The Constitutional Filter: Bayesian Estimation of Compliant Agents

Simon Kohaut[1,*], Felix Divo[1,*], Benedict Flade[2],
Devendra Singh Dhami[3], Julian Eggert[2], Kristian Kersting[1,4,5,6]

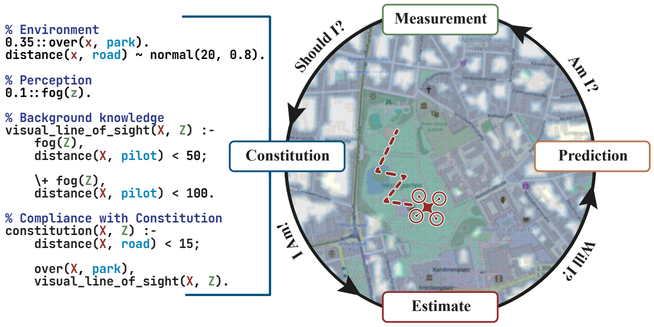
***Abstract*— Predicting agents impacted by legal policies, physical limitations, and operational preferences is inherently difficult. In recent years, neuro-symbolic methods have emerged, integrating machine learning and symbolic reasoning models into end-to-end learnable systems. Hereby, a promising avenue for expressing high-level constraints over multi-modal input data in robotics has opened up. This work introduces an approach for Bayesian estimation of agents expected to comply with a human-interpretable neuro-symbolic model we call its Constitution. Hence, we present the Constitutional Filter (CoFi), leading to improved tracking of agents by leveraging expert knowledge, incorporating deep learning architectures, and accounting for environmental uncertainties. CoFi extends the general, recursive Bayesian estimation setting, ensuring compatibility with a vast landscape of established techniques such as Particle Filters. To underpin the advantages of CoFi, we evaluate its performance on real-world marine traffic data. Beyond improved performance, we show how CoFi can learn to trust and adapt to the level of compliance of an agent, recovering baseline performance even if the assumed Constitution clashes with reality.**

***Index Terms*— Neuro-Symbolic Systems, Bayesian Estimation**

I. Introduction

PREDICTING a moving target without knowing its intentions or control methodology is inherently challenging. Specifically, when limited to the most basic state-space models, e.g., constant velocity or acceleration systems, only a vague understanding of the agent's trajectory can be formed. When this is the case, e.g., with human-operated vehicles such as cars or boats, one can leverage knowledge about the agent's environment to restrict the predicted state evolution. For instance, in the case of road vehicles, one can inform the forward model with the curvature of the road the agent is currently traversing [1]. However, the insight into purely geometric aspects does not fully leverage the frequently available background knowledge on legislative and logical constraints.

To this end, we present the Constitutional Filter (CoFi), a novel prediction model coupling an agent's motion to a neuro-symbolic model of its internal rules and perception of its state space. While such models have been employed as the basis for planning, e.g., mission design for Unmanned Aircraft Systems in regulated urban environments [2], this work explores how to apply such techniques for predicting other agents' motion without insights into their actual control and intentions. More specifically, we employ probabilistic first-order logic, capturing knowledge about the legal and physical constraints in an uncertainty-aware environment representation. This permits expressing rules over spatial relations between the agent's state-space and geographic features from potentially inaccurate map data.



```
% Environment
0.35::over(x, park).
distance(x, road) ~ normal(20, 0.8).

% Perception
0.1::fog(z).

% Background knowledge
visual_line_of_sight(X, Z) :-
    fog(Z),
    distance(X, pilot) < 50;

    \+ fog(Z),
    distance(X, pilot) < 100.

% Compliance with Constitution
constitution(X, Z) :-
    distance(X, road) < 15;

    over(X, park),
    visual_line_of_sight(X, Z).
```

Fig. 1: **The Constitutional Filter's estimation cycle:** A Constitution for a remote-controlled, unmanned aircraft system is shown on the left. The right-hand side shows the tracked drone and the likely compliant areas shaded in blue.

In summary, our key contributions are:

- We present the Constitutional Filter (CoFi), an extension to Bayesian estimation that tracks compliant agents informed by a neuro-symbolic model of the environment's and agent's rules as illustrated in Figure 1.
- We demonstrate how CoFi learns to trust observed agents, quantifying the degree to which they are expected to act according to the assumed Constitution.
- We show how CoFi adapts the impact of the Constitution according to the learned trust, assuring that a faulty constitution does not degrade filter performance.

Additionally, we provide an open-source implementation of CoFi as part of our framework for Probabilistic Mission Design in multi-modal mobility[1].

[*] Authors contributed equally
[1] Artificial Intelligence and Machine Learning Group,
Department of Computer Science,
TU Darmstadt, 64283 Darmstadt, Germany
[2] Honda Research Institute Europe GmbH,
Carl-Legien-Str. 30, 63073 Offenbach, Germany
[3] Uncertainty in Artificial Intelligence Group,
Department of Mathematics and Computer Science,
TU Eindhoven, 5600 MB Eindhoven, Netherlands
[4] Hessian AI
[5] Centre for Cognitive Science
[6] German Center for Artificial Intelligence (DFKI)

[1] github.com/HRI-EU/ProMis

II. Related Work

CoFi is a novel approach for the Bayesian estimation of compliant agents. To contextualize CoFi, we discuss neuro-symbolic systems (reasoning), probabilistic robotics (filtering), as well as perception and representation (mapping).

A. Neuro-Symbolic Systems

Neuro-symbolic systems are an emerging field aiming to intertwine programmatic reasoning on a symbolic level with the sub-symbolic capabilities of deep learning models.

One of the earliest programmatic reasoning systems based on first-order logic is Prolog [3]. To additionally embrace uncertainty into programmatic logic, systems such as Bayesian Logic Programs [4] and Probabilistic Logic Programs [5], [6] have been introduced. While they were not formulated for end-to-end learning with artificial neural networks, languages such as DeepProbLog [7], NeurASP [8], and SLASH [9] close this gap and combine the strengths of neural information processing and probabilistic reasoning. Recent work has demonstrated how such systems can be employed as a basis for Probabilistic Mission Design [2], [10] by encoding, e.g., traffic laws in hybrid probabilistic first-order logic [11] as a basis for planning or granting clearance to an agent.

CoFi follows the idea of encoding the laws and regulations that govern an agent's environment in a neuro-symbolic setting, formulating the agent's Constitution. In contrast to ProMis, this enables CoFi to integrate state estimation, sensor data, and high-level semantic information into a Bayes Filter for tracking targets with high-level rules of operation.

