which is a density and thus determined at a point by no finite set of detections, is instead estimated through the cell count.

B. Flow Dynamics Model

Each active voxel $\mathbf{i} \in \mathcal{A}^t$ allocates, on its first detection, a SW-GMM over the polar decomposition of an agent's velocity vector $(\theta, \rho) \in [0, 2\pi) \times \mathbb{R}_{>0}$ (its heading and speed respectively). The per-voxel flow distribution (3) is:

$$p_{\mathbf{i}}(\theta, \rho | t) \triangleq \sum_{k=1}^K \pi_{\mathbf{i}}^{(k)}(t) p_{\mathbf{i}}^{(k)}(\theta, \rho), \quad (10)$$

with time-varying mixing weights $\pi_{\mathbf{i}}^{(k)}(t) \geq 0$ that form a distribution at every time, $\sum_{k=1}^K \pi_{\mathbf{i}}^{(k)}(t) = 1$; the component densities $p_{\mathbf{i}}^{(k)}$ are time-invariant, so only the weights carry the time dependence. As θ is circular, each component k is a wrapped bivariate Gaussian, summed over angular windings:

$$p_{\mathbf{i}}^{(k)}(\theta, \rho) \triangleq \sum_{w=-W}^W \frac{1}{2\pi\sqrt{\det \Sigma}} \exp\left(-\frac{1}{2} \mathbf{d}_w^\top \Sigma^{-1} \mathbf{d}_w\right), \quad (11)$$

where $\mathbf{d}_w = (\theta + 2\pi w - \mu_{\theta}^{(k)}, \rho - \mu_{\rho, \mathbf{i}}^{(k)})^\top$, and we set $W=2$. Note that only the angular factor is wrapped (making the mixture semi-wrapped). The radial factor is an ordinary Gaussian on ρ , and since walking speeds lie several σ_{ρ} above zero, the mass at $\rho < 0$ is negligible and is left untruncated.

Unlike previous work [7], [15], we fix the number of components $K=8$. This makes the component geometry universal: slot k denotes the same heading band in every voxel, so each

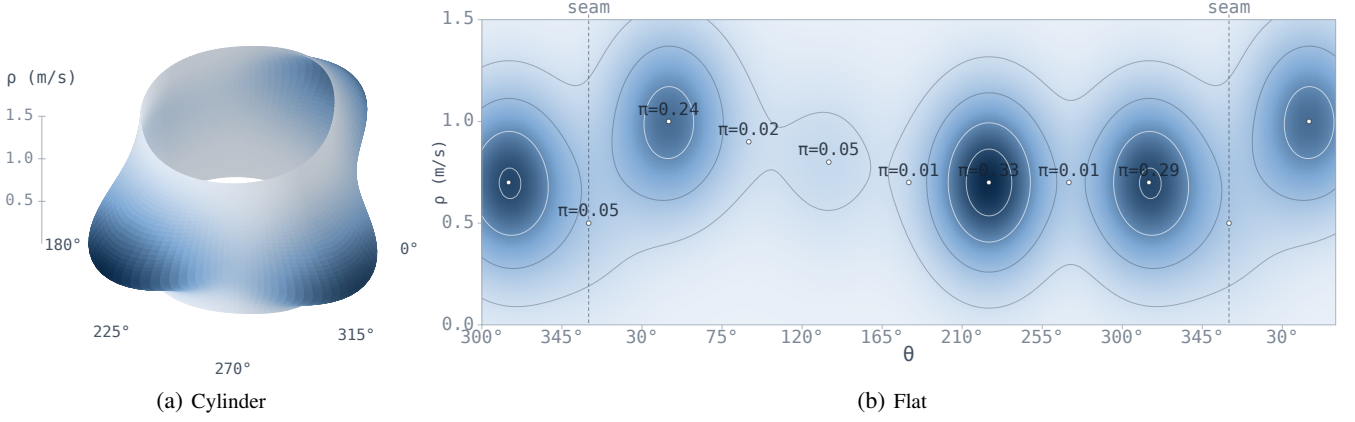


Fig. 2: Fixed-component SW-GMM for a single voxel, on two embeddings of its domain $[0, 2\pi) \times \mathbb{R}_{>0}$, a heading circle times a speed half-line. The $K=8$ components sit at cardinal headings $(k-1)\pi/4$ with shared spreads $\sigma_\theta = 0.4$ rad and $\sigma_\rho = 0.3$ m/s, so a component varies across voxels only in mean speed $\mu_{\rho,i}^{(k)}$, and across slots only in that mean and the weight $\pi_i^{(k)}$. Both panels render the same mixture.

weight is a scalar signal with a fixed meaning that a single predictor can track through time, and the weight vectors $\pi_i(t)$ are directly comparable across voxels. Each component k has mean heading $\mu_\theta^{(k)} = (k-1)\pi/4$ ($k = 1, \dots, K$) and diagonal covariance $\Sigma = \text{diag}(\sigma_\theta^2, \sigma_\rho^2)$, both shared across all voxels, while the mean speed $\mu_{\rho,i}^{(k)}$ is estimated per voxel: a component depends on the voxel through its radial mean alone (see Fig. 2).

We set the angular and radial spreads to $\sigma_\theta = 0.4$ rad and $\sigma_\rho = 0.3$ m/s, with zero cross-covariance, and identical across all components and voxels. The angular spread is therefore half the $\pi/4$ spacing between adjacent slots, so neighboring components overlap and a heading falling between two cardinals is shared by its two nearest slots. The fixed slots correspond to the orientation histograms of prior temporal maps [17], [16], realized as overlapping Gaussians rather than disjoint bins.

Heading and speed are not independent in general, and the mixture carries their dependence across slots through the per-slot speed means $\mu_{\rho,i}^{(k)}$. A cross-covariance would add dependence within the $\pi/4$ band a single slot spans, where too little heading variation remains for speed to correlate with, so we set it to zero. Marginalizing (10) over speed then leaves a mixture of wrapped normals over heading, and over heading a mixture of Gaussians over speed, under the same weights. Within this fixed geometry, a voxel's flow is described entirely by the mass each direction holds and the mean speed along it.

C. Temporal Forecasting

Because the component geometry is fixed, a voxel's flow distribution (10) changes over time only through its mixing weights. Forecasting the distribution forward in time therefore reduces to forecasting the K mixing-weight trajectories $\pi_i^{(k)}(t)$, one scalar time series per slot. A trajectory is forecastable past the observed interval only where it recurs, so we represent each one in a periodic basis. We adopt the Non-Uniform Discrete Fourier Transform (NUDFT) spectral model [2], [17], instantiating one *predictor* per scalar trajectory; each represents its signal as a constant mean plus harmonics at fixed candidate frequencies.

We fix a set of angular frequencies $\{\omega_f\}_{f=1}^F$, with $\omega_f = 2\pi/T_f$ for the periods T_f the environment is expected to exhibit. A predictor ingests timestamped samples (t_n, y_n) of the scalar signal it tracks. For slot k that signal is the fraction of each detection contributing to it: a detection (θ_n, ρ_n) is assigned softly across the K slots by its *responsibilities*

$$r_{i,n}^{(k)} \triangleq \frac{p_i^{(k)}(\theta_n, \rho_n)}{\sum_{\ell=1}^K p_i^{(\ell)}(\theta_n, \rho_n)}, \quad (12)$$

one per slot at the voxel the detection falls in, and slot k 's predictor receives one sample per crossing: a traversal c of the voxel contributes a single share $y_c^{(k)}$ (the mean of (12) over the detections of that traversal) at a time t_c within the visit. The unit of observation is therefore the passage of an agent, so the estimate depends neither on the rate at which detections arrive nor on how long an agent dwells in the cell.

Since the mixing weight is the time-average of these shares over the C crossings observed so far, $\pi_i^{(k)} = \frac{1}{C} \sum_{c=1}^C y_c^{(k)}$, the predictor's mean term γ_0 is exactly that running average, the static weight, and its coefficients γ_f capture how the share varies in time. Both are updated online as each sample arrives, without accumulating detections into fixed intervals [17]. Dropping the slot and voxel indices for a generic predictor, and writing γ_0^-, γ_f^- for the values before the update:

$$\gamma_0 \leftarrow \frac{c\gamma_0^- + y_c}{c+1}, \quad \gamma_f \leftarrow \frac{c\gamma_f^- + (y_c - \gamma_0^-)e^{-j\omega_f t_c}}{c+1}, \quad (13)$$

where j is the imaginary unit, and the γ_f update accumulates, for each frequency, the correlation between its sinusoid and the signal's deviation $(y_c - \gamma_0^-)$ from that mean.

Prediction at a future time t reconstructs the tracked signal from a *model order* $M \leq F$: the M of the F candidate frequencies whose coefficients have the largest magnitude [2]. The resulting forecast $\hat{y}(t)$ of the tracked signal (here slot k 's mixing weight, $\hat{y}(t) = \hat{\pi}_i^{(k)}(t)$) is

$$\hat{y}(t) \triangleq \gamma_0 + \sum_{m=1}^M 2|\gamma_{f_m}| \cos(\omega_{f_m} t + \angle \gamma_{f_m}), \quad (14)$$

where $|\gamma_{f_m}|$ and $\angle\gamma_{f_m}$ are the magnitude and phase of the m -th retained coefficient. A predictor is declared valid once it has accumulated at least $c_{\min}$ samples. The K slot predictors are fitted independently, so (14) guarantees neither non-negativity nor a unit sum, while (10) requires the weights to form a distribution. Therefore, the forecasts $\hat{\pi}_i^{(k)}(t)$ are clipped at zero and re-normalized to the probability simplex before evaluation. The predicted dominant heading $\hat{\theta}_{\text{dom}}(t)$ and speed $\hat{\rho}_{\text{dom}}(t)$ are the weight-averaged component means at the query time.

A voxel also stores the per-slot speed means $\mu_{\rho,i}^{(k)}$ (Section V-B). Unlike the mixing weights, these are not forecast but maintained as the responsibility-weighted running mean of the slot's speeds. Until a slot has enough observations of its own, it falls back to a global-mean prior: the running mean of speed pooled over all slots and voxels seen so far.

1) *Model Order Selection*: A fixed global spectral order M fits harmonics to noise wherever a cell's signal is stationary. Rather than validation-based order selection [2], [17], each predictor selects its order online through a predict-then-update *perquential gate*: as each sample arrives, and before it updates the coefficients, the gate scores every candidate order's one-step-ahead prediction of it, so each sample is held out against the state that predicts it. At query time the predictor uses the order with the lowest accumulated one-step error, defaulting to the static mean (order 0) until a higher order lowers it.

2) *Predictive Uncertainty*: A predictor returns every forecast with a predictive standard deviation formed from two independent sources: $\hat{\sigma}(t) \triangleq \sqrt{\sigma_{\text{epi}}^2(t) + \sigma_{\text{ale}}^2}$. The *epistemic* term $\sigma_{\text{epi}}^2(t)$ is the uncertainty remaining in the predictor's own coefficients, and contracts as samples accumulate. The *aleatoric* term σ_{ale}^2 is the spread of the samples about the value the predictor tracks.

At fixed phases $\angle\gamma_f$, the reconstruction (14) is linear in the mean term γ_0 and the harmonic amplitudes $a_f = 2|\gamma_f|$, over the known basis $[1, \cos(\omega_1 t + \angle\gamma_1), \dots, \cos(\omega_F t + \angle\gamma_F)]$. Under Gaussian observation noise and a Gaussian prior on the coefficients, conjugacy makes the posterior over those $F+1$ quantities Gaussian and updates it in closed form at each sample [32]. We approximate that posterior by its diagonal, one variance per coefficient, which holds the state and each rank-one update at $O(F)$ [33], and omits the correlations induced by the basis functions not being orthogonal over the sample times. The forecast is linear in these quantities, so its variance follows in closed form:

$$\sigma_{\text{epi}}^2(t) = \text{var}(\gamma_0) + \sum_{m=1}^M \text{var}(a_{f_m}) \cos^2(\omega_{f_m} t + \angle\gamma_{f_m}). \quad (15)$$

Each amplitude's variance appears scaled by the square of its basis function at the query time, so a harmonic contributes uncertainty in proportion to its weight in the forecast at that time. The variance state spans all F candidate frequencies while the readout runs over the M retained ones, and for a channel tracked by its mean term alone the sum is empty and the variance reduces to $\text{var}(\gamma_0)$.

Alongside γ_0 a predictor accumulates the running second moment m_2 of the values it receives, so $(m_2 - \gamma_0^2)$ gives their variance in constant memory. Measured on few samples this

variance is itself poorly determined, so it is combined with a prior variance σ_0^2 carrying the weight of κ samples of evidence,

$$\sigma_{\text{ale}}^2 \triangleq \frac{\kappa \sigma_0^2 + C(m_2 - \gamma_0^2)}{\kappa + C}, \quad (16)$$

so a cell reports the prior spread until it has evidence of its own, and its measured spread thereafter, with the prior's weight decaying as $\kappa/(\kappa + C)$.

This completes the state of a predictor: the mean term γ_0 , the second moment m_2 , F complex coefficients alongside their $F+1$ variances, and a sample count. The state is $O(F)$ scalars regardless of how long the stream it consumes runs, which is the constant-memory update Section V reduces (6) to.

D. Evidence Sharing

A voxel's mixing weights are estimated from its own crossings alone, and on few crossings that estimate is noisy. As flow varies smoothly over the free space agents traverse, a voxel's neighbors are informative about the same quantity it is estimating, and we treat them as a source of evidence.

We define the neighborhood $\mathcal{N}_i$ by the *traversability* relation of the topological substrate (Section IV-A): a voxel $j \neq i$ belongs to $\mathcal{N}_i$ if it lies in the support set of the navigational node $v \in \mathcal{V}_{\text{nav}}$ containing i , or that of a node connected to v . The support sets are disjoint and every voxel enters $\mathcal{N}_i$ at most once. Every such voxel is reachable from i without crossing a reconstructed surface, so no evidence is borrowed across obstacles.

Let $\gamma_{0,i}^{(k)}$ be the mean term of voxel i 's predictor for slot k , C_i its crossing count (Section V-C), and $\mathcal{N}_i$ its neighborhood as defined above, restricted to the voxels holding at least one crossing of their own. The neighborhood estimate pools the neighbors' mean terms for slot k , each weighted by the evidence behind it (how many agents have crossed it),

$$\bar{\gamma}_{0,i}^{(k)} \triangleq \frac{\sum_{j \in \mathcal{N}_i} C_j \gamma_{0,j}^{(k)}}{\sum_{j \in \mathcal{N}_i} C_j}, \quad (17)$$

the value a single voxel holding the neighborhood's pooled crossings would carry. Where no neighbor holds a crossing, or where the navigational layer does not yet cover i , the neighborhood is empty and the voxel reports its own mean.

Combining the voxel's own mean with the neighborhood estimate, the weight voxel i reports for slot k is:

$$\tilde{\gamma}_{0,i}^{(k)} \triangleq \frac{C_i \gamma_{0,i}^{(k)} + \nu \bar{\gamma}_{0,i}^{(k)}}{C_i + \nu}, \quad (18)$$

where ν is the weight the neighborhood carries, expressed in crossings and shared across slots and voxels. A voxel with few crossings returns a value close to the neighborhood estimate; one with many reports close to its own. Both means are distributions over the K slots and (18) is a convex combination of them, so the shared weights remain a distribution.

