Mapping Spatial Motion Patterns in Non-Stationary Environments

Junyi Shi[1], Zekun Wang[1], Ting Fu[1], Yufei Quan[1] and Tomasz Piotr Kucner[1,2]

***Abstract*—Spatial motion patterns provide a foundational representation for characterizing the behavioral dynamics of moving objects in complex scenes. This paper proposes a comprehensive framework for the systematic mapping of spatial motion patterns within non-stationary environments. We employ a grid-shifting mechanism to generate context-aware cells based on historical observations. Temporal patterns are modeled using Neural Stochastic Differential Equations, while a diffusion model is integrated to handle abrupt changes in motion patterns through an event-triggered mechanism. The resulting spatial motion maps offer a rich, time-indexed representation of environmental dynamics, providing essential situational awareness for downstream robotics tasks. Experimental results demonstrate the effectiveness of the proposed system in maintaining consistent motion patterns and generating high-fidelity pattern maps in real-world scenarios.**

I. Introduction

The majority of real-world environments are inherently dynamic, exhibiting continuous or discrete changes driven by the motion of objects or continuous media, as well as the consequences of such motion. These dynamics can manifest in various forms, ranging from the movement of pedestrians to the airflow. Such dynamic characteristics significantly influence the performance of autonomous systems, making it crucial for mobile robots to perceive and anticipate environmental dynamics during the motion planning and execution stages.

In recent years, a variety of techniques have been proposed to incorporate dynamic information into robotic perception systems. Among these methods, Maps of Dynamics (MoD) has emerged as a prominent approach, providing spatial or spatio-temporal representations of dynamic patterns within a specific environment [1]. MoDs aim to capture and retain information about patterns of change or motion based on historical observations. A key advantage of MoDs over other methods is their ability to transcend the limitations imposed by the robot's perceptual range. MoDs have found extensive applications, including human motion prediction [2], [3], motion planning [4], [5], task planning [6], and human-robot interaction [7].

Two widely used representations within the MoD framework are trajectory maps and directional maps. Trajectory maps operate under the assumption that objects in a given environment follow a finite set of paths. In contrast, directional maps focus on capturing local motion patterns around specific points, emphasizing the directional nature of motion, such as velocity magnitude and direction, while often neglecting global motion aspects.

[1]The authors are with the Department of Electrical Engineering and Automation, Aalto University, Espoo, Finland firstname.lastname@aalto.fi

[2]Tomasz Piotr Kucner is also with the Finnish Center for Artificial Intelligence, Finland

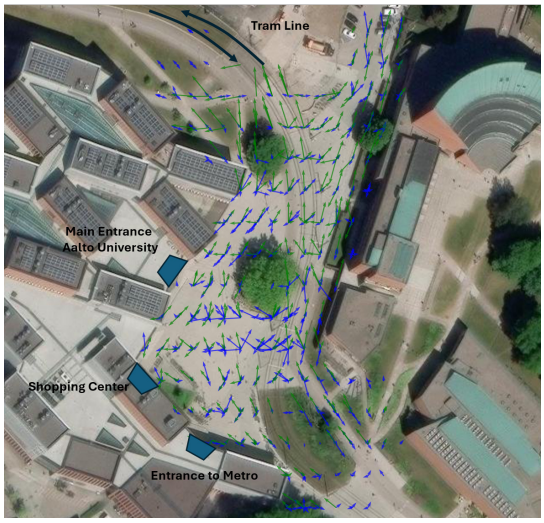

Fig. 1. Mapping results of motion patterns at Aalto University student square during midday. Blue arrows denote pedestrian motion patterns, while green arrows represent cyclist motion patterns.

Existing approaches to directional maps typically rely on fixed occupancy cells [8], where cell sizes are manually set rather than being adaptive to the environment. However, in practical scenarios, moving objects such as pedestrians are not uniformly distributed; they sometimes tend to cluster in certain areas and disperse in others. As a result, the intensity of motion patterns varies spatially, highlighting the need for a more flexible and context-aware approach to grid resolution.

