Inertial Navigation on Extremely Resource-Constrained Platforms: Methods, Opportunities and Challenges

[1]Swapnil Sayan Saha, [2]Yayun Du, [3]Sandeep Singh Sandha, [4]Luis Antonio Garcia, [5]Mohammad Khalid Jawed, and [6]Mani Srivastava

[1,5,6]*University of California, Los Angeles,* [2]*Northwestern University,* [3]*Abacus.AI,* [4]*University of Southern California*

[1,4,5,6]Los Angeles, CA, USA, [2]Evanston, IL, USA, [3] San Francisco, CA, USA

[1]swapnilsayan@g.ucla.edu, [2]duyayun1hit@northwestern.edu, [3]sandeep@abacus.ai, [4]lgarcia@isi.edu, [5]khalidjm@seas.ucla.edu, [6]mbs@ucla.edu

2023 IEEE/ION Position, Location and Navigation Symposium (PLANS) | 978-1-6654-1772-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/PLANS53410.2023.10139997

***Abstract*—Inertial navigation provides a small footprint, low-power, and low-cost pathway for localization in GPS-denied environments on extremely resource-constrained Internet-of-Things (IoT) platforms. Traditionally, application-specific heuristics and physics-based kinematic models are used to mitigate the curse of drift in inertial odometry. These techniques, albeit lightweight, fail to handle domain shifts and environmental non-linearities. Recently, deep neural-inertial sequence learning has shown superior odometric resolution in capturing non-linear motion dynamics without human knowledge over heuristic-based methods. These AI-based techniques are data-hungry, suffer from excessive resource usage, and cannot guarantee following the underlying system physics. This paper highlights the unique methods, opportunities, and challenges in porting real-time AI-enhanced inertial navigation algorithms onto IoT platforms. First, we discuss how platform-aware neural architecture search coupled with ultra-lightweight model backbones can yield neural-inertial odometry models that are 31-134$\times$ smaller yet achieve or exceed the localization resolution of state-of-the-art AI-enhanced techniques. The framework can generate models suitable for locating humans, animals, underwater sensors, aerial vehicles, and precision robots. Next, we showcase how techniques from neurosymbolic AI can yield physics-informed and interpretable neural-inertial navigation models. Afterward, we present opportunities for fine-tuning pre-trained odometry models in a new domain with as little as 1 minute of labeled data, while discussing inexpensive data collection and labeling techniques. Finally, we identify several open research challenges that demand careful consideration moving forward.**

This research was sponsored in part by the Air Force Office of Scientific Research (AFOSR) under Cooperative Agreement FA9550-22-1-0193; the IoBT REIGN Collaborative Research Alliance funded by the Army Research Laboratory (ARL) under Cooperative Agreement W911NF-17-2-0196; the NIH mHealth Center for Discovery, Optimization and Translation of Temporally-Precise Interventions (mDOT) under award 1P41EB028242; the National Science Foundation (NSF) under awards CNS-1705135, CNS-1822935, IIS-1925360, CNS-2213839, and CMMI-2047663; the National Institute of Food and Agriculture, USDA under awards 2021-67022-342000 and 2021-67022-34200; and, the CONIX Research Center, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the AFOSR, ARL, DARPA, NIH, NSF, SRC, USDA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

***Index Terms*—Bayesian, dead-reckoning, inertial, kalman filtering, neural architecture search, neural networks, neurosymbolic, odometry, platform-aware, sequence learning, TinyML**

I. Introduction

Inertial odometry uses accelerometer, gyroscope, or magnetometer data for indirectly estimating the change in an object's location and orientation with intermittent updates from infrastructure-dependent localization services [1]–[3]. Inertial navigation provides an always-available, small footprint, high resolution, and ultra-low-power pathway for dead-reckoning [1], [4]–[9]. MEMS inertial measurement units (IMU) encompass a broad domain spectrum. Underwater autonomous vehicles, picosatellites, micro-unmanned aerial vehicles, and precision agricultural robots use inertial sensors for fast, always-on, and energy-efficient state estimation and navigation [8], [10]–[23]. Smartphones and wearables feature on-board transportation mode recognition, pedestrian dead-reckoning (PDR), fitness monitoring, and human activity recognition [1], [4], [24]–[29]. Virtual reality headsets and earables estimate high-resolution head pose using IMU for haptics, gaming, and augmented reality applications [10], [24], [30]–[33]. Marine health trackers and wildlife tags use motion sensors for biologging tagged animal behavioral patterns within a tight power and weight budget [34]–[37]. Industrial robots and machinery employ multi-IMU arrays for vibration analysis, joint state estimation, and anomaly detection [38]–[41]. The market for inertial sensors was valued at USD 16.4 billion in 2020, with an expected value of USD 22.3 billion by 2026 at 5.4% CAGR[1].

Despite their widespread ubiquity, MEMS inertial sensors suffer from soft and hard iron distortions, additive white Gaussian noise (AWGN), angular random walk (ARW), and time-varying bias instability (BI) [42]. Thus, naive double integration (NDI) of accelerometer readings and magnetic heading estimation for dead-reckoning accumulates location error that

[1]https://www.reportlinker.com/p05798878

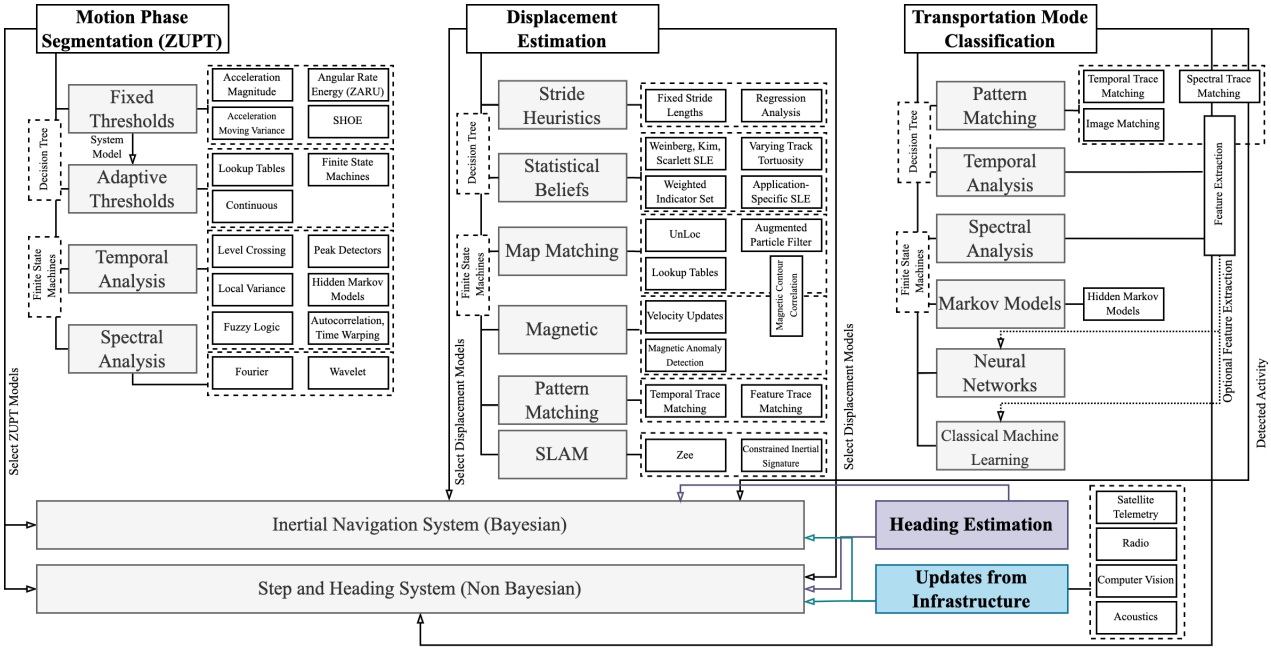

Fig. 1. Motion cycle segmentation, displacement estimation, and transportation mode recognition techniques used by classical inertial navigation algorithms.

drifts cubically with time [7], [43]–[45]. Traditionally, this drift is handled using belief-based velocity updates, heuristic drift reduction, map-matching, transportation phase detection, Bayesian filtering, and kinematic pattern matching [46]–[57]. While being computationally tractable, these system models are only linear approximations of the real-world state-evolution, failing to handle non-linear complex motions or domain shifts due to non-optimal parametrization and no "one size fits all" analytical solution [1], [5], [7], [18]–[20], [45], [58]–[61]. Recently, AI-enhanced inertial navigation has overcome the need for accurate system modeling, achieving superior long-term odometric resolution over classical methods [7], [19], [45], [58], [62]–[65]. Machine learning (ML) frameworks model sensor covariances and noise [18], [64], [66]–[68], estimate velocity profiles [62], [65], [69]–[71], provide state pseudo-measurements to Bayesian filters [63], [72]–[74], or estimate position end-to-end [1], [5], [7], [19], [58], [60]. These techniques, on their own, are unsuitable for real-time deployment on ultra-resource-constrained devices (e.g., microcontrollers) due to excessive memory and computational resource usage, need large amounts of labeled data in the target domain, suffer from inertial perturbations, and are unable to learn physics-based constraints and kinematic models [1], [5], [11], [61]. The real-time operation, deployability, interpretability, data efficiency, and physics awareness of AI-enhanced algorithms are difficult to guarantee, making their adoption on Internet of Things (IoT) platforms challenging.

In this article, we discuss several techniques for achieving feasible real-time AI-enhanced inertial navigation on resource-constrained platforms. *Firstly*, we introduce *TinyOdom*, a platform-aware framework for developing neural-inertial odometry models for microcontrollers and in-sensor processing units [1]. *TinyOdom* exploits advances in tiny machine learning (TinyML) to automatically optimize ultra-lightweight neural network (NN) backbones using platform-in-the-loop Bayesian neural architecture search (NAS), generating odometry models for various applications that are deployable on IoT platforms without sacrificing resolution significantly. *TinyOdom* talks to the target hardware during the optimization process to guarantee that the model fits within the platform constraints. *Secondly*, we showcase how neuro $\cup$ compile [symbolic] and symbolic [neuro] paradigms from neurosymbolic AI leads to physics-aware neural-inertial navigation models. Specifically, we develop a physics, velocity, and magnetometer-centric sequence learning formulation robust to gravity pollution, inertial disturbances, varying sensor placement, and heading rate singularity without needing large amounts of training data [1]. We also showcase a neural-Kalman filter for optimally combining interpretable physics-based models with NN [11]. *Thirdly*, we discuss transfer-learning techniques for fine-tuning pre-trained odometry models using small amounts of real-world labeled data in the target domain, while illustrating an automated video-processing and labeling pipeline for collecting high-resolution labeled IMU-location data [1], [11], [75]. *Lastly*, we outline several open challenges and ideas for future research.

The rest of the paper is organized as follows. Section II provides background on inertial navigation using both classical and AI-enhanced techniques. Section III delineates the platform-aware neural-inertial navigation framework and robust sequence learning formulation. Section IV presents the neural-Kalman filter. Section V details the transfer learning pipeline. Finally, Section VI provides concluding remarks and open research challenges.

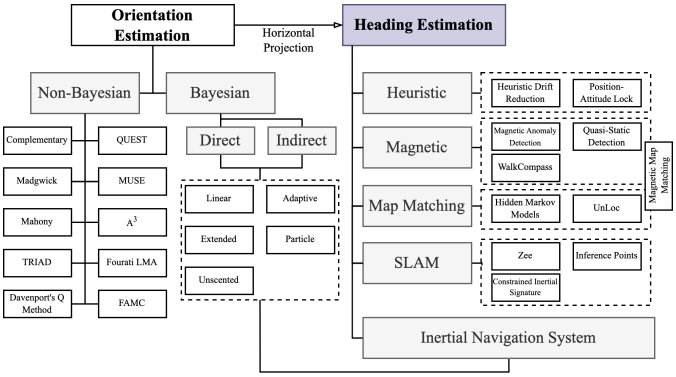

Fig. 2. Summary of heading estimation techniques used by classical inertial navigation algorithms.

II. Background

The accelerometer $\mathbf{f}^b$, gyroscope $\boldsymbol{\omega}_{ib}$, and magnetometer $\mathbf{b}^n$ within a MEMS IMU are modelled as [42]:

$$\mathbf{f}^b = \mathbf{R}^{bn}(\mathbf{a}^n_{nn} - \mathbf{g}) + \mathbf{b}_a + \mathbf{n}_a \tag{1}$$

$$\boldsymbol{\omega}_{ib} = \boldsymbol{\omega}_{nb} + \mathbf{b}_g + \mathbf{n}_g \tag{2}$$

$$\mathbf{b}^n = \mathbf{R}^{bn}\mathbf{m}^n + \mathbf{n}_b, \quad \mathbf{m}^n = [cos\delta \quad 0 \quad sin\delta] \tag{3}$$

where, $\mathbf{R}^{bn}$ is the rotation matrix from the navigation frame n to the body frame b, $\mathbf{a}_{nn}$ is the latent linear acceleration, $\mathbf{g}$ is the gravity vector, $\boldsymbol{\omega}_{nb}$ is the latent angular velocity, and δ is the geomagnetic inclination. $\dot{\mathbf{b}_a} \sim \mathcal{N}(0, \mathbf{Q}_a)$ and $\dot{\mathbf{b}_g} \sim \mathcal{N}(0, \mathbf{Q}_g)$ are BI gradients with covariance $\mathbf{Q}$. $\mathbf{n}_a \sim \mathcal{N}(0, \boldsymbol{\Sigma}_a)$, $\mathbf{n}_g \sim \mathcal{N}(0, \boldsymbol{\Sigma}_\omega)$, and $\mathbf{n}_b \sim \mathcal{N}(0, \boldsymbol{\Sigma}_m)$ are AWGN with covariance $\boldsymbol{\Sigma}$. Assuming the absence of geomagnetic perturbations, sensor placement offsets, and $\mathbf{b}_a$, the latitude (ϕ) and longitude (λ) of an object in the inertial frame I are calculated by NDI of accelerometer readings (a^I_x, a^I_y, a^I_z), using magnetometer readings for heading estimation (m^I_x, m^I_y, m^I_z) [34]:

$$\phi_t = \arcsin\left(\sin\phi_{t-1} \cdot \cos\tfrac{s_t}{R_E} + \cos\phi_{t-1} \cdot \sin\tfrac{s_t}{R_E} \cdot \cos H\right) \tag{4}$$

$$\lambda_t = \lambda_{t-1} + \arctan 2\left(\sin H \cdot \sin\tfrac{s_t}{R_E} \cdot cos\phi_{t-1}, \cos\tfrac{s_t}{R_E} - \sin\phi_{t-1} \cdot \sin\phi_t\right) \tag{5}$$

where,

$$H = \arctan\left(\frac{m^I_{y,t}}{-m^I_{x,t}}\right) \cdot \frac{180}{\pi} \tag{6}$$

$$s_t = \begin{cases} \beta\sqrt{{}^Ia^2_{x,t} + {}^Ia^2_{y,t} + {}^Ia^2_{z,t}} + \gamma, & \sqrt{{}^Ia^2_{x,t} + {}^Ia^2_{y,t} + {}^Ia^2_{z,t}} > \alpha \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

R_E is 6.371×10^6m, α is the noise rejection threshold, and β and γ are scaling constants. Without drift and noise compensation, H has an error of $\pm 100^\circ$ [76], while the error $\sigma(t)$ in ϕ and λ explodes cubically with time [43]:

$$\sigma(t) = s_t\sqrt{(2 \cdot ARW \cdot \sqrt{t^3}/3)^2 + (BI \cdot t^2/2)} \tag{8}$$

Inertial navigation systems (INS) counteract the curse of drift in the above formulation either using physics-based heuristic priors (Section II-A) or ML methods (Section II-B).