E. Flow Dependence

A voxel's mixture is a *spatially marginal* flow distribution, estimated independently of its neighbors. The sharing

of Section V-D acts on that marginal, sharpening a voxel's estimate of its own distribution from the evidence around it. Reasoning jointly over several voxels from marginals alone, however, admits combinations that continuous agent motion precludes (*e.g.*, an agent heading east at one voxel and west at the next). Continuous motion instead correlates flow direction across neighboring voxels. Directional and spectral maps of dynamics leave this dependence unmodeled, estimating each cell independently [2], [7], [15], [16]; we capture it with a pairwise coupling.

We assume this dependence is local: a voxel's flow is linked only to its immediate neighbors, so two non-adjacent voxels are conditionally independent given the voxels separating them, and couplings are stored only between directly adjacent voxels (*i.e.*, the state grows with the traversable edges rather than with pairs of cells).

As in Section V-D, adjacency is *traversable* adjacency, narrowed to face-sharing pairs. The dependence is induced by one passage crossing both cells in turn: face-sharing supplies that contiguity, while traversability confines the pair to one region of free space. The coupled field is then a Markov random field over the graph these pairs define, carried on the traversability relation the 3DSG already encodes.

1) *Joint and Conditional*: Let $k_i, k_j \in \{1, \dots, K\}$ denote the slots the flow occupies at voxels $\mathbf{i}$ and $\mathbf{j}$. For each slot pair, we introduce a coupling parameter $\phi_{ij}^{(k_i, k_j)}(t) \in \mathbb{R}$ and model the joint distribution over two adjacent voxels as their independently-estimated mixing weights, reweighted by it:

$$p(k_i, k_j | t) \triangleq \frac{1}{Z_{ij}(t)} \pi_i^{(k_i)}(t) \pi_j^{(k_j)}(t) \exp(\phi_{ij}^{(k_i, k_j)}(t)), \quad (19)$$

where $Z_{ij}(t)$ sums the numerator over the K^2 slot pairs (64 terms at $K = 8$, so evaluated directly). The coupling acts as a log-bias on the independent product: at $\phi = 0$ the joint reduces to the product of the two weight vectors (independent voxels), positive ϕ raises a slot pair's joint probability (co-occurrence, *e.g.*, aligned headings), and negative ϕ lowers it (incompatibility). Conditioning on $\mathbf{i}$ occupying slot k_i gives

$$p(k_j | k_i, t) \propto \pi_j^{(k_j)}(t) \exp(\phi_{ij}^{(k_i, k_j)}(t)), \quad (20)$$

that is, $\mathbf{j}$'s own weights are sharpened on the slots that tend to co-occur with k_i and suppressed elsewhere. The flow density at $\mathbf{j}$ follows as

$$p_j(\theta, \rho | k_i, t) = \sum_{k_j} p(k_j | k_i, t) p_j^{(k_j)}(\theta, \rho | t). \quad (21)$$

Conditioning is symmetric: fixing k_j instead yields $p(k_i | k_j, t)$ by the same construction. Where the marginal mixture (10) gives the flow at a voxel on its own, the conditional (21) gives the flow at $\mathbf{j}$ once the flow at $\mathbf{i}$ is known.

2) *Estimating the Coupling*: The coupling is learned online from paired observations on two timescales. The shorter one sets what constitutes a pair: detections in adjacent voxels $\mathbf{i}$ and $\mathbf{j}$ form a pair on edge $(\mathbf{i}, \mathbf{j})$ when they fall within a coherence window $\Delta t_{\text{coh}} = 1$ s of each other, short enough that both plausibly belong to the same passage of flow across the two cells. The longer one sets how often the coupling is measured:

pairs are grouped into consecutive *sampling windows* of length Δt_ϕ , each yielding one estimate of ϕ per slot pair. Within a window w , the outer product of each pair's responsibilities (12) accumulates into a $K \times K$ co-occurrence tensor

$$\Omega_{ij}^{w, (k_i, k_j)} \triangleq \sum_{n \in w} r_{i,n}^{(k_i)} r_{j,n}^{(k_j)}, \quad (22)$$

over the N_{ij}^w pairs the window holds. Each pair spreads a unit of mass across the tensor ($\sum_{k_i, k_j} r_{i,n}^{(k_i)} r_{j,n}^{(k_j)} = 1$), so Ω_{ij}^w is a soft co-occurrence count: the fractional number of pairs in the window in which voxel $\mathbf{i}$ occupied slot k_i while $\mathbf{j}$ occupied k_j . Because ϕ is exactly the log-departure from independence in (19), its natural estimator is the pointwise mutual information [34], the log-ratio of observed co-occurrence to the count expected under independence [35]:

$$\hat{\phi}_{ij}^{w, (k_i, k_j)} \triangleq \log \frac{\Omega_{ij}^{w, (k_i, k_j)} + \epsilon}{\Omega_{ij, \text{ind}}^{w, (k_i, k_j)} + \epsilon}, \quad (23)$$

where $\Omega_{ij, \text{ind}}^{w, (k_i, k_j)} = \bar{\pi}_i^{w, (k_i)} \bar{\pi}_j^{w, (k_j)} N_{ij}^w$ is the co-occurrence expected if the two slots were independent, with $\bar{\pi}^w$ the mean responsibility vectors over that same window and ϵ a small constant value for smoothing sparse pairs. When a window holds little evidence, both terms vanish together and the smoothing drives $\hat{\phi}_{ij} \rightarrow 0$, so a barely-observed edge defaults to independence. Swapping the two voxels transposes both tensors, so $\hat{\phi}_{ij}^{(k_i, k_j)} = \hat{\phi}_{ji}^{(k_j, k_i)}$: the edge is undirected, consistent with the symmetric joint (19).

Each closed window contributes one sample $\hat{\phi}_{ij}^w$, timestamped at the window's mid-time, to the corresponding NUDFT predictor (Section V-C). Every sample therefore measures the edge's departure from independence over one bounded stretch of time, and the predictor's mean term γ_0 averages those measurements while its harmonic terms fit their variation through the day. To resolve the rhythms being fitted we set the window from the shortest candidate period, $\Delta t_\phi = \min_f T_f / S$ with $S = 12$ samples per cycle, tying the sampling rate to $\{T_f\}$. A window holding a single pair returns $\hat{\phi} = 0$ identically, since its observed co-occurrence and its own marginal product coincide. We therefore keep a window that has not reached $N_{\min}$ pairs open past its nominal span until it does, so frequently observed edges sample at the nominal rate while sparse edges lengthen their own windows.

F. Presence Estimation

The mixture of Section V-B describes the flow distribution conditional on an agent being present; the occupancy intensity (2) of that voxel, and the presence probability derived from it, are estimated separately.

At any fixed location, an agent's motion reduces to three events: entering the voxel that represents the location, remaining for the time taken to cross it, and exiting. Across many agents, a voxel is entered, occupied, and vacated repeatedly, its occupancy set by how often agents enter and how long each stays. We formulate this motion with queuing theory [36]: the voxel is the system, each passing agent a customer, and the crossing time (the dwell) the service time. The superposition

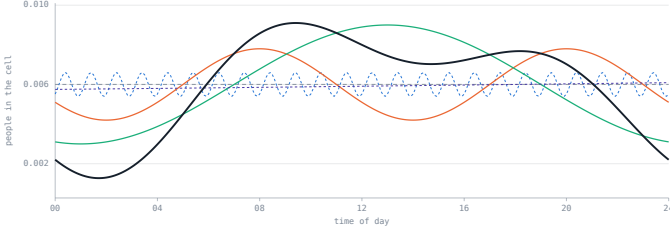


Fig. 3: **Presence channel** in a voxel's state read out across one day, for representative values. Dashed is the stored mean, and the four thin curves are the candidate rhythms of period 1 h (blue), 12 h (orange), 24 h (green) and one week (violet), solid where reported and dashed otherwise; the bold curve is the sum of the reported ones.

of many independent, sparsely visiting trajectories yields Poisson arrivals with time-varying rate $\lambda_a(t)$; crossing times are heterogeneous across agents, so the service-time distribution is left general; and agents traverse a voxel without queuing for access, so the number served concurrently is unbounded, an idealization that costs little since a fine cell holds about one agent at a time and a crowd appears as many occupied voxels. These three assumptions specify an $M_t/G/\infty$ queue.

Within the queuing formalism, the occupancy L (the mean number of agents present) is the arrival rate times the mean crossing time, $L = \lambda_a \tau_{\text{cross}}$ [37]. Under a time-varying rate this becomes $\int_0^\infty \lambda_a(t-s) (1-G(s)) ds$, of which the product is the quasi-stationary approximation, exact whenever the rate is constant over one crossing. A voxel is crossed in a fraction of a second, orders of magnitude below the shortest candidate period, so we apply the identity pointwise at each query time. We estimate L and τ_{cross} from observations and invert this relation to recover the arrival rate $\lambda_a(t) = L(t)/\tau_{\text{cross}}$.

Occupancy is estimated online per voxel with a conjugate Gamma-Poisson model [3]: a weakly informative Gamma prior on the detection rate per unit visible time, $\lambda_{\text{vis}} \sim \text{Gamma}(\alpha_0, \beta_0)$ with $\alpha_0 = \beta_0 = 1$. Conjugacy keeps the posterior Gamma, so both parameters update in closed form, $\alpha \leftarrow \alpha + 1$ per detection and $\beta \leftarrow \beta + \Delta t_{\text{vis}}$ per frame in which the voxel is visible, where Δt_{vis} is the frame's visible time in seconds, making β_0 one second of prior exposure. Multiplying the posterior mode $\hat{\lambda}_{\text{vis}} = (\alpha - 1)/\beta$ by the mean frame duration $\bar{\tau} = (\beta - \beta_0)/n_{\text{vis}}$ gives a mean count per frame, the occupancy $\hat{L}(t) = \hat{\lambda}_{\text{vis}}(t) \bar{\tau}$, realizing the occupancy intensity (2) over the voxel's extent. Once accumulated visible time dominates β_0 , the occupancy is just detections divided by frames in which the voxel was seen, $\hat{L} \approx (\alpha - 1)/n_{\text{vis}}$, empty frames included. We forecast $\hat{\lambda}_{\text{vis}}(t)$ with the spectral predictor (Section V-C), since the static count $(\alpha - 1)/n_{\text{vis}}$ cannot follow a time-varying rate (Fig. 3).

Writing $\bar{\rho}_i$ for the mean speed at voxel i , pooled across slots—a single time-invariant scalar per voxel—the crossing time follows from the geometry: an agent crossing a voxel of side s_v at speed $\bar{\rho}_i$ occupies it for $\tau_{\text{cross}} = s_v/\bar{\rho}_i$. The arrival rate is then $\lambda_a(t) = \hat{L}(t) \bar{\rho}_i/s_v$. In an $M_t/G/\infty$ queue the number of agents in the voxel at time t is Poisson with mean $\hat{L}(t)$, and arrivals after t are independent of it. The probability that at least one agent occupies the voxel at some point in a horizon

Δt from t therefore has both contributions in one exponent:

$$\begin{aligned} p_{\text{present}}(t, \Delta t) &= 1 - \exp(-\lambda_a(t) (\Delta t + \tau_{\text{cross}})) \\ &= 1 - \exp\left(-\hat{L}(t) \left(1 + \frac{\bar{\rho}_i \Delta t}{s_v}\right)\right). \end{aligned} \quad (24)$$

As $\Delta t \rightarrow 0$ this reduces to the instantaneous occupancy $1 - e^{-\hat{L}(t)}$, which is also the value returned when no speed estimate is available and the arrival term drops.

Presence and flow then combine into the full joint distribution as the two factors of a hurdle model, presence multiplying the flow density it conditions:

$$p(\theta, \rho, \text{present} \mid t, \Delta t) \triangleq p_{\text{present}}(t, \Delta t) p(\theta, \rho \mid \text{present}, t). \quad (25)$$

This instantiates the readout (6) at a cell. The first factor is the presence probability induced over the cell's extent and horizon Δt by the occupancy intensity λ of (2), realized per voxel as $\hat{L}(t)$; the second is the flow distribution of (3). The pair is the dynamics field $\hat{\mathcal{D}}$ at the queried cell and time, returned as a single joint over presence and velocity.

VI. HIERARCHICAL FLOW REPRESENTATION

The estimates of Section V are held at voxel resolution. We lift them up the layers of Section IV-A so that flow becomes a native attribute of the representation, carried by the navigational layer $\mathcal{V}_{\text{nav}}^t$ and the structure built on it.

A. Graph-Level Flow Dynamics

1) *Node-Level Aggregation*: Each navigational node aggregates the flow voxels of its support set $\mathcal{S}$ (Section IV-A). The flow distribution over a region is the intensity-weighted mixture of the distributions of the cells composing it,

$$p_{\mathcal{S}}(\theta, \rho \mid t) \triangleq \frac{\sum_{i \in \mathcal{S}} \hat{L}_i(t) p_i(\theta, \rho \mid t)}{\hat{L}_{\mathcal{S}}(t)}, \quad (26)$$

with $\hat{L}_{\mathcal{S}}(t) \triangleq \sum_{i \in \mathcal{S}} \hat{L}_i(t)$ the occupancy of the support region. The weights are the cells' occupancies because the joint intensity is additive over the region, $\Lambda_{\mathcal{S}} = \sum_i \lambda_i p_i$, and (26) is that sum normalized by $\lambda_{\mathcal{S}}$ as in (3). The mixture therefore carries no prediction horizon: how far ahead presence is queried does not enter the distribution over headings.

Given the fixed $K=8$ geometry, slot k spans the same heading band in every voxel, and the components differ across voxels only in their radial mean $\mu_{\rho, i}^{(k)}$ (Section V-B). Marginalized over speed, (26) is therefore a mixture of the same K wrapped normals, and the heading weights reduce exactly to a weighted sum per slot:

$$\pi_{\mathcal{S}}^{(k)}(t) = \frac{\sum_{i \in \mathcal{S}} \hat{L}_i(t) \pi_i^{(k)}(t)}{\hat{L}_{\mathcal{S}}(t)}, \quad k = 1, \dots, K, \quad (27)$$

so the aggregation is index-aligned. The joint over heading and speed, however, has $K|\mathcal{S}|$ distinct components, one per slot per voxel. We summarize it under the same K slots by aggregating the radial means with the same weights:

$$\mu_{\rho, \mathcal{S}}^{(k)}(t) \triangleq \frac{\sum_{i \in \mathcal{S}} \hat{L}_i(t) \pi_i^{(k)}(t) \mu_{\rho, i}^{(k)}}{\sum_{i \in \mathcal{S}} \hat{L}_i(t) \pi_i^{(k)}(t)}, \quad (28)$$

the speed mean of slot k pooled over the support region. This is exact in the mean of each slot and approximates its spread.

The presence channel aggregates through the expected count rather than per-voxel. Presence at region scale is

$$p_{\text{present},S}(t, \Delta t) \triangleq 1 - \exp\left(-\hat{L}_S(t)\left(1 + \frac{\bar{\rho}_S \Delta t}{\ell_S}\right)\right), \quad (29)$$

with ℓ_S the extent of the support region and $\bar{\rho}_S$ its occupancy-weighted mean speed. This reduces to (24) for a single-voxel support set. It is not the union of the cells' independent presences, which would count an agent once per cell it transits; the region crossing time $\ell_S/\bar{\rho}_S$ counts that agent once.