Moreover, motion patterns in real-world environments often exhibit temporal variations. While some prior work has attempted to address this by modeling the temporal aspects of MoDs [9], these approaches typically assume gradual changes in motion patterns over time. This assumption, however, still diverges from reality, where sudden changes in motion patterns are frequently observed, particularly during peak times in locations such as metro stations or busy streets. During such events, pedestrian speed and direction can change abruptly, challenging the assumption of gradual evolution.

In this work, we address these limitations by introducing an event-triggered MoD (ETMoD) framework that automatically identifies regions of interest and employs Neural Stochastic Differential Equations (SDEs) [10] to model temporal changes in motion patterns. To account for event-based pattern changes, we integrate a diffusion model [11] to generate intermediate data. An example of the mapping results is illustrated in Fig. 1. As indicated by the green arrows, cyclists exhibit higher velocities compared to pedestrians (blue arrows), a disparity that becomes pronounced in open areas away from buildings.

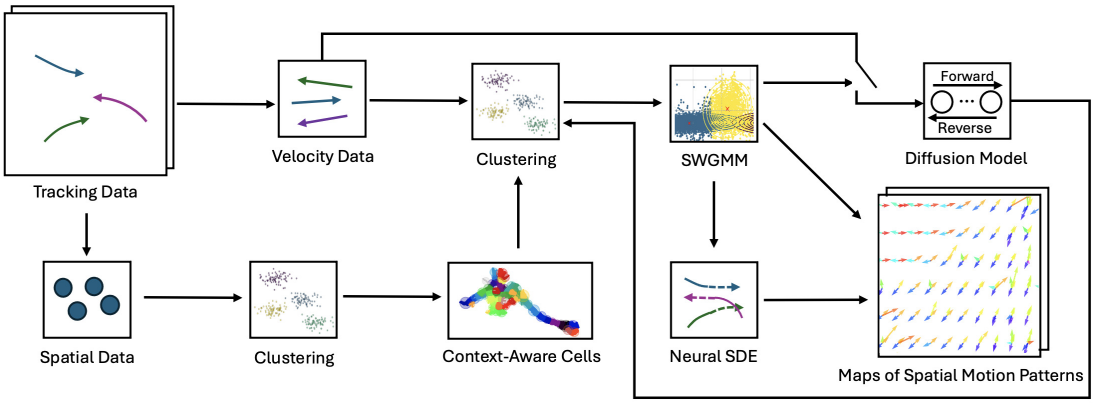

Fig. 2. Overview of our framework. Our framework processes tracking data as input, first clustering spatial data to generate context-aware cells, followed by velocity clustering to construct a SWGMM. Temporal patterns are modelled using a Neural SDE, while a diffusion model enables an event-triggered mechanism to handle abrupt behavioural changes.

Furthermore, the motion patterns of pedestrians demonstrate greater multimodality than those of cyclists, reflecting the higher variability inherent in pedestrian movement behavior.

II. Methodology

In this section, we present our methodology for modeling spatial motion patterns in non-stationary environments. As illustrated in Fig. 2, our framework begins by processing tracking data as input. First, spatial data is utilized for clustering to derive context-aware cells. Subsequently, velocity data is clustered to construct a SWGMM, capturing the multi-modal characteristics of motion patterns. To model temporal patterns, we employ a Neural SDE framework. Furthermore, a diffusion model is integrated to incorporate an event-triggered mechanism, enabling the system to adapt to abrupt changes in motion patterns.

