Neural Enhanced Belief Propagation for Data Association in Multiobject Tracking

Mingchao Liang and Florian Meyer
Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA
Email: {m3liang, flmeyer}@ucsd.edu

***Abstract*—Situation-aware technologies enabled by multi-object tracking (MOT) methods will create new services and applications in fields such as autonomous navigation and applied ocean sciences. Belief propagation (BP) is a state-of-the-art method for Bayesian MOT that relies on a statistical model and preprocessed sensor measurements. In this paper, we establish a hybrid method for model-based and data-driven MOT. The proposed neural enhanced belief propagation (NEBP) approach complements BP by information learned from raw sensor data with the goal to improve data association and to reject false alarm measurements. We evaluate the performance of our NEBP approach for MOT on the *nuScenes* autonomous driving dataset and demonstrate that it can outperform state-of-the-art reference methods.**

***Index Terms*—Belief propagation, graph neural networks, multiobject tracking, factor graph.**

I. Introduction

Multi-object tracking (MOT) [1] is a key aspect in a variety of applications including autonomous navigation and applied ocean sciences. In particular, in autonomous navigation systems accurate MOT enables tasks such as motion forecasting [2] and path planning [3]. The main challenge in MOT is data association uncertainty, i.e., the unknown association between measurements and objects. MOT is further complicated by the fact that the number of objects is unknown.

Most MOT methods adopt a detect-then-track approach where an object detector [4]–[13] is applied to the raw sensor data. The resulting object detections are then used as measurements for MOT. Many existing MOT methods follow a global nearest neighbor [1] approach where a Hungarian [14] or a greedy matching algorithm is used to perform "hard" data association. These types of methods rely on heuristics for track initialization and termination. To improve the reliability of hard data association, often discriminative features are extracted and incorporated by the matching algorithm [7], [15]–[22]. Another line of work formulates and solves MOT in the Bayesian estimation framework [1], [23]–[29]. This type of methods rely on statistical models for object birth, object motion, and sensor measurements [1], [23]–[29]. The statistical model makes it possible to perform a more robust probabilistic "soft" data association. In addition, heuristics for track initialization and termination can be avoided by modeling the existence of objects by binary random variables.

BP [30] can solve high-dimensional Bayesian estimation problems by "passing messages" on the edges of a factor graph [30] that represents the underlying statistical model. By exploiting the structure of the graph, BP-based MOT methods [23]–[25], [31], [32] are highly scalable. This makes it possible to generate and maintain a very large number of potential object tracks and, in turn, outperform existing MOT approaches [24], [25]. However, BP fully relies on a statistical model. When the factor graph does not accurately represent the true data generating process, MOT performance is reduced due to model mismatch. In addition, with the detect-then-track approach employed by BP-based MOT, important object-related information might be discarded by the object detector. We aim to overcome these limitations by introducing information learned from raw sensor data to BP-based MOT.

NEBP [33] is a hybrid method that combines the benefits of model-based and data-driven inference and addresses potential limitations of BP such as model mismatch and overconfident beliefs [33], [34]. NEBP has been successfully applied to decoding [33] and cooperative localization [34] tasks. In NEBP, a graph neural network (GNN) that matches the topology of the factor graph is introduced. The trained GNN enhances potentially inaccurate BP messages to ultimately improve object declaration and estimation accuracy.

In this paper, we propose NEBP for MOT. Here, BP messages calculated as input for probabilistic data association are combined with the output of the GNN. The GNN uses measurements (i.e., object detections) and shape features learned from raw sensor data as an input. For *false alarm rejection*, the GNN identifies which measurements are likely false alarms. If a measurement has been identified as a potential false alarm, the false alarm distribution in the statistical model of BP is locally increased. This reduces the probability that the measurement is associated with an existing object track or initializes a new object track. *Object shape association* computes improved association probabilities by also comparing shape features extracted for existing object tracks with shape features extracted for measurements. The resulting NEBP method for MOT can improve object declaration and estimation performance compared to BP for MOT as well as outperform further state-of-the-art methods.

The main contributions of this paper can be summarized as follows.

- We introduce an NEBP method for MOT where probabilistic data association is augmented by shape features learned from raw sensor data.
- We apply the proposed method to the nuScenes autonomous

driving dataset [35] and demonstrate state-of-the-art object tracking performance.

Our approach recognizes that in modern MOT problems with high-resolution sensors [32], [36]–[38], it is challenging to capture object shapes and the corresponding data generating process by a statistical model. Consequently, the influence of object shapes on data generation is best learned directly from data.

II. Background

In what follows, we will briefly review factor graphs and graph neural networks (GNNs).

A. Factor Graphs

A factor graph [30] is a bipartite undirected graph $\mathcal{G}_f = (\mathcal{V}_f, \mathcal{E}_f)$ that consists of a set of edges $\mathcal{E}_f$ and a set of vertices or nodes $\mathcal{V}_f = \mathcal{Q} \cup \mathcal{F}$. A variable node $q \in \mathcal{Q}$ represents a random variable $\mathbf{x}_q$ and a factor node $s \in \mathcal{F}$ represents a factor $f_s\big(\mathbf{x}^{(s)}\big)$. Here, each factor argument $\mathbf{x}^{(s)}$ comprises certain random variables $\mathbf{x}_q$ (each $\mathbf{x}_q$ can appear in several $\mathbf{x}^{(s)}$). Factor nodes and variable nodes are typically depicted by circles and boxes, respectively. The joint probability density function (PDF) represented by the factor graph reads $p(\mathbf{x}) \propto \prod_{s\in\mathcal{F}} f_s\big(\mathbf{x}^{(s)}\big)$ where $\propto$ indicates equality up to a constant factor.