These quantities are written into node metadata as the summary tuple $(\hat{L}_S, \pi_S, \theta_{\text{dom}}, \rho_{\text{dom}}, c_{\text{dir},S}, t_{\text{latest}})$, where $\theta_{\text{dom}}, \rho_{\text{dom}}$ are the dominant heading and speed, $c_{\text{dir},S}$ the directional concentration of the region, and t_{latest} the most recent time the node was observed. Storing the occupancy $\hat{L}_S$ rather than a presence probability lets a query supply its own Δt to (29) at read time.

The concentration is the occupancy-weighted mean over S , of each voxel's dominant-slot weight $\max_k \pi_i^{(k)}$:

$$c_{\text{dir},S}(t) \triangleq \frac{\sum_{i \in S} \hat{L}_i(t) \max_k \pi_i^{(k)}(t)}{\hat{L}_S(t)}. \quad (30)$$

It downweights rarely occupied voxels and measures how well-defined the flow is across the node's active region, ranging from $1/K$, where every voxel spreads its mass evenly over the headings, to 1, where each commits to a single slot.

These per-node attributes form a planner-facing interface: $\hat{L}_S$ and θ_{dom} let a route be scored by expected encounters and preferred travel direction, while π_S and $c_{\text{dir},S}$ expose the full directional mix and how sharply voxels commit to it.

2) *Edge Flow Annotation*: For each intra-layer edge (u, v) whose endpoints carry flow summaries, forward and reverse flow alignment scores are computed by projecting each slot's heading onto the edge direction:

$$F_{\text{fwd}}(t) \triangleq \frac{1}{2} \sum_{q \in \{u,v\}} \sum_k \pi_q^{(k)}(t) \hat{L}_q(t) \max(0, \hat{\mathbf{e}}_{uv} \cdot \mathbf{d}^{(k)}), \quad (31)$$

where $\hat{\mathbf{e}}_{uv}$ is the unit edge direction and the direction of slot k , $\mathbf{d}^{(k)} = (\cos \mu_\theta^{(k)}, \sin \mu_\theta^{(k)})^\top$, common to both endpoints since the slot geometry is shared across the map (Section V-B). The reverse score replaces $\hat{\mathbf{e}}_{uv}$ with $-\hat{\mathbf{e}}_{uv}$. Scaling by the occupancy keeps the score linear in the traffic each endpoint carries, so scores stay comparable across edges. These augment the navigational edges of Section IV-A, together with average speed and a bidirectionality term.

3) *Consistency under Optimization*: On every backend optimization the voxel set is re-aligned to the corrected geometry, each cell's index is moved and collisions are pooled. This realizes the positional action of (8) under any warp. Because the slot geometry is fixed globally, with slot k denoting the same heading band in every voxel (Section V-B), a correction that rotates one region relative to another both displaces that region's cells and turns the headings they store by the same angle. The rotation is therefore re-expressed on the fixed slots. Each cell takes the rotation the correction applies at its

own location from the control points it is deformed against, and its per-slot state is resampled onto the same slots —by cyclic permutation of the weight vector when the angle is multiple of the $\pi/4$ spacing, and by interpolation between adjacent slots otherwise. The mixing-weight predictors receive one responsibility vector per crossing at times common to every slot, and their mean and harmonic terms are linear in the value fed, so resampling those coefficients is therefore exact, and returns the same state as accumulating the rotated observations. Per-slot speeds are permuted to the nearest slot, since each speed predictor receives only the crossings assigned to its own slot; this leaves at most half the slot spacing of misassignment on that channel. Presence carries no direction and is unaffected. Both the move and the turn are computed against the cell's original index and frame, so (9) holds wherever the warp is locally a rigid motion whose rotation is a yaw, $\mathbf{J}_W \in SO(2)$. A stored heading is planar, so an out-of-plane component of a correction has no representation on the slots. Rotating a component of (10) by δ displaces its mean, so a component left in the frame its observations were accumulated in diverges from the corrected density by $D_{\text{KL}} = \delta^2/2\sigma_\theta^2$, and by convexity the mixture by at most that.

When voxels $\mathbf{i}_a, \mathbf{i}_b$ deform to the same corrected index, they are fused into a single estimate. Every accumulated quantity $x \in \{\gamma_0, \gamma_f, \mu_\rho^{(k)}\}$ is pooled as a sample-weighted average,

$$\text{pool}(x_a, x_b) \triangleq \frac{N_a x_a + N_b x_b}{N_a + N_b}, \quad (32)$$

the value a single voxel of $N_a + N_b$ samples would hold. The count is the one the pooled quantity is accumulated over: the crossings C of Section V-C on the mixing-weight channels, and the detections routed to a slot on that slot's speed channel. The Gamma-Poisson presence counts add, with the shared prior (α_0, β_0) subtracted once to avoid double-counting: $\alpha_{ab} = \alpha_a + \alpha_b - \alpha_0$ and $\beta_{ab} = \beta_a + \beta_b - \beta_0$. Couplings are addressed by their endpoints' indices, so the same re-keying carries them: an edge follows the corrected indices of the two cells it relates, and edges arriving at one pair pool their co-occurrence (22) and their coefficients as above.

B. Spatial Motion Semantics

To expose higher-level crowd structure, per-node flow summaries drive a *motion archetype* layer inserted above the navigational nodes. Each archetype node groups spatially adjacent navigational nodes sharing a motion-semantic class: *Static*, *Unimodal*, *Bimodal*, *Intersection/Multimodal*, or *Diffuse*.

Nodes whose support set holds no observed voxel are excluded. A node is *Static* if the presence probability (29) induced by its occupancy at a fixed classification horizon decreases below p_{min} ; since (29) is monotone in $\hat{L}_S(t)$, this is equivalently a bound on occupancy. The remaining classes follow from the number of dominant slots $d_S(t) \triangleq \sum_k \mathbf{1}[\pi_S^{(k)}(t) \geq \tau_w]$ with $\mathbf{1}(\cdot)$ the indicator function: *Unimodal*, *Bimodal*, or *Intersection/Multimodal* for $d_S(t) = 1, 2$, or ≥ 3 , and *Diffuse* for $d_S(t) = 0$, where the node is occupied but no heading dominates. The tests are evaluated in order and the classes are disjoint.

TABLE II: **Evaluation datasets.** TBD [38] is collected by a real moving robot and exercises the embodied pipeline end to end; ATC [1] and HB [39] supply the long-horizon recurrence TBD lacks, and are driven through a *grounded simulation*.

Dataset	Extent	Train / Test	Candidate periods	Primary role
TBD	~213 min over 3 mo.	3 / 3 days	{60, 300, 600} s	flow prediction, cost
ATC	92 days over ~1 yr	84 / 8 days	{1, 12, 24, 168} h	flow, sparse coverage, loop closure, coupling
HB	11 mo. continuous	227 / 28 days	{1, 12, 24, 168} h	presence, periodicity, coupling

VII. EXPERIMENTAL EVALUATION

We evaluate Kairos along several axes. Section VII-A describes the datasets and the online scoring protocol. We then compare the predicted flow distributions against grid- and graph-based baselines under identical scoring (Section VII-B) and under sparse embodied coverage (VII-C), and test whether the learned state survives map corrections by a SLAM backend (VII-D) —the property that distinguishes a grounded forecast from one held in a fixed external frame. We evaluate the presence forecasts against a dedicated spectral occupancy model (VII-E) and close with the computational cost the flow layer adds over the underlying 3DSG (VII-F). Finally, Section VII-G applies the representation to a planning task.

A. Datasets and Protocol

Table II summarizes the datasets, the train/test split, and the two observability regimes under which all results are reported.

1) *Datasets*: Evaluating Kairos requires a dataset that simultaneously offers (i) a *moving* sensing platform, (ii) RGB-D data, (iii) annotated or recoverable pedestrian trajectories, and (iv) multi-session temporal structure (repeated visits and time-of-day variation). No single dataset offers all four, so we evaluate on multiple, each covering a different aspect of the framework. The TBD pedestrian dataset [38] exercises the embodied pipeline end to end, from a real moving ZED stereo camera at 10 Hz with trajectories lifted directly to 3D (16 recordings, ~213 min over six days across three months). It supplies (i)-(iii) and the repeated sessions of (iv) but no recurrence at the periods those sessions resolve, so its results measure the end-to-end feasibility and the relative ordering of methods under identical online scoring.

Long-horizon forecasting is evaluated on ATC [1] and Havlíčkův Brod (HB) [39] via *grounded simulation*: a simulated robot with bounded field of view is driven over real recorded trajectories and floor maps, reproducing the online, partial-observability setting Kairos targets. The platform patrols a serpentine coverage loop over the navigation graph authored on the floor plans at 1.5 m/s, capturing a small fraction of the recorded detections while its short revisit period exposes every cell at every phase of the daily cycle. ATC is a large shopping mall, its pedestrian trajectories recorded at 10 Hz on 92 days across roughly a year. HB is a railway-station concourse recorded continuously for eleven months, with full wall-clock timestamps. Its ~1 Hz trajectories are interpolated per track to 10 Hz to match the other datasets.

2) *Evaluation Protocol*: On TBD we train on 3 days from the first two months and test on 3 held-out days from the final month, so tests are temporally disjoint from training

and separated from it by weeks. On ATC we train on 84 of the 92 recorded days and test on the 8 held-out. On HB we train on the 227 dates with complete camera coverage (Sept. 2022 through Apr. 2023) and test on the 28 days that follow. Following established convention [2], [17], candidate periods are set to each dataset’s recurrence scale, as shown in Table II.

Within these candidate sets the prequential gate of Section V-C selects a spectral order per cell online. On ATC and HB the weights gate validates the direction harmonics at a median order of 1 and the presence gate at orders 1–2. On TBD the presence gate holds at order 0, matching the absence of an occupancy rhythm, while the weights gate concentrates at the maximum order; since the test sessions are recorded weeks later, at which lag harmonics of 60–600 s carry arbitrary phase, this reflects the training signal rather than forecastable periodicity. Spatial and angular discretization is common to all methods: every baseline is evaluated at Kairos’s 0.4 m cell size and the histogram methods at 8 orientation bins.

3) *Observability Regimes*: For the grounded simulations we define two regimes. In the *embodied* regime the platform observes only detections inside its field of view (~4% of the ATC corpus, ~7% of HB); in the *full-visibility* regime, where the estimator is trained on every detection. All main results are embodied; full-visibility results appear only where marked, to measure how a mechanism responds to added observability.

B. Flow Prediction

1) *Baselines*: We compare Kairos’s flow distributions against four state-of-the-art methods for representing pedestrian flow, spanning the two spatial substrates Kairos draws on: grid-based and 3D scene graph. **CLiFF-map** [7]: an offline, batch SW-GMM with no temporal component; a temporally-agnostic SW-GMM reference for directional density estimation. **STeF-Map** [17]: a uniform-grid temporal model with per-cell spectral predictors over orientation histograms; the closest grid-based analogue to our temporal prediction. **Aion** [16]: a 3DSG method attaching discrete per-node orientation histograms with spectral temporal prediction; it isolates a scene-graph substrate carrying a *binned* directional representation rather than a continuous one. **Rheos** [15]: a 3DSG method using the same SW-GMM as Kairos, fit online by proximity-based association but without temporal forecasting; it isolates the contribution of temporal prediction.

2) *Evaluation Metrics*: Kairos predicts a full distribution over flow and presence at future times, so we evaluate with proper scoring rules [40] as they reward calibrated densities and penalize overconfidence.

Mean Log Predictive Density (MLPD), in nats. The mean log-density a model assigns to flow observations under its predicted distribution. For each held-out observation (θ_n, ρ_n) :

$$\text{MLPD} \triangleq \frac{1}{N_{\text{test}}} \sum_{n=1}^{N_{\text{test}}} \log p((\theta_n, \rho_n) \mid \hat{\mathcal{M}}_{\Xi(\mathbf{x}_n)}(t_n)), \quad (33)$$

where $\hat{\mathcal{M}}_{\Xi(\mathbf{x}_n)}(t_n)$ is the predicted mixture at the voxel containing $\mathbf{x}_n$, evaluated at query time t_n . We report it per channel —heading $(p(\theta), H)$, speed $(p(\rho), S)$, and joint $(p(\theta, \rho), J)$ —

TABLE III: **Flow prediction** on TBD [38], ATC [1] and HB [39]. MLPD (nats) per channel, circular CRPS of the heading forecast (rad) and speed MAE (m/s). *Cov.* is the fraction of test detections a method’s own representation covers; uncovered detections receive the uniform floor, whose value the *Uniform* row reports. Heading-only baselines carry no speed model (—); × marks a method that cannot be fit at a corpus’s scale. Bold marks the best method per column.

Method	TBD						ATC						H					
	MLPD ↑			CRPS ↓	MAE ↓	Cov. ↑	MLPD ↑			CRPS ↓	MAE ↓	Cov. ↑	MLPD ↑			CRPS ↓	MAE ↓	
	Cov. ↑	H	S				H	S	J				H	S	J			
<i>Uniform</i> (reference)	—	−1.84	−1.10	−2.94	0.79	—	—	−1.84	−1.10	−2.94	0.79	—	—	−1.84	−1.10	−2.94	0.79	—
CLiFF-map† [7]	1.0	−2.18	−2.14	−4.67	0.81	0.50	×	×	×	×	×	×	×	×	×	×	×	×
STeF-Map† [17]	1.0	−3.52	—	—	0.82	—	1.0	−1.43	—	—	0.71	—	1.0	−1.38	—	—	0.75	—
Aion‡ [16]	0.94	−5.64	—	—	0.91	—	1.0	−1.75	—	—	1.02	—	1.0	−1.75	—	—	0.89	—
Rheos‡ [15]	0.88	−4.78	−3.08	−7.07	0.88	0.50	×	×	×	×	×	×	×	×	×	×	×	×
Kairos‡ (ours)	0.93	−1.90	−0.94	−2.87	0.84	0.42	0.97	−1.48	−0.16	−1.61	0.71	0.19	0.97	−1.47	−0.65	−2.10	0.76	0.33

Notes: † indicates *full observability* regime; ‡ indicates *embodied path* regime.

so heading-only baselines (STeF-Map, Aion) are scored on the first alone and full-distribution methods on all three.

MLPD is computed over *all* held-out detections. A detection at a location unseen during training receives a uniform-density floor $\log p_{\text{floor}} = -\log(2\pi) - \log \rho_{\text{max}}$ over $[0, 2\pi) \times [0, \rho_{\text{max}}]$, with $\rho_{\text{max}} = 3.0$ m/s, the physical pedestrian speed cap. The floor is shared across methods so coverage is penalized identically and no method gains by predicting only where it is confident. The heading floor $-\log(2\pi) = -1.84$ and the joint floor $-\log(2\pi\rho_{\text{max}}) = -2.94$ are therefore the score charged for an uncovered detection. A covered detection is scored under the predicted density, clamped at 10^{-9} (−20.72 nats), for all methods so that no near-zero density dominates the mean. We report the match rate (fraction of detections covered) alongside MLPD.