A. Representation

In this work, we adopt the same setting of the representation with the CLiFF-Map [8]. The dynamic characteristics of an environment are characterized by velocity, a vector quantity defined by both direction and speed. At each sampling location, the velocity $\mathbf{v}$ is represented in polar coordinates, consisting of an orientation θ and a speed ρ, as follows:

$$\mathbf{v} = (\theta, \rho)^\top, \theta \in [0, 2\pi) \wedge \rho \in \mathbb{R}^T. \tag{1}$$

To model the variability of the velocity, a mixture of Semi-Wrapped Normal Distribution (SWND) is employed, as the direction θ is a circular variable and the speed ρ is a linear variable. At a given point, the semi-wrapped probability density function (PDF) over velocities can be visualized as a function on a cylinder, where the direction value θ is wrapped around the unit circle and the speed value ρ extends along the cylinder's length. The PDF of the above SWND is defined as:

$$\mathcal{N}^{\mathcal{SW}}_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\mathbf{v}) = \sum_{k\in\mathbb{Z}} \mathcal{N}_{\boldsymbol{\mu},\boldsymbol{\Sigma}}\left(\begin{bmatrix}\theta\\ \rho\end{bmatrix} + 2\pi\begin{bmatrix}k\\ 0\end{bmatrix}\right), \tag{2}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ denote the mean and covariance matrix of $(\theta, \rho)^T$, respectively. The integer k is a winding number that indicates the number of times the function is wrapped around the circle. In practice, k can be approximated to $k \in \{-1, 0, 1\}$, as suggested by Mardia and Jupp [12].

Since dynamic aspects such as pedestrian flows are often multi-modal in most areas, a SWGMM [13] is utilized to capture the different motion patterns. The SWGMM is represented by a weighted sum of J SWNDs:

$$p(\mathbf{v}|\boldsymbol{\xi}) = \sum_{j=1}^{J} \pi_j \mathcal{N}^{\mathcal{SW}}_{\boldsymbol{\Sigma}_j,\boldsymbol{\mu}_j}(\mathbf{v}), \tag{3}$$

where $\boldsymbol{\xi} = \{\xi_j = (\boldsymbol{\Sigma}_j, \boldsymbol{\mu}_j, \pi_j) | j \in \mathbb{Z}^+\}$ is a set of components of the SWGMM, and π_j is the mixing factor satisfying $0 \leq \pi_j \leq 1$. The expectation–maximization (EM) algorithm [14] is used to estimate the parameters of the SWGMM.

B. Mode Seeking and Clustering

To determine the number of modes present in a given area, mode seeking is a crucial step. In this work, we reformulate and adapt the GridShift (GS) [15] to suit our case. In the first step, we assume $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ represent the set of data points, where each $\mathbf{x}_i \in \mathbb{R}^{N\times 2}$ corresponds to 2D spatial data. The entire data space is divided into equal and disjoint grid cells of predefined length h, which serves as the bandwidth of the cell. Grid cells containing at least one data point are considered active cells, while the remaining cells are classified as inactive. Each active grid cell is labeled as a cluster, with its resident data points assigned as members of that cluster.

GS then shifts the locations of the active grid cells based on their new centroids. During this process, active grid cells that converge to the same location are merged. These steps are repeated iteratively until convergence is achieved. Upon convergence, GS returns the clustering results, including the centroids and membership assignments, in terms of the active grid cells. The detailed procedure of the algorithm can be found at our previous paper [16]. The data points for each cluster can be extracted based on their membership values. By applying GS to the spatial locations, we derive context-aware cells, as depicted in Fig. 3.

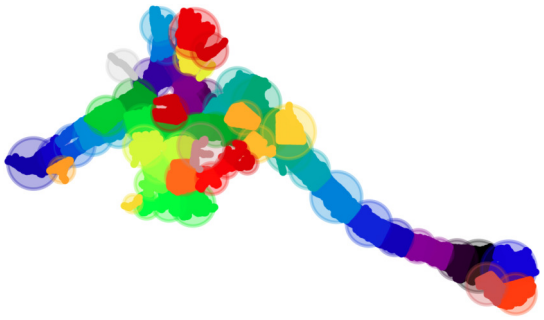

Fig. 3. Our method's measurement discretization procedure. Each color with the circle represents a different cell we generated from the ATC dataset [17]. The cells utilized in our model are determined based on the density of historical observations and their spatial distribution.