Belief propagation (BP) [30], also known as the sum-product algorithm can compute marginal PDFs $p(\mathbf{x}_q)$, $q \in \mathcal{Q}$ efficiently. BP performs local operations called "messages" on the edges of the factor graph. There are two types of messages: (i) messages passed from variable nodes to factor nodes given by

$$\phi_{\mathbf{x}_q \to f_s}(\mathbf{x}_q) = \prod_{a\in\mathcal{N}_{\mathcal{F}}(q)\setminus s} \phi_{f_a\to\mathbf{x}_q}(\mathbf{x}_q)$$

and (ii) messages passed from factor nodes to variable nodes defined as

$$\phi_{f_s\to\mathbf{x}_q}(\mathbf{x}_q) = \sum_{\mathbf{x}^{(s)}\setminus\mathbf{x}_q} f_s\big(\mathbf{x}^{(s)}\big) \prod_{m\in\mathcal{N}_{\mathcal{Q}}(s)\setminus q} \phi_{\mathbf{x}_m\to f_s}(\mathbf{x}_m).$$

Here, $\mathcal{N}_{\mathcal{Q}}(\cdot) \subseteq \mathcal{Q}$ and $\mathcal{N}_{\mathcal{F}}(\cdot) \subseteq \mathcal{F}$ denote the set of neighboring variable and factor nodes, respectively. After message passing is completed, one can subsequently obtain marginal PDFs $p(\mathbf{x}_q)$ as the product of all incoming messages from the neighboring factors, i.e., $p(\mathbf{x}_q) \propto \prod_{s\in\mathcal{N}_{\mathcal{F}}(q)} \phi_{f_s\to\mathbf{x}_q}(\mathbf{x}_q)$. In factor graph with loops, BP is applied in an iterative manner. It can then only provide approximations of marginal posterior pdfs $p(\mathbf{x}_q)$.

B. Graph Neural Networks

Graph Neural Networks (GNNs) [39] extend neural networks to graph-structured data. We consider message passing neural networks (MPNNs) [40] which are a variant of GNNs that generalizes graph convolutional networks [41] and provides a message passing mechanism similar to BP. A MPNN is defined on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{E}$ induces the sets of neighbors $\mathcal{N}(i) = \{j \in \mathcal{V} \big| (i,j) \in \mathcal{E}\}$.

Each node $i \in \mathcal{V}$ is associated with a vector $\mathbf{h}_i$ called node embedding. At message passing iteration $l \in \{1, \dots, L\}$, the following operations are performed for each node $i \in \mathcal{V}$ in parallel. First, messages are exchanged with neighboring nodes $j \in \mathcal{N}(i)$. In particular, the GNN message sent from node $i \in \mathcal{V}$ to its neighboring node $j \in \mathcal{N}(i)$ is given by

$$\mathbf{m}_{i\to j}^{(l)} = g_{i\to j}\big(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}, \mathbf{a}_{i\to j}\big).$$

In addition, the node embedding $\mathbf{h}_i^{(l)}$ is updated by incorporating the sum of received messages $\mathbf{m}_{j\to i}^{(l)}$, $j \in \mathcal{N}(i)$, i.e.,

$$\mathbf{h}_i^{(l+1)} = g_i\Big(\mathbf{h}_i^{(l)}, \sum_{j\in\mathcal{N}(i)} \mathbf{m}_{j\to i}^{(l)}, \mathbf{a}_i\Big).$$

Here, $g_i(\cdot), i \in \mathcal{V}$ and $g_{i\to j}(\cdot), (i,j) \in \mathcal{E}$ denote the node and edge networks, respectively. Furthermore, $\mathbf{a}_i$ and $\mathbf{a}_{i\to j}$ denote node and edge attributes.

III. Review of BP-based Multi-Object Tracking

The proposed NEBP approach is based on BP-based MOT introduced in [25]. The statistical model used by BP-based MOT is reviewed next.

A. Object States

At each time frame k, an object detector $g_{\text{det}}(\cdot)$ extracts J_k measurements $\mathbf{z}_k \triangleq [\mathbf{z}_{k,1}^{\mathrm{T}} \cdots \mathbf{z}_{k,J_k}^{\mathrm{T}}]^{\mathrm{T}}$ from raw sensor data $\mathcal{Z}_k$, i.e., $\mathbf{z}_k = g_{\text{det}}(\mathcal{Z}_k)$. All measurements extracted up to time frame k are denoted as $\mathbf{z}_{1:k} \triangleq [\mathbf{z}_1^{\mathrm{T}} \cdots \mathbf{z}_k^{\mathrm{T}}]^{\mathrm{T}}$. Since the number objects is unknown, potential object (PO) states are introduced. The number of PO states N_k is the maximum possible number of objects that have generated a measurement up to time frame k. At time frame k, the existence of a PO $n \in \{1, \cdots, N_k\}$ is modeled by a binary random existence variable $r_{k,n} \in \{0, 1\}$, i.e., PO n exists if and only if $r_{k,n} = 1$. The state of PO n is modeled by the random vector $\mathbf{x}_{k,n}$. The augmented PO state vector is denoted by $\mathbf{y}_{k,n} \triangleq [\mathbf{x}_{k,n}^{\mathrm{T}}\ r_{k,n}]^{\mathrm{T}}$ and the joint PO state vector by $\mathbf{y}_k \triangleq [\mathbf{y}_{k,1}^{\mathrm{T}} \cdots \mathbf{y}_{k,N_k}^{\mathrm{T}}]^{\mathrm{T}}$. There are two types of POs:

- *New POs* denoted $\overline{\mathbf{y}}_{k,j} = [\overline{\mathbf{x}}_{k,j}^{\mathrm{T}}\ \overline{r}_{k,j}]^{\mathrm{T}}$, $j \in \{1, \cdots, J_k\}$ represent objects that at time frame k generated a measurement for the first time. Each measurement $\mathbf{z}_{k,j}, j \in \{1, \cdots, J_k\}$ introduces a new PO j with state $\overline{\mathbf{y}}_{k,j}$.
- *Legacy POs* denoted $\underline{\mathbf{y}}_{k,i} = [\underline{\mathbf{x}}_{k,i}^{\mathrm{T}}\ \underline{r}_{k,i}]^{\mathrm{T}}$, $i = \{1, \cdots, I_k\}$ represent objects that have generated a measurement for the first time at a previous time frame $k' < k$.