A. Classical Inertial Navigation Techniques

Heuristic techniques decompose the dead-reckoning problem into heading estimation, motion cycle segmentation, transportation mode classification, and displacement estimation [4], [6], [8], [76], summarized in Fig. 1 and Fig. 2. The selected models are combined either using INS or step and heading systems (SHS) for legged objects.

Heading Estimation: For strapdown inertial navigation, the heading is obtained by horizontal projection $\vec{\mathbf{u}_t}^{N,horz}$ of the object attitude $\mathbf{q}_t$ [53]:

$$\vec{\mathbf{u}_t}^{N,horz} = \begin{bmatrix} \mathbf{I}_{2\times 2} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} \mathbf{C}^N_B(\mathbf{q}_t)\vec{\mathbf{u}_t}^B \tag{9}$$

$\mathbf{C}^N_B$ is the direction cosine matrix (DCM) and $\mathbf{u}_t{}^B$ is the unit vector nonparallel to the ground plane perpendicular. The simplest algorithm (complementary) fuses low-pass filtered accelerometer data and high-pass filtered gyroscope data [77]:

$$\text{q2e}(\mathbf{q}_t) = (1-\alpha)\left(\text{q2e}(\mathbf{q}_{t-1}) + \Delta t\mathbf{R}^{bn}\boldsymbol{\omega}_t\right) + \alpha \begin{bmatrix} \arctan 2(a^I_{y,t}, a^I_{z,t}) \\ \arctan 2(-a^I_{x,t}, \sqrt{{a^I_{y,t}}^2 + {a^I_{z,t}}^2}) \\ 0 \end{bmatrix} \tag{10}$$

q2e$(\cdot)$ is the quaternion to Euler conversion. Non-linear complementary filters [78] correct gyroscope quaternion e2q($\Delta t\mathbf{R}^{bn}\boldsymbol{\omega}_t$) error using accelerometer-magnetometer fusion via proportional-integral compensation (Mahony) [79], gradient-descent (Madgwick) [80], quasi-static motion detection (A^3) [81], and Levenberg-Marquardt algorithm (Fourati LMA) [82]. Least-square solvers (e.g., TRIAD [83], QUEST [84], Davenport's Q Method [85], and Fast Accelerometer-Magnetometer Combination (FAMC) [86]) pose the attitude estimation problem as Wahba's problem [87], finding DCM between a reference vector and an observation vector. MUSE [44] uses geomagnetic North as the primary anchor instead of gravity. For optimal linear attitude estimation, a Kalman filter (KF) fuses noisy sensors under Bayesian variations using a prediction and update model [88]. The extended KF (EKF) [89] and unscented KF (UKF) [90] linearizes non-linear MARG models using Jacobians and sigma points around the running moments [91]. Indirect KF model sensor errors as state vectors, while direct KF output object attitude [53], [92]–[95]. Adaptive KF allows the filter covariance and noise parameters to be estimated on the fly [96]–[98].

To counteract drift in $\vec{\mathbf{u}_t}^{N,horz}$, heuristic drift reduction and position-attitude lock discards heading updates during periods of negligible translational motion or straight walking [46], [51], [99] with the aid of transportation mode classification and curve angle segmentation. Magnetic drift reduction techniques correct accelerometer and gyroscope errors in presence of geomagnetic perturbations and sensor placement offset using iterative magnetic triangulation (WalkCompass) or by detecting periods of flat and abnormal magnetic field gradients [100]–[103]. Known anomaly earth magnetic fields, magnetic contours, and accelerometer-gyroscope signatures may serve as virtual landmarks within a known localization space [49], [104]–[107], forming a map and dramatically limiting the possible heading angles [108]. Particle filters allow

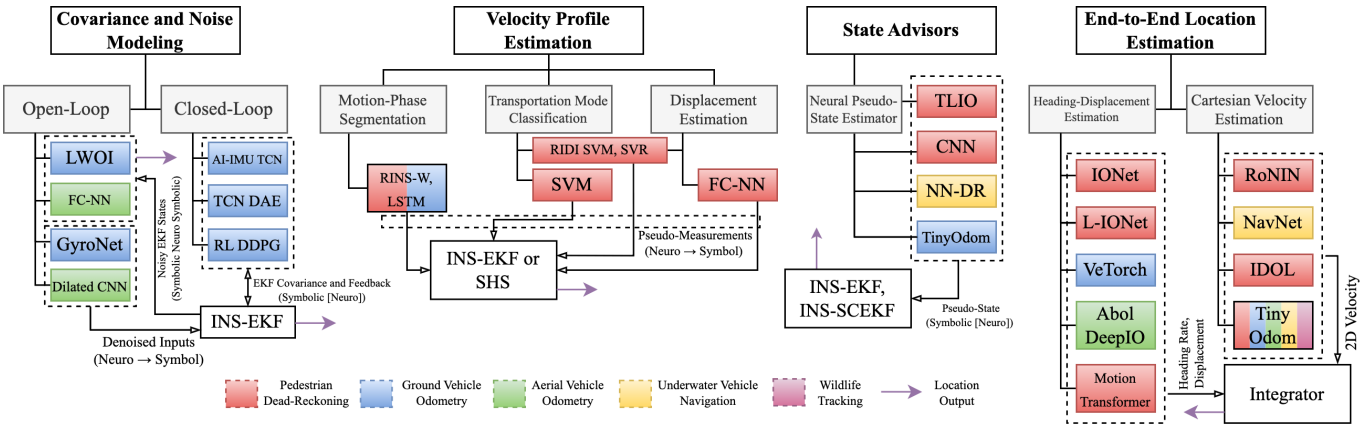

Fig. 3. Summary of AI-enhanced inertial navigation techniques and intended applications. INS implies inertial navigation system, SHS implies step and heading system, EKF is an extended Kalman filter, and SCEKF refers to stochastic cloning EKF.

these semantic walking direction segments (Zee), inference points, and inertial landmarks to be detected on the fly using simultaneous localization and mapping (SLAM) [103], [109]–[111], indirectly modeling the IMU noise and biases as filter disturbances. Finally, INS can optimally combine several attitude and heading estimation models using Kalman or information filter [112] primitives. Information filter is faster than KF but mathematically equivalent [113] and preferred for real-time navigation.

Motion Cycle Segmentation: Motion phase segmentation or zero velocity update (ZUPT) detects whether sufficient translational motion has occurred or not to activate the displacement estimation module, based on the gait or motion cycle [4], [26], [51], [52]. Static belief-based threshold detectors such as the accelerometer vector sum, accelerometer rolling variance, angular rate energy (ZARU), and cascade (SHOE) segregate the motion cycle for a single transportation mode [47], [51], [52], [101], [102], [114]–[120]. Adaptive thresholds are robust across transportation modes, dynamically varying fixed threshold hyperparameters discretely (finite state machines (FSM) and lookup tables) or continuously [121]–[128]. Temporal detectors exploit the periodic properties of the motion cycle to extract recurrent contextual dynamics in the gait cycle, further improving applicability under varying activity primitives [4], [26], [76]. Hard temporal detectors include g-crossings [53], [129]–[131], local extrema [49], [110], [111], [129], [132]–[136], and local variance [118], [119], [124], [126], [137]–[139]. Fuzzy logic and hidden Markov models (HMM) turn hard constraints into generalizable, statistically magnified, and granular soft thresholds [139]–[148]. In particular, HMM robustly model the transition and time evolution between non-observable motion states probabilistically, with the phases forming a Markov chain. Autocorrelation and dynamic time warping allow temporal analysis on the same motion cycle at different time-scales [109], [110], [136]. Orthogonally, spectral analysis identifies harmonics associated with particular phases, filtering out inertial disturbances [140], [141]. Finally, several ZUPT models can jointly form a decision tree or FSM, or be fed to an INS or SHS.

Displacement Estimation: The average stride length derived from object statics or step frequency via regression analysis forms the simplest displacement estimation algorithm, with the regression coefficients solved using recursive least squares, particle filters, and online/offline calibration [108], [110], [123], [126], [130]–[134], [149], [150]. Dynamic stride length estimators (SLE) operate on accelerometer statistics, step frequency, and lookup indicator sets, aided by transportation mode detectors and object dynamics priors to vary track tortuosity [34], [53], [116], [121], [127], [129], [135], [138], [151]–[155]. Inertial, magnetic, and geometric landmarks associated with a particular space can adaptively calibrate and constrain stride lengths, either discovered via SLAM on-the-fly (Zee) or known pre-deployment and stored as graphs, spatial maps, or lookup tables [49], [108], [109], [117], [141]. In particular, stored anomaly earth magnetic fields and magnetic contours can be correlated with geophysical fields to form the basis for geomagnetic self-localization in multirotor, ships, and aircrafts [104]–[107], [156]. Known inertial traces signifying object strides and time periods can be mapped to memory and matched with incoming IMU sequences [57], [157]. Select displacement models are fed to an uncertainty-aware INS using Kalman filters [51], [95], [102], [114], [137], [139], [140], [148] or non-Bayesian SHS [115], [119], [124].

Transportation Mode Classification: Transportation mode classifiers make motion phase segmentation, displacement estimation, and heading estimation robust to activity primitives, sensor placements, and topography [26] [76]. Classical activity detectors form decision trees and FSM from temporal and spectral features, match known inertial traces with incoming IMU data, or use HMM to form probabilistic FSM. Recent transportation mode classification frameworks opt for model-free and data-driven ML and NN-based mobility classification [4], [26], [76], [76], [158]. IMU data is collected in diverse settings, pre-processed, and labeled. The dataset is then partitioned and features are optionally extracted, on which the ML algorithms are trained [25], [27], [159]–[163].

B. Neural Inertial Navigation Techniques

Fig. 3 outlines four classes of AI-enhanced inertial odometry techniques. The methods either aid in supplying model-free and application-agnostic velocity profiles, covariance matrices, noise parameters, or pseudo-states to classical INS or SHS, or estimate location end-to-end [1], [11]. These methods are generally superior at handling micro-vibrations, IMU noise, sensor offset, domain shifts, different activity primitives, varying topography, inertial perturbations, and non-linearities over classical methods without needing human-designed system models [1], [5], [7], [18]–[20], [45], [58]–[61].

Covariance and Noise Modeling: Conventional INS estimate the KF covariance matrices adaptively or offline using statistical techniques [164]. Statistical approaches for estimating the KF process and measurement noise include maximum likelihood estimation [165], covariance least squares [98], [166], Bayesian updates [167], and correlation technique [168]. These techniques, based on statistical tests on measurement error residuals, do not optimally capture the effects of varying motion dynamics on the covariance parameters [18]. NN are adept at finding black-box relations between covariance matrices and motion primitives [18], [169]. NN-based systems can be open-loop or closed-loop. Open-loop frameworks either remove noise and drift from IMU readings before supplying to an INS using GyroNet [170] or dilated convolutional NN (CNN) [66], or remove the effects of noise and drift from the INS position output using LWOI [64] or fully-connected NN (FC-NN) [68]. To counteract statistically non-optimal denoising under non-ideal gradient descent convergence, closed-loop frameworks use temporal convolutional networks (TCN), TCN with denoising autoencoders (DAE), or reinforcement learning (RL) to dynamically and tightly update KF covariance and noise parameters while receiving feedback about the correction performance [18], [169], [171]–[173]. Closed-loop systems follow the *symbolic [neuro]* neurosymbolic paradigm of combining NN with physics-based models (e.g., KF), while open-loop frameworks follow *neuro→symbol* or *symbolic neuro symbolic* paradigms [174], [175].

Velocity Profile Estimation: Long short-term memory (LSTM) networks are trained to segment motion cycles [62], [65]. A support vector machine is trained to classify activity primitives [45], [69], [71]. FC-NN or support vector regressors (SVR) are trained to perform stride length estimation [45], [70]. The model outputs are fed to INS or SHS for location estimation. This technique follows the *neuro→symbol* paradigm.

State Advisors: State advisors provide pseudo-state information (such as velocity and heading) to INS, aiding the KF and following the tightly coupled *symbolic [neuro]* paradigm. TLIO [63] and Cortes *et al.* [72] use CNN to provide a strapdown INS with displacement and KF covariance matrix during measurement update. NN-DR [73], [74] uses an FC-NN to provide heading pseudo-states to an EKF. TinyOdom [1] fuses an end-to-end neural inertial navigation system model with the GNSS measurement update model via a neural-EKF.

End-to-End Location Estimation: The absence of domain-specific INS or SHS makes this the preferred AI-enhanced inertial navigation technique [1], [11]. Deep inertial sequence learning is used to train a NN to either regress the heading and displacement [5], [19], [58], [61], [176] or the 2D Cartesian velocities [1], [7], [20], [60] of an object directly from inertial sensor windows. The latter formulation is robust to heading rate singularities [1], [11]. An integrator converts the model outputs to location. This approach is purely neural.

III. Platform-Aware Neural-Inertial Navigation

IONet [58] proposed the first end-to-end AI-enhanced inertial navigation technique based on deep inertial sequence learning, using the heading-displacement formulation:

$$(\Delta l_k, \Delta\psi_k) = y_{\theta^*}(\mathbf{v}^I(0), \mathbf{g}_0^I, \hat{\mathbf{a}}_{q:q+n}^I, \hat{\mathbf{w}}_{q:q+n}^I). \tag{11}$$

The NN y with parameters θ^* estimates the heading rate $\Delta\psi_k$ and the displacement rate Δl_k in polar coordinates from accelerometer $\hat{\mathbf{a}}_{q:q+n}^I$ and gyroscope windows $\hat{\mathbf{w}}_{q:q+n}^I$) of length n and stride s in the inertial frame I. The implicit task is to estimate the latent initial velocity $\mathbf{v}^I(0)$ and gravity vector direction $\mathbf{g}_0^I$ in each window at epoch k. The position (L_x, L_y) and learning algorithm to obtain θ^* are given as:

$$\begin{cases} L_{x,k} = L_{x,k-1} + \Delta l_k \cos(\Delta\psi_{k-1} + \Delta\psi_k) \\ L_{y,k} = L_{y,k-1} + \Delta l_k \sin(\Delta\psi_{k-1} + \Delta\psi_k) \end{cases} \tag{12}$$

$$\theta^* = \arg\min_{\theta} \mathcal{L}(y_{\theta^*}(\mathbf{X}), \mathbf{Y}) \tag{13}$$

$$\mathcal{L}(\cdot) = \sum ||\Delta l_{g,k} - \Delta l_k||_2^2 + \kappa||\Delta\psi_{g,k} - \Delta\psi_k||_2^2 \tag{14}$$

$\Delta l_{g,k}$ and $\Delta\psi_{g,k}$ are the ground truth displacement and heading rates, respectively. $\mathbf{X}$ and $\mathbf{Y}$ are the training data (IMU) and ground truth labels. κ is a weighing factor. The heading-displacement formulation has three shortcomings. *Firstly*, as $\Delta L_{y,g,k} \wedge \Delta L_{x,g,k} \rightarrow 0$, $\Delta\psi_{g,k} \uparrow$, leading to large spikes in $\Delta\psi_{g,k}$. This is known as *heading-rate singularity*, causing y to fail for motion dominated by rotational artifacts with little translational changes [1], [7], [20], [60]. *Secondly*, $\mathbf{g}_0^I$ polluted by coupling of linear and gravitational acceleration. *Gravity pollution* and gyroscope drift induce errors in the latent attitude estimate, degenerating coordinate frame normalization, and velocity projection [7], [44]. *Thirdly*, y is prone to supplying invalid outputs caused by high-frequency inertial signatures stemming from sensor placement offset, IMU noise, and rotational artefacts [1]. In addition, the architectures of y in existing frameworks are suitable for deployment on smartphones [5], [18], [61], [66] but not on low-end IoT platforms. A microcontroller typically has only 128 kB SRAM and 1 MB eFlash while a smartphone may have 4 GB RAM and 64 GB flash [177]. Reducing the memory-compute footprint of smartphone-class models using deep compression (pruning, quantization, and encoding) [178]–[180] or TinyML compiler optimizations [177], [181]–[184] for microcontrollers is challenging [177], [181], [185], [186].