C. Learning Temporal Patterns

In this step, we employ Neural SDEs to learn the temporal patterns. SDEs describe systems influenced by both deterministic trends and random fluctuations. In this work, we incorporate the mean speed $\bar{\rho}$ and mean angle $\bar{\theta}$ from the SWGMM into the SDE:

$$\mathrm{d}(\bar{\rho}, \bar{\theta}) = f^{\gamma}\left((\bar{\rho}, \bar{\theta}), t\right) \mathrm{d}t + g^{\gamma}\left((\bar{\rho}, \bar{\theta}), t\right) \mathrm{d}W_t, \tag{4}$$

where W_t represents independent Wiener processes. We train the Neural SDEs to predict motion patterns within each cell across different time steps, learning the underlying dynamics. The model takes the first time step as input and is trained using sequentially estimated SWGMM outputs as ground truth. We employ the Milstein method [18] to solve the SDEs, obtaining predictions $\bar{\rho}^{\gamma}$ and $\bar{\theta}^{\gamma}$ at each time period. The loss function is defined as:

$$\mathcal{L}(\gamma) = \frac{1}{T} \sum_{t=1}^{T} \left\| (\bar{\rho}^{t}, \bar{\theta}^{t}) - (\bar{\rho}^{t,\gamma}, \bar{\theta}^{t,\gamma}) \right\|^2, \tag{5}$$

where T is the total number of time steps.

This formulation ensures that motion dynamics are influenced by the statistical properties of the clusters while allowing for stochastic variations. In our implementation, both the deterministic and stochastic components are parameterized by Multi-Layer Perceptrons (MLPs). Learning from SDEs, the model captures the underlying dynamics between time.

D. Event-Triggered Mechanism

Although SDEs can capture random fluctuations through the diffusion term, they still cannot effectively learn drastic transitions in objects' motion when the amount of data is insufficient. To tackle this challenge, we leverage a diffusion model to generate intermediate data between two-time points where sudden behavioral shifts are observed. This diffusion model aids in smoothing the training data and facilitates the learning of dynamics through SDEs, enhancing their capacity to capture rapid changes in motion patterns.

We assume that these abrupt changes are often driven by real-world events. Inspired from the control theory, we incoporate the event-triggered mechanism [19] that determines when the diffusion model should generate intermediate data. The mechanism in our work is defined as follows:

$$\begin{aligned} &\Delta\left((\bar{\rho}^{t+1}, \bar{\theta}^{t+1}), (\bar{\rho}^{t}, \bar{\theta}^{t})\right) - \\ &\frac{1}{T-1} \sum_{t=1}^{T-1} \Delta\left((\bar{\rho}^{t+1}, \bar{\theta}^{t+1}), (\bar{\rho}^{t}, \bar{\theta}^{t})\right) \geq \delta, \end{aligned} \tag{6}$$

where δ is a predefined threshold. The event is triggered when the abrupt change exceeds the average change by at least δ. This means that the diffusion model will be used only when the observed change is significantly larger than usual.

When the change in the output of the SWGMM satisfies Equation (6) at timestamp $t+1$, the diffusion model processes data from consecutive timestamps $t+1$ and t, represented as $\boldsymbol{v}$. In the forward process, the model gradually transforms $\boldsymbol{v}$ into Gaussian noise through a series of steps. This transformation is described by the following equation:

$$q(\boldsymbol{v}^{k+1}|\boldsymbol{v}^{k}) = \mathcal{N}(\boldsymbol{v}^{k+1}; \sqrt{\alpha^{k}}\boldsymbol{v}^{k}, (1-\alpha^{k})I), \tag{7}$$