New POs become legacy POs when the measurements of the next time frame are considered. Thus, the number of legacy POs at time frame k is $I_k = I_{k-1} + J_{k-1} = N_{k-1}$ and the total number of POs is $N_k = I_k + J_k$. We further denote the joint new PO state by $\overline{\mathbf{y}}_k \triangleq [\overline{\mathbf{y}}_{k,1}^{\mathrm{T}} \cdots \overline{\mathbf{y}}_{k,J_k}^{\mathrm{T}}]^{\mathrm{T}}$ and the joint legacy PO state by $\underline{\mathbf{y}}_k \triangleq [\underline{\mathbf{y}}_{k,1}^{\mathrm{T}} \cdots \underline{\mathbf{y}}_{k,I_k}^{\mathrm{T}}]^{\mathrm{T}}$, i.e., $\mathbf{y}_k = [\underline{\mathbf{y}}_k^{\mathrm{T}}\ \overline{\mathbf{y}}_k^{\mathrm{T}}]^{\mathrm{T}}$.

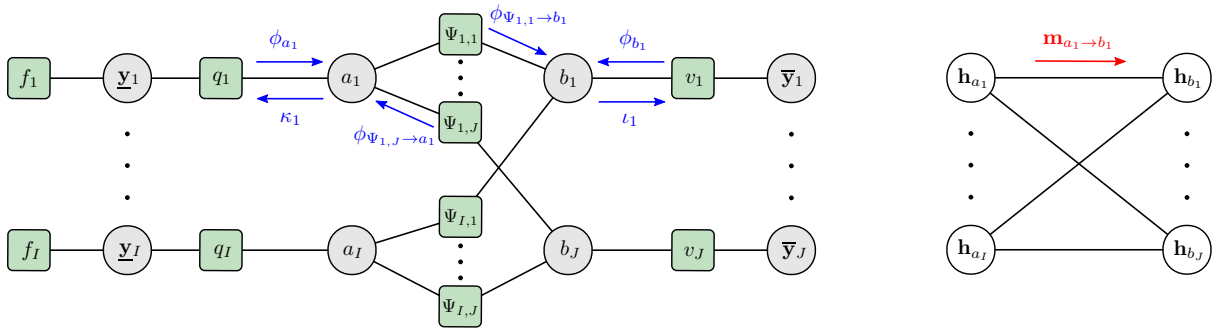

Fig. 1: Factor graph for MOT (a) and corresponding graph neural network (GNN) (b) for a single time frame k. In (a), BP messages that correspond to data association (DA) are shown in blue color. These messages are enhanced by the proposed NEBP approach are shown; (b) shows the corresponding GNN messages. The time index k is omitted.

B. Measurement Model

The origin of measurements $\mathbf{z}_{k,j}$, $j \in \{1, \cdots, J_k\}$ is unknown. A measurement can originate from a PO or can be a false alarm. Furthermore, a PO may also not generate any measurements (missed detection). With the assumption that a PO can generate at most one measurement and a measurement is originated from at most one PO, we model data association uncertainty as follows [25]. The PO-measurement association at time frame k can be described by an "object-oriented" DA vector $\mathbf{a}_k = [a_{k,1} \cdots a_{k,I_k}]^{\mathrm{T}}$. Here, the association variable $a_{k,i} = j \in \{1, \cdots, J_k\}$ indicates that legacy PO i generated measurement j and $a_{k,i} = 0$ indicated that legacy PO i did not generate any measurement at time k. Following [42], we also introduce the "measurement-oriented" DA vector $\mathbf{b}_k = [b_{k,1} \cdots b_{k,J_k}]^{\mathrm{T}}$ with $b_{k,j} = i \in \{1, \cdots, I_k\}$ if measurement j was generated by legacy PO i, or $b_{k,j} = 0$ if measurement j was not generated by any legacy PO. Note that there is a one-to-one mapping between $\mathbf{a}_k$ and $\mathbf{b}_k$ and vice versa. Introducing $\mathbf{b}_k$ in addition to $\mathbf{a}_k$ makes it possible to develop scalable MOT [25]. Finally, we establish the notation $\mathbf{a}_{1:k} \triangleq [\mathbf{a}_1^{\mathrm{T}} \cdots \mathbf{a}_k^{\mathrm{T}}]^{\mathrm{T}}$ and $\mathbf{b}_{1:k} \triangleq [\mathbf{b}_1^{\mathrm{T}} \cdots \mathbf{b}_k^{\mathrm{T}}]^{\mathrm{T}}$.

If legacy PO i exists, it generates a measurement (i.e. $a_{k,i} = j \in \{1, \cdots, J_k\}$) with probability p_{d}. Furthermore, the probability that it also exists at the next time frame $k+1$ is p_{s}. The number of false alarms is modeled by a Poisson distribution with mean μ_{fa} and false alarm measurements are independent and identically distributed according to $f_{\mathrm{fa}}(\mathbf{z}_{k,j})$. Before the measurements $\{\mathbf{z}_{k,j}\}_{j=1}^{J_k}$ are observed, the number of new POs is unknown. The number of newly detected objects is Poisson distributed with mean μ_{n}, while the states of newly detected objects are a priori independent and identically distributed according to $f_{\mathrm{n}}(\overline{\mathbf{x}}_{k,j})$. Following the assumptions presented in [25, Sec. VIII-A], the joint posterior PDF $f(\mathbf{y}_{1:k}, \mathbf{a}_{1:k}, \mathbf{b}_{1:k} | \mathbf{z}_{1:k})$ can be derived [25, Sec. VIII-G]. The factorization of this joint posterior pdf is visualized by the factor graph shown in Fig. 1a. Note that legacy POs are connected to object-oriented association variables and new POs are connected to measurement-oriented association variables.