A. Robust Neural-Inertial Sequence Learning

TinyOdom proposes a physics, velocity, and magnetometer-centric sequence learning formulation to handle the shortcomings of the heading displacement formulation [1], [11], [75].

Handling Heading Rate Singularity: We train y to predict 2D Cartesian velocities (v_x, v_y) instead of heading-displacement. Given the ground truth velocities $(v_{x,g}, v_{y,g})$, the sequence learning formulation, position, and the loss function (strided loss [7]) in the learning algorithm are given as:

$$(v_{x,k}, v_{y,k}) = y_{\theta^*}(\mathbf{v}^I(0), \mathbf{g}_0^I, \hat{\mathbf{a}}_{q:q+n}^I, \hat{\mathbf{w}}_{q:q+n}^I). \tag{15}$$

$$\begin{cases} L_{x,k} = L_{x,k-1} + \frac{s \cdot v_{x,k}}{n-s}, \\ L_{y,k} = L_{y,k-1} + \frac{s \cdot v_{y,k}}{n-s}. \end{cases} \tag{16}$$

$$\mathcal{L}_y = \mathbb{E}[(v_{x,g,k} - v_{x,k})^2] + \kappa \mathbb{E}[(v_{y,g,k} - v_{y,k})^2]. \tag{17}$$

Reducing Dependence on Gravitational Anchor: We supply y with magnetometer data $\hat{\mathbf{m}}_{q:q+n}^I$ to provide the geomagnetic North as an additional latent attitude anchor $\mathbf{N}_0^I$ beyond $\mathbf{g}_0^I$.

$$(v_{x,k}, v_{y,k}) = y_{\theta^*}(\mathbf{v}^I(0), \mathbf{g}_0^I, \mathbf{N}_0^I, \hat{\mathbf{a}}_{q:q+n}^I, \hat{\mathbf{w}}_{q:q+n}^I, \hat{\mathbf{m}}_{q:q+n}^I) \tag{18}$$

$\mathbf{N}_0^I$ is robust to gravity pollution, gyroscope drift, sensor placement offset, and rotational artifacts, implicitly correcting and constraining $\mathbf{g}_0^I$ and $\mathbf{v}^I(0)$.

Physics-Aware Velocity Estimation: We follow the *neuro* $\cup$ *compile [symbolic]* paradigm from neurosymbolic ML to ingrain physics or constraints in y via a *physics metadata channel* $c_k(\cdot)$. Specifically, we supply y with a local-variance step detector binary mask [47] or mean Fourier transform coefficients [28] of accelerometer vector sum, signifying ZUPT or transportation modes. y uses this information to provide velocity updates only when significant translational movements have occurred.

$$(v_{x,k}, v_{y,k}) = y_{\theta^*}\left(\mathbf{v}^I(0), \mathbf{g}_0^I, \mathbf{N}_0^I, \hat{\mathbf{a}}_{q:q+n}^I, \hat{\mathbf{w}}_{q:q+n}^I, \hat{\mathbf{m}}_{q:q+n}^I, c_k({}^I\hat{\mathbf{a}})\right) \tag{19}$$

$$c_k^{\text{legged}}({}^I\hat{\mathbf{a}}) = \begin{cases} 1, \hat{\mathbf{a}}_{L,\Delta t}^I > \zeta \cdot \sqrt{\frac{\sum_{j \in \Delta t}\left(\hat{\mathbf{a}}_{L,j}^I - \overline{\hat{\mathbf{a}}_{L,\Delta t}^I}\right)^2}{n}} \\ 0, \text{otherwise} \end{cases} \tag{20}$$

$$c_k^{\text{non-legged}}({}^I\hat{\mathbf{a}}) = \left|\overline{|\text{FFT}(|\hat{\mathbf{a}}_{q:q+n}^I|)|}\right| \tag{21}$$

$\hat{\mathbf{a}}_{L,\Delta t}^I = G_{5,f_c}(|\hat{\mathbf{a}}_{\Delta t}^I|) - G_{5,f_c}(\overline{|\hat{\mathbf{a}}_{\Delta t}^I|})$, $\Delta t = q : q+n$, ζ is a tunable parameter and $G_{5,f_c}(\cdot)$ is a 5th order low-pass filter with cutoff f_c. $c(\cdot)$ supplies f with *latent vald motion metadata* derived from Newtonian kinematics.

B. Platform-Aware Neural Architecture Search

TinyOdom uses automated platform-in-the-loop NAS on a lightweight model backbone search space $\mathbf{\Omega}$ to generate y that are performant, yet fit within the memory-compute bounds of the target IoT platform.

NAS Program Formulation: The search is modeled as a parallelizable black-box Bayesian optimization problem to minimize the latency and validation loss within the SRAM and flash capacity of the microcontroller [1], [75]:

$$f_{\text{opt}} = \lambda_1 f_{\text{error}}(\mathbf{\Omega}) + \lambda_2 f_{\text{flash}}(\mathbf{\Omega}) + \lambda_3 f_{\text{SRAM}}(\mathbf{\Omega}) + \lambda_4 f_{\text{latency}}(\mathbf{\Omega}) \tag{22}$$

where,

$$f_{\text{error}}(\mathbf{\Omega}) = \mathcal{L}_{\text{validation}}(\mathbf{\Omega}) \tag{23}$$

$$f_{\text{flash}}(\mathbf{\Omega}) = \begin{cases} \gamma_f \Leftrightarrow \left(|\gamma_f| < 1 \wedge \underbrace{\epsilon_{\text{flag}} = 0}_{\text{fault flag}}\right), \ \gamma_f = -\frac{\text{Compiler-reported flash}}{\text{flash}_{\max}} \\ \alpha_f, \ \ \alpha_f \gg \text{flash}_{\max} \end{cases} \tag{24}$$

$$f_{\text{SRAM}}(\mathbf{\Omega}) = \begin{cases} \gamma_s \Leftrightarrow \left(|\gamma_s| < 1 \wedge \underbrace{\epsilon_{\text{flag}} = 0}_{\text{fault flag}}\right), \ \gamma_s = -\frac{\text{Compiler-reported SRAM}}{\text{SRAM}_{\max}} \\ \alpha_s, \alpha_s \gg \text{SRAM}_{\max} \end{cases} \tag{25}$$

$$f_{\text{latency}}(\mathbf{\Omega}) = \begin{cases} \frac{\text{RTOS-reported latency}}{\text{latency}_{\text{target}}} \Leftrightarrow \underbrace{\epsilon_{\text{flag}} = 0}_{\text{fault flag}} \\ \alpha_l, \alpha_l \gg \text{latency}_{\text{target}} \end{cases} \tag{26}$$

f_{opt} seeks a Pareto-optimal configuration of parameters Ω^* under competing objectives [187]:

$$f_k(\Omega^*) <= f_k(\Omega) \ \ \forall k, \Omega \ \ \wedge \exists j : f_j(\Omega^*) < f_j(\Omega) \ \ \forall \Omega \neq \Omega^* \tag{27}$$

The NAS program constructs candidate models from Ω, which is composed of NN weights, hyperparameters, ML operations, and connections denoted as a directed acyclic graph. The program converts the TensorFlow model [188] to the native coding scheme of the IoT platform using TensorFlow Lite Micro (TFLM) [182]. The flatbuffer serialized model schema, along with the TFLM and real-time operating system (RTOS) file system is then flashed onto the target microcontroller. If the model fits and there are no runtime faults, unsupported operators, or compilation errors (denoted by the fault flag), the candidate model is trained to obtain $f_{\text{error}}(\cdot)$. $f_{\text{SRAM}}(\cdot)$ and $f_{\text{flash}}(\cdot)$ are designed to perform *full device capability exploitation* via *hard thresholding*, such that the SRAM and flash consumption is maximized within the memory bounds to improve $f_{\text{error}}(\cdot)$ with larger models. If the model does not fit or induces faults (denoted by ϵ_{flag}), then the hardware metrics are set to a value α much larger than the device capacity or target latency to penalize the NAS program. If the target hardware is unavailable, the NAS program uses the size of the flatbuffer model schema as a proxy for γ_f [182], the standard RAM usage model (intermediate layer-wise activation maps and tensors are stored in the SRAM) as a proxy for γ_s [187], and FLOPS as a proxy for latency [185]. λ allows the user to specify the weight of each element in f_{opt}.

Search Algorithm: The NAS program uses Gaussian Process $\mathcal{GP}$ as the surrogate model to approximate f_{opt} while using Monte Carlo sampling with Upper-Confidence Bounds as the acquisition function to sample the next set of parameters from $\mathbf{\Omega}$ during the search process [189], [190]:

$$\hat{f}(\mathbf{\Omega}) \sim \mathcal{GP}(\mu(\mathbf{\Omega}), k(\mathbf{\Omega}, \mathbf{\Omega}')) \tag{28}$$

$$\mathbf{\Omega}_t = \arg\max_{\mathbf{\Omega}}(\mu_{t-1}(\mathbf{\Omega}) + \beta^{0.5}\sigma_{t-1}(\mathbf{\Omega})) \tag{29}$$

TABLE I

(LEFT) LOCALIZATION PERFORMANCE AND RESOURCE USAGE OF TINYODOM VERSUS COMPETING BASELINES ON PREVIOUSLY UNSEEN TRAJECTORIES. (RIGHT) ABLATION STUDY OF ROBUST NEURAL-INERTIAL SEQUENCE LEARNING FORMULATION.

Dataset	Method	SRAM (kB)	Flash (kB)	ATE (m)	RTE (m)	Difference
OxIOD [5]	PDR [138]	10.8	49.6	3.47	3.24	ATE 1.2× ↑, RTE 2.6× ↑
	NDI [34]	1.2	28.1	9120	248	ATE 3260× ↑, RTE 200× ↑
	IONET [58]	766	670	5.95	2.84	ATE 2.1× ↑, Size 9.4× ↑
	L-IONet [5]	154	183	4.37	2.82	ATE 1.6× ↑, Size 2.6× ↑
	RoNIN TCN [7]	2046	2196	1.95	0.42	ATE 1.4× ↓, Size 31× ↑
	TinyOdom	52.4-90.1	71.0-117	2.80-6.82	1.24-1.37	
RoNIN [5]	PDR [138]	10.8	49.6	34.8	23.6	ATE 1.5× ↑, RTE 4.0× ↑
	NDI [34]	1.2	28.1	12398	59.9	ATE 519× ↑, RTE 10× ↑
	IONET [58]	976	782	22.5	7.63	RTE 1.3× ↑, Size 15× ↑
	L-IONet [5]	159	182	24.7	14.8	RTE 2.5× ↑, Size 3.6× ↑
	RoNIN TCN [7]	2046	2196	4.73	1.21	ATE 5.0× ↓, Size 43× ↑
	TinyOdom	36.2-257	50.8-254	23.9-28.3	5.84-7.76	
AQUALOC [196]	NavNet [20]	1364	1397	3.80	2.98	ATE 1.1× ↑, Size 68× ↑
	TinyOdom	17.3-36.7	20.5-34.2	3.32-4.99	2.45-3.30	
EuRoC MAV [195]	AbolDeepIO [19]	4218	4218	11.2	14.0	ATE 5.1× ↑, Size 134× ↑
	VeTorch [61]	7325	7294	13.5	15.2	ATE 6.1× ↑, Size 232× ↑
	TinyOdom	43.7-90	31.4-110	2.19-2.90	2.02-2.55	
GunDog [197]	GunDog [34], [197]	8.5	32.5	28.5	10.6	ATE 2.5× ↑, RTE 71× ↑
	TinyOdom	32.4-84.3	55.9-96.0	11.4-60.8	0.15-0.35	
AgroBot [11]	IONET [58]	N/A	600	10.1	0.57	ATE 1.1× ↑, Size 1.7× ↑
	L-IONet [5]		183	18.6	1.40	ATE 2.0× ↑, Size 1.9× ↓
	AbolDeepIO [19]		4160	20.6	0.93	ATE 2.3× ↑, Size 12× ↑
	VeTorch [61]		∼9000	15.6	0.84	ATE 1.7× ↑, Size 25× ↑
	TinyOdom		356	9.12	1.55	

Dataset	Formulation			ATE (m)
	Velocity	Physics	Magnetometer	
OxIOD [5]*	Yes	No	No	9.95
	No	Yes	No	5.36
	No	No	Yes	4.15
	Yes	Yes	No	7.51
	Yes	No	Yes	4.02
	No	Yes	Yes	5.13
	Yes	Yes	Yes	3.30
AQUALOC [196]*	Yes	No	No	3.93
	No	Yes	No	6.41
	No	No	Yes	5.46
	Yes	Yes	No	3.71
	Yes	No	Yes	3.63
	No	Yes	Yes	7.95
	Yes	Yes	Yes	3.28
AgroBot [11]@	No	No	No	16.6
	Yes	No	No	21.0
	No	Yes	No	16.3
	No	No	Yes	16.3
	Yes	Yes	No	29.8
	Yes	No	Yes	9.13
	No	Yes	Yes	16.4
	Yes	Yes	Yes	7.85

* For selected hardware, @ for phase 1 of the dataset.

$\mathcal{GP}$ provides tractable and non-parameterized assessment of prediction uncertainty incorporating the effect of data scarcity [191]. Monte-Carlo sampling optimizes the non-gradient-friendly f_{opt} on $\mathbf{\Omega}$ containing categorical, discrete, continuous, and conditional elements. The acquisition function also balances exploration and exploitation via clustering search to ensure Pareto-optimal convergence, while the gradient-free formulation accelerates the search process without evaluating at invalid configurations in $\mathbf{\Omega}$ [189], [190], [192].