B. Probabilistic Robotics

One cannot deny the critical role of probabilistic methods in robotics, commonly based on Bayesian beliefs [12]. For instance, state-of-the-art localization usually relies on employing members of the family of Bayes Filters (Section III-A), such as the Kalman Filter [13], providing a closed-form solution to recursive tracking in linear and Gaussian systems.

To enable tracking in non-linear systems while keeping the Gaussian assumption, generalizations such as the Extended Kalman Filter [14]–[16] and the Unscented Kalman Filter [17], [18] have been introduced. While the former relies on a linearization of the model, the latter works by forwarding samples through the so-called unscented transform to conserve the Gaussian nature of the estimate. When dropping linearity and distribution assumptions on the underlying process, one can employ the Particle Filter [19], [20] instead. Here, approximate results are obtained using sequential Monte Carlo techniques.

CoFi is also a member of the Bayes Filters. By introducing a third step to the paradigm of prediction-correction schemes, namely the Constitution step, CoFi guides the estimation process through the likelihood of a satisfied Constitution given the state, measurement, and high-level semantic information. Because CoFi does not alter the roles of the process and measurement models, it stays compatible with a wide range of prior work on filter techniques that can complement CoFi's insights based on probabilistic first-order logic.

C. Perception and Environment Representation

Numerous sensors have been developed to perceive oneself and the environment. They range from proprioceptive sensors, e.g., the Global Navigation Satellite System [21] and Inertial Measurement Units [22], [23] which inform about the agent themselves, to exteroceptive sensors such as cameras [24], [25], Lidar [26], [27], and Radar [28], [29] which inform about the environment.

These perceptions can be leveraged to improve filter performance significantly [30]. For instance, in the autonomous driving context, this has been achieved by coupling observations of lane markings [31] or stop lines [32] with a vehicle's proprioception. Similarly, knowledge of the road geometry can be employed to successfully predict vehicle motion along a local Frenet coordinate system [1], assuming the agent follows local traffic regulations. Likewise, high-level semantic information on the road infrastructure and rules can be leveraged to improve motion prediction [33], e.g., lane count and directionality. Like transportation systems need to follow road laws, tracking pedestrians profits from integrating data about local laws and environment features such as movement patterns and social interactions [34].

Along these lines of research, CoFi integrates high-level concepts and semantics of the environment by employing a Statistical Relational Map [35] (StaR Map, see Section III-B), assigning heterogeneous levels of uncertainty to geographic features [36]. Hence, in its Constitution, probabilistic inference across likely states and measurements results in an additional distribution weighting the estimation process, encoding how likely the agent's local laws and assumed preferences are satisfied.

III. Preliminaries

We present the general formulations and notation of recursive estimation of dynamic systems in a Bayesian setting and the uncertainty-aware, semantic environment representation of Statistical Relational Maps, as they will be required later in Sec. IV for building the Constitutional Filter.

A. Bayesian Filters

Bayesian filters recursively estimate the state $\mathbf{x}_t \in \mathcal{X}$ of a dynamical system at time $t \in \mathbb{N}$. While the state is hidden, a combination of prediction through a process model f with control inputs $\mathbf{u}_t \in \mathbb{R}^U, U \in \mathbb{N}$ and belief updates through an observation model h with measurements $\mathbf{z}_t \in \mathcal{Z}$ is employed to keep track of $\mathbf{x}_t$. More specifically, the system can be written as

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{e}_{x,t}) \text{ and } \mathbf{z}_t = h(\mathbf{x}_t, \mathbf{u}_t, \mathbf{e}_{z,t}). \quad (1)$$

Intuitively speaking, f expresses an expectation on a state's evolution within a discrete time-step, while h encodes the values produced by a sensor. Here, both models are subject to i.i.d. noise $\mathbf{e}_{x,t}$ and $\mathbf{e}_{z,t}$ respectively. Depending on the application, this formulation can slightly change, e.g., often the input $\mathbf{u}_t$ is not part of the computation of the system's output $\mathbf{z}_t$. We will omit $\mathbf{u}_t$ for simplicity in the following.

A Bayesian filter operates in two steps. First, a prior belief is obtained using the last estimate $\mathbf{x}_{t-1}$ as

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) = \int \overbrace{p(\mathbf{x}_t|\mathbf{x}_{t-1})}^{\text{Process Model } f} p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}) d\mathbf{x}_{t-1}. \quad (2)$$

Second, once a measurement $\mathbf{z}_t$ is available, the prediction is updated using Bayes' rule

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) = \frac{\overbrace{p(\mathbf{z}_t|\mathbf{x}_t)}^{\text{Observation Model } h} p(\mathbf{x}_t|\mathbf{z}_{1:t-1})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1})}, \text{ where} \quad (3)$$

$$p(\mathbf{z}_t|\mathbf{z}_{1:t-1}) = \int p(\mathbf{z}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) d\mathbf{x}_t. \quad (4)$$

As is the case for the Kalman filter, closed-form solutions to the equations above exist depending on assumptions about the observed process, e.g., linear state transitions and Gaussian noise. Without such strong assumptions, approaches such as the Particle filter allow the tracking of the hidden state using sampling techniques.

As a final remark, in recursive filtering, we assume the Markov property for states and measurements, i.e.,

$$p(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t-1}) = p(\mathbf{x}_t|\mathbf{x}_{t-1}) \text{ and} \quad (5)$$

$$p(\mathbf{z}_t|\mathbf{x}_{1:t}) = p(\mathbf{z}_t|\mathbf{x}_t). \quad (6)$$

Intuitively, the previous state contains sufficient information for prediction to discard all other historical data.

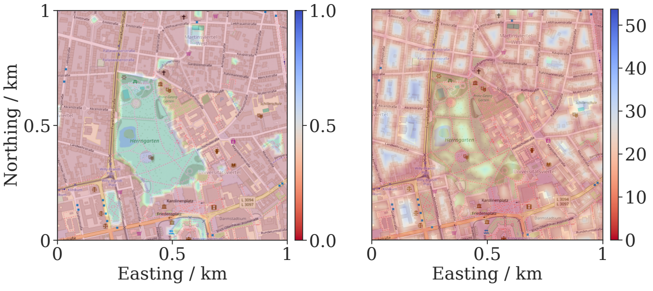
B. Statistical Relational Maps

Statistical Relational Maps (StaR Maps) have been recently introduced to represent uncertain environments consisting of semantic features in a relational and probabilistic manner [35]. Rather than containing a graphical representation of the environment, a StaR Map parameterizes hybrid probabilistic (discrete and continuous) spatial relations.