Circular Continuous Ranked Probability Score (CRPS), in radians. The CRPS scores the entire predicted heading distribution by its arc distance to the observed heading, so mass placed near the observation improves the score even when the observation itself is not matched exactly. It complements MLPD, which evaluates the density only at the observed heading and cannot distinguish mass placed 30° from mass placed 180° away. For circular variables [41],

$$\text{CRPS}(F, \theta_n) \triangleq \mathbb{E}_F d(\Theta, \theta_n) - \frac{1}{2} \mathbb{E}_F d(\Theta, \Theta'), \quad (34)$$

where d is arc distance on the circle and $\Theta, \Theta' \sim F$ are independent draws from the predicted heading distribution: the expected arc distance of the forecast mass from the observation θ_n , minus a spread term that keeps the score proper. It is distance-aware and bounded by π , and a uniform forecast scores $\pi/4 = 0.79$ rad, which is also the floor assigned to uncovered detections, mirroring the MLPD coverage policy.

Speed Mean Absolute Error (MAE), in m/s. The mean absolute error between predicted and observed per-detection speed; a point error is meaningful here, unlike for heading, since per-slot speed is unimodal. Unlike previous metrics, it is computed on matched detections only, so it carries no coverage penalty and the subsets differ across methods. STeF-Map and Aion carry no speed model and are omitted from this metric.

3) *Results:* On TBD, Kairos achieves the best MLPD in every channel and the lowest speed MAE (Table III), despite running embodied against baselines given full observability.

Against STeF-Map, Kairos lowers the heading loss from −3.52 to −1.90, a 1.62 nat gain corresponding to roughly $5\times$ the density assigned to held-out headings; and its speed MAE of 0.42 m/s is 16% below CLiFF-map’s 0.50 m/s while covering fewer detections (0.93 against the grids’ 1.0). No method improves on the uniform reference of heading (−1.90 against −1.84) or CRPS (0.81–0.91 against 0.79): the hall’s flow carries no recurring directional structure at the periods 213 minutes of recording can resolve. Thus, the informative channels for TBD are speed and the joint.

Rheos shares Kairos’s SW-GMM representation yet loses 4.2 nats on the joint (−7.07 against −2.87). The cause is on-line covariance estimation: a fraction of per-voxel covariances become near-degenerate, concentrating the joint density on a one-dimensional ridge that assigns near-zero probability to off-ridge observations. The batch fit of CLiFF-map is unaffected, placing the cause in the incremental estimation; Kairos’ fixed diagonal components (Section V-B) have no off-diagonal term that can degenerate.

On ATC and HB, STeF-Map achieves a slightly better heading MLPD (−1.43/−1.38 versus −1.48/−1.47). The gap traces to the fixed-width heading kernel ($\sigma_\theta = 0.4$ rad), which spreads each training vote and each query density across adjacent slots: Kairos assigns each cell’s rarely-used sectors more mass than their empirical share, and correspondingly less mass to the dominant directions that carry most of the traffic, where STeF-Map’s histogram concentrates. On the distance-aware CRPS the two methods are at parity, and the table’s marginal gap comes from the uniform floor assigned to Kairos’s unmatched detections. Kairos additionally predicts the heading marginal jointly with calibrated speed and presence densities, neither of which the grid baseline provides.

The training corpora reach 3.3×10^9 (ATC) and 3.2×10^8 (HB). CLiFF-map’s per-cell mean-shift and EM buffer the full corpus and scales quadratically in the samples per cell, beyond practical memory at this scale [7]; Rheos fits the same SW-GMM incrementally, but scoring the full joint density at ATC’s $\sim 3.4 \times 10^8$ test detections requires weeks of compute, and HB’s training stream alone exceeds a month. Both therefore appear on TBD only (× in Table III).

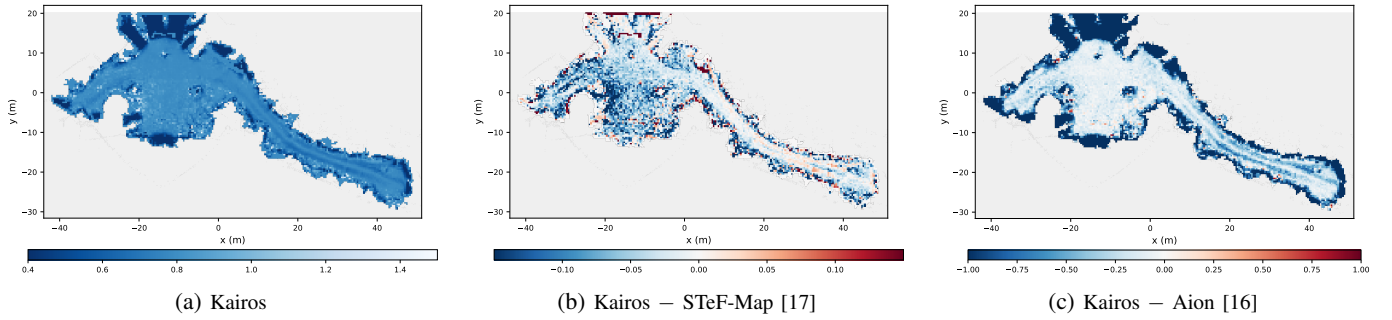


Fig. 4: **Per-cell map quality under sparse embodied coverage on ATC [1].** Circular heading CRPS per cell; gray is unobserved structure. (a) Kairos (ours). (b)–(c) per-cell difference against each baseline.

TABLE IV: **Map quality under sparse embodied coverage on ATC [1].** Heading CRPS and heading MLPD (H).

Method	CRPS $\downarrow$ [95% CI]	H MLPD $\uparrow$
STeF-Map [17]	0.76 [0.75, 0.76]	−4.28
Aion [16]	1.33 [1.32, 1.34]	−5.21
Kairos (ours)	0.72 [0.71, 0.72]	−1.62

TABLE V: **Loop-closure robustness on ATC [1],** under a drift model that both displaces and rotates the pose estimate.

	Per detection			Per node	
	H $\uparrow$	J $\uparrow \downarrow$	CRPS $\downarrow$	$\Delta\theta_{\text{dom}}$ ($^\circ$) $\downarrow$	$> 22.5^\circ$ (%) $\downarrow$
Odometry					
<i>Drift-free</i>	−1.47	−1.56	0.71	—	—
Uncorrected	−1.58	−1.69	0.75	22.8	50.5
Loop closure	−1.49	−1.58	0.72	9.5	26.1

C. Robustness Under Sparse Embodied Coverage

The comparison of Section VII-B weights every held-out detection equally, so the few high-traffic cells that every method learns dominate the mean, and Kairos and STeF-Map reach equal CRPS. This section instead weights every cell equally, in a sparse-mapping regime [17], so that rarely-observed cells contribute equally with high-traffic ones. The ATC floor is partitioned by k -means into 24 regions of comparable area and the recorded day into 10 min decision intervals. The patrol observes one region per interval and nothing for the remainder of the day, giving an observation budget measured in intervals, set to 4 of a day’s ~ 67 intervals (6.25%). At each decision it drives to the chosen region over the traversable graph at 1 m/s and covers it with a serpentine loop until the next one. Over the 84 training days each region is observed approximately 14 times: one 10-minute window every six days.

Kairos achieves the lowest per-cell heading CRPS (Table IV), 0.72 against STeF-Map’s 0.76 and Aion’s 1.33. The gap widens in MLPD: a sparse patrol (6.25%) leaves few detections in a peripheral cell, so most of its 45° sectors receive no mass and a held-out heading falling in one of them is scored far below uniform; recorded as -4.28 and -5.21 against Kairos’s -1.62 , with the uniform reference at -1.84 . Kairos’s fixed-width kernel distributes graded mass around each observed heading, retaining a calibrated forecast in cells with few observations. Fig. 4 resolves the comparison spatially: Kairos obtains the lower CRPS in 71% and 96% of cells against STeF-Map and Aion respectively, with the largest margins in the sparsely observed periphery and side passages.

D. Loop Closure Robustness

Odometry drift displaces the map a robot builds, and each loop closure warps it from the previous estimate onto the corrected one. Kairos keys its flow state to the 3DSG, so the flow voxels follow that correction (Section VI-A).

We inject drift into the position and the heading of the pose estimate of the ATC grounded simulation while the platform patrols in the embodied regime of Section VII-A. A loop closure is registered at every physical revisit, where corrections are computed from the injected drift and delivered as the time-stamped control-point warp the backend emits. The same detection stream, spanning the full 84 training days, is processed three times: under perfect odometry (*drift-free*), under drift corrected at every loop closure, and under drift left uncorrected within the day. The uncorrected run is re-anchored only at the start of the next day, as a platform re-localized on start-up would be.

The corrected run drifts by up to 0.6 m (1.5 voxels) and 0.77° over each 659 m patrol loop, and every completed loop triggers a correction. The uncorrected run is assigned a twentieth of each rate, compounding to ≈ 3 m (7.5 voxels) and 4.2° by the end of the day. Both rates are conservative: state-of-the-art odometry drifts by 0.1% to 0.5% of distance traveled and by 0.0013° to 0.0033° per metre, for LiDAR-inertial and visual-inertial systems respectively [42], [43], [44], so over the ≈ 65 km the platform covers in a day even the lower rate would displace the map by more than the building’s extent. The corrected run drifts at 0.09% between corrections and the uncorrected run at a twentieth of that, below any published system; its 4.2° heading error alone displaces the far side of the 54 m floor by a further 4.0 m.

Under loop closure the corrected state stays within 0.03 nats of the reference (Table V) in heading MLPD. Uncorrected, and at a drift rate twenty times lower, the degradation is nearly four times larger at 0.11 nats, and circular CRPS rises from the reference’s 0.71 to 0.75. Each run’s state is also aggregated onto the nodes of the navigation graph and compared against the reference’s at the nodes both support, through the angle $\Delta\theta_{\text{dom}}$ between the two dominant directions, reported as its median and as the share of nodes exceeding 22.5° , half a

TABLE VI: **Presence estimation** on HB [39] across event windows Δt : MLPP (in nats), reliability (Rel), and resolution (Res); all in units of 10^{-3} . Reliability and resolution are shown from 60 s onward, the windows in which they separate the estimators.

Regime	Model	$\Delta t = 5$ s	$\Delta t = 10$ s	$\Delta t = 60$ s			$\Delta t = 300$ s			$\Delta t = 600$ s		
		MLPP $\uparrow$	MLPP $\uparrow$	MLPP $\uparrow$	Rel $\downarrow$	Res $\uparrow$	MLPP $\uparrow$	Rel $\downarrow$	Res $\uparrow$	MLPP $\uparrow$	Rel $\downarrow$	Res $\uparrow$
<i>Embodied</i> <i>5 m FoV</i>	Constant base rate	−38	−64	−222	0.1	0.0	−489	0.8	0.0	−601	1.6	0.0
	Kairos (no dwell)	−45	−85	−421	3.3	0.0	−1421	36.0	0.0	−2153	80.2	0.0
	Kairos (static rate)	−34	−57	−190	0.1	3.6	−407	3.6	26.1	−508	9.0	44.3
	Kairos	− 33	− 56	− 182	0.1	5.1	− 379	1.8	35.5	− 460	4.6	58.9
<i>Full</i> <i>visibility</i>	Constant base rate	−37	−63	−217	0.1	0.0	−481	0.5	0.0	−592	1.0	0.0
	FreMEn [2] occupancy $\dagger$	− 32	− 53	− 172	0.0	5.6	− 343	0.2	41.5	− 393	0.2	72.5
	Kairos (no dwell)	−44	−83	−409	3.1	0.0	−1380	34.0	0.0	−2090	75.9	0.0
	Kairos (static rate)	−33	−56	−185	0.1	3.6	−396	3.8	26.4	−495	9.3	45.2
	Kairos	− 32	−54	−175	0.1	5.2	−361	1.9	37.4	−434	4.6	63.9

Notes: $\dagger$ FreMEn divides each cell’s occupied time by total elapsed time, so time in which the cell was not observed is indistinguishable from time in which it was empty. It is therefore fit only under *full visibility*.

direction slot. Under loop closure the median is 9.5° and a quarter of the nodes exceed the slot; uncorrected, the median is 22.8° and half of them do. Corrections of the form the backend produces therefore preserve the learned state, and the re-keying they require is necessary at any realistic drift rate.

E. Presence Estimation

A 0.4 m voxel is crossed in a fraction of a second, so instantaneous occupancy is near zero and carries no useful signal and the presence forecast is windowed. The target is the event that a cell holds at least one detection within the window $\omega = [t, t + \Delta t]$; we call Δt the *horizon* and evaluate $\Delta t \in \{5, 10, 60, 300, 600\}$ s, the scale at which a navigating robot queries the cells ahead of it. The set is capped at 600 s so every window stays below the shortest candidate period on HB (1 h). Scoring follows the protocol established for spectral occupancy models [2]: the trained state is frozen, and at every step t each covered cell’s predicted occupancy probability for ω is scored against the realized occupancy of that window.

1) *Baselines*: We compare against two baselines and two ablations of our own estimator. **Constant base rate**: the training corpus’s occupancy frequency at each horizon. Calibrated by construction, it carries no cell-level information. **FreMEn [2]**: an independent per-cell spectral occupancy model, the standard temporal-map baseline. **Kairos (no dwell)**: the presence estimator without the dwell correction from the static rate, *i.e.*, $\tau_{\text{cross}} = 0$, isolating what the correction contributes. **Kairos (static rate)**: the presence estimator with the dwell correction but the rate held time-invariant, dropping the gate-selected harmonics and isolating the periodic component.

2) *Evaluation Metrics*: Presence forecasts are probability forecasts of a binary event, so we score them with a proper scoring rule and decompose that score into its calibration and information components.

Mean Log Predictive Probability (MLPP), in nats. The mean logarithmic score [40] of the realized occupancy of ω under the predicted probability. It is maximized only by the true probability and dominated by confident errors on the rare occupied cells.

Calibration and Informativeness. Alongside the MLPP we report *reliability* and *resolution* terms of the vector partition of the Brier score [45], computed on the same forecast. Reliability is the mean squared gap between predicted probabilities

and the occupancy frequencies observed at those probabilities (0 for a perfectly calibrated forecast); resolution is the spread of those observed frequencies around the overall occupancy rate (0 for a constant forecast). Together they separate whether an estimator is calibrated from whether it is informative. We report both from the 60 s horizon onward: at 5 and 10 s every estimator’s reliability is at most 0.1 and its resolution at most 0.2, so neither term separates the estimators there.

3) *Results*: Presence is evaluated on HB: the occupancy channel carries its strongest temporal structure at the concourse, whose continuous recording contains the full daily cycle. ATC is recorded during opening hours only. TBD is visited too sparsely to support an occupancy forecast at all, and is retained for the flow metrics.

In the embodied regime Kairos achieves the best MLPP at every horizon (Table VI), from −33 at 5 to −460 at 600, ahead of its static-rate variant by 1 and by 48 respectively. Kairos is the most informative forecast and the best calibrated: at 600 s, its resolution is 58.9 against the static-rate variant’s 44.3, and its reliability 4.6 against 9.0. These margins grow with the horizon because a longer window integrates the rate over more of the daily profile, where time-varying and state rates differ most. The margin is carried by the harmonics the presence gate validates, at a median order of 1–2 on HB (Section VII-A2): the daily cycle and its first overtone, the component the static-rate variant holds constant by construction.