C. Object Declaration and State Estimation

In the Bayesian setting, declaration of object existence and object state estimation are based on the marginal existence probabilities $p(r_{k,n} = 1 | \mathbf{z}_{1:k})$ and the conditional PDFs $f(\mathbf{x}_{k,n} | r_{k,n} = 1, \mathbf{z}_{1:k})$. In particular, declaration of object existence is performed by comparing $p(r_{k,n} = 1 | \mathbf{z}_{1:k})$ to a threshold T_{dec}. In addition, for objects n that are declared to exist, an estimate of $\mathbf{x}_{k,n}$ is provided by the minimum mean-square error (MMSE) estimator

$$\hat{\mathbf{x}}_{k,n}^{\mathrm{MMSE}} = \int \mathbf{x}_{k,n} f(\mathbf{x}_{k,n} | r_{k,n} = 1, \mathbf{z}_{1:k}) \mathrm{d}\mathbf{x}_{k,n}.$$

Note that declaration of object existence is based on the existence probability $p(r_{k,n} = 1 | \mathbf{z}_{1:k}) = \int f(\mathbf{x}_{k,n}, r_{k,n} = 1 | \mathbf{z}_{1:k}) \mathrm{d}\mathbf{x}_{k,n}$. In addition, object state estimation makes use of

$$f(\mathbf{x}_{k,n} | r_{k,n} = 1, \mathbf{z}_{1:k}) = \frac{f(\mathbf{y}_{k,n} | \mathbf{z}_{1:k})}{p(r_{k,n} = 1 | \mathbf{z}_{1:k})}.$$

Thus, both tasks rely on the calculation of marginal posterior PDFs $f(\mathbf{y}_{k,n} | \mathbf{z}_{1:k}) \triangleq f(\mathbf{x}_{k,n}, r_{k,n} | \mathbf{z}_{1:k})$. By applying BP following [25, Sec. VIII-IX], accurate approximations (a.k.a. "beliefs") $\tilde{f}(\mathbf{y}_{k,n}) \approx f(\mathbf{y}_{k,n} | \mathbf{z}_{1:k})$ of marginal posterior PDFs can be calculated efficiently. For future reference we introduce the notation $\hat{r}_{k,n} = p(r_{k,n} = 1 | \mathbf{z}_{1:k})$.

Note that since we introduce a new PO for each measurement, the number of POs grows with time k. Therefore, legacy and new POs whose approximate existence probabilities are below a threshold T_{pru} are pruned, i.e., removed from the state space.

IV. NEBP-based Multi-Object Tracking

To further improve the performance of BP-based MOT, we augment the factor graph in Fig. 1a by a GNN. The GNN uses features extracted from previous estimates, measurements, and raw sensor information as an input. Since we limit the following discussion to a single time frame, we will omit the time index k.

A. Feature Extraction

First, we discuss how features are learned from raw sensor data for legacy POs and measurements. We consider motion and shape features. The motion features for legacy PO i and measurement j are computed as $\mathbf{h}_{a_i,\text{motion}} = g_{\text{motion},1}(\hat{\underline{\mathbf{x}}}_i^-, \hat{\underline{r}}_i^-)$ and $\mathbf{h}_{b_j,\text{motion}} = g_{\text{motion},2}(\mathbf{z}_j)$, respectively. Here, $g_{\text{motion},1}(\cdot)$ as well as $g_{\text{motion},2}(\cdot)$ are neural networks. In addition, $\hat{\underline{\mathbf{x}}}_i^-$ and $\hat{\underline{r}}_i^-$ are the approximate MMSE state estimate and existence probability of legacy PO i at the previous time frame. Similarly, the shape features, denoted by $\mathbf{h}_{a_i,\text{shape}}$ and $\mathbf{h}_{b_j,\text{shape}}$, are extracted from raw sensor data $\mathcal{Z}^-$ and $\mathcal{Z}$ at previous and current time, respectively, i.e., $\mathbf{h}_{a_i,\text{shape}} = g_{\text{shape},1}(\mathcal{Z}^-, \hat{\underline{\mathbf{x}}}_i^-)$ and $\mathbf{h}_{b_j,\text{shape}} = g_{\text{shape},2}(\mathcal{Z}, \mathbf{z}_j)$. Here, $g_{\text{shape},1}(\cdot)$ as well as $g_{\text{shape},2}(\cdot)$ are again neural networks. We will discuss one particular instance of shape feature extraction in Sec. V-A.

B. The Proposed Message Passing Algorithm

For neural enhanced DA, we introduce a GNN that matches the topology of the DA section of the factor graph in Fig. 1a. The resulting GNN is shown in Fig. 1b. In addition to the output of the detector, the GNN also uses raw sensor information as an input. The goal is to use this additional information to reject false alarm measurements and obtain improved DA probabilities by enhancing BP messages with the output of the GNN. NEBP for MOT consists of the following steps:

1) Conventional BP: First, conventional BP-based MOT is run until convergence. This results in the BP messages $\phi_{a_i} = [\phi_{a_i}(0) \cdots \phi_{a_i}(J)]^{\mathrm{T}} \in \mathbb{R}^{J+1}$, $\phi_{b_j} = [\phi_{b_j}(0) \cdots \phi_{b_j}(I)]^{\mathrm{T}} \in \mathbb{R}^{I+1}$, $\phi_{\Psi_{i,j}\to b_j} \in \mathbb{R}$, and $\phi_{\Psi_{i,j}\to a_i} \in \mathbb{R}$ (cf. [25, Sec. IX-A1–IX-A3]).