Using a StaR Map, the so-obtained spatial relations can be translated to distributional ground atoms in first-order logic, exemplified by the Listing in Figure 1. Furthermore, one can visualize the probabilistic spatial relations, e.g., by computing the expected values across the mapped space as shown in Figure 2.

Consider the exemplified relations *over(x, g)* and *distance(x, g)* between a point $\mathbf{x} \in \mathbb{R}^D$ to a set of environment features $g \in \mathcal{G}$. The specifics, such as the dimensionality of the mapped space and the sets of features, will depend on the application. For example, in the aerial mobility scenario illustrated in Figure 1, $\mathcal{G} = \{park, road, pilot\}$ and $D = 2$. Let us discuss how StaR Maps estimate the parameters of such relations.

Consider a map $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \rho)$ be a triple of vertices $\mathcal{V}$, edges $\mathcal{E}$, and a tagging function $\rho : \mathcal{V} \to \mathcal{P}(\mathcal{G})$. The function $\rho(\mathbf{v}) \subseteq \mathcal{G}$ annotates vertex $\mathbf{v} \in \mathcal{V}$ with a set of semantic tags. If a path exists between two vertices in $\mathcal{V}$ across edges in $\mathcal{E}$, StaR Maps consider them part of the same *feature*, hence ρ assigns the same type to each.



(a) $E\left[over(X, park)\right]$ (b) $E\left[distance(X, road)\right]$

Fig. 2: **StaR Maps parameters in an Advanced Aerial Mobility setting**: Here, the expected values of two basic probabilistic spatial relations are shown in an urban environment, namely *over* and *distance* (used in Figure 1).

To address inaccuracies of the map data, e.g., due to low-cost sensors or crowd-sourced data, StaR Maps employ a stochastic error model analogously to prior work [36]. That is, for each $\mathbf{v}_{i,j} \in \mathcal{V}$, being the j-th vertex of the i-th feature, the following generates $N \in \mathbb{N}$ samples:

$$\mathbf{\Phi}^{(n)} \sim \phi_i \quad \text{(Transformation)}$$

$$\mathbf{t}^{(n)} \sim \kappa_i \quad \text{(Translation)}$$

$$\mathbf{v}_{i,j}^{(n)} = \mathbf{\Phi}^{(n)} \cdot \mathbf{v}_{i,j} + \mathbf{t}^{(n)} \quad \text{(Generation)}$$

Here, ϕ_i and κ_i are feature-wise distributions of linear maps $\mathbf{\Phi}^{(n)}$, e.g., rotation, scaling or shearing, and translations $\mathbf{t}^{(n)}$. With $\mathcal{M}^{(n)}$ being generated by taking the n-th sample of each vertex and a copy of the edges $\mathcal{E}$, this collection of randomized maps allows StaR Maps to compute the statistical moments of its spatial relations.

Let $r(\mathcal{M}^{(n)}, \mathbf{x}, g)$ be a deterministic function evaluating a spatial relation on map $\mathcal{M}^{(n)}$ at location $\mathbf{x} \in \mathbb{R}^d$ limited to features for which $\rho(\mathbf{v}) = g$. Through this sampling process, we can empirically estimate statistical moments, e.g., mean and variance, with respect to the chosen relation r, location $\mathbf{x}$, and tag g:

$$\widehat{\mu_r} = \frac{1}{N} \sum_n r(\mathcal{M}^{(n)}, \mathbf{x}, g) \quad (7)$$

$$\widehat{\sigma_r^2} = \frac{1}{N-1} \sum_n \left(r(\mathcal{M}^{(n)}, \mathbf{x}, g) - \widehat{\mu_r} \right)^2 \quad (8)$$

For example, assume *distance*$(\mathbf{x}, g) \sim \mathcal{N}(\mu_d, \sigma_d^2)$ to be a normally distributed random variable with the deterministic function r_d. That is, the function $r_d(\mathcal{M}^{(n)}, \mathbf{x}, g)$ computes the Euclidean distance from $\mathbf{x}$ to the closest feature in $\mathcal{M}^{(n)}$.

IV. Methods

We present the Constitutional Filter (CoFi) for tracking compliant agents with unknown internal control and planning mechanisms. This includes the discussion of (*A*) CoFi's novel architecture, introducing neuro-symbolic concepts to recursive Bayesian estimation, (*B*) the construction and (*C*) semantics of the Constitution as a probabilistic reasoning model, and (*D*) how to obtain a density of the constitutional likelihood to be integrated into (*E*) the constitutional belief update. Finally, we show how (*F*) CoFi learns and integrates a level of trust about how compliant observed agents are.

A. *Architecture*

As illustrated in Figure 3, CoFi integrates neuro-symbolic representations of the agent's internal preferences, physical limitations, and environment policies, such as traffic laws, into recursive Bayesian estimation. Hence, CoFi leverages high-level expert knowledge about the task, environment, and semantic information beyond simple sensor readings. Specifically, besides a process model f and observation model h, CoFi is equipped with the Constitution $\mathcal{C}_t$, allowing it to map states and measurements into the probability of the Constitution being satisfied. Through a learned trust ratio τ, CoFi automatically regulates the influence the Constitution has on the belief update, recovering the equations of an unconstitutional filter if the observed agent proves incompliant.

B. *Constitutions as Deep Probabilistic Logic Programs*

We encode the Constitution as a deep probabilistic first-order logic program $\mathcal{C}_t = \mathcal{B}_t \cup \mathcal{P}_t \cup \mathcal{E}_t$, combined of sub-programs encoding background knowledge $\mathcal{B}_t$, perception $\mathcal{P}_t$, and environment representation $\mathcal{E}_t$ as illustrated in Figure 1. For CoFi, we assume $\mathcal{B}_t$ to be provided by a domain expert, i.e., by formalizing their knowledge as first-order logic or by translating natural language descriptions through, e.g., a Large Language Model. Further, $\mathcal{P}_t$ encodes the semantics of the environment as experienced during the agent's lifetime, e.g., information from other agents in the environment. Finally, $\mathcal{E}_t$ comprises spatial relations queried from a StaR Map as outlined in Section III-B.

Constitutions consist of *clauses*, each being made up of a *head*, optional *body*, and *distribution*. Consider the following two clauses.

$$p \ :: \ r_1(a_1, \dots, a_n) \text{ :- } l_1, \ \dots, \ l_m. \qquad \text{(Categorical)}$$

$$r_2(a_1, \dots, a_i) \sim p(\theta) \text{ :- } l_1, \ \dots, \ l_j. \qquad \text{(Continuous)}$$

In the first case, the head r_1 is true with a probability p given that all the *literals* l_k in the body are true. Analogously, in the second case, head r_2 is distributed according to the density $p(\theta)$ with parameters θ if its body is true. If the right-hand side is empty, the head is regarded as a fact and distributed independently of any other symbols.