The dwell correction is what makes this channel informative. Without it (24) reduces to the instantaneous occupancy $1 - e^{-\hat{L}}$, and $\hat{L} \ll 1$ at every cell, so forecasts fall in the near-zero bin regardless of query time or cell. Resolution is 0 at all horizons and MLPP falls to −2153, the only estimator below the constant base rate in both regimes.

At full visibility FreMEn achieves the best MLPP at every horizon (−32 to −393), under two conditions: it is fit on the complete occupancy record of every cell, and the occupancy event is the only measure it models, while Kairos’s state additionally yields expected counts and the direction-resolved flow. Kairos matches it at 5 s (−32) and falls below it as the window grows, from −54 against −53 at 10 s to −434 against −393 at 600 s. The dwell-corrected estimators remain well calibrated: reliability is at most 9.3 for the Kairos rows and at most 0.2 for FreMEn’s, both far below the resolution attained at the long horizons (63.9 and 72.5 at 600 s).

TABLE VII: **Computational cost** on TBD [38] over all 16 scenes. Kairos’s *per-frame* row is its full per-frame cost; the remaining rows are unit costs and do not sum to it.

		mean $\pm$ std [ms]	med [ms]	p95 [ms]
<i>Base</i>	Frontend total	76.9 $\pm$ 45.1	74.7	263.2
	Backend optimization	292.0 $\pm$ 158.2	263.8	989.8
<i>Flow dynamics</i>	Per frame (10 Hz)	0.427 $\pm$ 0.181	0.441	0.689
	Cell update (per det.)	0.004 $\pm$ 0.004	0.004	0.010
	Sharing (per neighborhood)	0.001 $\pm$ 0.001	0.000	0.003
	Coupling (per frame it runs)	0.007 $\pm$ 0.007	0.006	0.019
	Node summary (per event)	11.596 $\pm$ 4.727	12.540	16.314
	Deformation remap (per event)	0.328 $\pm$ 3.079	0.175	0.230

F. Computational Cost

We profile Kairos following [31], timing every component per frame against a 200 ms key-frame budget (5 Hz; the TBD detection stream arrives at 10 Hz) on an Intel Core i9-14900KF CPU (Table VII). The 3DSG front-end (reconstruction and place extraction) runs at a mean of 76.9 ms, inside the budget; the backend (room detection and pose-graph optimization) runs asynchronously and, as in Hydra [31], does not block it. The flow layer adds 0.43 ms per frame, two orders of magnitude below the front-end and 4 μ s per detection. Node aggregation (11.6 ms) and the deformation remap (0.3 ms) run per backend event against its mean 292 ms, adding 4.1% per re-optimization. Enabling the layer leaves the front-end timing, tail included, unchanged.

G. Downstream Task: Encounter-probability Planning

We evaluate Kairos on the encounter-probability planning task of [46]. Starting from a home position at a given departure time, the robot has a 600 s deadline to reach within 2 m of a person. A finite-horizon dynamic program decides at each step whether to wait or move to an adjacent cell of the traversable graph; the reward for occupying a cell is the information source’s predicted activity there at that time, and the policy maximizes accumulated reward within the deadline.

1) *Baselines*: The information sources cover the alternatives a robot could carry. **None (wait)** and **None (random walk)** use no information and serve as uninformed baselines. **Static rate** uses each cell’s training-mean detection rate: where people are on average, with no notion of time. **FreMEn** [2] under its full-visibility fit.

2) *Evaluation Protocol*: We follow the protocol of [47]: a single fixed planner, with only the information source varying across variants. Whereas prior evaluations [46], [47] adopted a simulator with scripted agents and on binary room-occupancy records, we replay every episode against the recorded pedestrians of a held-out day. Episodes start every 30 minutes within operating hours on each held-out day from a single fixed home node, giving 952 episodes over 28 held-out days on HB and 192 over 8 days on ATC. For each episode we record whether a person is reached before the deadline (Success), the time to the first encounter (if successful, Med. t), and the number of distinct people encountered (Enc.). Only the presence channel is evaluated, since an encounter depends on presence alone.

The informed sources drive a deterministic planner over a frozen trained state, so each episode’s outcome is fixed.

TABLE VIII: **Encounter-probability planning** on HB [39] and ATC [1]: success rate, median time to the first encounter and mean number of distinct people encountered per episode.

	HB			ATC		
Information source	Succ. $\uparrow$	Med. t [s] $\downarrow$	Enc. $\uparrow$	Succ. $\uparrow$	Med. t [s] $\downarrow$	Enc. $\uparrow$
None (wait)	0.484	138.0	1.5	0.818	111.0	3.2
None (random walk)	0.893	72.0	9.2	0.875	45.2	37.9
Static rate	0.909	18.5	23.5	0.885	6.5	14.3
FreMEn	0.937	30.0	27.4	0.896	7.0	152.9
Kairos	0.922	30.0	26.8	0.891	7.0	53.0

Because every day repeats the same grid of departure times, the held-out day rather than the individual episode is the unit of replication: we report each success rate with a 95% confidence interval formed as a t -interval over the D held-out days. These intervals therefore quantify day-to-day generalization; the single fixed home node means they do not capture sensitivity to the start location. Median time-to-encounter is a discrete statistic and is reported as a point value.

3) *Results*: On HB (Table VIII), the static rate succeeds on 91% of episodes and encounters 23.5 people, against 89% and 9.2 for the random walk and 48% and 1.5 for waiting at the home position, showing that knowing where people are on average raises the number of people met rather than the probability of meeting any. Conditioning on time of day improves both: FreMEn succeeds on 94% of episodes (95% CI [0.92, 0.95] over the 28 test days) and Kairos on 92% ([0.91, 0.94]), their intervals overlapping, meeting 27.4 and 26.8 people (17% and 14% above the static rate). The static source reaches its first person sooner (18.5 s median against 30 s) but is the least consistent across days: its interval is much wider ([0.85, 0.97]) and its per-day success ranges from 0.18 to 0.97, because its single greatest-average-rate cell is fixed (failing on days that cell is quiet) whereas the time-conditioned sources relocate by hour and hold near 0.9 on every day. Kairos remains within 1.6% and 0.6 encounters of FreMEn, which is fit on the full detection record while Kairos observes $\approx$ 7% of the stream on HB and $\approx$ 4% on ATC.

On ATC all informed sources succeed nearly identically (88%–90%, with fully overlapping 95% intervals: [0.86, 0.91] for the static rate and [0.87, 0.92] for Kairos over the 8 test days) and reach the first person within seconds (6.5–7.0 s median). ATC’s within-day structure is close to spatially uniform, so conditioning on time does not change where the robot should stand: at this pedestrian density place alone suffices. Distinct-people counts vary widely across sources (3.2 for waiting to 152.9 for FreMEn), reflecting movement and how often d plan relocates through a dense crowd rather than the quality of its information.

VIII. ABLATION STUDIES

A. Temporal forecasting of the direction weights

This ablation isolates the spectral forecasting of the direction mixing weights against an order-0 arm holding them at their running mean, on HB (Table IX). We report hourly columns between 05–23 h, since overnight hours carry negligible traffic. The embodied regime is the operating condition;

TABLE IX: **Temporal forecasting of the direction mixing weights** on HB [39]: change in joint MLPD from replacing static weights (order-0) with the gated NUDFT forecast (order-1), by hour of day. Entries are in units of 10^{-3} nats; positive favors the forecast. *All* is the whole test set over the full 24-hour day.

Regime	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	All
Embodied (5 m FoV)	21.4	15.4	1.0	-1.2	12.6	9.8	13.8	11.0	21.9	13.2	11.7	12.1	10.4	16.2	18.0	-1.6	14.8	-6.2	23.8	11.7
Full visibility	24.1	17.8	1.3	-0.7	14.3	11.5	16.3	13.8	25.5	14.8	14.2	14.5	13.4	19.5	21.1	0.2	17.7	-5.7	24.4	13.9

TABLE X: **Evidence sharing ablation** on TBD [38], ATC [1], and HB [39]: joint MLPD by per-voxel coverage and global MLPD. Coverage strata are dataset-specific, reflecting the corpora’s different densities.

Method	TBD				ATC				HB			
	sparse	medium	dense	global	sparse	medium	dense	global	sparse	medium	dense	global
N detections	211	4 541	337 338	342 090	2 508 395	23 079 563	304 305 539	329 893 497	325	3 794	40 438 584	40 442 703
Reference	-8.128	-4.340	-2.931	-2.953	-1.573	-1.722	-1.550	-1.562	-6.801	-5.036	-2.053	-2.054
Sharing	-7.367	-4.017	-2.823	-2.840	-1.572	-1.722	-1.550	-1.561	-6.274	-4.787	-2.053	-2.051

full-visibility regime measures how the same mechanism responds to added observability.

The spectral predictor improves the joint MLPD in both regimes, and the margin grows with observability: by 0.0117 embodied (-2.0770 against -2.0887) and 0.0139 at full visibility (-2.0728 against -2.0867). Across the day the improvement is positive at all but three hours (marginal dips at 08:00, 20:00, and 22:00, none beyond 0.007; Table IX), and largest at 05:00, 13:00 and 23:00 (up to $+0.024$ embodied, $+0.026$ at full visibility). The spectral arm converges to the static weights where the rhythm is absent, so the harmonic terms add resolution where the data supports them and leave the forecast unchanged elsewhere.

A model-free measurement of the direction rhythm on the training detections, the traffic-weighted divergence between each hour’s heading histogram and the cell’s all-hour marginal, places the recoverable direction-mix signal at 0.011 bits (0.008 nats) on HB and 0.0099 bits on ATC, which are the cells a detection-weighted MLPD average reports. The 0.0117 and 0.0139 nats recovered here (last column in Table IX) therefore sit at the upper end of the available signal.

B. Evidence Sharing

The sharing of Section V-D gives a voxel’s neighborhood estimate the weight of ν crossings, so the neighborhood holds a share $\nu/(C+\nu)$ of the weights it reports and that share falls as its own crossings C accumulate. Because the shrinkage is applied at read time, both settings score the same trained state. We compare $\nu = 3$ (Sharing) against $\nu = 0$ (Reference, each voxel reporting its own mean alone) and report MLPD stratified by per-voxel coverage (Table X). TBD supplies the sparse embodied regime the mechanism targets; ATC and HB the dense regime in which the shrinkage should be negligible. Strata are fixed per-voxel detection counts set per dataset: sparse, medium and dense hold below 10, 10 to 50, and above 50 detections on TBD and HB, and below 5 000, 5 000 to 20 000, and above 20 000 on ATC.

The improvement is ordered by coverage on every dataset, as the share $\nu/(C+\nu)$ predicts: 0.761, 0.323 and 0.108 from the sparse to the dense stratum on TBD, and 0.527 and 0.249 in HB’s two sparse strata against no change in

its dense one. On ATC the least-covered stratum already contains 2.5 million detections, resulting in a 0.001 gain. The global MLPD understates all of this: HB improves by 0.003 globally because its two sparse strata hold four thousand of 40.4 million detections, so a whole-dataset average reports the dense stratum and conceals the 0.527 gain. Across the three datasets no stratum loses more than 0.0001, so sharing is enabled by default: it supplies a prior where a voxel’s own evidence is thin and withdraws where it is not.

C. Flow Dependence

The coupling term (Section V-E) promotes the per-voxel marginals to a joint distribution over adjacent-voxel pairs. Samples are taken at $S = 12$ per cycle of the shortest candidate period, giving a sampling window $\Delta t_\phi = 5$ s on TBD and 300 s on ATC and HB, with $N_{\min} = 2$ pairs per window.

The ablation evaluates the coupling term at three settings: removed ($\phi_{ij}^{(k_i, k_j)} \equiv 0$, Kairos without coupling), fixed (a prescribed same-slot alignment with no learned parameters), and learned. The primary metric is the joint-MLPD gain ΔJ over the removed setting: the log-density of the two simultaneous observations under the pair model, against independent-marginal scoring. The secondary metric is the *conditional heading MAE*: the error at voxel j when its heading is predicted from the simultaneous observation at adjacent voxel i . Pairs are further stratified by $D_{\text{KL}}(\pi_i \parallel \pi_j)$, the Kullback–Leibler divergence between the two voxels’ training-time mixing weights, which measures how differently the two cells flow overall: small values mark pairs that share a flow profile.

Coupling in either form improves both metrics on ATC and HB in both regimes, and marginally on TBD, and the two metrics separate what it contributes (Table XI). The conditional heading MAE carries the large effect: predicting a voxel’s heading from a simultaneous observation at its neighbor reduces the error by 31.0° on ATC and 56.5° on HB. The joint gain ranges from $+0.02$ on TBD to $+0.63$ on HB. TBD dissociates the two metrics: its conditional error falls by 30.2° while the pair density remains within 0.02 of the uncoupled reference. Coupling contributes primarily by conditioning one voxel’s prediction on its neighbor, and secondarily by the pairwise structure the joint density measures.

TABLE XI: **Flow dependence ablation** on TBD [38], ATC [1] and HB [39], over adjacent-voxel pairs. Results report gains over the uncoupled model ($\phi \equiv 0$): ΔJ in joint MLPD, ΔMAE in conditional heading error, so higher is better for both. The lower block stratifies the learned coupling by $D_{KL}(\pi_i || \pi_j)$.

Pairs	Coupling	TBD		ATC (emb.)		ATC (full)		HB (emb.)		HB (full)	
		ΔJ	ΔMAE	ΔJ	ΔMAE	ΔJ	ΔMAE	ΔJ	ΔMAE	ΔJ	ΔMAE
<i>All</i>	Fixed alignment prior	0.25	26.2°	0.29	33.5°	0.28	32.9°	0.42	51.0°	0.43	52.1°
	Learned coupling	0.02	30.2°	0.27	31.0°	0.38	35.9°	0.63	56.5°	0.62	57.1°
<i>Learned, by D_{KL}</i>	$D_{KL} < 0.2$	0.03	32.4°	0.28	32.6°	0.39	38.0°	0.63	56.6°	0.62	57.0°
	$0.2 \leq D_{KL} < 1.0$	-0.11	14.0°	0.05	8.2°	0.13	8.2°	0.33	23.5°	0.20	22.9°
	$D_{KL} \geq 1.0$	-0.11	3.4°	-0.29	0.4°	-0.11	-5.4°	-0.32	12.3°	-0.62	3.9°

Whether to learn the coupling is decided by per-edge evidence: the learned arm beats the prior only where that evidence is dense (as in HB, and ATC at full visibility). Within ATC the learned increment over the fixed prior grows with observability, from -0.02 embodied to $+0.10$ at full visibility. On TBD the learned joint gain is $+0.02$ against the prior’s $+0.25$: per-edge estimation requires pair observations that the shortest deployments do not provide, and the parameter-free prior is preferable there. Where evidence is present learning the coupling performs better on both metrics, by $+0.21$ in joint gain and 5.5° in conditional MAE on embodied HB.