2) GNN Messages: Next, GNN message passing is executed iteratively. In particular, at iteration $l \in \{1, \ldots, L\}$ the following operations are performed:

$$\mathbf{m}_{a_i\to b_j}^{(l)} = g_{\text{e}}\Big(\mathbf{h}_{a_i}^{(l)}, \mathbf{h}_{b_j}^{(l)}, \phi_{a_i}(j), \phi_{\Psi_{i,j}\to b_j}\Big) \tag{1}$$

$$\mathbf{m}_{b_j\to a_i}^{(l)} = g_{\text{e}}\Big(\mathbf{h}_{a_i}^{(l)}, \mathbf{h}_{b_j}^{(l)}, \phi_{a_i}(j), \phi_{\Psi_{i,j}\to a_i}\Big)$$

$$\mathbf{h}_{a_i}^{(l+1)} = g_{\text{n}}\bigg(\mathbf{h}_{a_i}^{(l)}, \sum_{j\in\mathcal{N}(i)} \mathbf{m}_{b_j\to a_i}^{(l)}, \phi_{a_i}(0)\bigg)$$

$$\mathbf{h}_{b_j}^{(l+1)} = g_{\text{n}}\bigg(\mathbf{h}_{b_j}^{(l)}, \sum_{j\in\mathcal{N}(i)} \mathbf{m}_{a_i\to b_j}^{(l)}, \phi_{b_j}(0)\bigg). \tag{2}$$

Here, $g_{\text{e}}(\cdot)$ is the edge neural network and $g_{\text{n}}(\cdot)$ is the node neural network. The edge neural network $g_{\text{e}}(\cdot)$ provides messages passed along the edges of the GNN.

The node embeddings are initialized as the concatenation of respective motion and shape features, i.e., $\mathbf{h}_{a_i}^{(1)} = [\mathbf{h}_{a_i,\text{motion}}^{\mathrm{T}}\ \mathbf{h}_{a_i,\text{shape}}^{\mathrm{T}}]^{\mathrm{T}}$ and $\mathbf{h}_{b_j}^{(1)} = [\mathbf{h}_{b_j,\text{motion}}^{\mathrm{T}}\ \mathbf{h}_{b_j,\text{shape}}^{\mathrm{T}}]^{\mathrm{T}}$. Finally, for each $j \in \{1, \cdots, J\}$, the correction factors $\beta_j = g_{\text{r}}(\mathbf{h}_{b_j}^{(L)}) \in (0, 1]$ and $\gamma_i(j) = g_{\text{a}}\big(\mathbf{m}_{b_j\to a_i}^{(L)}\big) \in \mathbb{R}$ are computed based on the two additional neural networks $g_{\text{r}}(\cdot)$ and $g_{\text{a}}(\cdot)$. As discussed next, these correction factors provided by the GNN are used to implement false alarm rejection and object shape association, respectively.

3) NEBP Messages: After computing (1)–(2) for L iterations, neural enhanced message passing is performed as follows. First, neural enhanced versions $\tilde{\phi}_{a_i}$ of the messages ϕ_{a_i} are obtained by computing

$$\tilde{\phi}_{a_i}(j) = \beta_j \bar{\phi}_{a_i}(j) + \text{ReLU}\big(\gamma_i(j)\big), \quad j \in \{1, \cdots, J\} \tag{3}$$

and setting $\tilde{\phi}_{a_i}(0) = \phi_{a_i}(0)$. Here, $\text{ReLU}(\cdot)$ is a rectified linear unit and $\bar{\phi}_{a_i}$ is a normalized[1] version of ϕ_{a_i} (cf. [25, Sec. IX-A2]), i.e.,

$$\bar{\phi}_{a_i}(j) = \frac{\phi_{a_i}(j)}{\sum_{j'=0}^{J} \phi_{a_i}(j')}, \quad j \in \{1, \cdots, J\}.$$

Note that $\phi_{a_i}(j)$, $j \in \{1, \ldots, J\}$ represents the likelihood that the legacy PO i is associated to measurement j [25]. Consequently, the $\text{ReLU}\big(\gamma_i(j)\big) > 0$ term in (3) provided by the GNN implements object shape association, i.e., the likelihood that the legacy PO i is associated to the measurement j is increased if the shape features extracted for legacy POs resembles the shape features extracted for measurements.

Next, neural enhanced versions $\tilde{\phi}_{b_j}$ of the messages ϕ_{b_j} are obtained by computing

$$\tilde{\phi}_{b_j}(0) = \beta_j\big(\phi_{b_j}(0) - 1\big) + 1 \tag{4}$$

and setting $\tilde{\phi}_{b_j}(i) = \phi_{b_j}(i), i \in \{1, \cdots, I\}$. We recall that $\phi_{b_j}(0)$ is given by (cf. [25, Sec. IX-A2])

$$\phi_{b_j}(0) = \frac{\mu_{\text{n}}}{\mu_{\text{fa}} f_{\text{fa}}(\mathbf{z}_j)} \int f_{\text{n}}(\overline{\mathbf{x}}_j)\, f(\mathbf{z}_j|\overline{\mathbf{x}}_j)\, \mathrm{d}\overline{\mathbf{x}}_j + 1\,.$$

The scalar $\beta_j \in (0, 1)$ in (3) and (4) provided by the GNN implements false alarm rejection. In particular, $\beta_j < 1$ is equal to the local increase of the false alarm distribution given by $\tilde{f}_{\text{fa}}(\mathbf{z}_j) = \frac{1}{\beta_j} f_{\text{fa}}(\mathbf{z}_j)$. This local increase of the false alarm distribution makes it less likely that the measurement $\mathbf{z}_j$ is associated to a legacy PO and reduces the existence probability of the new PO introduced for the measurement $\mathbf{z}_j$.