For example, let us consider the running examples from Section III-B as probabilistic clauses:

$$0.95 \ :: \ \text{over}(\text{x}, \text{park}). \tag{9}$$

$$\text{distance}(\text{x}, \text{road}) \sim \mathit{normal}(100, 1). \tag{10}$$

This means that at the location referenced with the term x, there is a park with a probability of 0.95, and the distance to the nearest road is expected to be about $100\,\text{m}$.

A logic program must be solved, i.e., the models under which a chosen clause is satisfied must be found to perform probabilistic inference. While details will differ across off-the-shelf solving pipelines such as Prolog [37] or clingo [38], the rough process can be described in two steps. First, the program is grounded, replacing all variables with the possible values from the program's domain. Second, the solver processes the grounded program, enumerating all solutions consisting of models $j \in \mathcal{J}$ of ground atoms $a \in \mathcal{A}$.

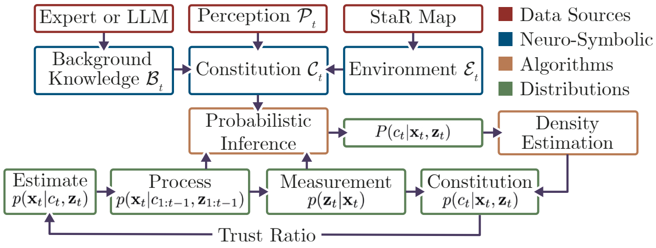

Expert or LLM
Perception $\mathcal{P}_t$
StaR Map
Data Sources
Neuro-Symbolic
Algorithms
Distributions
Background Knowledge $\mathcal{B}_t$
Constitution $\mathcal{C}_t$
Environment $\mathcal{E}_t$
Probabilistic Inference
$P(c_t|\mathbf{x}_t, \mathbf{z}_t)$
Density Estimation
Estimate $p(\mathbf{x}_t|c_t, \mathbf{z}_t)$
Process $p(\mathbf{x}_t|c_{1:t-1}, \mathbf{z}_{1:t-1})$
Measurement $p(\mathbf{z}_t|\mathbf{x}_t)$
Constitution $p(c_t|\mathbf{x}_t, \mathbf{z}_t)$
Trust Ratio

Fig. 3: **The Constitutional Filter's architecture:** CoFi extends recursive Bayesian estimation using a neuro-symbolic model of the tracked agent's decision-making constraints, integrating background knowledge, perception, and a probabilistic environment representation.

C. *Exact Probabilistic Inference*

We now aim to compute the probability of a pair $(\mathbf{x}_t, \mathbf{z}_t)$ satisfying the first-order logic program of $\mathcal{C}_t$ when solving for the *constitution(X, Y)* clause. For exact probabilistic inference, one needs to assign the probabilities $P(a = j(a)|\mathbf{x}_t, \mathbf{z}_t, \mathcal{C}_t)$ of atom a to take on the value assigned by model j given the current state $\mathbf{x}_t$, measurement $\mathbf{z}_t$ and constitution $\mathcal{C}_t$. For CoFi, we refer to a StaR Map (see Section III-B) to provide the distribution parameters to ground atoms expressing spatial relations as in Eqs. 9 and 10. Further, we assume $\mathcal{B}_t$ to be parameterized according to expert knowledge or learned and $\mathcal{P}_t$ to encode uncertainties associated with the respective perceived properties (see Figure 1 and Listing 1). In turn, the probability $P(C_t|\mathbf{x}_t, \mathbf{z}_t)$ of the Constitution being satisfied given the state and measurement at time t is then obtained via the sum-product

$$P(C_t|\mathbf{x}_t, \mathbf{z}_t) = \sum\nolimits_{j \in \mathcal{J}} \prod\nolimits_{a \in \mathcal{A}} P(a = j(a)|\mathbf{x}_t, \mathbf{z}_t, \mathcal{C}_t). \tag{11}$$

A knowledge compiler is often employed to compress this sum-product using a heuristic search for a minimal formula, often leading to substantial inference speedups [39].

D. *Density Estimation of the Constitutional Likelihood*

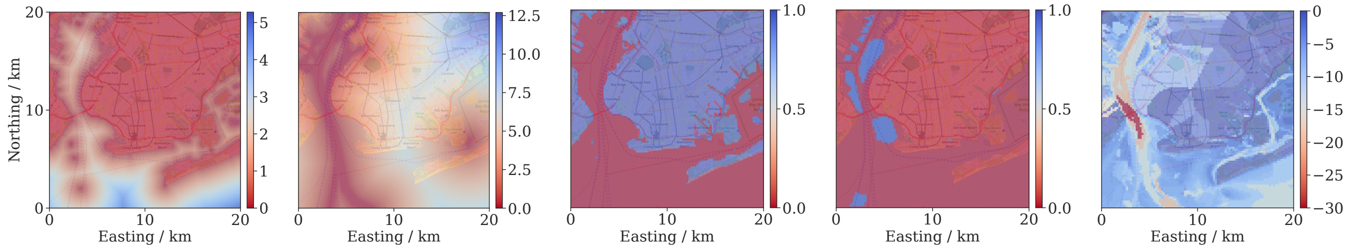
Note that $P(C_t|\mathbf{x}_t, \mathbf{z}_t)$ of Eq. 11 is a discrete distribution, encoding the probability of the constitution being satisfied given a state and measurement. To integrate the Constitution into a Bayesian belief update in CoFi's process, we must find the probability density function $p(c_t|\mathbf{x}_t, \mathbf{z}_t)$ of a sample satisfying the Constitution. To do so, we compute the set of probabilities of the constitution being satisfied

$$\mathcal{S} = \left\{ P(C_t|\mathbf{x}_t^{(n)}, \mathbf{z}_t^{(n)}) \right\}_{n \in \{1, \dots, N\}}, \tag{12}$$

where $\mathbf{x}_t^{(n)} \sim p(\mathbf{x}_t|c_{1:t-1}, \mathbf{z}_{1:t-1})$ and $\mathbf{z}_t^{(n)} \sim p(\mathbf{z}_t|\mathbf{x}_t)$. One can then approximate the density $p(c_t|\mathbf{x}_t, \mathbf{z}_t)$ from $\mathcal{S}$, for instance, using Kernel Density Estimation [40], [41].

E. *Constitutional Bayesian Belief Update*

We have shown how one can compute the probability that the Constitution is satisfied for a state $\mathbf{x}_t$ and measurement $\mathbf{z}_t$, yielding $p(c_t|\mathbf{x}_t, \mathbf{z}_t)$. Hence, we can integrate the Constitution into a general Bayesian belief update.