where α^{k} is a hyperparameter that controls the variance schedule, determining how much noise is added at each step k. After K steps, the final state $\boldsymbol{v}^{K}$ becomes a sample from a standard Gaussian distribution. The reverse process aims to denoise $\boldsymbol{v}^{k}$ step by step, moving backwards from the noisy state to the original data. This is modeled as:

$$p^{\phi}(\boldsymbol{v}^{k-1}|\boldsymbol{v}^{k}) = \mathcal{N}(\boldsymbol{v}^{k-1}|\mu^{\phi}(\boldsymbol{v}^{k}, k), \Sigma^{\phi}(\boldsymbol{v}^{k}, k)), \tag{8}$$

where μ^{ϕ} and Σ^{ϕ} are the mean and covariance functions, respectively, parameterized by neural networks. These functions predict the mean and covariance of the distribution at each step k, allowing the model to reverse the noise addition process. The loss function for training the diffusion model is based on the denoising diffusion probabilistic models framework. It is given by [20]:

$$\mathcal{L}(\phi) = \mathbb{E}_{k, \boldsymbol{v}_0, \epsilon}\left[\|\epsilon - \epsilon^{\phi}(\boldsymbol{v}^{k}, k)\|^2\right], \tag{9}$$

where $\epsilon^{\phi}(\boldsymbol{v}^{k}, k)$ is from the neural network that predicts the noise $\epsilon \sim \mathcal{N}(0, I)$ introduced in the forward process.

Finally, the data generated by the diffusion model is modeled using the SWGMM after clustering to obtain the intermediate state, which is then utilized for training the SDEs.

III. Experiments

A. Experiment Setting

To evaluate the mapping capability of our framework, we conducted experiments using the ATC pedestrian tracking dataset [17], which provides real-world pedestrian data collected from the Asia and Pacific Trade Shopping Center in Osaka, Japan. The goal of the evaluation is to assess the quality of the map representation for pedestrian motion patterns across varying time intervals in non-stationary environments. The original dataset spans from 9:40 to 20:20. To temporally utilize the data, we established a half-hour time step, dividing the day into 20 intervals for presenting mapping results.

In terms of implementation, the Neural SDE and diffusion model were developed using PyTorch [21]. For both drift

function and diffusion function, there are 3 MLP layers for each input and each hidden layer includes 64 neurons. The SDE model was optimized using the Adam optimizer [22] with a learning rate of 0.01. The model was trained over 100 epochs on a single NVIDIA RTX A2000 GPU, with an average training time of approximately 3 minutes per cell. All cells were trained simultaneously to ensure efficiency.

For the diffusion model, our model was implemented based on U-Net [23] architecture with 64 neurons for each layer. The diffusion steps were 1000. We utilized the AdamW optimizer [24] with a learning rate of 10^{-4}. The model was trained with a batch size of 128 over 100 epochs. The average training time for each cell was approximately 10 minutes, and similar to the SDE model, all cells were trained concurrently.

B. Baseline and Metrics

For comparative analysis, two state-of-the-art methods CLiFF-Map [8] and MSSM [9] were employed as baseline methods. All of the methods were trained and evaluated under the same time horizon framework to ensure consistency and fairness in the comparison.

The divergence estimator proposed by Wang et al. [25] was employed to assess the quality of the map in absolute values. At each location within the map, there is associated a true physical process which is generating observations from an unknown PDF $\boldsymbol{\xi}^t$. Wang's divergence estimator calculates the differences between the model's output and the original data. For each query location in the map, a motion distribution is obtained from the output of the model. Concurrently, there is a set of observations $\{\mathbf{v}_1^o, \ldots, \mathbf{v}_n^o\}$ from the original dataset. Wang's divergence estimator is then applied to estimate the divergence between these two distributions, considering only the samples derived from them. The estimator is computed as follows:

$$\hat{D}_{n,m}\left(\boldsymbol{\xi}^t \| \boldsymbol{\xi}\right) = \frac{d}{n}\sum_{i=1}^{n}\log_2\frac{v_k(i)}{\rho_k(i)} + \log_2\frac{m}{n-1} \tag{10}$$