Lightweight Model Backbones: $\mathbf{\Omega}$ is composed of a TCN backbone. TCN use dilated causal convolutions with gated residual blocks. Compared to vanilla CNN or LSTM, TCN can discover high-resolution, long-range, and non-linear spatial and temporal context in long inertial windows without overfitting, high parameter cost, or high memory usage [193], [194]. The search parameters include the filter count, layer count, layer-wise dilation factors, dropout rate, use of skip connections, and normalization category.

C. Evaluation

We evaluate *TinyOdom* on the OxIOD [5] and RoNIN [7] datasets for PDR, EuRoC MAV [195] dataset for aerial vehicle odometry, AQUALOC [196] dataset for underwater vehicle navigation, GunDog dataset [197] for wildlife tracking, and the AgroBot dataset [11] for precision agricultural robot localization. We run the NAS program to generate models for four different STM32 ARM Cortex-M microcontrollers with 128-320 kB SRAM and 0.5-1 MB flash. We use the absolute trajectory error (ATE) and relative trajectory error (RTE) to quantify localization performance. ATE is the mean root-mean-squared-error between the predicted and ground truth locations for the whole trajectory, while RTE is the ATE for 1 minute [7], [63]. For each dataset, the NAS program was run for 50 epochs, while each candidate model was trained for 300-900 epochs. The dataset splits, dataset features, window size, stride, NAS search space, baselines, and training infrastructure are delineated in [1] and [11].

Table I (Left) shows the odometric resolution and resource usage of *TinyOdom* and competing baselines across the six datasets spanning five applications. The models generated by *TinyOdom* are 31-134× smaller (flash usage 27-72 kB) than competing neural-inertial navigation models, yet provide 1.15× higher resolution on average. *TinyOdom* provides a location estimate within 2.5-12m for trajectories of length 12-1160 m or spanning 60 seconds. Classical inertial navigation techniques such as SHS-PDR [51], NDI [34] and GunDog [197] are 1.5× smaller than *TinyOdom*, but have 750× lower resolution. Baselines using the heading-displacement formulation struggle in presence of sharp turns, rotational artifacts, and sensor placement offset. RoNIN [7] outperforms *TinyOdom* in terms of resolution, but at the cost of 34× more weights, availability of device attitude, and 50% more training data. The hardware-aware auto TinyML formulation of *TinyOdom* maintains superior odometric resolution both in the long and short term for various microcontrollers across heterogeneous applications.

Table I (Right) illustrates an ablation study showing the importance of different components in the robust neural-inertial sequence learning formulation. Our formulation lowers the ATE by 1.1-3.1×. The addition of magnetometer data and making y regress $(v_{x,k}, v_{y,k})$ instead of $(\Delta l_k, \Delta\psi_k)$ reduces the ATE the most. The velocity-centric formulation mitigates the heading rate singularity issue. The magnetometer data counteracts the effects of gravity pollution, IMU drift, rotational artifacts, and sensor placement offset. The physics channel implicitly forces the network to output zero velocities when sufficient translational motion is absent. All three components contribute to obtain the highest odometric resolution.

Fig. 4 (Left) and (Center) show predicted trajectories by *TinyOdom* and competing baselines against ground truth trajectory on unseen data. Classical methods have large error bounds, while the heading-displacement formulation baselines struggle during sharp turns and rough patches. Competing baselines over smooth the trajectories, drift rapidly and suffer

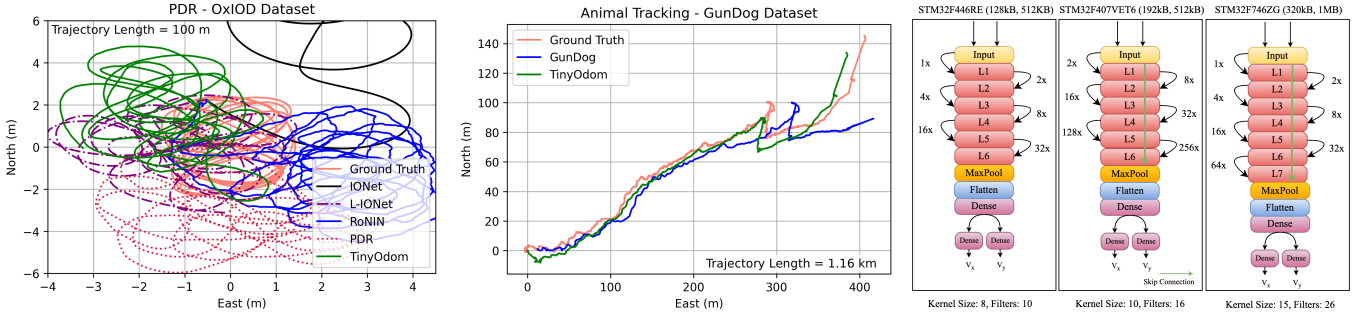

Fig. 4. (Left) and (Center) Selected previously unseen trajectory reconstruction by *TinyOdom* and competing baselines on the OxIOD and GunDog datasets. (Right) Architectural adaptation and device capability exploitation by *TinyOdom* on the RoNIN dataset for various microcontrollers. The SRAM and flash capacity of each microcontroller are given in parentheses.

TABLE II
GPS FUSION PERFORMANCE AND FLASH USAGE OF NEURAL-EKF VERSUS CONVENTIONAL INS.

Method	Flash	ATE (m)			RTE (m)			Difference
		P1	P2	P3	P1	P2	P3	
UKF INS+GPS [198]	192	4.06	4.35	8.09	0.18	0.21	1.07	ATE 3.1× ↑
EKF INS+GPS [199]	77	2.22	2.24	5.28	0.35	0.35	1.05	ATE 1.8× ↑
GPS only	None	1.90	1.88	1.89	0.40	0.40	0.45	ATE 1.2× ↑
Neural-EKF	376	1.07	2.20	2.17	0.30	1.16	0.45	

P refers to the dataset phase, flash size is in kB.

from rotational artifacts, noise, and varying topography.

Fig. 4 (Right) shows how *TinyOdom* adapts the NN for the same dataset for different microcontrollers depending on SRAM and flash availability. When more resources are available, *TinyOdom* automatically increases the parameter count, the number of layers, filter count, and the kernel size of the network to lower the ATE and RTE. *TinyOdom* also mitigates the lack of parameters for models running on ultra-low-end microcontrollers by assigning small dilation factors to lower layers and large dilation factors to higher layers, allowing tiny networks to model both local and global contexts. To prevent overfitting and exploding-and-vanishing gradients, *TinyOdom* adds skip connections to larger networks.

IV. NEURAL-KALMAN FILTERING

TinyOdom automatically learns complex inertial dynamics from the training data without depending on human-engineered approximations used in INS or SHS. Thus, NN-based state evolution models are preferred over classical INS/SHS formulation used in AI-enhanced navigation systems that model covariance and noise, estimate velocity profiles, or use NN as state advisors. However, *TinyOdom* is black-box and non-interpretable to the user. The *neuro* $\cup$ *compile [symbolic]* paradigm of injecting physics-based constraints does not guarantee that the constraints will be followed by the NN. Given a neural system model, it is unclear how to combine physics-based measurement updates in an interpretable, near-optimal, and strictly-enforced fashion. We develop the concept of neural-Kalman filtering using a modified *symbolic[neuro]* paradigm and use the filter to inject *TinyOdom* with intermittent GPS/GNSS updates.

A. Filter Formulation

The discrete-time EKF [200]–[202] propagate and update equations are:

$$\hat{\mathbf{x}}_{k+1|k} = f(\hat{\mathbf{x}}_k, \mathbf{u}_{k+1}, \mathbf{w}_{k+1}),$$
$$\mathbf{P}_{k+1|k} = \mathbf{F}_{k+1}\mathbf{P}_k\mathbf{F}_{k+1}^T + \mathbf{G}_{k+1}\mathbf{Q}_k\mathbf{G}_{k+1}^T,$$
$$\mathbf{F}_{k+1} = \left.\frac{\partial f}{\partial x}\right|_{\hat{\mathbf{x}}_k, \mathbf{u}_{k+1}, \mathbf{w}_{k+1}}, \quad \mathbf{G}_{k+1} = \left.\frac{\partial f}{\partial w}\right|_{\hat{\mathbf{x}}_k, \mathbf{u}_{k+1}, \mathbf{w}_{k+1}}, \quad (30)$$

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k}\mathbf{H}_{k+1}^T \left(\underbrace{\mathbf{H}_{k+1}\mathbf{P}_{k+1|k}\mathbf{H}_{k+1}^T + \mathbf{R}_{k+1}}_{\text{innovation covariance}} \right)^{-1},$$
$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1} \left(\underbrace{\mathbf{z}_{k+1} - h(\hat{\mathbf{x}}_{k+1|k}, \mathbf{v}_k)}_{\text{measurement residual}} \right),$$
$$\mathbf{P}_{k+1|k+1} = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})\,\mathbf{P}_{k+1|k}, \quad \mathbf{H}_{k+1} = \left.\frac{\partial h}{\partial x}\right|_{\hat{\mathbf{x}}_{k+1|k}} \quad (31)$$

During state propagation, $\hat{\mathbf{x}}$ is the predicted state, which is a non-linear mapping f of the past state, control input $\mathbf{u}$, and AWGN $\mathbf{w}$ with covariance $\mathbf{Q}$. $\hat{\mathbf{P}}$ is the predicted process covariance and given by the Lyapunov equation [203]. The diagonal elements of P encode the variances for the state estimation errors for each element in $\hat{\mathbf{x}}$, while the off-diagonal elements encode correlations. Minimizing the trace of $\mathbf{P}$ minimizes the mean squared error of the state estimates [204]. The equation contains Jacobians of f with respect to $\hat{\mathbf{x}}$ and $\mathbf{w}$ at k, which linearizes the non-linear f about $\hat{\mathbf{x}}_k$. $\mathbf{K}$ is the Kalman gain, which is computed during the measurement update by linearizing the observation model h and inverting the innovation covariance matrix. $\mathbf{z}$ represents measurements mixed with AWGN $\mathbf{v}$ (noise covariance $\mathbf{R}$). $\mathbf{K}$ is multiplied with the measurement residual to update $\hat{\mathbf{x}}$. $\hat{\mathbf{P}}$ is then updated using algebraic Riccati recursion [203].

Now, consider a dynamical system such that $T_{\mathbf{A}} : \hat{\mathbf{x}}_{k+1|k} \rightarrow \hat{\mathbf{x}}_k \mid T_{\mathbf{A}}$ is linear, and $g : \hat{\mathbf{x}}_{k+1|k} \rightarrow \mathbf{u}_{k+1} \mid g$ is non-linear.

If $T_{\mathbf{A}}$ and g are linearly separable, then the EKF propagation equation is [11]:

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{A}\hat{\mathbf{x}}_k + g(\mathbf{u}_{k+1}),$$
$$\mathbf{P}_{k+1|k} = \mathbf{A}\mathbf{P}_k\mathbf{A}^T + \mathbf{B}_{k+1}\mathbf{U}_k\mathbf{B}_{k+1}^T, \quad \mathbf{B}_{k+1} = \frac{\partial g}{\partial u}\bigg|_{\hat{\mathbf{x}}_k, \mathbf{u}_{k+1}} \tag{32}$$

If $\hat{\mathbf{x}}$ contains loose nonholonomic location and velocity estimates, and g represents the *TinyOdom*-integrator combination:

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{L}_x \\ \hat{L}_y \\ v_x \\ v_y \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} \mathbf{a}^I_{q:q+n} \\ \mathbf{w}^I_{q:q+n} \\ \mathbf{m}^I_{q:q+n} \\ c(\mathbf{a}^I_{q:q+n}) \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} \mathbf{I}_{2\times 2} & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 2} \end{bmatrix}, \tag{33}$$

$$g(\cdot) = \begin{bmatrix} \Delta t \cdot \mathbf{I}_{2\times 2} \\ \mathbf{I}_{2\times 2} \end{bmatrix} \cdot y_{\theta^*}(\cdot), \quad \Delta t = \frac{s}{n-s} \tag{34}$$

$$\mathbf{B}_{k+1} = \begin{bmatrix} \frac{\Delta t \partial y_\theta(\cdot)_x}{\partial \mathbf{a}^I_{q:q+n}} & \frac{\Delta t \partial y_\theta(\cdot)_x}{\partial \mathbf{w}^I_{q:q+n}} & \frac{\Delta t \partial y_\theta(\cdot)_x}{\partial \mathbf{m}^I_{q:q+n}} & \frac{\Delta t \partial y_\theta(\cdot)_x}{\partial c(\mathbf{a}^I_{q:q+n})} \\ \frac{\Delta t \partial y_\theta(\cdot)_y}{\partial \mathbf{a}^I_{q:q+n}} & \frac{\Delta t \partial y_\theta(\cdot)_y}{\partial \mathbf{w}^I_{q:q+n}} & \frac{\Delta t \partial y_\theta(\cdot)_y}{\partial \mathbf{m}^I_{q:q+n}} & \frac{\Delta t \partial y_\theta(\cdot)_y}{\partial c(\mathbf{a}^I_{q:q+n})} \\ \frac{\partial y_\theta(\cdot)_x}{\partial \mathbf{a}^I_{q:q+n}} & \frac{\partial y_\theta(\cdot)_x}{\partial \mathbf{w}^I_{q:q+n}} & \frac{\partial y_\theta(\cdot)_x}{\partial \mathbf{m}^I_{q:q+n}} & \frac{\partial y_\theta(\cdot)_x}{\partial c(\mathbf{a}^I_{q:q+n})} \\ \frac{\partial y_\theta(\cdot)_y}{\partial \mathbf{a}^I_{q:q+n}} & \frac{\partial y_\theta(\cdot)_y}{\partial \mathbf{w}^I_{q:q+n}} & \frac{\partial y_\theta(\cdot)_y}{\partial \mathbf{m}^I_{q:q+n}} & \frac{\partial y_\theta(\cdot)_y}{\partial c(\mathbf{a}^I_{q:q+n})} \end{bmatrix} \tag{35}$$

$y_{\theta^*}(\cdot)$ is the NN from *TinyOdom*, providing a non-linear black-box mapping between $\mathbf{u}$ and $\hat{\mathbf{x}}$. According to EKF primitives, $\mathbf{B}_{k+1}$ contains Jacobians of g with respect to $\mathbf{u}$ around the current inertial window. $\mathbf{U}$ contains the Allan variance parameters [205] of the IMU, which include the accelerometer noise variance ($\sigma^2(n_{\mathbf{a}^I})$), variance in gyroscope ARW ($\sigma^2(\mathbf{w}^I_{\text{ARW}}\sqrt{\Delta t})$) and BI ($\sigma^2(\mathbf{w}^I_{\text{BI}})$), gyroscope noise variance ($\sigma^2(n_{\mathbf{w}^I})$), and the magnetometer noise variance $\sigma^2(n_{\mathbf{m}^I})$.

$$\mathbf{U} = \text{diag}\big(\sigma^2(n_{\mathbf{a}^I}), \sigma^2(\mathbf{w}^I_{\text{ARW}}\sqrt{\Delta t}) + \sigma^2(\mathbf{w}^I_{\text{BI}}) + \sigma^2(n_{\mathbf{w}^I}), \sigma^2(n_{\mathbf{m}^I}), \sum \sigma^2(n_{\mathbf{a}^I})\big) \tag{36}$$

Given the EKF system model representation of *TinyOdom*, the physics-based models can be combined with *TinyOdom* in the form of measurement updates. $\mathbf{R}$ is the covariance associated with velocity and position estimation error from these models. The EKF is non-agnostic to both $\mathbf{R}$ and $\mathbf{U}$, optimally balancing complementary properties of the NN and the measurement models (e.g., smoothness of neural-inertial navigation with long-term precision of GPS/GNSS updates). In addition, if the position estimation from *TinyOdom* drifts too far, $\hat{\mathbf{x}}$ and $\mathbf{P}$ can be reset using the measurement updates.