The D_{KL} stratification locates the gain, and both metrics concentrate on the same strata: on pairs whose marginals are similar the joint gain is largest ($+0.28$ on ATC embodied at $D_{KL} < 0.2$, $+0.39$ at full visibility) and so is the error reduction (32.6° and 38.0°). On divergent pairs the joint gain is negative (-0.29 and -0.11 at $D_{KL} \geq 1.0$) and the error reduction shrinks (0.4° and -5.4°). HB shows the same ordering ($+0.63$ on similar pairs, -0.32 on divergent ones, embodied). Coupling is therefore informative where the neighbor’s flow resembles the target cell’s, and restricting ϕ to low-divergence pairs is a direct refinement.

D. Calibration of Predictive Uncertainty

1) *Speed*: The speed channel emits a per-detection interval $\hat{\sigma}_\rho$. In the evaluated configuration the per-slot speed estimate is a running mean, so this interval is the aleatoric variance (16) alone: each cell’s measured speed scatter, regularized by the prior where evidence is sparse. We report empirical coverage at the 68% and 95% levels, the Expected Calibration Error (ECE), and sharpness (the mean interval width $2\hat{\sigma}_\rho$).

Coverage is 0.76 and 0.95 on ATC and 0.66 and 0.95 on HB against nominal 0.68 and 0.95, at ECE 0.06 and 0.01 (Table XII). The interval widths correspond to a per-detection standard deviation of 0.27 m/s on ATC and 0.40 m/s on HB, a difference no per-site parameter produces: (16) blends each cell’s measured scatter with a prior common to both datasets. Coverage at the 95% level is exact at both sites; at 68% the intervals are conservative, the safe direction for a planner reading them. The interval thus follows each environment’s scatter without recalibration.

2) *Weights*: The weight channel emits, per slot, the mixing weight together with its epistemic variance (15), propagated through the simplex normalization. A mixing weight is the fraction of a voxel’s crossings belonging to a slot. That fraction is not directly observable, so we compare the predicted weight

TABLE XII: **Uncertainty calibration** on ATC [1] and HB [39]: coverage, expected calibration error and sharpness.

	ATC				HB			
	Cov. 68	Cov. 95	ECE ↓	Sharp. ↓	Cov. 68	Cov. 95	ECE ↓	Sharp. ↓
Speed	0.76	0.95	0.06	0.54	0.66	0.95	0.01	0.80
Weights	0.71	0.93	0.02	0.24	0.69	0.96	0.01	0.37

against the mean soft responsibility (12) over the crossings falling in a non-overlapping time window at one voxel, rebuilt with the trainer’s own crossing definition. A window holds finitely many crossings, so its mean responsibility departs from the voxel’s mixing weight by an amount governed by that count. That departure is the channel’s aleatoric term, formed per window from that count, and the reported credible interval combines it with the epistemic variance.

The weight channel is well calibrated on both datasets. Coverage is 0.707 and 0.935 on ATC and 0.699 and 0.964 on HB against the nominal 0.68 and 0.95, at ECE 0.022 and 0.011. A cluster bootstrap over 36,168 and 2,783 voxel-day units (the resampling unit since windows sharing pedestrians are not independent samples) places the ECE within $[0.022, 0.023]$ and $[0.010, 0.011]$, against the 0.0002 and 0.0004 of a perfect model. Attributing each crossing to its nearest component rather than splitting it across components, the same intervals cover 0.891 and 0.922 at the 95% level.

We decompose the predictive variance by the training evidence a voxel carries, on ATC since HB supports no such stratification. The epistemic term supplies 0.7% of the predictive variance across the 8.6 million windows above 200 training crossings, 19.0% across the 12,384 below 200, and 52.9% across the 2,440 of those falling below 50, with no per-cell parameter setting the transition: the credible interval reduces to its aleatoric term where the mixing weights are determined by the data and is governed by the epistemic term where they are not. Coverage falls across the same strata, from 0.936 to 0.866 and 0.832 against nominal 0.95, so the epistemic term widens the interval in the right direction without fully accounting for the miscoverage on thin evidence.

IX. CONCLUSION

We presented Kairos, a predictive directional-flow memory that extends a hierarchical 3D scene graph to a 4D scene graph. At any traversable location and any future query time, Kairos predicts whether people are likely to be present and how they are likely to be moving, as a full directional distribution

anchored to the reconstructed geometry. We evaluated it on three real pedestrian environments, from minutes of robot operation to a station concourse recorded continuously for eleven months, against dedicated occupancy and flow baselines. Kairos attains the best scores in every channel on the robot-collected dataset and is the only compared method that predicts the speed, joint, and presence densities together. Its presence forecasts are competitive with a dedicated occupancy model trained on the full detection stream, while observing a small fraction of it. Its credible intervals are calibrated, and its learned state remains consistent under loop-closure corrections. On a downstream person-finding task, planning over Kairos's forecasts succeeds more often and encounters more people than planning over any time-invariant map.

Limitations. Three limitations bear on the estimates the system reports. First, the mixing-weight and presence predictors accumulate cumulative means, so every observation carries equal weight: a place whose flow changes permanently is corrected at a rate of one over its accumulated sample count, and no test separates such a change from sampling noise. Second, the learned coupling requires per-edge observations that short deployments do not supply so the parameter-free prior is preferable there. Third, the weight credible interval under-covers where training evidence is thin since a Gaussian interval propagated through the simplex normalization does not respect the bounded support of a mixing weight.

Future Work. Each limitation admits a direct refinement. Recency weighting, available on the speed channel, extends to the weight and presence predictors, paired with a change test that resets a cell rather than diluting it. Restricting ϕ to low-divergence pairs confines the coupling to where it is informative, and lifting it from adjacent voxels to the place topology would represent a corridor as a single field. A Dirichlet posterior over the slots, formed from the crossing counts the state already carries, replaces the normalized Gaussian interval with one that respects the simplex and widens as those counts fall. Finally, motion archetypes are derived from the flow state but not consumed by it; using a place's archetype as the cold-start prior for a new voxel would extend evidence sharing from spatial neighbors to places that behave alike.

REFERENCES

- [1] D. Bršćić, T. Kanda, T. Ikeda, and T. Miyashita, "Person tracking in large public spaces using 3D range sensors," *IEEE Trans. Human-Mach. Syst.*, vol. 43, no. 6, pp. 522–534, 2013.
- [2] T. Krajník, J. P. Fentanes, J. M. Santos, and T. Duckett, "FreMEN: Frequency map enhancement for long-term mobile robot autonomy in changing environments," *IEEE Trans. Robot.*, vol. 33, no. 4, pp. 964–977, 2017.
- [3] F. Jovan, J. Wyatt, N. Hawes, and T. Krajník, "A Poisson-spectral model for modelling temporal patterns in human data observed by a robot," in *IEEE/RSJ Int. Conf. on Intell. Robots & Systems*, 2016, pp. 4013–4018.
- [4] T. P. Kucner *et al.*, "Survey of maps of dynamics for mobile robots," *Int. J. Robot. Res.*, vol. 42, no. 11, pp. 977–1006, 2023.
- [5] T. Krajník, J. Santos, B. Seemann, and T. Duckett, "Fractomap: An efficient spatio-temporal environment representation," *Adv. Auton. Robot. Syst.*, p. 269, 2014.
- [6] T. Kucner, J. Saarinen, M. Magnusson, and A. J. Lilienthal, "Conditional transition maps: Learning motion patterns in dynamic environments," in *IEEE/RSJ Int. Conf. on Intell. Robots & Systems*, 2013, pp. 1196–1201.
- [7] T. P. Kucner, M. Magnusson, E. Schaffernicht, V. H. Bennetts, and A. J. Lilienthal, "Enabling flow awareness for mobile robots in partially observable environments," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 1093–1100, 2017.
- [8] R. Senanayake and F. Ramos, "Directional grid maps: modeling multi-modal angular uncertainty in dynamic environments," in *IEEE/RSJ Int. Conf. on Intell. Robots & Systems*, 2018, pp. 3241–3248.
- [9] I. Catalano, C. C. Zumaya, J. A. Placed, J. Civera, W. M. Bessa, and J. Peña-Queralt, "3D scene graphs in robotics: A unified representation bridging geometry, semantics, and action," *IEEE Trans. Robot.*, 2026.
- [10] A. Rosinol *et al.*, "Kimera: From SLAM to spatial perception with 3D dynamic scene graphs," *Int. J. Robot. Res.*, vol. 40, no. 12–14, pp. 1510–1546, 2021.
- [11] E. Greve, M. Büchner, N. Vödisch, W. Burgard, and A. Valada, "Collaborative dynamic 3D scene graphs for automated driving," in *IEEE Int. Conf. on Robotics & Autom.*, 2024, pp. 11 118–11 124.
- [12] S. Looper, J. Rodriguez-Puigvert, R. Siegwart, C. Cadena, and L. Schmid, "3D VSG: Long-term semantic scene change prediction through 3D variable scene graphs," in *IEEE Int. Conf. on Robotics & Autom.*, 2023, pp. 8179–8186.
- [13] L. Schmid, M. Abate, Y. Chang, and L. Carlone, "Khronos: A unified approach for spatio-temporal metric-semantic SLAM in dynamic environments," in *Robotics: Science & Systems*, 2024.
- [14] N. Gorlo, L. Schmid, and L. Carlone, "Describe anything anywhere at any moment," in *IEEE/CVF Conf. on Computer Vision & Pattern Recognition*, 2026.
- [15] I. Catalano, F. Verdoja, J. Civera, J. Peña-Queralt, and J. A. Placed, "Rheos: Modelling continuous motion dynamics in hierarchical 3D scene graphs," in *IEEE/RSJ Int. Conf. on Intell. Robots & Systems*, 2026.
- [16] I. Catalano, E. Montijano, J. Civera, J. A. Placed, and J. Peña-Queralt, "Aion: Towards hierarchical 4D scene graphs with temporal flow dynamics," in *IEEE Int. Conf. on Robotics & Autom.*, 2026.
- [17] S. Molina, G. Cielniak, and T. Duckett, "Robotic exploration for learning human motion patterns," *IEEE Trans. Robot.*, vol. 38, no. 2, pp. 1304–1318, 2021.
- [18] Y. Zhu, A. Rudenko, L. Palmieri, L. Heuer, A. J. Lilienthal, and M. Magnusson, "Fast online learning of CLIFF-maps in changing environments," in *IEEE Int. Conf. on Robotics & Autom.*, 2025.
- [19] Y. Zhu, S.-M. Yang, A. Rudenko, T. P. Kucner, A. J. Lilienthal, and M. Magnusson, "NeMo-map: Neural implicit flow fields for spatio-temporal motion mapping," in *Int. Conf. on Learning Repr.*, 2026.
- [20] M. Stuede and M. Schappler, "Non-parametric modeling of spatio-temporal human activity based on mobile robot observations," in *IEEE/RSJ Int. Conf. on Intell. Robots & Systems*, 2022, pp. 126–133.
- [21] T. Krajník, J. P. Fentanes, G. Cielniak, C. Dondrup, and T. Duckett, "Spectral analysis for long-term robotic mapping," in *IEEE Int. Conf. on Robotics & Autom.*, 2014, pp. 3706–3711.
- [22] J. M. Santos, T. Krajník, J. P. Fentanes, and T. Duckett, "Life-long information-driven exploration to complete and refine 4D spatio-temporal maps," *IEEE Robot. Autom. Lett.*, vol. 1, no. 2, pp. 684–691, 2016.
- [23] T. Vint *et al.*, "Time-varying pedestrian flow models for service robots," in *European Conf. on Mobile Robots*, 2019, pp. 1–7.
- [24] J. Shi and T. P. Kucner, "Learning temporal maps of dynamics for mobile robots," *Robot. Auton. Syst.*, vol. 184, p. 104853, 2025.
- [25] Z. Yan, J. Shi, and T. P. Kucner, "STeF-LSTM: A hybrid framework integrating periodic and sequential modeling for mapping motion patterns," in *European Conf. on Mobile Robots*, 2025, pp. 1–8.
- [26] J. Shi, Q. Guo, and T. P. Kucner, "Event-triggered maps of dynamics: A framework for modeling spatial motion patterns in non-stationary environments," in *IEEE/RSJ Int. Conf. on Intell. Robots & Systems*, 2025, pp. 12924–12931.
- [27] F. Verdoja, T. P. Kucner, and V. Kyri, "Bayesian floor field: Transferring people flow predictions across environments," in *IEEE/RSJ Int. Conf. on Intell. Robots & Systems*, 2024, pp. 12 801–12 807.
- [28] I. Catalano, D. Morilla-Cabello, J. Peña-Queralt, and E. Montijano, "EgoMoD: Predicting global maps of dynamics from local egocentric observations," *arXiv:2603.00167*, 2026.
- [29] Z. Ravichandran, L. Peng, N. Hughes, J. D. Griffith, and L. Carlone, "Hierarchical representations and explicit memory: Learning effective navigation policies on 3d scene graphs using graph neural networks," in *IEEE Int. Conf. on Robotics & Autom.*, 2022, pp. 9272–9279.
- [30] C. Agia *et al.*, "Taskography: Evaluating robot task planning over large 3D scene graphs," in *Conf. on Robot Learning*, 2022, pp. 46–58.
- [31] N. Hughes *et al.*, "Foundations of spatial perception for robotics: Hierarchical representations and real-time systems," *Int. J. Robot. Res.*, vol. 43, no. 10, pp. 1457–1505, 2024.

- [32] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.
- [33] J. Sherman and W. J. Morrison, "Adjustment of an inverse matrix corresponding to a change in one element of a given matrix," *Ann. Math. Statist.*, vol. 21, no. 1, pp. 124–127, 1950.
- [34] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Wiley-Interscience, 2006.
- [35] K. W. Church and P. Hanks, "Word association norms, mutual information, and lexicography," *Comput. Linguist.*, vol. 16, no. 1, pp. 22–29, 1990.
- [36] L. Kleinrock, *Queueing Systems, Volume 1: Theory*. Wiley, 1975.
- [37] J. D. C. Little, "A proof for the queuing formula: $l = \lambda w$," *Oper. Res.*, vol. 9, no. 3, pp. 383–387, 1961.
- [38] A. Wang, D. Sato, Y. Corzo, S. Simkin, A. Biswas, and A. Steinfeld, "TBD pedestrian data collection: Towards rich, portable, and large-scale natural pedestrian data," in *IEEE Int. Conf. on Robotics & Autom.*, 2024.
- [39] T. Apeltauer, O. Uhlík, J. Apeltauer, and V. Juřík, "Spatiotemporal trajectories of pedestrian mobility at the train station: evidence of 24 million trajectories," *Sci. Data*, vol. 11, no. 1, p. 1254, 2024.
- [40] T. Gneiting and A. E. Raftery, "Strictly proper scoring rules, prediction, and estimation," *J. Amer. Statist. Assoc.*, vol. 102, no. 477, pp. 359–378, 2007.
- [41] E. P. Grimit, T. Gneiting, V. J. Berrocal, and N. A. Johnson, "The continuous ranked probability score for circular variables and its application to mesoscale forecast ensemble verification," *Quart. J. Roy. Meteorol. Soc.*, vol. 132, no. 621C, pp. 2925–2942, 2006.
- [42] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: Fast direct LiDAR-inertial odometry," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2053–2073, 2022.
- [43] J. Zuo, Z. Zhang, and S. Shen, "VINS-Mono: Robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 37, no. 4, pp. 1231–1248, 2021.
- [44] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *IEEE/CVF Conf. on Computer Vision & Pattern Recognition*, 2012, pp. 3354–3361.
- [45] A. H. Murphy, "A new vector partition of the probability score," *J. Appl. Meteorol.*, vol. 12, no. 4, pp. 595–600, 1973.
- [46] G. D. Tipaldi and K. O. Arras, "I want my coffee hot! Learning to find people under spatio-temporal constraints," in *IEEE Int. Conf. on Robotics & Autom.*, 2011, pp. 1217–1222.
- [47] T. Krajník, M. Kulich, L. Mudrová, R. Ambrus, and T. Duckett, "Where's Waldo at time t ? Using spatio-temporal models for mobile robot search," in *IEEE Int. Conf. on Robotics & Autom.*, 2015, pp. 2140–2146.
- [48] C. A. Pouw, G. G. van der Vleuten, A. Corbetta, and F. Toschi, "Data-driven physics-based modeling of pedestrian dynamics," *arXiv:2407.20794*, 2024.
- [49] B. Majecka, "Statistical models of pedestrian behaviour in the forum," MSc Dissertation, School of Informatics, University of Edinburgh, 2009.