In the divergence estimation process, the distance $\rho_k(i)$ between $\mathbf{v}_i^o$ and its k-nearest neighbors (k-NN) within $\left\{\mathbf{v}_j^o\right\} j \neq i$ is compared with the distance $v_k(i)$ between $\mathbf{v}_i^o$ and its k-NN in $\left\{\mathbf{v}_j^q\right\}$, where $\left\{\mathbf{v}_j^q\right\}$ denotes the observations queried from one model. In practice, the parameter k is set to be 1, indicating that the algorithm considers only the nearest single neighbour to the new data point.

C. Results

Evaluations were conducted across various time horizons using a dataset from November 14, 2012. The quantitative evaluation results of the mapping are presented in Table I. Our proposed method outperforms CLiFF-Map and MSSM in most time horizons, particularly in the initial few horizons. This enhanced performance is possibly attributed to the model's ability to detect abrupt changes, which frequently occur during rush hours. This suggests that our method is more effective at capturing patterns and abrupt changes in flow density, especially during periods of high activity. Such capability is crucial for reducing the deployment effort of robots in dynamic environments and making them more suitable for practical real-world applications.

TABLE I
QUANTITATIVE EVALUATION RESULTS.

Horizon	Observations	Divergence [bit]		
		CLiFF [8]	MSSM [9]	ETMoD
10:00 - 10:30	517360	0.3765	0.3518	**0.1498**
10:30 - 11:00	668494	0.3518	0.3397	**0.2052**
11:00 - 11:30	655704	0.4207	0.3982	**0.2552**
11:30 - 12:00	792447	0.3797	**0.3589**	0.4045
12:00 - 12:30	1046258	0.3614	0.3568	**0.2312**
12:30 - 13:00	1138716	0.3429	0.3451	**0.2116**
13:00 - 13:30	892426	0.3647	0.3639	**0.3308**
13:30 - 14:00	772909	0.4122	**0.3826**	0.4432
14:00 - 14:30	770234	0.3953	0.3674	**0.3184**
14:30 - 15:00	810805	0.3555	0.3429	**0.2995**
15:00 - 15:30	831551	0.3776	0.3567	**0.2374**
15:30 - 16:00	863024	0.3412	**0.3407**	0.3639
16:00 - 16:30	800019	0.3708	**0.3598**	0.4242
16:30 - 17:00	828187	0.3829	**0.3691**	0.4535
17:00 - 17:30	629797	0.4243	0.3971	**0.3821**
17:30 - 18:00	773548	**0.3657**	0.3668	0.4306
18:00 - 18:30	702478	0.4060	**0.3896**	0.4090
18:30 - 19:00	529632	0.4397	0.4107	**0.3960**
19:00 - 19:30	321988	0.5183	**0.4823**	0.4892
19:30 - 20:00	327273	0.5032	**0.4812**	0.5077
Average		0.3985	0.3761	**0.3472**

All methods are influenced by variations in observation density, as this factor significantly impacts the distribution acquisition. MSSM exhibits more robustness than our method in scenarios with fewer observations. This is probably due to the fact that diffusion models introduce randomness during data generation. While they improve performance on average, at specific time intervals, the generated samples might deviate slightly due to stochastic variations.

In contrast, our model exhibits improved performance in scenarios with higher observation density, indicating its ability to aggregate information effectively when more data points are available. This suggests that our model excels in aggregating and leveraging large amounts of data, making it suitable for applications in densely observed environments. Notably, our SDE model is trained using the outcomes of SWGMM, without the direct dependency for large amounts of real-world data. This improves the training efficiency while still enabling the model to effectively capture the underlying dynamics and remain consistent with the motion patterns.