(a) $distance(X, land)$ (b) $distance(X, way)$ (c) $over(X, land)$ (d) $over(X, anchorage)$ (e) $depth(X, water)$

Fig. 4: **Statistical relational maps for marine traffic:** Expectations of StaR Map relations employed in Listing 1 at New York's harbor, considering land, waterways, anchorage areas, and depth. Distances are shown in kilometers.

CoFi's prior of the state follows the standard formulation while considering past evaluations of the agent's Constitution, similar to Eq. 2:

$$p(\mathbf{x}_t | c_{1:t-1}, \mathbf{z}_{1:t-1}) = \int p(\mathbf{x}_t | \mathbf{x}_{t-1}) \, p(\mathbf{x}_{t-1} | c_{1:t-1}, \mathbf{z}_{1:t-1}) d\mathbf{x}_{t-1}. \tag{13}$$

Further, we obtain the posterior of an agent's location given the Constitution by plugging Eq. 13 into Bayes' rule as in Eq. 3:

$$p(\mathbf{x}_t | c_{1:t}, \mathbf{z}_{1:t}) = \frac{p(c_t, \mathbf{z}_t | \mathbf{x}_t) p(\mathbf{x}_t | c_{1:t-1}, \mathbf{z}_{1:t-1})}{\int p(c_t, \mathbf{z}_t | \mathbf{x}_t) p(\mathbf{x}_t | c_{1:t-1}, \mathbf{z}_{1:t-1}) d\mathbf{x}_t}. \tag{14}$$

Finally, by applying the chain rule of probability to the conditional joint probability $p(c_t, \mathbf{z}_t | \mathbf{x}_t)$, we get the Constitutional Bayesian belief update such that

$$p(\mathbf{x}_t | c_{1:t}, \mathbf{z}_{1:t}) = \frac{\overbrace{p(c_t | \mathbf{x}_t, \mathbf{z}_t)}^{\text{Constitution}} \overbrace{p(\mathbf{z}_t | \mathbf{x}_t)}^{\text{Measurement}} \overbrace{p(\mathbf{x}_t | c_{1:t-1}, \mathbf{z}_{1:t-1})}^{\text{Prediction}}}{\int p(c_t | \mathbf{x}_t, \mathbf{z}_t) p(\mathbf{z}_t | \mathbf{x}_t) p(\mathbf{x}_t | c_{1:t-1}, \mathbf{z}_{1:t-1}) d\mathbf{x}_t}. \tag{15}$$

F. Constitutional Trust Ratio

One cannot blindly assume that the proclaimed Constitution perfectly matches an agent's behavior. First, the Constitution might be incomplete or erroneous. Second, the agent might not diligently follow the rules of the environment, e.g., by disregarding local policies. Hence, we introduce the trust ratio $\tau(\boldsymbol{\psi})$, allowing CoFi to blend between the Constitutional likelihood and a uniform distribution:

$$p_\tau(c_t | \mathbf{x}_t, \mathbf{z}_t) = \tau(\boldsymbol{\psi}_t) p(c_t | \mathbf{x}_t, \mathbf{z}_t) + (1 - \tau(\boldsymbol{\psi}_t)) U(0, 1) \tag{16}$$

$$\boldsymbol{\psi}_t = \begin{bmatrix} \psi_1(\mathbf{x}_t, \mathbf{z}_t) & \cdots & \psi_K(\mathbf{x}_t, \mathbf{z}_t) \end{bmatrix} \tag{17}$$

Here, the vector $\boldsymbol{\psi}$ captures trust features, e.g., the type of a vehicle or its current velocity, that the trust depends on.

The density $p_\tau(c_t | \mathbf{x}_t, \mathbf{z}_t)$ can then replace the standard Constitutional likelihood in Eq. 15. Note that for $\tau = 1$, we fully trust the correctness of the constitution, and for $\tau = 0$, we obtain a standard Bayesian belief update that discards the constitution entirely (unconstitutional baseline). As demonstrated in Sec. V-E, maximizing CoFi's accuracy on real-world data can determine an appropriate trust ratio.

V. Experiments

We answer the following questions using real-world marine localization from Autonomous Identification System (AIS) recordings and official charts.

(Q1) How can we encode expert knowledge about marine vessels, like traffic regulations, into a Constitution?

(Q2) What are the computational demands of integrating neuro-symbolic reasoning into Bayesian estimation?

(Q3) Can we learn how much to trust marine vessels given the Constitution and historical AIS data?

(Q4) Does a trust-calibrated CoFi lead to improved tracking accuracy compared to the unconstitutional baseline?

A. Experimental Setup

Throughout the experiments, we assume a constant velocity model $\mathbf{x}_t = (\mathbf{p}_t, \mathbf{v}_t)$ in a two-dimensional space, i.e., $\mathbf{p}_t, \mathbf{v}_t \in \mathbb{R}^2$ and $\dot{\mathbf{p}}_t = \mathbf{v}_t$, where only the position $\mathbf{p}_t$ is measured. Hence, given the difference δt between time steps, the dynamic system can be written as

$$\mathbf{x}_t = \begin{bmatrix} 1 & \delta t \\ 0 & 1 \end{bmatrix} \mathbf{x}_{t-1} \quad \text{and} \quad \mathbf{z}_t = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}_t.$$

All experiments were conducted on an AMD Ryzen Threadripper 1950X 16-Core processor with 128GB of memory. Where applicable, we report the means and standard deviations obtained from multiple runs with varying seeds.

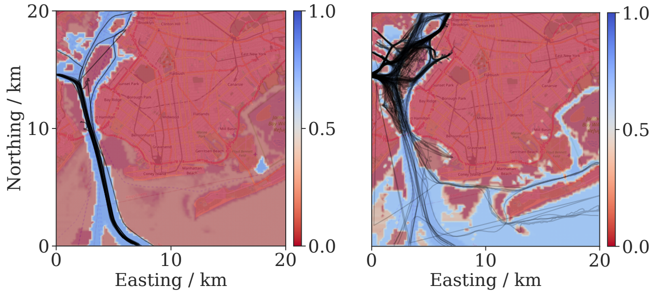
B. A Constitution for Maritime Traffic Tracking

We address **(Q1)** in three steps. First, we integrate messages retrieved via AIS [42], transmitting navigation and ship data in marine applications, into the Constitution's perception subprogram $\mathcal{P}_t$ and trust features $\boldsymbol{\psi}$. Second, common safety and navigational constraints considering the vessel's purpose were encoded into the agents' background knowledge $\mathcal{B}_t$. Finally, the environment $\mathcal{E}_t$ is represented by a StaR Map processing official Electronic Navigation Charts (ENC) provided by the United States National Oceanic and Atmospheric Administration (NOAA) [43]. Figure 4 shows the resulting spatial relations, detailing geographic and navigational features surrounding the New York Harbor area. The composition of $\mathcal{P}_t$, $\mathcal{B}_t$, and $\mathcal{E}_t$ is shown in Listing 1.