B. Evaluation

We evaluate the neural-EKF for locating a precision-agricultural robot using the AgroBot dataset [11], introduced in Section III-C. In our evaluation, $\mathbf{z}$ is the GPS/GNSS position and velocity update. h is the inverse mapping from longitude-latitude to 2D Cartesian coordinates and velocities using the WGS-84 ellipsoid geodetic model [206]. We also evaluate neural-EKF for tracking underwater animals with intermittent GPS updates [75].

TABLE III
RTE (M) OF NEURAL-INERTIAL MODELS ACROSS DIFFERENT DATASETS (LEFT) AND APPLICATIONS (RIGHT) WITHOUT FINE TUNING. THE TRAINING DATASET OR APPLICATION IS SHOWN IN PARENTHESIS.

Method	OxIOD	RoNIN
IONet (OxIOD)	2.84	4.7
IONet (RoNIN)	7.65	7.63
LIONet (OxIOD)	2.82	4.7
LIONet (RoNIN)	8.35	14.84
RoNIN TCN (OxIOD)	0.42	13.4
RoNIN TCN (RoNIN)	10.3	1.21
TinyOdom (OxIOD)	1.26	97.2
TinyOdom (RoNIN)	3.16	6.74

Method	PDR	UUV
IONet (PDR)	2.84	5.15
LIONet (PDR)	2.82	3.94
RoNIN TCN (PDR)	0.42	14.96
TinyOdom (PDR)	1.26	7.83
NavNet (UUV)	93	2.96
TinyOdom (UUV)	5.82	2.45

Table II summarizes the odometric resolution of neural-EKF and conventional INS on the AgroBot dataset on previously unseen trajectories, while Fig. 5 (Left) shows sample trajectory plots. Neural-EKF provides 1.2-3.1× lower ATE over traditional INS-GPS fusion algorithms. Traditional INS puts more trust in the noisy GPS updates than the dead-reckoning algorithm, leading to noisy trajectories. Neural-EKF balances the short-term smoothness of neural-inertial navigation with the long-term GNSS precision, leading to a smooth trajectory that closely follows the ground truth.

Fig 5 (Center) and (Right) show the error evolution of neural-EKF and traditional INS with increasing GPS outage intervals on the AgroBot dataset and for locating marine animals. For the first case, neural-EKF constrains the ATE to 2.75 m even with 20 minutes of GPS outage. For the second case, the ATE is 14.4 m (0.4% of the entire trajectory of length 4 km) with 1 minute GPS update rate and 57.1 m (1.4% of the entire trajectory) with 6 minute GPS update rate. Without GPS updates, traditional INS drifts due to NDI cubic error accumulation. Pure GPS updates, on the other hand, are noisy and unusable. For underwater tracking, neural-EKF combines the robustness of *TinyOdom* against wave turbulence, varying depth, ship vibrations, and rotational artifacts with GPS precision, while for agricultural robot tracking, neural-EKF exploits the robustness of *TinyOdom* against motor vibrations, wheel slippage, and bumpy topography.

V. FINE-TUNING PRE-TRAINED MODELS

Neural-inertial navigation techniques need large amounts of high-resolution training data in the target domain for providing acceptable odometric resolution [1], [11], [75]. Table III shows an example of resolution degradation of neural-inertial models across domains. In general, these models perform well within the learned data distribution but fail to adapt to different motion artifacts and IMU characteristics in a new domain due to differences in learned physical embeddings across domains. Moreover, under parameterized systems such as *TinyOdom* overfit on the dataset-specific spatiotemporal features due to a lack of redundant weights, poorly generalizing across domains.

A. Data-Efficient Transfer-Learning

Instead of training new models from scratch using large amounts of training data in the new domain, we propose using transfer learning [207] to fine-tune pre-trained models using

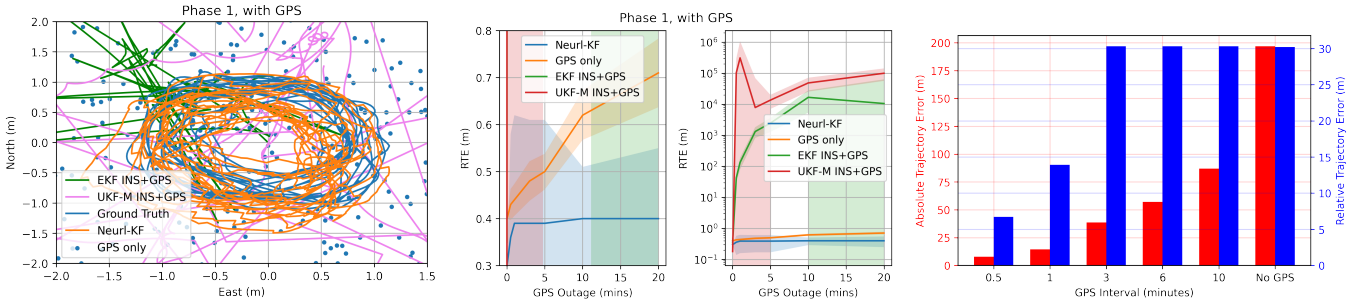

Fig. 5. (Left) Trajectory reconstruction of Neural-EKF (1 Hz GPS, 100 m trajectory) and competing baselines on the AgroBot dataset (phase 1). (Center) Evolution of RTE (m) with different GPS outage intervals Neural-EKF and competing baselines on the AgroBot dataset (phase 1). (Right) Evolution of ATE (m) and RTE (m) with different GPS outage intervals Neural-EKF for tracking marine animals (4 km trajectory)

TABLE IV
FINE-TUNING PRE-TRAINED MODELS ACROSS DIFFERENT PHASES OF THE AGROBOT DATASET

Training Dataset	RTE (m) on Inference Dataset (Unseen Trajectory)					
	P1	P1 (FT)	P2	P2 (FT)	P3	P3 (FT)
P1	1.10		14.5	1.45	15.0	4.96
P2	2.71	1.09	0.97		4.25	2.63
P3	1.85	0.76	1.93	1.15	2.58	

Method	RTE (m) with T minutes of data in the new domain*		
	T = 1	T = 5	T = 20
Train from scratch	26.8	3.29	2.55
Fine-tune*	1.92	1.62	1.45

P1: Phase 1, P2: Phase 2, P3: Phase 3
FT: Fine-tuning with 20 minutes of data in the new domain for 100 epochs
* The pre-trained model was trained on Phase 1 data; target dataset: Phase 2

TABLE V
FINE-TUNING PRE-TRAINED OXIOD *TinyOdom* MODEL ON THE AGROBOT DATASET

Method	RTE (m) with T minutes of data in the new domain*							
	T= 1	T= 3	T= 5	T= 10	T= 20	T= 40	T= 75	T= 100
Train from scratch	3.26	1.39	1.09	0.94	0.86	0.83	0.81	0.80
Fine-tune	1.09	0.85	0.81	0.80	0.80	0.80	0.80	0.80

as little as 1 minute of labeled data in the target domain. We freeze some of the lower layers of the pre-trained NN and make the higher layers trainable.

Table IV and Table V showcase the data efficiency and resolution improvement brought on by transfer learning. For the first case, fine-tuning reduces RTE by 1.6 - 13.6× in the target domain, while increasing data efficiency by >20×. In other words, 1 minute of fine-tuning exceeds training from scratch on 20 minutes of labeled data by 1.3×. For the second case, 1 minute of fine-tuning reduces the RTE by 8× for a pre-trained model with no fine-tuning, while 5 minutes of fine-tuning equals training from scratch on 100 minutes of labeled data. Pre-trained models already have some notion of inertial dynamics in a different domain, which models trained from scratch must learn, resulting in fine-tuning being data-efficient.

B. Collecting Labeled Data in New Domain

Fine-tuning still requires some high-resolution labeled inertial odometry data. Specialized motion capture systems suffer from limited coverage, high cost, use of specialized software, high computational requirements, and ambient lighting conditions [208]. Vanilla GPS is noisy, with a maximum resolution of around 2 m [11], [75] (unless differential GPS is used, which can achieve centimeter-level accuracy at the cost of limited coverage, complexity, and time delay [209]). We develop an automated data extraction framework operating on overhead quadrotor video feeds [11] that mitigates the aforementioned limitations. Fig. 6 illustrates the automated inertial odometry data extraction pipeline. The user places several printed checkerboard patterns as reference landmarks at the boundaries of the quadrotor camera's field of view (FOV). The horizontal distance, h between landmark i and j, and the vertical distance, v between landmarks k and l are

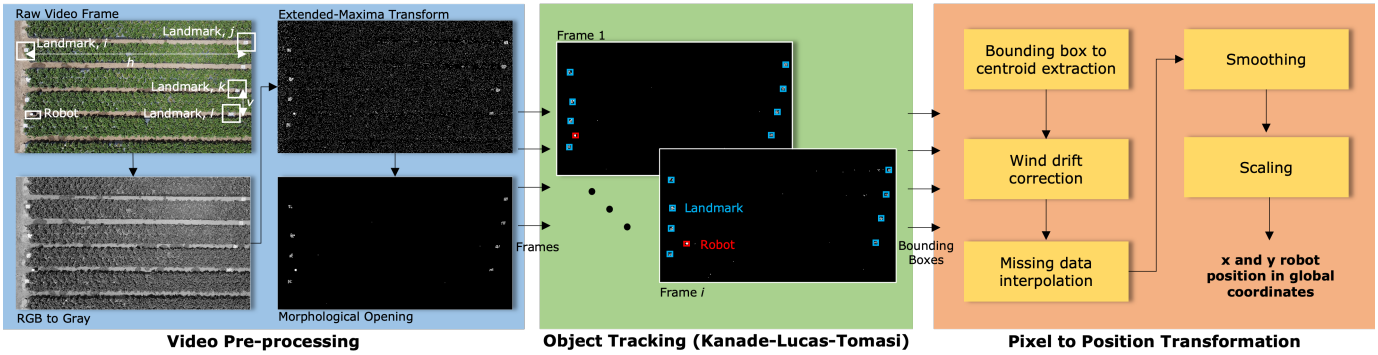

Fig. 6. Automated pipeline to extract labeled inertial odometry data from monocular quadrotor video feeds.

measured and noted. The object to be tracked is then moved within the FOV of the quadrotor camera. The IMU data is logged onboard the object, while the quadrotor camera records the object moving. The camera frames are synchronized with the IMU data using "static-rotate-static" motion patterns. The pipeline has three steps:

Video Pre-Processing: After converting the RGB video frames to grayscale, extended maxima transform [210] and morphological opening [211] are applied to leave only the landmark and the object to be tracked in the frame.

Object Tracking: The user marks the bounding boxes for the object and the landmarks in the first frame. The Kanade-Lucas-Tomasi tracker [212], coupled with a minimum eigenvalue feature extractor [213], tracks the bounding boxes across subsequent frames.

Pixel to Position Transformation: The centroids of the bounding boxes are derived from the corner points, which are corrected for quadrotor wind drift by observing the movement of static landmarks. The user provides warm starts when the tracker loses the object or the landmarks due to light intensity changes. Linear interpolation fills the gaps between warm-starts and the last known object location. Median filtering [214] cancels high-frequency tracking noise. Finally, the derived pixel positions are scaled with the scale factor s_x, s_y to convert to global coordinates as follows:

$$s_x = \frac{|C^i_{x,1} - C^j_{x,1}|}{h}, \quad s_y = \frac{|C^k_{y,1} - C^l_{y,1}|}{v}, \quad s_x \approx s_y. \quad (37)$$

$C^c_{a,b}$ is the centroid of landmark c at frame b for axis a. The pipeline, executable on commodity computers, provides ground truth locations at a resolution of ± 5.0 cm.

VI. Challenges and Conclusion

This paper provides a quantitative, methodological, and qualitative review of several techniques conceived by us to port AI-enhanced inertial navigation algorithms to IoT platforms for real-time adoption. We showcase how advances in TinyML and platform-aware NAS can yield models whose performance exceeds both classical and existing AI-enhanced INS, yet fit within the tight resource bounds of microcontrollers. Neurosymbolic AI paradigms yield models that are physics-aware, interpretable, and satisfy user constraints. Transfer learning drastically reduces the data inefficiency of NN-based navigation algorithms, allowing the usage of user-friendly video pipelines to collect inertial odometry training datasets and deployment of pre-trained models in a new domain. However, there are several open research challenges.

On-device Domain Adaptation: Collecting labeled data in the target domain for fine-tuning may not always be possible. The onboard navigation models need to be adaptively personalized using unlabeled data streams to account for domain shifts. Possible solutions include the use of onboard BERT-like unsupervised pre-trained IMU embeddings [215], [216] and on-device training [181], [217]–[219]. AI-enhanced navigation algorithms also need to be context-aware, modifying NN dynamics depending on the application, noise, and environmental conditions.

Injecting Physics-Aware Embeddings: The *neuro* $\cup$ *compile [symbolic]* paradigm does not guarantee strict enforcement of user constraints, which are lost within the NN embeddings. The *symbolic [neuro]* paradigm allows the fusion of neural and physics-based components through an EKF, but the NN is agnostic to the heuristic rules, physics, and bounds being managed by the EKF. The preferred paradigm is neuro [symbolic], where the NN architecture is embedded with special symbolic reasoning layers. The symbolic layer is biased to strictly enforce user-defined models and mimic a logical reasoning module [174], [175]. Since the loss flows through the whole NN, the NN becomes aware of the intricacies of the symbolic layer. More work is required to implement such physics-aware reasoning layers [220]–[224] into onboard neural-inertial navigation models.

Uncertainty Awareness: IMU data in the wild suffer from missing data, cross-channel timestamp misalignment, and window jitter [225]–[227], which can degrade the NN odometric resolution. Possible solutions include using uncertainty-aware training frameworks [225], [227] and on-chip hardware enhancements [228].

References

[1] S. S. Saha, S. S. Sandha, L. A. Garcia, and M. Srivastava, "Tinyodom: Hardware-aware efficient neural inertial navigation," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 6, no. 2, pp. 1–32, 2022.

[2] J. Gui, D. Gu, S. Wang, and H. Hu, "A review of visual inertial odometry from filtering and optimisation perspectives," *Advanced Robotics*, vol. 29, no. 20, pp. 1289–1301, 2015.

[3] H. Badino, A. Yamamoto, and T. Kanade, "Visual odometry by multi-frame feature integration," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2013, pp. 222–229.

[4] Y. Wu, H.-B. Zhu, Q.-X. Du, and S.-M. Tang, "A survey of the research status of pedestrian dead reckoning systems based on inertial sensors," *International Journal of Automation and Computing*, vol. 16, no. 1, pp. 65–83, 2019.