TABLE XIII: **Datasets** analyzed in the temporal structure of pedestrian flow study.

Dataset	Environment	Time Span	Rate
ATC [1]	shopping mall	84 days / ~ 1 yr	10 Hz
Eindhoven [48]	train platform	60 days	10 Hz
Havlíčkův Brod [39]	station concourse	11 months, 24/7	~ 1 Hz
Pardubice [39]	station concourse	11 months, 24/7	~ 1 Hz
TBD [38]	university hall	~ 213 min/16 scenes	10 Hz
Edinburgh [49]	university hall	~ 7 months	~ 9 Hz

APPENDIX A

TEMPORAL STRUCTURE OF PEDESTRIAN FLOW

We report a model-free measurement of how pedestrian activity, heading, and speed vary across six real environments over selected temporal cycles. Its purpose is to establish which of the two quantities Kairos factorizes (presence and the direction mix) varies with time, at which periods, and how strongly.

A. Datasets and Candidate Periods

We select candidate periods from the type of environment (Table XIII) and the temporal coverage of its recordings, following the approach used for the predictors in Section VII-A. ATC [1] and the station concourses Havlíčkův Brod (HB) and Pardubice (PB) [39] datasets provide calendar timestamps, allowing daily and weekly folds and separate weekday and weekend analyses. Eindhoven [48] provides a continuous sequence without date labels, allowing repeated short-cycle measurements but limiting calendar-based interpretation. Edinburgh’s [49] approximate timestamps support analysis at resolutions of five minutes or longer. TBD [38] contains short sessions, totaling approximately 213 minutes, and is evaluated at periods of 1, 5, and 10 minutes.

We consider daily and weekly candidate periods at ATC; daily, weekly, morning/afternoon, and 30- and 60-minute cycles at HB and PB; a daily and 30- and 15-minute cycles at Eindhoven; and a daily period at Edinburgh. At Eindhoven, the 30-minute period represents the expected timetable cadence and the 15-minute provides a comparison.

B. Evaluation Protocol

a) Spatial cells and temporal phases: We partition each environment into 0.4 m cells (Kairos’ flow voxel resolution). For a candidate period T we define a *bin resolution* that partitions the phase into *phase bins* of equal width, and each observation is assigned accordingly (*i.e.*, at hourly resolution of a daily period, each bin contains the observations recorded at the same hour across days). We bin under the assumption that the distributions below are roughly constant across a bin width.

b) Observation units and measurements: We select moving detections with speed above 0.5 m/s. The observation unit is the *passage of a cell*. At 10 fps a single crossing yields tens of near-identical detections and a lingering agent thousands, so counting detections would conflate time spent in a cell with the number of agents crossing it. We therefore retain

one observation per track, cell, and phase. We define a cell’s *activity* in a phase bin as the number of observations lying in it. We assign each observation’s heading to one of K *heading bins* (eight, one per cardinal direction). For each cell and each phase bin we then form the normalized histogram over those bins and over binned speed. We also measure the fraction of activity within $\pm 45^\circ$ of the direction opposite the cell’s dominant heading, which distinguishes predominantly one-way from bidirectional locations.

For heading and speed we retain moving detections with speed above 0.5 m/s, and their observation unit is the *passage of a cell*. At 10 fps a single crossing yields tens of near-identical detections and a lingering agent thousands, so counting detections would conflate time spent in a cell with the number of agents crossing it. We assign each detection’s heading to one of K *heading bins* (eight, one per cardinal direction) and its speed to one of six bins of equal width over $[0.5, 3.0]$ m/s, and retain one observation per track, cell, phase bin, and value bin, so a passage whose detections span two heading bins contributes to both. A cell’s *traffic* is the number of observations retained there, counted separately for heading and for speed. For each cell and each phase bin we form the normalized histogram over the heading and speed bins. We define *activity* the count every detection at the source frame rate, stationary people included. We also measure the fraction of a cell’s traffic assigned to the heading bin opposite its dominant one and the two bins adjacent to it, pooled over all phase bins, which distinguishes predominantly one-way from bidirectional locations.

c) Signal and reference divergence: For heading and speed, we compare the cell’s distribution in each phase bin with its distribution pooled across phases, using Jensen–Shannon (JS) divergence with base-two logarithms, and average the divergences over phase bins, weighting each by its share of the cell’s traffic. The resulting **signal** measures how much that distribution changes with phase. For activity, the signal is instead the divergence between the cell’s normalized activity profile and a uniform profile. We divide the recording into alternating days, weeks, or cycles, as appropriate for the candidate period. For heading and speed, we take the JS divergence between the two halves’ distributions at each phase bin and average it as above; for activity, we take the divergence between the two halves’ phase profiles, each normalized to sum to one. The resulting **reference divergence** measures disagreement between repeated observations of the same phase, including finite-sample variation, differences between days or cycles, and any dependence remaining within the observations. We report the signal divided by the reference divergence as S/R. Since both are computed from finite counts, neither reaches zero even when the underlying distributions agree. The signal compares a phase bin against a marginal pooled over every bin, whereas the reference compares two halves of that one bin, each holding half its observations. The reference therefore has up to four times the sampling variance of the signal, and a cell with no phase-dependent variation yields between 0.25 and 0.5 rather than 1, approaching the lower end as the number of phase bins grows. We take 0.3 as that null, and read a ratio above 3 as variation beyond what

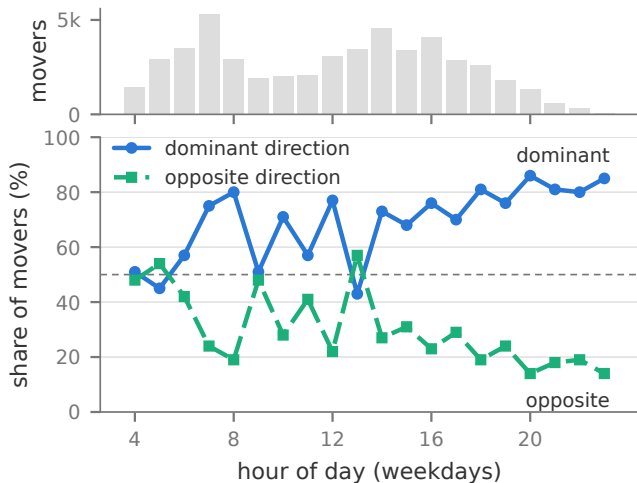


Fig. 5: **Hourly heading profile** at the HB cell selected for its strong variation, restricted to weekdays and hours with at least 20 movers. **Top:** movers per hour. **Bottom:** shares within $\pm 45^\circ$ of the cell's dominant and opposite headings.

repeated observation produces. A coarser division of the period pools more observations into each bin and lowers the reference divergence, so it detects more at the cost of any variation that changes within a bin.

We also report a site-level ratio: the mean signal over the 200 cells with the most observations divided by their mean reference divergence, weighted by those same counts.

C. Variation in Activity

Activity varies most with hour of day at HB and PB, where all 200 analyzed cells at each station exceed S/R 3, with a median per-cell S/R near 150. ATC, Eindhoven, and Edinburgh also vary with the hour, at 22, 6.0, and 14.2. Activity also varies with day of week at ATC (24) and weakly at HB (4.4), while PB stays below the threshold (1.1). At Eindhoven it varies within the 30-minute period as well, at S/R 5.0, changing by about a factor of 4.8 between the quietest and busiest phases. The hour-of-day results use weekdays only at HB, PB, and Edinburgh, and all recorded days at ATC.

The activity divergence is 0.34 bits at ATC and about 0.13 bits at HB and PB. ATC's is the larger despite its much lower S/R, because ATC is recorded only during opening hours and its daily profile therefore includes hours of zero, while HB and PB are recorded continuously.

D. Variation in Heading

HB. At HB on weekdays the site-level S/R is 2.5, below the threshold, on a signal of 0.011 bits. Per cell, the median S/R is 2.1, while 49 of the 200 cells exceed the threshold and hold 34% of the traffic. These cells have an S/R of 4–9 and a heading signal of 0.007–0.018 bits: detectable, but far below the 0.13 bits of activity variation at the same station. Heading therefore varies with hour of day in a quarter of the cells and not in the site-wide measurement. Dividing the day into morning and afternoon instead puts 60 cells above the

threshold, while at periods of 30 and 60 minutes the site-level S/R is 0.3 and 1.1.

A cell's S/R averages over the day, so individual hours can depart from the cell's overall distribution by more than it indicates. Fig. 5 shows the hourly profile of a cell selected for its strong variation. Its dominant heading, fixed over the whole recording, holds 86% of the traffic at some hours and 45% at others: nearly one-way at one time of day, close to balanced at another, with the reverse flow strongest in the early-commuter and midday hours.

PB. In contrast, PB shows no detectable variation by hour of day and clear variation between morning and afternoon. Its hourly S/R is 1.2, below the threshold at every cell; dividing the day into morning and afternoon instead puts 86 of its 200 cells above it. At periods of 30 and 60 minutes the S/R is 0.4 and 0.6.

ATC. At ATC the site-level S/R is approximately 15 and 197 of the 200 cells exceed the threshold, accounting for 98% of the traffic. The signal is 0.0099 bits against a reference divergence of approximately 0.0007 bits. By day of week the signal is lower, at 0.0025 bits and S/R 6.4. In several busy corridor cells the share of traffic running opposite the dominant heading is 47–56% in the first recorded hour, against 28–35% later in the day, so these cells are close to balanced early and one-way afterwards.

Eindhoven. Eindhoven varies within the 30-minute period and not by hour of day: S/R 8.5 against 1.45. Within that period 141 of the 200 cells exceed the threshold and account for 55% of the traffic. At bidirectional mid-platform cells the along-platform share runs from about 15% to 80%, while predominantly one-way corridors vary in activity without a comparable change in heading. At a 15-minute period the S/R is 0.24, since folding a 30-minute rhythm at half its length superimposes the two halves of each cycle. Eindhoven and the concourses are therefore opposite cases: the period that reveals structure at one shows none at the other.

Edinburgh and TBD. Neither presents variation with hour of day with an S/R of 0.6 and 0.2–0.3, and no cell above the threshold.

E. Variation in Speed

Speed varies with hour of day at ATC, where the signal is 0.006 bits at S/R 15.9 and 199 of the 200 cells exceed the threshold. At HB the site-level S/R is 1.6, below the threshold, on a signal of 0.004 bits, with 13 cells above it. At TBD the S/R is 0.2, with no cell above the threshold.

The variation is detectable but small. At ATC's cells with the most observations the mean speed changes by 0.02–0.06 m/s across the day, with roughly stable dispersion, against the 0.3 m/s standard deviation of the model's speed components. Speed also has the smallest divergence of the three quantities, below heading's 0.0099 bits at the same site.

F. Interpretation and Model Design

Activity and heading describe different aspects of pedestrian motion: activity can increase while directional proportions remain unchanged, and activity divergences exceed heading

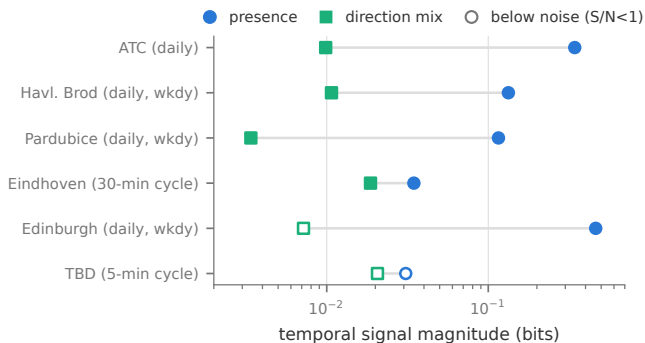


Fig. 6: **Temporal divergences** for activity and heading at each site’s primary period. Open markers denote aggregate S/N below 1.

TABLE XIV: **Heading and speed dependence** within and across slots.

Dataset	within slot			across slots
	med r	r 5th/95th	r^2	η^2
TBD	−0.018	−0.49 / 0.47	0.058	0.181
ATC	−0.003	−0.25 / 0.26	0.019	0.070
HB	0.009	−0.42 / 0.43	0.057	0.186

and speed divergences throughout (Fig. 6). The measurements therefore support estimating presence separately from the conditional flow distribution.

Directional changes are concentrated at particular locations and phases rather than spread evenly through the period, which supports selecting the spectral order per cell.

Repeated detections also require appropriate observation units: the heading estimator uses crossings, while the presence estimator accounts separately for visible time and dwell.

The hourly speed changes are small against the width of the model’s speed components, which supports holding the per-slot speed means time-invariant.

APPENDIX B HEADING AND SPEED DEPENDENCE WITHIN A COMPONENT

Kairos uses diagonal component covariances and a separate mean speed for each directional slot. Within one slot, a detection’s heading is at most $\pi/8$ from the slot centre; we measure the linear heading–speed dependence within slots with the Pearson correlation r between that offset and the detection’s speed. We evaluate whether speed differs between directions with the correlation ratio η^2 . Each detection is assigned to a single nearest slot, where the model spreads it across neighbouring components.

Within slots. The dependence between a detection’s offset and its speed has no consistent sign. The median correlation over all voxels and slots is −0.018 on TBD, −0.003 on ATC, and +0.009 on HB, whereas the middle 90% of correlations spans −0.49 to 0.47 on TBD, −0.25 to 0.26 on ATC, and −0.42 to 0.43 on HB (Table XIV). The dependence is therefore *local*, taking one sign in some voxels and slots and the opposite in others. Moreover, its magnitude is small: the

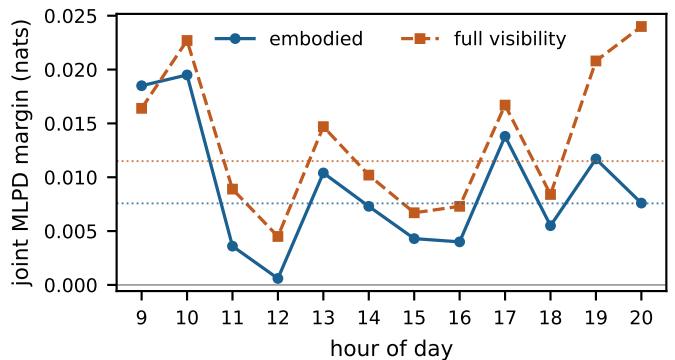


Fig. 7: **Joint-MLPD difference** between temporal and static weights on ATC in both regimes (*embodied* and *full visibility*).

offset accounts for 5.8% of within-slot speed variance on TBD, 1.9% on ATC, and 5.7% on HB. The component covariance is a single matrix shared by every voxel (Section V-B), so a cross-covariance term would take one value throughout the map. The value minimizing the error across all of them is close to zero, which is what the diagonal covariance already specifies.