We solve Listing 1 for the *constitution(X, Z)* clause across the $20 \times 20\,\mathrm{km}^2$ area by plugging in the respective StaR

```
% Trust features from (deep) models
trust_feature({cargo, towing, sar}, ais_class(z)).
trust_feature(on_waterways, waterway_bound(z)).
trust_feature(anchoring, anchoring_classifier(z)).

% Environment from nautical StaR Map
distance(x, land) ~ normal(150, 15).
distance(x, waterway) ~ normal(300, 20).
0.0::over(x, land).
0.1::over(x, anchorage).
depth(x, water) ~ normal(20, 1).

% Perception from sensors
draught(z) ~ normal(3, 1).

% Background knowledge
sufficient_depth(X, Z) :-
    A is depth(X, water), B is draught(Z),
    A + B < -1.5, \+ over(X, land).

0.95::safe(X) :-
    sufficient_depth(X), distance(X, land) > 50.

proper_anchorage(X, Z) :-
    anchoring(X, Z), over(X, anchorage);
    \+ anchoring(X, Z), \+ over(X, anchorage).

0.90::respects_waterways(X, Z) :-
    on_waterways(Z), distance(X, waterway) < 400;
    \+ on_waterways(Z).

% Compliance with Constitution
constitution(X, Z) :-
    proper_anchorage(X, Z), safe(X),
    respects_waterways(X, Z).

% Query P(C | x, z)
query(constitution(x, z)).
```

Listing 1: A marine vessel's constitution, modeling compliance with safety and traffic rules.



(a) Cargo

(b) Towing

Fig. 5: **Marine traffic over constitutional probabilities:** Here, we show real-world AIS traces from (a) often compliant cargo and (b) often incompliant towing vessels on top of the probabilities for each to satisfy the constitution.

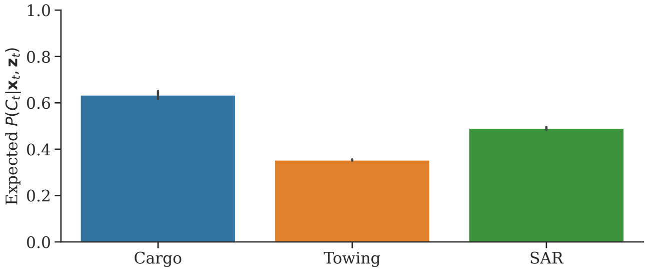


Fig. 6: **Trust depending on observations:** Not every tracked agent can be expected to satisfy the assumed constitution. Here, the average value of $P(C_t|\mathbf{x}_t, \mathbf{z}_t)$ is shown aggregated for different vessel types.

Maps parameters and observations. We show how this Constitution successfully models the environment and movement of vessels in the area in Figure 5. One can see how, although some trajectories lead through assumed incompliant areas, the traffic patterns overall follow the model. Similarly, Figure 6 shows how the probability of agents sticking to the Constitution depends on, e.g., their employment type. Note how, for example, towing vehicles are overall less likely to follow the Constitution, as seen in Figure 5b. We will revisit this observation in Section V-E when determining appropriate trust ratios.

C. *Resource Requirements*

To address **(Q2)**, we (i) break down the costs of computing a StaR Map from ENCs and (ii) compare the runtime of tracking vessel motion with a standard Particle Filter and with CoFi. Computing StaR Maps at a resolution of 100×100 points for each probabilistic spatial relation took 630.69 ± 8.51 seconds. When querying the Constitution, we employ linear interpolation for parameters not on the grid. While we compute StaR Maps parameters once in advance, more dynamic applications may need updates due to environmental changes, e.g., a *distance* to a moving target.

The StaR Map parameters have been visualized in Fig. 4, including the *distance* and *over* relations analogously to the original paper and a new *depth* relation introduced here. While both mean and standard deviation were computed as described in Section III-B, we visualized the probabilistic spatial relations through their expected values.

We implement CoFi as a Constitutional Particle Filter using sequential Monte Carlo approximation with 2000 particles. This allows us to compare CoFi to a common baseline, i.e., we effectively ablate the evaluation of the Constitution and its guidance for the tracking process by controlling for the influence of computing the Constitutional likelihood.

Without the neuro-symbolic reasoning of CoFi, a standard Particle Filter update can be obtained in 0.004 ± 0.001 seconds. By sampling 100 times locally according to the particle distribution, CoFi can run constitutional belief updates in 0.829 ± 0.024 seconds. Else, one may precompute the nearby scalar field $P(C_t|\mathbf{x}_t, \mathbf{z}_t)$ if the environment and measurements are sufficiently static. Then, CoFi's updates run near baseline speed in 0.007 ± 0.001 seconds. Hence, while CoFi can easily keep up in the maritime context, optimizations are necessary for highly dynamic settings.

D. *Calibrating the Trust Ratio*

We resolve **(Q3)** by modeling a marine Constitution as laid out in Section V-B, considering three trust features. Namely, the type of the vessel extracted from its AIS report, whether it is bound to waterways through a heuristic rule, and a deep learning classifier deciding whether it is currently anchoring. To this end, we access real-world AIS recordings provided

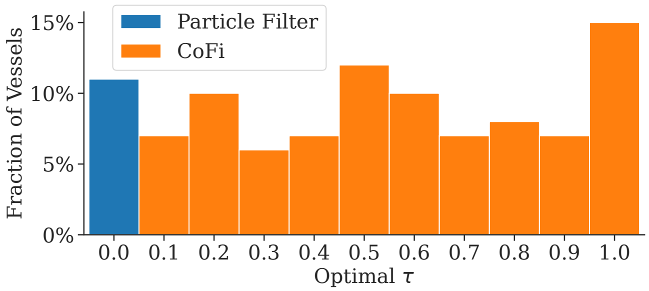


Fig. 7: **A majority of vessels can be trusted:** Here, the rate at which a specific value of τ is optimal (given observed trust features) is shown for the entire dataset. Note how CoFi rarely needs to fall back to its internal Particle Filter.