4) Belief Calculation: Finally, conventional BP-based MOT is again run until convergence by replacing ϕ_{a_i} with its neural enhanced counterpart $\tilde{\phi}_{a_i}$. This results in the enhanced output messages $\tilde{\kappa}_i = [\tilde{\kappa}_i(0) \cdots \tilde{\kappa}_i(J)]^{\mathrm{T}} \in \mathbb{R}^{J+1}$ and $\tilde{\iota}_j = [\tilde{\iota}_i(0) \cdots \tilde{\iota}_i(I)]^{\mathrm{T}} \in \mathbb{R}^{I+1}$ (cf. Fig. 1). After performing the normalization

$$\tilde{\kappa}_i'(j) = \frac{\tilde{\phi}_{a_i}(j)}{\phi_{a_i}(j)} \tilde{\kappa}_i(j), \quad j \in \{0, \cdots, J\}$$

the resulting messages $\tilde{\kappa}_i'$ are used for the calculation of legacy PO beliefs $\tilde{f}(\underline{\mathbf{y}}_i)$, $i \in \{1, \cdots, I\}$ (cf. [25, Sec. IX-A4–IX-A6]). Similarly, the enhanced messages $\tilde{\iota}_j$ are directly used for the calculation of new PO beliefs $\tilde{f}(\overline{\mathbf{y}}_j), j \in \{1, \cdots, J\}$.

[1] Multiplying BP messages by a constant factor does not alter the resulting beliefs [30].

C. The Loss Function

For supervised learning, it is assumed that ground truth object tracks are available in the training set. Ground truth object tracks consist of a sequence of object positions and object identities (IDs). During the training of the GNN, the parameters of all neural networks are updated through backpropagation, which computes the gradient of the loss function. The loss function has the form $\mathcal{L} = \mathcal{L}_{\mathrm{r}} + \mathcal{L}_{\mathrm{a}}$. Here, the two contributions $\mathcal{L}_{\mathrm{r}}$ and $\mathcal{L}_{\mathrm{a}}$ establish false alarm rejection and object shape association, respectively.

False alarm rejection, introduces the binary cross-entropy loss [43, Chapter 4.3]

$$\mathcal{L}_{\mathrm{r}} = \frac{-1}{J} \sum_{j=1}^{J} \beta_j^{\mathrm{gt}} \ln(\beta_j) + \epsilon(1 - \beta_j^{\mathrm{gt}}) \ln(1 - \beta_j) \qquad (5)$$

where $\beta_j^{\mathrm{gt}} \in \{0, 1\}$ is the pseudo ground truth label for each measurement and $\epsilon \in \mathbb{R}^+$ is a tuning parameter. β_j^{gt} is 1 if the distance between the measurement and any ground truth position is not larger than T_{dist} and 0 otherwise.

The tuning parameter $\epsilon \in \mathbb{R}^+$ is motivated as follows. Since missing an object is typically more severe than producing a false alarm, object detectors often output many detections and produce more false alarm measurements than true measurements. The tuning parameter $\epsilon \in \mathbb{R}^+$ addresses this imbalance problem which is well studied in the context of learning-based binary classification [44].

Since $\tilde{\phi}_{a_i}(j)$ in (3) represents the likelihood that the legacy PO i is associated to the measurement j, ideally $\mathrm{ReLU}\big(\gamma_i(j)\big)$ is large if PO i is associated to the measurement j, and is equal to zero if they are not associated. Thus, object shape association introduces the following binary cross-entropy loss

$$\mathcal{L}_{\mathrm{a}} = \frac{-1}{IJ} \sum_{i=1}^{I} \sum_{j=1}^{J} \gamma_i^{\mathrm{gt}}(j) \ln\big(\sigma(\gamma_i(j))\big) + \big(1 - \gamma_i^{\mathrm{gt}}(j)\big) \ln\big(1 - \sigma(\gamma_i(j))\big) \qquad (6)$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the sigmoid function and $\gamma_i^{\mathrm{gt}} = [\gamma_i^{\mathrm{gt}}(1) \cdots \gamma_i^{\mathrm{gt}}(J)]^{\mathrm{T}} \in \{0, 1\}^J$ is the pseudo ground truth association vector of legacy PO $i \in \{1, \ldots, I\}$. In each pseudo ground truth association vector γ_i^{gt}, at most one element is equal to one and all the other elements are equal to zero.

Note that in (6), we do not apply the ReLU to the $\gamma_i(j)$, since this would result in the gradients $\partial \mathcal{L}_{\mathrm{a}} / \partial \gamma_i(j)$ to be zero for negative values of $\gamma_i(j)$. It was observed that performing backpropagation by also making use of the gradients related to the negative values of $\gamma_i(j)$, leads to a more efficient training of the GNN. At each time frame, pseudo ground truth association vectors are constructed from measurements and ground truth object tracks based on the following rules:

- *Get Measurement IDs:* Compute the Euclidean distance between all ground truth positions and measurements and run the Hungarian algorithm [1] to find the best association between ground truth positions and measurements. All measurements that have been associated with a ground truth position and have a distance to that ground truth position that is smaller than T_{dist} inherit the ID of the ground truth position. All other measurements do not have an ID.
- *Update Legacy PO IDs:* Legacy POs inherit the ID from the previous time frame. If a legacy PO with ID has a distance not larger than T_{dist} to a ground truth position with the same ID, it keeps its ID. The for a legacy PO $i \in \{1, \ldots, I\}$ that has the same ID as measurement $j \in \{1, \ldots, J\}$, the entry $\gamma_i^{\mathrm{gt}}(j)$ is set to one. All other entries $\gamma_i^{\mathrm{gt}}(j)$, $i \in \{1, \ldots, I\}$, $j \in \{1, \ldots, J\}$ are set to zero.
- *Introduce New PO IDs:* For any measurement $j \in \{1, \ldots, J\}$ with an ID that does not share its ID with a legacy object, the corresponding new PO inherits the ground truth ID from the measurement. All other new POs do not have an ID.