[5] C. Chen, P. Zhao, C. X. Lu, W. Wang, A. Markham, and N. Trigoni, "Deep-learning-based pedestrian inertial navigation: Methods, data set, and on-device inference," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4431–4441, 2020.

[6] R. Harle, "A survey of indoor inertial positioning systems for pedestrians," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1281–1293, 2013.

[7] S. Herath, H. Yan, and Y. Furukawa, "Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3146–3152.

[8] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. Mccullough, and A. Mouzakitis, "A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 829–846, 2018.

[9] D. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology*. IET, 2004, vol. 17.

[10] F. Höflinger, J. Müller, R. Zhang, L. M. Reindl, and W. Burgard, "A wireless micro inertial measurement unit (imu)," *IEEE Transactions on instrumentation and measurement*, vol. 62, no. 9, 2013.

[11] Y. Du, S. S. Saha, S. S. Sandha, A. Lovekin, J. Wu, S. Siddharth, M. Chowdhary, M. K. Jawed, and M. Srivastava, "Neural-kalman gnss/ins navigation for precision agriculture," *International Conference on Robotics and Automation (ICRA)*, 2023.

[12] J. Das, G. Cross, C. Qu, A. Makineni, P. Tokekar, Y. Mulgaonkar, and V. Kumar, "Devices, systems, and methods for automated monitoring enabling precision agriculture," in *2015 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, 2015, pp. 462–469.

[13] Y. Zhang, F. Gao, and L. Tian, "Ins/gps integrated navigation for wheeled agricultural robot based on sigma-point kalman filter," in *2008 Asia Simulation Conference-7th International Conference on System Simulation and Scientific Computing*. IEEE, 2008, pp. 1425–1431.

[14] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct ekf-based approach," in *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2015, pp. 298–304.

[15] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to mav navigation," in *2013 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2013, pp. 3923–3929.

[16] O. Hegrenas, E. Berglund, and O. Hallingstad, "Model-aided inertial navigation for underwater vehicles," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 1069–1076.

[17] J. Li, M. Post, T. Wright, and R. Lee, "Design of attitude control systems for cubesat-class nanosatellite," *Journal of Control Science and Engineering*, vol. 2013, 2013.

[18] M. Brossard, A. Barrau, and S. Bonnabel, "Ai-imu dead-reckoning," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 4, 2020.

[19] M. A. Esfahani, H. Wang, K. Wu, and S. Yuan, "Aboldeepio: A novel deep inertial odometry network for autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1941–1950, 2019.

[20] X. Zhang, B. He, G. Li, X. Mu, Y. Zhou, and T. Mang, "Navnet: Auv navigation through deep sequential learning," *IEEE Access*, vol. 8, pp. 59 845–59 861, 2020.

[21] V. Chandrasekhar, W. K. Seah, Y. S. Choo, and H. V. Ee, "Localization in underwater sensor networks: survey and challenges," in *Proceedings of the 1st ACM international workshop on Underwater networks*, 2006, pp. 33–40.

[22] L. Paull, S. Saeedi, M. Seto, and H. Li, "Auv navigation and localization: A review," *IEEE Journal of oceanic engineering*, vol. 39, no. 1, pp. 131–149, 2013.

[23] M. Chowdhary, "Method and apparatus for inertial guidance for an automobile navigation system," Aug. 28 2001, uS Patent 6,282,496.

[24] S. S. Saha, S. S. Sandha, S. Pei, V. Jain, Z. Wang, Y. Li, A. Sarker, and M. Srivastava, "Auritus: An open-source optimization toolkit for training and development of human movement models and filters using earables," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 6, no. 2, pp. 1–34, 2022.

[25] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE communications surveys & tutorials*, vol. 15, no. 3, pp. 1192–1209, 2012.

[26] X. Hou and J. Bergmann, "Pedestrian dead reckoning with wearable sensors: A systematic review," *IEEE Sensors Journal*, vol. 21, no. 1, pp. 143–152, 2020.

[27] M. A. R. Ahad, A. D. Antar, and M. Ahmed, *IoT Sensor-based Activity Recognition*. Springer, 2020.

[28] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Using mobile phones to determine transportation modes," *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 2, pp. 1–27, 2010.

[29] M. O'Reilly, B. Caulfield, T. Ward, W. Johnston, and C. Doherty, "Wearable inertial sensor systems for lower limb exercise detection and evaluation: a systematic review," *Sports Medicine*, vol. 48, no. 5, pp. 1221–1246, 2018.

[30] C. M. Brigante, N. Abbate, A. Basile, A. C. Faulisi, and S. Sessa, "Towards miniaturization of a mems-based wearable motion capture system," *IEEE Transactions on industrial electronics*, vol. 58, no. 8, pp. 3234–3241, 2011.

[31] T. Oskiper, S. Samarasekera, and R. Kumar, "Multi-sensor navigation algorithm using monocular camera, imu and gps for large scale augmented reality," in *2012 IEEE international symposium on mixed and augmented reality (ISMAR)*. IEEE, 2012, pp. 71–80.

[32] J. D. Hol, T. Schon, F. Gustafsson, and P. J. Slycke, "Sensor fusion for augmented reality," in *2006 9th International Conference on Information Fusion*. IEEE, 2006, pp. 1–6.

[33] E. Foxlin, Y. Altshuler, L. Naimark, and M. Harrington, "Flighttracker: A novel optical/inertial tracker for cockpit enhanced vision," in *Third IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE, 2004, pp. 212–221.

[34] R. P. Wilson, N. Liebsch, I. M. Davies, F. Quintana, H. Weimerskirch, S. Storch, K. Lucke, U. Siebert, S. Zankl, G. Müller *et al.*, "All at sea with animal tracks; methodological and analytical solutions for the resolution of movement," *Deep Sea Research Part II: Topical Studies in Oceanography*, vol. 54, no. 3-4, pp. 193–210, 2007.

[35] R. P. Wilson, E. Shepard, and N. Liebsch, "Prying into the intimate details of animal lives: Use of a daily diary on animals," *Endangered Species Research*, vol. 4, no. 1-2, pp. 123–137, 2008.

[36] H. J. Williams, L. A. Taylor, S. Benhamou, A. I. Bijleveld, T. A. Clay, S. de Grissac, U. Demšar, H. M. English, N. Franconi, A. Gómez-Laich *et al.*, "Optimizing the use of biologgers for movement ecology research," *Journal of Animal Ecology*, vol. 89, no. 1, 2020.

[37] H. J. Williams, M. D. Holton, E. L. Shepard, N. Largey, B. Norman, P. G. Ryan, O. Duriez, M. Scantlebury, F. Quintana, E. A. Magowan *et al.*, "Identification of animal movement patterns using tri-axial magnetometry," *Movement ecology*, vol. 5, no. 1, pp. 1–14, 2017.

[38] M. Bloesch, M. Hutter, M. A. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart, "State estimation for legged robots-consistent fusion of leg kinematics and imu," *Robotics*, vol. 17, pp. 17–24, 2013.

[39] B. Steenwinckel, D. De Paepe, S. V. Hautte, P. Heyvaert, M. Bentefrit, P. Moens, A. Dimou, B. Van Den Bossche, F. De Turck, S. Van Hoecke *et al.*, "Flags: A methodology for adaptive anomaly detection and root cause analysis on sensor data streams by fusing expert knowledge with machine learning," *Future Generation Computer Systems*, vol. 116, pp. 30–48, 2021.

[40] F. Seraj, B. J. v. d. Zwaag, A. Dilo, T. Luarasi, and P. Havinga, "Roads: A road pavement monitoring system for anomaly detection using smart phones," in *Big data analytics in the social and ubiquitous context*. Springer, 2015, pp. 128–146.

[41] J.-A. Ting, E. Theodorou, and S. Schaal, "A kalman filter for robust outlier detection," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 1514–1519.

[42] M. Kok, J. Hol, and T. Schön, "Using inertial sensors for position and orientation estimation," *Foundations and Trends in Signal Processing*, vol. 11, pp. 1–153, 2017.

[43] I. P. Prikhodko, B. Bearss, C. Merritt, J. Bergeron, and C. Blackmer, "Towards self-navigating cars using mems imu: Challenges and opportunities," in *2018 IEEE International Symposium on Inertial Sensors and Systems*. IEEE, 2018, pp. 1–4.

[44] S. Shen, M. Gowda, and R. Roy Choudhury, "Closing the gaps in inertial motion tracking," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, 2018, pp. 429–444.

[45] H. Yan, Q. Shan, and Y. Furukawa, "Ridi: Robust imu double integration," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 621–636.

[46] J. Borenstein, L. Ojeda, and S. Kwanmuang, "Heuristic reduction of gyro drift for personnel tracking systems," *The Journal of navigation*, vol. 62, no. 1, pp. 41–58, 2009.

[47] I. Skog, P. Handel, J.-O. Nilsson, and J. Rantakokko, "Zero-velocity detection—an algorithm evaluation," *IEEE transactions on biomedical engineering*, vol. 57, no. 11, pp. 2657–2666, 2010.

[48] F. Zampella, M. Khider, P. Robertson, and A. Jiménez, "Unscented kalman filter and magnetic angular rate update (maru) for an improved pedestrian dead-reckoning," in *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*. IEEE, 2012.

[49] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: Unsupervised indoor localization," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*, 2012, pp. 197–210.

[50] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, and E. Aboutanios, "Recent advances in indoor localization: A survey on theoretical approaches and applications," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1327–1346, 2016.

[51] A. R. Jiménez, F. Seco, J. C. Prieto, and J. Guevara, "Indoor pedestrian navigation using an ins/ekf framework for yaw drift reduction and a foot-mounted imu," in *2010 7th workshop on positioning, navigation and communication*. IEEE, 2010, pp. 135–143.

[52] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *IEEE Computer graphics and applications*, vol. 25, no. 6, pp. 38–46, 2005.

[53] P. Goyal, V. J. Ribeiro, H. Saran, and A. Kumar, "Strap-down pedestrian dead-reckoning system," in *2011 international conference on indoor positioning and indoor navigation*. IEEE, 2011, pp. 1–7.

[54] Y. Fuke and E. Krotkov, "Dead reckoning for a lunar rover on uneven terrain," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1. IEEE, 1996, pp. 411–416.

[55] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, Fitchburg, MA, 2008.

[56] M. S. Grewal, L. R. Weill, and A. P. Andrews, *Global positioning systems, inertial navigation, and integration*. John Wiley & Sons, 2007.

[57] J. S. Walker, M. W. Jones, R. S. Laramee, M. D. Holton, E. L. Shepard, H. J. Williams, D. M. Scantlebury, N. Marks, E. A. Magowan, I. E. Maguire *et al.*, "Prying into the intimate secrets of animal lives; software beyond hardware for comprehensive annotation in 'daily diary'tags," *Movement ecology*, vol. 3, no. 1, pp. 1–16, 2015.

[58] C. Chen, X. Lu, A. Markham, and N. Trigoni, "Ionet: Learning to cure the curse of drift in inertial odometry," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[59] S. Herath, D. Caruso, C. Liu, Y. Chen, and Y. Furukawa, "Neural inertial localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6604–6613.

[60] S. Sun, D. Melamed, and K. Kitani, "Idol: Inertial deep orientation-estimation and localization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 7, 2021, pp. 6128–6137.

[61] R. Gao, X. Xiao, S. Zhu, W. Xing, C. Li, L. Liu, L. Ma, and H. Chai, "Glow in the dark: Smartphone inertial odometry for vehicle tracking in gps blocked environments," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12 955–12 967, 2021.

[62] B. Wagstaff and J. Kelly, "Lstm-based zero-velocity detection for robust inertial navigation," in *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2018, pp. 1–8.

[63] W. Liu, D. Caruso, E. Ilg, J. Dong, A. I. Mourikis, K. Daniilidis, V. Kumar, and J. Engel, "Tlio: Tight learned inertial odometry," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5653–5660, 2020.

[64] M. Brossard and S. Bonnabel, "Learning wheel odometry and imu errors for localization," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 291–297.

[65] M. Brossard, A. Barrau, and S. Bonnabel, "Rins-w: Robust inertial navigation system on wheels," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 2068–2075.

[66] M. Brossard, S. Bonnabel, and A. Barrau, "Denoising imu gyroscopes with deep learning for open-loop attitude estimation," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4796–4803, 2020.

[67] F. Xu and J. Fang, "Velocity and position error compensation using strapdown inertial navigation system/celestial navigation system integration based on ensemble neural network," *Aerospace Science and Technology*, vol. 12, no. 4, pp. 302–307, 2008.

[68] M. K. Al-Sharman, Y. Zweiri, M. A. K. Jaradat, R. Al-Husari, D. Gan, and L. D. Seneviratne, "Deep-learning-based neural network training for state estimation enhancement: application to attitude estimation," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 1, pp. 24–34, 2020.

[69] B. Wagstaff, V. Peretroukhin, and J. Kelly, "Robust data-driven zero-velocity detection for foot-mounted inertial navigation," *IEEE Sensors Journal*, vol. 20, no. 2, pp. 957–967, 2019.

[70] S. Beauregard, "A helmet-mounted pedestrian dead reckoning system," in *3rd International Forum on Applied Wearable Computing 2006*. VDE, 2006, pp. 1–11.

[71] B. Wagstaff, V. Peretroukhin, and J. Kelly, "Improving foot-mounted inertial navigation through real-time motion classification," in *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2017, pp. 1–8.

[72] S. Cortés, A. Solin, and J. Kannala, "Deep learning based speed estimation for constraining strapdown inertial navigation on smartphones," in *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2018, pp. 1–6.

[73] S. Song, J. Liu, J. Guo, J. Wang, Y. Xie, and J.-H. Cui, "Neural-network based auv navigation for fast-changing environments," *IEEE Internet of Things Journal*, 2020.

[74] Y. Xie, J. Liu, C.-q. Hu, J.-h. Cui, and H. Xu, "Auv dead-reckoning navigation based on neural network using a single accelerometer," in *Proceedings of the 11th ACM International Conference on Underwater Networks & Systems*, 2016, pp. 1–5.

[75] S. S. Saha, C. Davis, S. S. Sandha, J. Park, J. Geronimo, L. A. Garcia, and M. Srivastava, "Locomote: Ai-driven sensor tags for fine-grained undersea localization and sensing," *IEEE Internet of Things Journal*, 2023.

[76] Z. Yang, C. Wu, Z. Zhou, X. Zhang, X. Wang, and Y. Liu, "Mobility increases localizability: A survey on wireless indoor localization using inertial sensors," *ACM Computing Surveys (CSUR)*, vol. 47, no. 3, pp. 1–34, 2015.

[77] W. T. Higgins, "A comparison of complementary and kalman filtering," *IEEE Transactions on Aerospace and Electronic Systems*, no. 3, pp. 321–325, 1975.

[78] R. Mahony, T. Hamel, and J.-M. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Transactions on automatic control*, vol. 53, no. 5, pp. 1203–1218, 2008.

[79] M. Euston, P. Coote, R. Mahony, J. Kim, and T. Hamel, "A complementary filter for attitude estimation of a fixed-wing uav," in *2008 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2008, pp. 340–345.