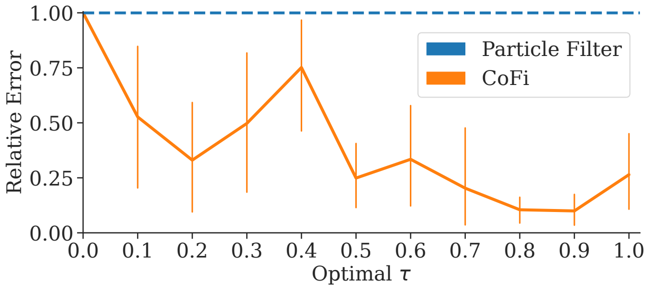


Fig. 8: **CoFi adapting to appropriate trust:** We show the relative mean absolute error, comparing CoFi's performance with the baseline Particle Filter. Once a vessel profits from $\tau > 0$, CoFi achieves greatly improved accuracy.

by NOAA detailing, among others, the vessel's position and characteristics, such as high-level usage type or draft.

We can henceforth observe the alignment of individual agents' tracks with the Constitution as previously discussed in Section V-B and visualized by Figures 5 and 6. Note how CoFi decides the trust level τ based on all three trust features, not only on the shown vessel type. Here, without loss of generality, we consider the trust features and the appropriate trust to be time-invariant throughout a recorded journey.

CoFi chooses τ to maximize its tracking accuracy for agents with the respective trust features. In our experiments, we perform this computation as an offline learning task on historical AIS data, comparing performance across discrete choices for τ. In Figure 7, we demonstrate the distribution of optimal choices for τ across the entire dataset population. It becomes evident again that, although simple, the assumed Constitution matches most agents' behavior. That is, about 89% of agents are tracked best for a trust $\tau > 0$, benefiting from CoFi's improved estimation, as further discussed next.

E. Constitutional Filtering

Finally, to answer **(Q4)**, we employ the obtained optimal trust τ for respective trust features $\boldsymbol{\psi}$ to assess how CoFi improves estimation accuracy if properly calibrated. Depending on the learned optimal trust for a vessel's trust features, the Constitution of CoFi improves accuracy or recovers baseline performance. So, if trust features indicate the vessel does not comply with the assumed model and the optimal τ is 0, the belief update equals a Particle Filter without Constitution. As Figure 8 shows, CoFi provides more accurate tracking as soon as $\tau > 0$, leveraging the information provided by the Constitutional likelihood in CoFi's belief update.

Note that there are limitations to these results, considering the choice of the Constitution. While a trust-calibrated CoFi clearly can lead to improved results and assures an upper bound to its error, not every environment and application will allow for formulating a Constitution that clearly separates likely and unlikely states. For example, while roads, pavements, and traffic lights give way to predicting traffic participants' motion in an urban environment, the same cannot be said for areas where free motion is predominant.

VI. Discussion

We propose the Constitutional Filter (CoFi), a neuro-symbolic Bayesian estimation scheme based on a model of an agent's internal rules over their behavior. CoFi improves traditional filtering approaches by reasoning on an agent's Constitution. In experiments on real-world map and localization data, we have demonstrated how the Constitution can substantially improve the filter's accuracy and how its impact can be tuned to an agent's individual behavior. Furthermore, through the symbolic nature of the Constitution written in probabilistic first-order logic, an interpretable and adaptable interface to the filtering process is provided.

Although CoFi brings several advantages, as shown in Section V, they depend on application-specific background knowledge which must be obtained first. While we have shown a hand-crafted Constitution, real-world applications may need to explore options such as rule-learning or automated code generation via Large Language Models. Furthermore, prediction with CoFi entails additional computational costs, possibly prohibiting its application in environments where particularly rapid inference is key.

In the future, end-to-end learning with the agent's Constitution, i.e., learning the rules' conditional probabilities and bodies from data, as well as the weights of neural models feeding into the Constitution, could be explored. Similarly, the automatic discovery of an agent's reasons for non-compliance with the assumed Constitution is a promising direction forward, i.e., learning if the Constitution itself is faulty or if the agent deliberately does not comply.

Acknowledgments

Simon Kohaut gratefully acknowledges the financial support from Honda Research Institute Europe. This work received funding from the German BMBF project "KompAKI" within the "The Future of Value Creation – Research on Production, Services and Work" program (grant 02L19C150), managed by the PTKA. The TU Eindhoven authors received support from their Department of Mathematics and Computer Science and the Eindhoven Artificial Intelligence Systems Institute. Map data © OpenStreetMap contributors, licensed under the ODbL and available from openstreetmap.org.

References

[1] R. Toledo-Moreo, D. Betaille, and F. Peyret, "Lane-Level Integrity Provision for Navigation and Map Matching With GNSS, Dead Reckoning, and Enhanced Maps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 1, pp. 100–112, mar 2010.

[2] S. Kohaut, B. Flade, D. S. Dhami, J. Eggert, and K. Kersting, "Mission design for unmanned aerial vehicles using hybrid probabilistic logic programs," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2023, pp. 1506–1513.

[3] A. Colmerauer, "An introduction to prolog III," *Communications of the ACM*, vol. 33, no. 7, pp. 69–90, 1990.

[4] K. Kersting and L. De Raedt, "Bayesian logic programming: Theory and tool," *Introduction to Statistical Relational Learning*, p. 291, 2007.

[5] L. De Raedt, A. Kimmig, and H. Toivonen, "Problog: A probabilistic prolog and its application in link discovery." in *IJCAI*, vol. 7. Hyderabad, 2007, pp. 2462–2467.

[6] D. Fierens, G. Van den Broeck, J. Renkens, D. Shterionov, B. Gutmann, I. Thon, G. Janssens, and L. De Raedt, "Inference and learning in probabilistic logic programs using weighted boolean formulas," *Theory and Practice of Logic Programming*, vol. 15, no. 3, pp. 358–401, 2015.

[7] R. Manhaeve, S. Dumancic, A. Kimmig, T. Demeester, and L. De Raedt, "DeepProbLog: Neural probabilistic logic programming," *advances in neural information processing systems*, vol. 31, 2018.

[8] Z. Yang, A. Ishay, and J. Lee, "Neurasp: Embracing neural networks into answer set programming," in *29th International Joint Conference on Artificial Intelligence (IJCAI 2020)*, 2020.

[9] A. Skryagin, D. Ochs, D. S. Dhami, and K. Kersting, "Scalable neural-probabilistic answer set programming," *Journal of Artificial Intelligence Research*, vol. 78, pp. 579–617, 2023.

[10] S. Kohaut, B. Flade, D. S. Dhami, J. Eggert, and K. Kersting, "Towards probabilistic clearance, explanation and optimization," in *2024 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2024, pp. 911–916.