IV. CONCLUSIONS

In this paper, we introduced the ETMoD framework, a novel method for modeling spatial motion patterns in non-stationary environments. By combining a grid-shifting mechanism, Neural SDEs and an event-triggered mechanism, ETMoD effectively captures abrupt changes in motion patterns, outperforming state-of-the-art methods in accuracy and training efficiency. Experimental results on real-world tracking datasets demonstrate its robustness in dynamic scenarios, particularly during peak activity periods.

REFERENCES

[1] T. P. Kucner, M. Magnusson, S. Mghames, L. Palmieri, F. Verdoja, C. S. Swaminathan, T. Krajník, E. Schaffernicht, N. Bellotto, M. Hanheide *et al.*, "Survey of maps of dynamics for mobile robots," *The International Journal of Robotics Research*, vol. 42, no. 11, pp. 977–1006, 2023.

[2] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 895–935, 2020.

[3] Y. Zhu, A. Rudenko, T. P. Kucner, L. Palmieri, K. O. Arras, A. J. Lilienthal, and M. Magnusson, "Cliff-lhmp: Using spatial dynamics patterns for long-term human motion prediction," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 3795–3802.

[4] C. S. Swaminathan, T. P. Kucner, M. Magnusson, L. Palmieri, and A. J. Lilienthal, "Down the cliff: Flow-aware trajectory planning under motion pattern uncertainty," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 7403–7409.

[5] L. Palmieri, T. P. Kucner, M. Magnusson, A. J. Lilienthal, and K. O. Arras, "Kinodynamic motion planning on gaussian mixture fields," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 6176–6181.

[6] F. Surma, T. P. Kucner, and M. Mansouri, "Multiple robots avoid humans to get the jobs done: An approach to human-aware task allocation," in *2021 European Conference on Mobile Robots (ECMR)*. IEEE, pp. 1–6.

[7] M. Hanheide, D. Hebesberger, and T. Krajnik, "The when, where, and how: An adaptive robotic info-terminal for care home residents - a long-term study," in *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2017, pp. 341–349.

[8] 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 Robotics and Automation Letters*, vol. 2, no. 2, pp. 1093–1100, 2017.

[9] J. Shi and T. P. Kucner, "Learning temporal maps of dynamics for mobile robots," *Robotics and Autonomous Systems*, vol. 184, p. 104853, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889024002379

[10] P. Kidger, J. Foster, X. C. Li, and T. Lyons, "Efficient and accurate gradients for neural sdes," *Advances in Neural Information Processing Systems*, vol. 34, pp. 18 747–18 761, 2021.

[11] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[12] K. V. Mardia and P. E. Jupp, *Directional statistics*. John Wiley & Sons, 2009.

[13] A. Roy, S. K. Parui, and U. Roy, "Swgmm: a semi-wrapped gaussian mixture model for clustering of circular–linear data," *Pattern Analysis and Applications*, vol. 19, pp. 631–645, 2016.

[14] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society: series B (methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[15] A. Kumar, O. S. Ajani, S. Das, and R. Mallipeddi, "Gridshift: A faster mode-seeking algorithm for image segmentation and object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8131–8139.

[16] 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 *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2025, pp. 12 924–12 931.

[17] D. Brščić and T. Kanda, "Changes in usage of an indoor public space: Analysis of one year of person tracking," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 2, pp. 228–237, 2014.

[18] G. Mil'shtejn, "Approximate integration of stochastic differential equations," *Theory of Probability & Its Applications*, vol. 19, no. 3, pp. 557–562, 1975.

[19] W. M. Heemels, K. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," 12 2012, pp. 3270–3285.

[20] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *ArXiv*, vol. abs/2006.11239, 2020.

[21] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Springer International Publishing, 2015, pp. 234–241.

[24] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=Bkg6RiCqY7

[25] Q. Wang, S. R. Kulkarni, and S. Verdú, "Divergence estimation for multidimensional densities via k-nearest-neighbor distances," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2392–2405, 2009.