V. Experimental Results

We present results in an urban autonomous driving scenario to validate our method. In particular, we use data provided by a LiDAR sensor mounted on the roof of an autonomous vehicle. This data is part of the *nuScenes* dataset [35].

A. System Model and Implementation Details

The nuScenes dataset consists of 1000 autonomous driving scenes and seven object classes. We use the official split of the dataset, where there are 700 scenes for training, 150 for validation, and 150 for testing. Each scene has a length of roughly 20 seconds and contains keyframes (frames with ground truth object annotations) sampled at 2Hz. Object detections extracted by the *CenterPoint* [7] detector are used as measurements, which are then preprocessed using non-maximum suppression (NMS) [45]. Each measurement has a class index and the proposed MOT method is performed for each class individually.

We define the states of POs as $\mathbf{x}_{k,n} \in \mathbb{R}^4$ which include their 2D position and 2D velocity. The measurements $\mathbf{z}_{k,j} \in \mathbb{R}^5$ consist of the 2D position and velocity obtained as well as a score $0 < s_{k,j} \leq 1$ from the object detector. The dynamics of objects are modeled by a constant-velocity model [46]. The region of interest (ROI) is given by $[x_e - 54, x_e + 54] \times [y_e - 54, y_e + 54]$, where (x_e, y_e) is the 2D position of the autonomous vehicle. The prior PDF of false alarms $f_{\mathrm{fa}}(\cdot)$ and newly detected objects $f_{\mathrm{n}}(\cdot)$ are uniform over the ROI. All other parameters used in the system model are estimated from the training data. The thresholds for object declaration was set to $T_{\mathrm{dec}} = 0.5$ for legacy POs and to a class dependent value for new POs. The pruning threshold was set to $T_{\mathrm{pru}} = 10^{-3}$.

The neural networks $g_{\mathrm{e}}(\cdot)$, $g_{\mathrm{n}}(\cdot)$, $g_{\mathrm{a}}(\cdot)$, $g_{\mathrm{motion}}(\cdot) \triangleq g_{\mathrm{motion},1}(\cdot) = g_{\mathrm{motion},2}(\cdot)$ are multi-layer perceptrons (MLPs) with a single hidden layer and leaky ReLU activation function. Furthermore, $g_{\mathrm{r}}(\cdot)$ is a single-hidden-layer MLP with sigmoid activation at the output layer. Finally, $g_{\mathrm{shape}}(\cdot) \triangleq g_{\mathrm{shape},1}(\cdot) = g_{\mathrm{shape},2}(\cdot)$ consists of two convolutional layers followed by a single-hidden-layer MLP. At each time frame, we use the

Methods	Modalities	AMOTA ↑	IDS ↓	Frag ↓
CenterPoint [7]	LiDAR	0.665	562	424
Chiu et al. [17]	LiDAR+Camera	0.687	-	-
Zaech et al. [18]	LiDAR	0.693	262	332
BP	LiDAR	0.698	**161**	**250**
NEBP (proposed)	LiDAR	**0.708**	172	271

TABLE I: Performance results on nuScenes validation set. "-" indicates that the metric is not reported.

output $\mathcal{Z}$ of *VoxelNet* [4] to extract shape features as discussed in Section IV-A. The used VoxelNet has been pre-trained by the CenterPoint method [7]. Its parameters remain unchanged during the training of the proposed NEBP method. NEBP training is performed by employing the *Adam optimizer* [47]. The number of GNN iteration is $L = 3$. The batch size was set to 1, the learning rate to 10^{-4}, and the number of training epochs to 8. The tuning parameter ϵ in (5) was set to 0.1 and the threshold T_{dist} for the pseudo ground truth extraction discussed in Section IV-C was set to 2 meters.

B. Performance Evaluation

We use the widely used CLEAR metrics [48] that include the number of false positives (FP), identity switches (IDS) and fragments (Frag). In addition, we also consider the average multi-object tracking accuracy (AMOTA) metric proposed in [15]. Note that the AMOTA is also the primary metric used for the nuScenes tracking challenge [35].

Evaluation of the AMOTA requires a score for each estimated object. It was observed that a high AMOTA performance is obtained by calculating the estimated object score as a combination of existence probability and measurement score. In particular, for legacy PO i the estimated object score is calculated as

$$\underline{s}_i = \tilde{p}(\underline{r}_i = 1) + \sum_{j=1}^{J} \tilde{p}_{a_i}(j)\, s_j,$$

where $\tilde{p}_{a_i}(j) \propto \phi_{a_i}(j)\kappa_i(j)$ are approximate marginal association probabilities [25]. Finally, for new PO j the estimated object score is given by $\overline{s}_j = \tilde{p}(\overline{r}_j = 1) + s_j$.

For a fair comparison, we use state-of-the-art reference methods that all rely on the CenterPoint detector [7]. In particular, BP refers to the traditional BP-based MOT method [25] that uses object detections provided by the CenterPoint detector as measurements. Furthermore, the tracking method proposed in [7] uses a heuristic to create tracks and a greedy matching algorithm based on the Euclidean distance to associate CenterPoint object detections to tracks. Chiu et al. [17] follows a similar strategy but makes use of a hybrid distance that combines the Mahalanobis distance and the so-called deep feature distance. Finally, the method introduced by Zaech et al. [18] utilizes a network flow formulation and transforms the DA problem into a classification problem.