[80] S. O. Madgwick, A. J. Harrison, and R. Vaidyanathan, "Estimation of imu and marg orientation using a gradient descent algorithm," in *2011 IEEE international conference on rehabilitation robotics*. IEEE, 2011, pp. 1–7.

[81] P. Zhou, M. Li, and G. Shen, "Use it free: Instantly knowing your phone attitude," in *Proceedings of the 20th annual international conference on Mobile computing and networking*, 2014, pp. 605–616.

[82] H. Fourati, N. Manamanni, L. Afilal, and Y. Handrich, "A nonlinear filtering approach for the attitude and dynamic body acceleration estimation based on inertial and magnetic sensors: Bio-logging application," *IEEE Sensors Journal*, vol. 11, no. 1, pp. 233–244, 2010.

[83] H. D. Black, "A passive system for determining the attitude of a satellite," *AIAA journal*, vol. 2, no. 7, pp. 1350–1351, 1964.

[84] M. D. Shuster and S. D. Oh, "Three-axis attitude determination from vector observations," *Journal of guidance and Control*, vol. 4, no. 1, pp. 70–77, 1981.

[85] P. B. Davenport, *A vector approach to the algebra of rotations with applications*. National Aeronautics and Space Administration, 1968, vol. 4696.

[86] Z. Liu, W. Liu, X. Gong, and J. Wu, "Simplified attitude determination algorithm using accelerometer and magnetometer with extremely low execution time," *Journal of Sensors*, vol. 2018, 2018.

[87] G. Wahba, "A least squares estimate of satellite attitude," *SIAM review*, vol. 7, no. 3, pp. 409–409, 1965.

[88] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 03 1960.

[89] L. Ljung, "Asymptotic behavior of the extended kalman filter as a parameter estimator for linear systems," *IEEE Transactions on Automatic Control*, vol. 24, no. 1, pp. 36–50, 1979.

[90] R. Van Der Merwe, A. Doucet, N. De Freitas, and E. A. Wan, "The unscented particle filter," in *Advances in neural information processing systems*, 2001, pp. 584–590.

[91] C. Yang and E. Blasch, "Kalman filtering with nonlinear state constraints," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 1, pp. 70–84, 2009.

[92] N. Trawny and S. I. Roumeliotis, "Indirect kalman filter for 3d attitude estimation," *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep*, vol. 2, 2005.

[93] A. M. Sabatini, "Quaternion-based extended kalman filter for determining orientation by inertial and magnetic sensing," *IEEE transactions on Biomedical Engineering*, vol. 53, no. 7, pp. 1346–1356, 2006.

[94] P. Zhang, J. Gu, E. E. Milios, and P. Huynh, "Navigation with imu/gps/digital compass with unscented kalman filter," in *IEEE International Conference Mechatronics and Automation, 2005*, vol. 3. IEEE, 2005, pp. 1497–1502.

[95] A. Giannitrapani, N. Ceccarelli, F. Scortecci, and A. Garulli, "Comparison of ekf and ukf for spacecraft localization via angle measurements," *IEEE Transactions on aerospace and electronic systems*, vol. 47, no. 1, pp. 75–84, 2011.

[96] R. Mehra, "On the identification of variances and adaptive kalman filtering," *IEEE Transactions on automatic control*, vol. 15, no. 2, pp. 175–184, 1970.

[97] W. Li and J. Wang, "Effective adaptive kalman filter for mems-imu/magnetometers integrated attitude and heading reference systems," *The Journal of Navigation*, vol. 66, no. 1, pp. 99–113, 2013.

[98] A. Mohamed and K. Schwarz, "Adaptive kalman filtering for ins/gps," *Journal of geodesy*, vol. 73, no. 4, pp. 193–203, 1999.

[99] E. M. Diaz, A. L. M. Gonzalez, and F. de Ponte Müller, "Standalone inertial pocket navigation system," in *2014 IEEE/ION Position, Location and Navigation Symposium-PLANS 2014*. IEEE, 2014, pp. 241–251.

[100] M. H. Afzal, V. Renaudin, and G. Lachapelle, "Use of earth's magnetic field for mitigating gyroscope errors regardless of magnetic perturbation," *Sensors*, vol. 11, no. 12, pp. 11 390–11 414, 2011.

[101] M. Ma, Q. Song, Y.-h. Li, Y. Gu, and Z.-m. Zhou, "A heading error estimation approach based on improved quasi-static magnetic field detection," in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2016, pp. 1–8.

[102] M. Ilyas, K. Cho, S.-H. Baeg, and S. Park, "Drift reduction in pedestrian navigation system by exploiting motion constraints and magnetic field," *Sensors*, vol. 16, no. 9, p. 1455, 2016.

[103] N. Roy, H. Wang, and R. Roy Choudhury, "I am a smartphone and i can tell my user's walking direction," in *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, 2014, pp. 329–342.

[104] B. Brzozowski, K. Kaźmierczak, Z. Rochala, M. Wojda, and K. Wojtowicz, "A concept of uav indoor navigation system based on magnetic field measurements," in *2016 IEEE Metrology for Aerospace (MetroAeroSpace)*. IEEE, 2016, pp. 636–640.

[105] F. Goldenberg, "Geomagnetic navigation beyond the magnetic compass," in *Proceedings of IEEE/ION PLANS 2006*, 2006, pp. 684–694.

[106] F. C. Teixeira and A. Pascoal, "Magnetic navigation and tracking of underwater vehicles," *IFAC Proceedings Volumes*, vol. 46, no. 33, pp. 239–244, 2013.

[107] A. D. McAulay, "Computerized model demonstrating magnetic submarine localization," *IEEE Transactions on Aerospace and Electronic Systems*, no. 3, pp. 246–254, 1977.

[108] K.-C. Lan and W.-Y. Shih, "On calibrating the sensor errors of a pdr-based indoor localization system," *Sensors*, vol. 13, no. 4, pp. 4781–4810, 2013.

[109] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *Proceedings of the 18th annual international conference on Mobile computing and networking*, 2012, pp. 293–304.

[110] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao, "A reliable and accurate indoor localization method using phone inertial sensors," in *Proceedings of the 2012 ACM conference on ubiquitous computing*, 2012, pp. 421–430.

[111] S. Sen, J. Lee, K.-H. Kim, and P. Congdon, "Avoiding multipath to revive inbuilding wifi localization," in *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, 2013, pp. 249–262.

[112] B. Jia, K. D. Pham, E. Blasch, D. Shen, Z. Wang, and G. Chen, "Cooperative space object tracking using space-based optical sensors via consensus-based filters," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 4, pp. 1908–1936, 2016.

[113] N. Assimakis, M. Adam, and A. Douladiris, "Information filter and kalman filter comparison: Selection of the faster filter," in *Information Engineering*, vol. 2, no. 1, 2012, pp. 1–5.

[114] F. Montorsi, F. Pancaldi, and G. M. Vitetta, "Design and implementation of an inertial navigation system for pedestrians based on a low-cost mems imu," in *2013 IEEE International Conference on Communications Workshops (ICC)*. IEEE, 2013, pp. 57–61.

[115] M. Benoussaad, B. Sijobert, K. Mombaur, and C. Azevedo Coste, "Robust foot clearance estimation based on the integration of foot-mounted imu acceleration data," *Sensors*, vol. 16, no. 1, p. 12, 2016.

[116] Z.-Q. Zhang and X. Meng, "Use of an inertial/magnetic sensor module for pedestrian tracking during normal walking," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 3, pp. 776–783, 2014.

[117] Q. Wang, H. Luo, F. Zhao, and W. Shao, "An indoor self-localization algorithm using the calibration of the online magnetic fingerprints and indoor landmarks," in *2016 international conference on indoor positioning and indoor navigation (IPIN)*. IEEE, 2016, pp. 1–8.

[118] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: wireless indoor localization with little human intervention," in *Proceedings of the 18th annual international conference on Mobile computing and networking*, 2012, pp. 269–280.

[119] H. Fourati, N. Manamanni, L. Afilal, and Y. Handrich, "Position estimation approach by complementary filter-aided imu for indoor environment," in *2013 European Control Conference (ECC)*. IEEE, 2013, pp. 4208–4213.

[120] I. Skog, J.-O. Nilsson, and P. Händel, "Evaluation of zero-velocity detectors for foot-mounted inertial navigation systems," in *2010 International Conference on Indoor Positioning and Indoor Navigation*. IEEE, 2010, pp. 1–6.

[121] A. Mikov, A. Moschevikin, A. Fedorov, and A. Sikora, "A localization system using inertial measurement units from wireless commercial hand-held devices," in *International Conference on Indoor Positioning and Indoor Navigation*. IEEE, 2013, pp. 1–7.

[122] U. Ryu, K. Ahn, E. Kim, M. Kim, B. Kim, S. Woo, and Y. Chang, "Adaptive step detection algorithm for wireless smart step counter," in *2013 International Conference on Information Science and Applications (ICISA)*. IEEE, 2013, pp. 1–4.

[123] Z. Li, C. Song, J. Cai, R. Hua, and P. Yu, "An improved pedestrian navigation system using imu and magnetometer," in *2017 International Conference on Computer Systems, Electronics and Control (ICCSEC)*. IEEE, 2017, pp. 1639–1642.

[124] L.-F. Shi, Y.-L. Zhao, G.-X. Liu, S. Chen, Y. Wang, and Y.-F. Shi, "A robust pedestrian dead reckoning system using low-cost magnetic and inertial sensors," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 8, pp. 2996–3003, 2018.

[125] F. Pasolini and I. Binda, "Pedometer device and step detection method using an algorithm for self-adaptive computation of acceleration thresholds," Dec. 9 2008, uS Patent 7,463,997.

[126] S. Shin, C. Park, J. Kim, H. Hong, and J. Lee, "Adaptive step length estimation algorithm using low-cost mems inertial sensors," in *2007 ieee sensors applications symposium*. IEEE, 2007, pp. 1–5.

[127] J. W. Kim, H. J. Jang, D.-H. Hwang, and C. Park, "A step, stride and heading determination for the pedestrian navigation system," *Journal of Global Positioning Systems*, vol. 3, no. 1-2, pp. 273–279, 2004.

[128] M. Alzantot and M. Youssef, "Uptime: Ubiquitous pedestrian tracking using mobile phones," in *2012 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2012, pp. 3204–3209.

[129] J. Park, Y. Kim, and J. Lee, "Waist mounted pedestrian dead-reckoning system," in *2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*. IEEE, 2012, pp. 335–336.

[130] A. Perttula, H. Leppäkoski, M. Kirkko-Jaakkola, P. Davidson, J. Collin, and J. Takala, "Distributed indoor positioning system with inertial measurements and map matching," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 11, pp. 2682–2695, 2014.

[131] X. Zhu, Q. Li, and G. Chen, "Apt: Accurate outdoor pedestrian tracking with smartphones," in *2013 Proceedings IEEE INFOCOM*. IEEE, 2013, pp. 2508–2516.

[132] D. Gusenbauer, C. Isert, and J. Krösche, "Self-contained indoor positioning on off-the-shelf mobile devices," in *2010 International Conference on Indoor Positioning and Indoor Navigation*. IEEE, 2010, pp. 1–9.

[133] C. Randell, C. Djiallis, and H. Muller, "Personal position measurement using dead reckoning," in *Seventh IEEE International Symposium on Wearable Computers, 2003. Proceedings.* IEEE, 2003, pp. 166–173.

[134] B. Shin, S. Lee, C. Kim, J. Kim, T. Lee, C. Kee, S. Heo, and H. Rhee, "Implementation and performance analysis of smartphone-based 3d pdr system with hybrid motion and heading classifier," in *2014 IEEE/ION Position, Location and Navigation Symposium-PLANS 2014*. IEEE, 2014, pp. 201–204.

[135] W. Kang and Y. Han, "Smartpdr: Smartphone-based pedestrian dead reckoning for indoor localization," *IEEE Sensors journal*, vol. 15, no. 5, pp. 2906–2916, 2014.

[136] L. Rong, D. Zhiguo, Z. Jianzhong, and L. Ming, "Identification of individual walking patterns using gait acceleration," in *2007 1st international Conference on Bioinformatics and Biomedical Engineering*. IEEE, 2007, pp. 543–546.

[137] H. Guo, M. Uradziński, H. Yin, and M. Yu, "Indoor positioning based on foot-mounted imu," *Bulletin of the Polish Academy of Sciences. Technical Sciences*, vol. 63, no. 3, pp. 629–634, 2015.

[138] A. R. Jimenez, F. Seco, C. Prieto, and J. Guevara, "A comparison of pedestrian dead-reckoning algorithms using a low-cost mems imu," in *2009 IEEE International Symposium on Intelligent Signal Processing*. IEEE, 2009, pp. 37–42.

[139] W. Zhang, X. Li, D. Wei, X. Ji, and H. Yuan, "A foot-mounted pdr system based on imu/ekf+ hmm+ zupt+ zaru+ hdr+ compass algorithm," in *2017 International conference on indoor positioning and indoor navigation (IPIN)*. IEEE, 2017, pp. 1–5.

[140] N. Castaneda and S. Lamy-Perbal, "An improved shoe-mounted inertial navigation system," in *2010 International Conference on Indoor Positioning and Indoor Navigation*. IEEE, 2010, pp. 1–6.

[141] S.-W. Lee and K. Mase, "Activity and location recognition using wearable sensors," *IEEE pervasive computing*, vol. 1, no. 3, pp. 24–32, 2002.

[142] L. A. Zadeh, "Fuzzy logic," *Computer*, vol. 21, no. 4, pp. 83–93, 1988.

[143] A. Mannini and A. M. Sabatini, "Gait phase detection and discrimination between walking–jogging activities using hidden markov models applied to foot motion data from a gyroscope," *Gait & posture*, vol. 36, no. 4, pp. 657–661, 2012.

[144] A. Sundaresan, A. RoyChowdhury, and R. Chellappa, "A hidden markov model based framework for recognition of humans from gait sequences," in *Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429)*, vol. 2. IEEE, 2003, pp. II–93.

[145] A. Kale, A. Sundaresan, A. Rajagopalan, N. P. Cuntoor, A. K. Roy-Chowdhury, V. Kruger, and R. Chellappa, "Identification of humans using gait," *IEEE Transactions on image processing*, vol. 13, no. 9, pp. 1163–1173, 2004.

[146] S. K. Park and Y. S. Suh, "A zero velocity detection algorithm using inertial sensors for pedestrian navigation systems," *Sensors*, vol. 10, no. 10, pp. 9163–9178, 2010.

[147] G. Panahandeh, N. Mohammadiha, A. Leijon, and P. Händel, "Continuous hidden markov model for pedestrian activity classification and gait analysis," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 5, pp. 1073–1083, 2013.

[148] M. Ren, K. Pan, Y. Liu, H. Guo, X. Zhang, and P. Wang, "A novel pedestrian navigation algorithm for a foot-mounted inertial-sensor-based system," *Sensors*, vol. 16, no. 1, p. 139, 2016.

[149] R. W. Levi and T. Judd, "Dead reckoning navigational system using accelerometer to measure foot impacts," Dec. 10 1996, uS Patent 5,583,776.

[150] V. Renaudin, M. Susi, and G. Lachapelle, "Step length estimation using handheld inertial sensors," *Sensors*, vol. 12, no. 7, pp. 8507–8525, 2012.