[11] D. Nitti, T. De Laet, and L. De Raedt, "Probabilistic logic programming for hybrid relational domains," *Machine Learning*, vol. 103, pp. 407–449, 2016.

[12] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, ser. Intelligent robotics and autonomous agents. MIT Press, 2005.

[13] 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. [Online]. Available: https://doi.org/10.1115/1.3662552

[14] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, jun 2007.

[15] G. P. C. Junior, A. M. C. Rezende, V. R. F. Miranda, R. Fernandes, H. Azpurua, A. A. Neto, G. Pessin, and G. M. Freitas, "EKF-LOAM: An Adaptive Fusion of LiDAR SLAM With Wheel Odometry and Inertial Data for Confined Spaces With Few Geometric Features," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 1458–1471, jul 2022.

[16] H. Fang, M. A. Haile, and Y. Wang, "Robust Extended Kalman Filtering for Systems With Measurement Outliers," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 2, pp. 795–802, mar 2022.

[17] R. Martinez-Cantin and J. A. Castellanos, "Unscented SLAM for large-scale outdoor environments," *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pp. 3427–3432, 2005.

[18] J. Kuti, I. J. Rudas, H. Gao, and P. Galambos, "Computationally Relaxed Unscented Kalman Filter," *IEEE Transactions on Cybernetics*, vol. 53, no. 3, pp. 1557–1565, mar 2023.

[19] P. Del Moral, "Nonlinear filtering: Interacting particle resolution," *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*, vol. 325, no. 6, pp. 653–658, 1997.

[20] J. Gao and H. Zhao, "An Improved Particle Filter for UAV Passive Tracking Based on RSS," in *2020 IEEE 92nd Vehicular Technology Conference*. IEEE, nov 2020, pp. 1–6.

[21] N. Joubert, T. G. Reid, and F. Noble, "Developments in modern GNSS and its impact on autonomous vehicle architectures," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 2029–2036.

[22] F. Zhang, H. Stahle, G. Chen, C. C. C. Simon, C. Buckl, and A. Knoll, "A sensor fusion approach for localization with cumulative error elimination," in *2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. IEEE, sep 2012, pp. 1–6.

[23] L. Gao, L. Xiong, X. Xia, Y. Lu, Z. Yu, and A. Khajepour, "Improved Vehicle Localization Using On-Board Sensors and Vehicle Lateral Velocity," *IEEE Sensors Journal*, vol. 22, no. 7, pp. 6818–6831, apr 2022.

[24] Y. Alkendi, L. Seneviratne, and Y. Zweiri, "State of the Art in Vision-Based Localization Techniques for Autonomous Navigation Systems," *IEEE Access*, vol. 9, pp. 76 847–76 874, 2021.

[25] B. Flade, M. Nieto, G. Velez, and J. Eggert, "Lane Detection Based Camera to Map Alignment Using Open-Source Map Data," in *21st IEEE International Conference on Intelligent Transportation Systems*, 2018, pp. 890–897.

[26] M. Demir and K. Fujimura, "Robust Localization with Low-Mounted Multiple LiDARs in Urban Environments," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, oct 2019, pp. 3288–3293.

[27] M. Petrlik, T. Krajnik, and M. Saska, "LIDAR-based Stabilization, Navigation and Localization for UAVs Operating in Dark Indoor Environments," in *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, no. 871479. IEEE, jun 2021, pp. 243–251.

[28] E. Ward and J. Folkesson, "Vehicle localization with low cost radar sensors," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, vol. 2016-Augus, no. Iv. IEEE, jun 2016, pp. 864–870.

[29] J. Michalczyk, R. Jung, and S. Weiss, "Tightly-Coupled EKF-Based Radar-Inertial Odometry," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, oct 2022, pp. 12 336–12 343.

[30] P. Karaked, W. Saengphet, and S. Tantrairatn, "Multi-Sensor Fusion with Extended Kalman Filter for Indoor Localization system of Multirotor UAV," in *2022 19th International Joint Conference on Computer Science and Software Engineering (JCSSE)*. IEEE, jun 2022, pp. 1–5.

[31] B. Flade, A. Koppert, G. Velez, A. Das, D. Betaille, G. Dubbelman, O. Otaegui, and J. Eggert, "Vision-Enhanced Low-Cost Localization in Crowdsourced Maps," *IEEE Intelligent Transportation Systems Magazine*, vol. 12, no. 3, pp. 70–80, 2020.

[32] A. Barth, J. Siegemund, and J. Schwehr, "Fast and precise localization at stop intersections," in *2013 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, jun 2013, pp. 82–87.

[33] D. Svensson and J. Sorstedt, "Ego lane estimation using vehicle observations and map information," in *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, jun 2016, pp. 909–914.

[34] K. Krishanth, X. Chen, R. Tharmarasa, T. Kirubarajan, and M. McDonald, "The Social Force PHD Filter for Tracking Pedestrians," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 4, pp. 2045–2059, aug 2017.

[35] B. Flade, S. Kohaut, J. Eggert, D. S. Dhami, and K. Kersting, "Star maps: Unveiling uncertainty in geospatial relations," in *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2024.

[36] B. Flade, S. Kohaut, and J. Eggert, "Error decomposition for hybrid localization systems," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 149–156.

[37] J. Wielemaker, T. Schrijvers, M. Triska, and T. Lager, "Swi-prolog," *Theory and Practice of Logic Programming*, vol. 12, no. 1-2, pp. 67–96, 2012.

[38] M. Gebser, R. Kaminski, B. Kaufmann, and T. Schaub, "Multi-shot ASP solving with clingo," *CoRR*, vol. abs/1705.09811, 2017.

[39] C. Muise, S. A. McIlraith, J. C. Beck, and E. I. Hsu, "D sharp: Fast d-DNNF compilation with sharpSAT," in *Advances in Artificial Intelligence: 25th Canadian Conference on Artificial Intelligence, Canadian AI 2012, Toronto, ON, Canada, May 28-30, 2012. Proceedings 25*. Springer, 2012, pp. 356–361.

[40] R. A. Davis, K.-S. Lii, and D. N. Politis, "Remarks on some nonparametric estimates of a density function," *Selected Works of Murray Rosenblatt*, pp. 95–100, 2011.

[41] E. Parzen, "On estimation of a probability density function and mode," *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[42] M. Series, "Technical characteristics for an automatic identification system using time-division multiple access in the VHF maritime mobile band," *Recommendation ITU: Geneva, Switzerland*, pp. 1371–1375, 2014.

[43] US NOAA Office for Coastal Management, "Marine Cadastre Vessel Traffic," https://hub.marinecadastre.gov/pages/vesseltraffic, accessed: March 2025.