Across slots. Speed differs between directions: η^2 is 0.181 on TBD, 0.070 on ATC, and 0.186 on HB. We represent these differences through per-slot mean speeds.

APPENDIX C TEMPORAL FORECASTING OF DIRECTION WEIGHTS A. Contribution of Temporal Weights on ATC

We compare Kairos with its static-weight variant on the same held-out observations, keeping the component speed model fixed. This isolates the effect of forecasting the mixing weights on joint MLPD.

On ATC, temporal weights improve joint MLPD by 0.0076 nats in the embodied regime and 0.0115 at full visibility, with every reported hour positive (Table XV). The gain varies by hour (Fig. 7): largest at 09:00–10:00 (0.0185–0.0195 nats) and smallest at 12:00 (0.0006). At full visibility the largest difference is 0.0240 nats at 20:00. HB’s aggregate gains are 0.0117 and 0.0139 nats (Table IX).

B. Heading Difference with STeF-Map

We examine whether STeF-Map’s heading advantage comes from Kairos’s fixed-width angular kernel, which spreads probability away from the dominant directions (Section VII-B3). For each detection we evaluate the predicted probability in the 45° sector containing its observed heading, where an uninformative uniform forecast assigns 0.125 to every sector.

Both methods place less than $1/8$ on the observed sector for a similar minority of detections: 23.7% for Kairos and 21.7% for STeF-Map on ATC, 17.8% overlapping, and 19.9% against 17.3% on HB, 15.2% overlapping. Where only Kairos places less than $1/8$, its median sector mass is 0.114 on ATC and 0.115 on HB, against the uniform 0.125.

Kairos spreads probability across neighboring directions, both when assigning an observation to a component and

TABLE XV: **Temporal forecasting of the direction mixing weights** on ATC [1]: change in joint MLPD from replacing static weights (order-0) with the gated NUDFT forecast (order-1), by hour of day. Entries are in units of 10^{-3} nats; positive favors the forecast. *All* is the whole test set.

Regime	9	10	11	12	13	14	15	16	17	18	19	20	All
Embodied (5 m FoV)	18.5	19.5	3.6	0.6	10.4	7.3	4.3	4.0	13.8	5.5	11.7	7.6	7.6
Full visibility	16.4	22.7	8.9	4.5	14.7	10.2	6.7	7.3	16.7	8.4	20.8	24.0	11.5

TABLE XVI: **Presence by hour of day** on HB at the 600 s horizon: mean log predictive probability (MLPP) per hour, with each hour's occupancy rate for context. The aggregate table of the main text summarizes these columns; the orderings are the same at the other horizons.

Hour	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23
<i>Embodied (5 m field of view)</i>																								
Constant base rate	-0.28	-0.28	-0.29	-0.33	-0.45	-0.63	-0.69	-0.76	-0.66	-0.71	-0.68	-0.72	-0.72	-0.76	-0.76	-0.77	-0.75	-0.71	-0.63	-0.58	-0.48	-0.43	-0.37	-0.34
Kairos (static rate)	-0.60	-0.60	-0.59	-0.56	-0.50	-0.51	-0.52	-0.53	-0.48	-0.49	-0.48	-0.50	-0.48	-0.53	-0.51	-0.51	-0.50	-0.48	-0.46	-0.45	-0.47	-0.50	-0.53	-0.56
Kairos	-0.32	-0.33	-0.36	-0.39	-0.42	-0.50	-0.52	-0.53	-0.50	-0.50	-0.50	-0.51	-0.50	-0.53	-0.51	-0.51	-0.49	-0.48	-0.46	-0.44	-0.42	-0.40	-0.37	-0.33
<i>Full visibility</i>																								
Constant base rate	-0.28	-0.28	-0.29	-0.33	-0.45	-0.62	-0.68	-0.75	-0.65	-0.70	-0.67	-0.71	-0.70	-0.75	-0.75	-0.75	-0.73	-0.70	-0.62	-0.57	-0.48	-0.43	-0.37	-0.34
FreMen occupancy	-0.04	-0.02	-0.08	-0.17	-0.32	-0.48	-0.51	-0.51	-0.47	-0.47	-0.47	-0.47	-0.47	-0.46	-0.48	-0.47	-0.47	-0.47	-0.44	-0.42	-0.35	-0.31	-0.23	-0.21
Kairos (static rate)	-0.59	-0.60	-0.59	-0.55	-0.49	-0.50	-0.51	-0.52	-0.47	-0.48	-0.46	-0.48	-0.47	-0.51	-0.50	-0.50	-0.48	-0.46	-0.44	-0.44	-0.46	-0.49	-0.53	-0.55
Kairos	-0.25	-0.26	-0.29	-0.34	-0.39	-0.48	-0.50	-0.50	-0.49	-0.48	-0.49	-0.49	-0.48	-0.50	-0.49	-0.48	-0.47	-0.46	-0.44	-0.43	-0.39	-0.36	-0.31	-0.27

when evaluating the component densities: at $\sigma_\theta = 0.4$ rad about a third of a component's mass is assigned outside the 45° sector centered on it. Sectors carrying less than 5% of a cell's traffic receive mean masses of 0.087 from Kairos and 0.078 from STeF-Map on ATC, against the 0.028 they actually contain, and 0.088 and 0.076 on HB against 0.031. On HB, Kairos's per-detection MLPP is 0.14 nats above STeF-Map's for detections below 0.25 m/s and 0.08–0.15 nats below it between 0.5 and 2 m/s. A heading measured on a near-stationary pedestrian is dominated by noise and is uncorrelated with the cell's flow, where spread mass loses less than a sharp peak; above 0.5 m/s headings are reliable and fall in the dominant directions.

On ATC the mean paired CRPS difference is below 0.001 rad; on HB it is 0.002 rad. Integrating Kairos's forecast into the same 45° sectors before scoring changes its CRPS by less than 0.002 rad on ATC and less than 0.001 rad on HB, showing two methods are indistinguishable. STeF-Map's advantage on heading MLPP is therefore specific to the log score, which is evaluated at the observed heading alone, whereas circular CRPS penalizes mass in proportion to its angular distance from the observation. Its histogram concentrates mass on the directions carrying more traffic, while Kairos's fixed-width components retain mass on the remaining directions. The tradeoff is between sharpness on common directions and coverage of uncommon ones, which motivates cell-specific angular widths.

APPENDIX D PRESENCE ESTIMATION

A. Presence Estimation by Hour of Day at HB

We examine presence MLPP at the 600 s event window by hour of day, using the protocol of Section VII-E.

In the embodied regime the temporal model's MLPP is -320 at 00:00 against the static rate's -600 , and -330 at 23:00 against -560 ; the occupancy rate in those hours is 0.00 and 0.05. The advantage narrows to 10 at 05:00, and from 08:00 through 12:00 the static rate leads by 10 to 20. Over the whole day the values are -460 and -508 (Table VI).

The gain is produced by the hours in which the cell is almost never occupied, where a rate held constant across the day overpredicts activity; through the busiest hours the two are within 20 of each other.

B. Presence Estimation on ATC

ATC is recorded during opening hours only, so its record omits the closed part of the daily cycle (Section VII-E3). The baselines follow Section VII-E1. FreMen uses a full-visibility fit, without per-cell order selection.

In the embodied regime MLPP is -192 for the static-rate variant and -202 for temporal Kairos at 5 s, and -1719 and -1887 at 600 s; at full visibility the 600 s pair is -459 and -1042 . The static variant is ahead at every window in both regimes, and the deficit grows from 10 to 168 embodied and to 583 at full visibility. When no event occurs in the window, the score for that query is $-\hat{\lambda}\Delta t$ nats (24). The cost of a given rate error therefore grows with the window duration.

The constant base rate has the best embodied MLPP from 60 s onward, -693 at 600 s against -1719 for the best cell-level variant, while at 5 and 10 s the cell-level estimates are better in both regimes. At full visibility the ordering reverses and the static-rate variant exceeds the base rate by 75 at 60 s and 242 at 600 s. The constant base rate estimates one rate from the whole record, so its error does not depend on how often an individual cell was observed. The per-cell rate supports the forecast at short windows under either coverage, and at long windows only under full visibility.

At 600 s FreMen has the highest resolution in the table, 138.3 embodied and 149.7 at full visibility, with MLPP of -1896 and -1745 . Removing the dwell correction gives resolution of at most 0.4 and reliability of 294.7.

On HB the gated temporal model has the best embodied MLPP at every window, -460 against -508 for its static variant at 600 s (Table VI); on ATC that ordering reverses at every window. HB is recorded continuously over eleven months, ATC over 84 days spread across a year (Table XIII). What limits the presence forecast on ATC is the rate estimate rather than the temporal model. A per-cell window limit, or

TABLE XVII: **Presence estimation on ATC [1]** across event windows Δt : MLPP (in nats), reliability (Rel), and resolution (Res); all in units of 10^{-3} . Reliability and resolution are shown from 60 s onward, the windows in which they separate the estimators. Rows and units match Table VI: *Kairos* is the full gated estimator and *Kairos (static rate)* the variant holding the rate time-invariant.

Regime	Model	$\Delta t = 5$ s	$\Delta t = 10$ s	$\Delta t = 60$ s			$\Delta t = 300$ s			$\Delta t = 600$ s		
		MLPP $\uparrow$	MLPP $\uparrow$	MLPP $\uparrow$	Rel $\downarrow$	Res $\uparrow$	MLPP $\uparrow$	Rel $\downarrow$	Res $\uparrow$	MLPP $\uparrow$	Rel $\downarrow$	Res $\uparrow$
<i>Embodied</i> <i>5 m FoV</i>	Constant base rate	-207	-299	-579	3.0	0.0	-698	3.4	0.0	-693	2.9	0.0
	Kairos (no dwell)	-218	-362	-1113	61.7	0.2	-2134	211.7	0.3	-2562	294.7	0.4
	Kairos (static rate)	-192	-285	-729	74.4	50.9	-1464	81.4	87.6	-1719	63.7	98.0
	Kairos	-202	-303	-806	63.1	51.4	-1597	65.6	83.3	-1887	54.5	89.6
<i>Full</i> <i>visibility</i>	Constant base rate	-195	-283	-557	3.3	0.0	-693	4.4	0.0	-701	4.0	0.0
	FreMEn [2] occupancy $\dagger$	-197	-492	-1478	57.5	76.0	-1948	45.6	134.1	-1745	35.8	149.7
	Kairos (no dwell)	-302	-508	-1562	56.3	0.0	-2948	186.7	0.0	-3516	258.2	0.0
	Kairos (static rate)	-191	-273	-482	31.1	51.7	-489	23.4	104.6	-459	12.4	122.0
	Kairos	-195	-283	-570	29.3	53.7	-866	30.4	97.3	-1042	27.4	105.2

Notes: $\dagger$ FreMEn divides each cell's occupied time by total elapsed time, so time in which the cell was not observed is indistinguishable from time in which it was empty. It is therefore fit only under *full visibility*.

TABLE XVIII: **Weight interval by training evidence.** Windows are stratified by the number of crossings the voxel accumulated during training.

Dataset	Crossings	Windows	Epi. share	Cov. 95
ATC [1]	≥ 200	8,605,880	0.007	0.936
	< 200	12,384	0.190	0.866
	< 50	2,440	0.529	0.832
HB [39]	≥ 200	2,963,424	0.002	0.964
	< 200	704	0.048	0.939
	< 50	64	0.145	0.969

TABLE XIX: **Speed uncertainty: adapted variance against a constant prior** on ATC [1] and HB [39]. The constant row fixes $\hat{\sigma}_\rho$ to the aleatoric prior $\sigma_\rho = 0.3$ m/s for every detection. Bold marks the better value per column.

	ATC [1]				HB [39]			
	Cov. 68	Cov. 95	ECE $\downarrow$	Sharp. $\downarrow$	Cov. 68	Cov. 95	ECE $\downarrow$	Sharp. $\downarrow$
Constant aleatoric prior	0.79	0.96	0.09	0.60	0.55	0.84	0.08	0.60
Adapted aleatoric variance	0.76	0.95	0.06	0.54	0.66	0.95	0.01	0.80

a fallback to the global rate in cells with few observations, would address that limit directly.

APPENDIX E PREDICTIVE UNCERTAINTY

Weight Intervals. The weight forecast is evaluated against the mean soft responsibility of crossings observed within a voxel-time window. Its uncertainty has two sources: uncertainty in the predicted mean weight, and variation in the finite set of crossings used to evaluate it.

Let p be the predicted weight and v its propagated epistemic variance. For $0 < p < 1$ and $0 < v < p(1 - p)$, a moment-matched Beta distribution has parameters

$$a = p\kappa, \quad b = (1 - p)\kappa, \quad \kappa = \frac{p(1 - p)}{v} - 1. \quad (35)$$

The variance is capped below the Bernoulli bound before matching. Conditional on a latent share $P \sim \text{Beta}(a, b)$, an integer count of slot assignments among n independent crossings satisfies $X \mid P \sim \text{Binomial}(n, P)$. Marginalizing

P gives the beta-binomial predictive distribution, whose share variance is

$$\text{Var}(X/n) = \frac{p(1 - p)}{n} + \left(1 - \frac{1}{n}\right)v. \quad (36)$$

The first term is the finite-window variation and the second the uncertainty in the mean.

For an integer observation x , the randomized probability integral transform is $F(x^-) + U\text{Pr}(X = x)$, where U is uniform on $[0, 1]$ and seeded per window and slot. Under a correctly specified count distribution the transformed values are uniform. The evaluated target is a sum of fractional responsibilities, so the beta-binomial is a count-based approximation and the uniformity holds only approximately. The propagation omits cross-slot covariance, although the responsibilities sum to one, and crossings within a voxel-day can be correlated; the reported intervals resample voxel-day units.

Below 200 training crossings HB has 704 evaluation windows, and below 50 it has 64, against ATC's 12,384 and 2,440 (Table XVIII). Its epistemic shares in those strata are 0.048 and 0.145, and its coverage is 0.939 and 0.969 against the nominal 0.95. These strata are too small to support the comparison reported on ATC in Section VIII-D.

Speed Intervals. Sharpness is the mean interval width $2\hat{\sigma}_\rho$. With the constant 0.3 m/s prior, coverage is 0.79 and 0.96 on ATC and 0.55 and 0.84 on HB, against the nominal 0.68 and 0.95 (Table XIX). With the adapted variance it is 0.76 and 0.95 on ATC and 0.66 and 0.95 on HB. The mean width decreases from 0.60 to 0.54 m/s on ATC and increases from 0.60 to 0.80 m/s on HB. The measured scatter is 0.27 m/s on ATC and 0.40 m/s on HB, on either side of the 0.3 m/s prior. The estimator therefore corrects the interval in opposite directions at the two sites from the same prior value. A fixed width fits one site or the other, not both.