[151] H. Weinberg, "Using the adxl202 in pedometer and personal navigation applications," *Analog Devices AN-602 application note*, vol. 2, no. 2, pp. 1–6, 2002.

[152] L. Fang, P. J. Antsaklis, L. A. Montestruque, M. B. McMickell, M. Lemmon, Y. Sun, H. Fang, I. Koutroulis, M. Haenggi, M. Xie *et al.*, "Design of a wireless assisted pedestrian dead reckoning system-the navmote experience," *IEEE transactions on Instrumentation and Measurement*, vol. 54, no. 6, pp. 2342–2358, 2005.

[153] J. Scarlett, "Enhancing the performance of pedometers using a single accelerometer," *Application Note, Analog Devices*, no. AN-900, 2007.

[154] M. Reinstein and M. Hoffmann, "Dead reckoning in a dynamic quadruped robot: Inertial navigation system aided by a legged odometer," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 617–624.

[155] A. Shurin and I. Klein, "Qdr: A quadrotor dead reckoning framework," *IEEE Access*, vol. 8, pp. 204 433–204 440, 2020.

[156] S. Zahran, A. M. Moussa, A. B. Sesay, and N. El-Sheimy, "A new velocity meter based on hall effect sensors for uav indoor navigation," *IEEE Sensors Journal*, vol. 19, no. 8, pp. 3067–3076, 2018.

[157] R. P. Wilson, M. D. Holton, A. di Virgilio, H. Williams, E. L. C. Shepard, S. Lambertucci, F. Quintana, J. E. Sala, B. Balaji, E. S. Lee, M. Srivastava, D. M. Scantlebury, and C. M. Duarte, "Give the machine a hand: A boolean time-based decision-tree template for rapidly finding animal behaviours in multisensor data," *Methods in Ecology and Evolution*, vol. 9, no. 11, pp. 2206–2215, 2018.

[158] C. Wan, L. Wang, and V. V. Phoha, "A survey on gait recognition," *ACM Computing Surveys (CSUR)*, vol. 51, no. 5, pp. 1–35, 2018.

[159] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Computing Surveys (CSUR)*, vol. 43, no. 3, pp. 1–43, 2011.

[160] F. Attal, S. Mohammed, M. Dedabrishvili, F. Chamroukhi, L. Oukhellou, and Y. Amirat, "Physical human activity recognition using wearable sensors," *Sensors*, vol. 15, no. 12, pp. 31 314–31 338, 2015.

[161] C. Chen, R. Jafari, and N. Kehtarnavaz, "A survey of depth and inertial sensor fusion for human action recognition," *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 4405–4425, 2017.

[162] N. Y. Hammerla, S. Halloran, and T. Plötz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2016, pp. 1533–1540.

[163] S. Ramasamy Ramamurthy and N. Roy, "Recent trends in machine learning for human activity recognition—a survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1254, 2018.

[164] C. Yang, L. Kaplan, and E. Blasch, "Performance measures of covariance and information matrices in resource management for target state estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 3, pp. 2594–2613, 2012.

[165] V. A. Bavdekar, A. P. Deshpande, and S. C. Patwardhan, "Identification of process and measurement noise covariance for state and parameter estimation using extended kalman filter," *Journal of Process control*, vol. 21, no. 4, pp. 585–601, 2011.

[166] K. Myers and B. Tapley, "Adaptive sequential estimation with unknown noise statistics," *IEEE transactions on automatic control*, vol. 21, no. 4, pp. 520–523, 1976.

[167] C. G. Hilborn and D. G. Lainiotis, "Optimal estimation in the presence of unknown parameters," *IEEE Transactions on Systems Science and Cybernetics*, vol. 5, no. 1, pp. 38–43, 1969.

[168] B. Carew and P. Belanger, "Identification of optimum filter steady-state gain for systems with unknown noise covariances," *IEEE Transactions on Automatic Control*, vol. 18, no. 6, pp. 582–587, 1973.

[169] X. Gao, H. Luo, B. Ning, F. Zhao, L. Bao, Y. Gong, Y. Xiao, and J. Jiang, "Rl-akf: An adaptive kalman filter navigation algorithm based on reinforcement learning for ground vehicles," *Remote Sensing*, vol. 12, no. 11, p. 1704, 2020.

[170] X. Zhao, C. Deng, X. Kong, J. Xu, and Y. Liu, "Learning to compensate for the drift and error of gyroscope in vehicle localization," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 852–857.

[171] F. Wu, H. Luo, H. Jia, F. Zhao, Y. Xiao, and X. Gao, "Predicting the noise covariance with a multitask learning model for kalman filter-based gnss/ins integrated navigation," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–13, 2020.

[172] K. A. Kramer, S. C. Stubberud, and J. A. Geremia, "Target registration correction using the neural extended kalman filter," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 7, pp. 1964–1971, 2009.

[173] S. E. Kozhaya, J. A. Haidar-Ahmad, A. A. Abdallah, Z. M. Kassas, and S. S. Saab, "Comparison of neural network architectures for simultaneous tracking and navigation with leo satellites," in *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021)*, 2021, pp. 2507–2520.

[174] H. Kautz, "The third ai summer: Aaai robert s. engelmore memorial lecture," *AI Magazine*, vol. 43, no. 1, pp. 93–104, 2022.

[175] M. K. Sarker, L. Zhou, A. Eberhart, and P. Hitzler, "Neuro-symbolic artificial intelligence," *AI Communications*, no. Preprint, pp. 1–13, 2021.

[176] C. Chen, Y. Miao, C. X. Lu, L. Xie, P. Blunsom, A. Markham, and N. Trigoni, "Motiontransformer: Transferring neural inertial tracking between domains," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 8009–8016.

[177] J. Lin, W.-M. Chen, Y. Lin, C. Gan, S. Han *et al.*, "Mcunet: Tiny deep learning on iot devices," *Advances in Neural Information Processing Systems*, vol. 33, pp. 11 711–11 722, 2020.

[178] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," in *International Conference on Learning Representations (ICLR)*, 2016.

[179] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," *Advances in neural information processing systems*, vol. 28, 2015.

[180] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep learning with limited numerical precision," in *International conference on machine learning*. PMLR, 2015, pp. 1737–1746.

[181] S. S. Saha, S. S. Sandha, and M. Srivastava, "Machine learning for microcontroller-class hardware: A review," *IEEE Sensors Journal*, vol. 22, no. 22, pp. 21 362–21 390, 2022.

[182] R. David, J. Duke, A. Jain, V. Janapa Reddi, N. Jeffries, J. Li, N. Kreeger, I. Nappier, M. Natraj, T. Wang *et al.*, "Tensorflow lite micro: Embedded machine learning for tinyml systems," *Proceedings of Machine Learning and Systems*, vol. 3, pp. 800–811, 2021.

[183] T. Chen, T. Moreau, Z. Jiang, L. Zheng, E. Yan, H. Shen, M. Cowan, L. Wang, Y. Hu, L. Ceze *et al.*, "{TVM}: An automated {End-to-End} optimizing compiler for deep learning," in *13th USENIX Symposium*

on Operating Systems Design and Implementation (OSDI 18), 2018, pp. 578–594.
[184] G. Gobieski, B. Lucia, and N. Beckmann, "Intelligence beyond the edge: Inference on intermittent embedded systems," in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, 2019.
[185] C. Banbury, C. Zhou, I. Fedorov, R. Matas, U. Thakker, D. Gope, V. Janapa Reddi, M. Mattina, and P. Whatmough, "Micronets: Neural network architectures for deploying tinyml applications on commodity microcontrollers," *Proceedings of Machine Learning and Systems*, vol. 3, pp. 517–532, 2021.
[186] E. Liberis, Ł. Dudziak, and N. D. Lane, "μnas: Constrained neural architecture search for microcontrollers," in *Proceedings of the 1st Workshop on Machine Learning and Systems*, 2021, pp. 70–79.
[187] I. Fedorov, R. P. Adams, M. Mattina, and P. N. Whatmough, "Sparse: Sparse architecture search for cnns on resource-constrained microcontrollers," *Advances in Neural Information Processing Systems*, 2019.
[188] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "{TensorFlow}: a system for {Large-Scale} machine learning," in *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, 2016.
[189] S. S. Sandha, M. Aggarwal, I. Fedorov, and M. Srivastava, "Mango: A python library for parallel hyperparameter tuning," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3987–3991.
[190] S. S. Sandha, M. Aggarwal, S. S. Saha, and M. Srivastava, "Enabling hyperparameter tuning of machine learning classifiers in production," in *2021 IEEE Third International Conference on Cognitive Machine Intelligence (CogMI)*. IEEE, 2021, pp. 262–271.
[191] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," *Advances in neural information processing systems*, vol. 25, 2012.
[192] T. Desautels, A. Krause, and J. W. Burdick, "Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization," *Journal of Machine Learning Research*, vol. 15, pp. 3873–3923, 2014.
[193] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," in *9th ISCA Speech Synthesis Workshop*, 2016.
[194] C. Lea, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks: A unified approach to action segmentation," in *European Conference on Computer Vision*. Springer, 2016, pp. 47–54.
[195] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, 2016.
[196] M. Ferrera, V. Creuze, J. Moras, and P. Trouvé-Peloux, "Aqualoc: An underwater dataset for visual–inertial–pressure localization," *The International Journal of Robotics Research*, vol. 38, no. 14, 2019.
[197] R. M. Gunner, M. D. Holton, M. D. Scantlebury, O. L. V. Schalkwyk, H. M. English, H. J. Williams, P. Hopkins, F. Quintana, A. Gómez-Laich, L. Börger, and et al., "Dead-reckoning animal movements in r: a reappraisal using gundog.tracks," *Animal Biotelemetry*, vol. 9, 2021.
[198] M. Brossard, S. Bonnabel, and J.-P. Condomines, "Unscented kalman filtering on lie groups," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 2485–2491.
[199] H. Qi and J. B. Moore, "Direct kalman filtering approach for gps/ins integration," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 2, pp. 687–693, 2002.
[200] L. McGee, S. Schmidt, and G. Smith, "Applications of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle," *NASA Technical Report R-135, Tech. Rep*, 1962.
[201] H. W. Sorenson, "On the development of practical nonlinear filters," *Information Sciences*, vol. 7, pp. 253–270, 1974.
[202] M. I. Ribeiro, "Kalman and extended kalman filters: Concept, derivation and properties," *Institute for Systems and Robotics*, vol. 43, 2004.
[203] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.
[204] N. Thacker and A. Lacey, "Tutorial: The kalman filter," *Imaging Science and Biomedical Engineering Division, Medical School, University of Manchester*, vol. 61, 1998.
[205] N. El-Sheimy, H. Hou, and X. Niu, "Analysis and modeling of inertial sensors using allan variance," *IEEE Transactions on instrumentation and measurement*, vol. 57, no. 1, pp. 140–149, 2007.
[206] M. Kumar, "World geodetic system 1984: A modern and accurate global reference frame," *Marine Geodesy*, vol. 12, no. 2, 1988.
[207] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, 2010.
[208] D. Shao, X. Liu, B. Cheng, O. Wang, and T. Hoang, "Edge4real: A cost-effective edge computing based human behaviour recognition system for human-centric software engineering," in *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering*, 2020, pp. 1287–1291.
[209] G. Morgan-Owen and G. Johnston, "Differential gps positioning," *Electronics & Communication Engineering Journal*, vol. 7, no. 1, 1995.
[210] P. Soille *et al.*, *Morphological image analysis: principles and applications*. Springer, 1999, vol. 2, no. 3.
[211] R. Van Den Boomgaard and R. Van Balen, "Methods for fast morphological image transforms using bitmapped binary images," *CVGIP: Graphical Models and Image Processing*, vol. 54, no. 3, 1992.
[212] C. Tomasi and T. Kanade, "Detection and tracking of point," *Int J Comput Vis*, vol. 9, pp. 137–154, 1991.
[213] J. Shi and C. Tomasi, "Good features to track," in *1994 Proceedings of IEEE conference on computer vision and pattern recognition*. IEEE, 1994, pp. 593–600.
[214] B. Justusson, "Median filtering: Statistical properties," *Two-Dimensional Digital Signal Prcessing II*, pp. 161–196, 1981.
[215] H. Xu, P. Zhou, R. Tan, M. Li, and G. Shen, "Limu-bert: Unleashing the potential of unlabeled data for imu sensing applications," in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, 2021, pp. 220–233.
[216] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.
[217] S. Dhar, J. Guo, J. Liu, S. Tripathi, U. Kurup, and M. Shah, "A survey of on-device machine learning: An algorithms and learning theory perspective," *ACM Transactions on Internet of Things*, vol. 2, no. 3, pp. 1–49, 2021.
[218] H. Cai, C. Gan, L. Zhu, and S. Han, "Tinytl: Reduce memory, not parameters for efficient on-device learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 11 285–11 297, 2020.
[219] H. Ren, D. Anicic, and T. A. Runkler, "Tinyol: Tinyml with online-learning on microcontrollers," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–8.
[220] S. Greydanus, M. Dzamba, and J. Yosinski, "Hamiltonian neural networks," *Advances in Neural Information Processing Systems*, vol. 32, pp. 15 379–15 389, 2019.
[221] M. Cranmer, S. Greydanus, S. Hoyer, P. Battaglia, D. Spergel, and S. Ho, "Lagrangian neural networks," in *ICLR 2020 WKSH on Integration of Deep Neural Models and Differential Equations*, 2020.
[222] S. Yao, A. Piao, W. Jiang, Y. Zhao, H. Shao, S. Liu, D. Liu, J. Li, T. Wang, S. Hu *et al.*, "Stfnets: Learning sensing signals from the time-frequency perspective with short-time fourier neural networks," in *The World Wide Web Conference*, 2019, pp. 2192–2202.
[223] S. Li, R. R. Chowdhury, J. Shang, R. K. Gupta, and D. Hong, "Units: Short-time fourier inspired neural networks for sensory time series classification," in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, 2021, pp. 234–247.
[224] S. Badreddine, A. d. Garcez, L. Serafini, and M. Spranger, "Logic tensor networks," *Artificial Intelligence*, vol. 303, p. 103649, 2022.
[225] S. S. Saha, S. S. Sandha, and M. Srivastava, "Deep convolutional bidirectional lstm for complex activity recognition with missing data," in *Human Activity Recognition Challenge*. Springer, 2020, pp. 39–53.
[226] A. Stisen, H. Blunck, S. Bhattacharya, T. S. Prentow, M. B. Kjærgaard, A. Dey, T. Sonne, and M. M. Jensen, "Smart devices are different: Assessing and mitigating mobile sensing heterogeneities for activity recognition," in *Proceedings of the 13th ACM Conference on embedded networked Sensor Systems*, 2015, pp. 127–140.
[227] S. S. Sandha, J. Noor, F. M. Anwar, and M. Srivastava, "Time awareness in deep learning-based multimodal fusion across smartphone platforms," in *2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI)*. IEEE, 2020.
[228] S. S. Sandha, F. M. Anwar, J. Noor, and M. Srivastava, "Exploiting smartphone peripherals for precise time synchronization," in *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2019.