In Table I, it can be seen that the proposed NEBP approach outperforms all reference methods in terms of AMOTA performance. Furthermore, it can be observed, that BP and NEBP achieve a much lower IDS and Frag metric compared to the reference methods. This is because both BP and NEBP make use of a statistical model to determine the initialization and termination of tracks [25] which is more robust compared to the heuristic track management performed by other reference methods. The improved AMOTA performance of NEBP over BP comes at the cost of a slightly increased IDS and Frag.

TABLE II shows the AMOTA performance as well as number of FP for the bicycle and motorcycle class. To ensure a fair comparison, all the FP values are evaluated for the same percentage of true positives referred to as "recall". In particular, for each class, the recall that leads to the largest multi-object tracking accuracy [48] for NEBP was used.

Method	bicycle		motorcycle	
	AMOTA ↑	FP ↓	AMOTA ↑	FP ↓
CenterPoint [7]	0.458	390	0.615	792
BP	0.505	168	0.725	349
NEBP (proposed)	**0.550**	**120**	**0.739**	**208**

TABLE II: Evaluation results on nuScenes validation set in terms of AMOTA and FP for the bicycle and motorcycle class.

For the considered two classes, NEBP yields the largest improvement in terms of AMOTA performance over BP. Compared to BP, the proposed NEBP method also has a reduced number of FP. In conclusion, false alarm rejection and object shape association introduced by the proposed NEBP method can make effective use of features learned from raw sensor data and substantially improve MOT performance.

VI. Final Remarks

In this paper, we present a NEBP method for MOT that enhances probabilistic data association by features learned from raw sensor data. A GNN is introduced that matches the topology of the factor graph for model-based data association. In addition to the preprocessed measurements employed by BP, the GNN also makes use of object features learned from raw sensor data. For false alarm rejection, the GNN identifies which measurements are likely false alarms. For object shape association, the GNN computes improved association probabilities. The proposed method can improve the object declaration and state estimation performance of BP while preserving its low computational complexity. Performance evaluation based on the nuScenes autonomous driving dataset demonstrated state-of-the-art object tracking performance.

ACKNOWLEDGEMENT

This work was supported by the National Science Foundation (NSF) under CAREER Award No. 2146261.

REFERENCES

[1] Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and Data Fusion: A Handbook of Algorithms*. Storrs, CT: Yaakov Bar-Shalom, 2011.

[2] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, "DESIRE: Distant future prediction in dynamic scenes with interacting agents," in *Proc. CVPR-17*, Jul. 2017, pp. 336–345.

[3] W. Zeng, W. Luo, S. Suo, A. Sadat, B. Yang, S. Casas, and R. Urtasun, "End-to-end interpretable neural motion planner," in *Proc. CVPR-19*, Jun. 2019, pp. 8660–8669.

[4] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3D object detection," in *Proc. CVPR-18*, Jun. 2018, pp. 4490–4499.

[5] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. CVPR-19*, Jun. 2019, pp. 12 697–12 705.

[6] B. Zhu, Z. Jiang, X. Zhou, Z. Li, and G. Yu, "Class-balanced grouping and sampling for point cloud 3D object detection," *arXiv:1908.09492*, 2019.

[7] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3D object detection and tracking," in *Proc. CVPR-21*, Jun. 2021, pp. 11 784–11 793.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. NeurIPS-15*, vol. 28, Dec. 2015, pp. 91–99.

[9] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *Proc. CVPR-19*, Jun. 2019, pp. 770–779.

[10] A. Simonelli, S. R. Bulo, L. Porzi, M. López-Antequera, and P. Kontschieder, "Disentangling monocular 3D object detection," in *Proc. CVPR-19*, Jun. 2019, pp. 1991–1999.

[11] S. Bartoletti, Z. Liu, M. Z. Win, and A. Conti, "Device-free localization of multiple targets in cluttered environments," *IEEE Trans. Aerosp. Electron. Syst.*, 2022, to appear.

[12] F. Meyer and K. L. Gemba, "Probabilistic focalization for shallow water localization," *J. Acoust. Soc. Am.*, vol. 150, no. 2, pp. 1057–1066, 2021.

[13] M. Brambilla, D. Gaglione, G. Soldi, R. Mendrzik, G. Ferri, K. D. LePage, M. Nicoli, P. Willett, P. Braca, and M. Z. Win, "Cooperative localization and multitarget tracking in agent networks with the sum-product algorithm," *IEEE O. J. Signal Process.*, vol. 3, pp. 169–195, 2022.

[14] H. W. Kuhn, "The Hungarian method for the assignment problem," *NRL quarterly*, vol. 2, no. 1-2, pp. 83–97, Mar. 1955.

[15] X. Weng, J. Wang, D. Held, and K. Kitani, "3D multi-object tracking: A baseline and new evaluation metrics," in *Proc. IROS-20*, Oct. 2020, pp. 10 359–10 366.

[16] X. Weng, Y. Wang, Y. Man, and K. M. Kitani, "GNN3DMOT: Graph neural network for 3D multi-object tracking with 2D-3D multi-feature learning," in *Proc. CVPR-20*, Jun. 2020, pp. 6499–6508.

[17] H.-k. Chiu, J. Li, R. Ambrus, and J. Bohg, "Probabilistic 3D multi-modal, multi-object tracking for autonomous driving," in *Proc. ICRA-21*, Jun. 2021, pp. 14 227–14 233.

[18] J.-N. Zaech, D. Dai, A. Liniger, M. Danelljan, and L. Van Gool, "Learnable online graph representations for 3D multi-object tracking," *IEEE Robot. Autom. Lett.*, Jan. 2022.

[19] A. Rangesh, P. Maheshwari, M. Gebre, S. Mhatre, V. Ramezani, and M. M. Trivedi, "TrackMPNN: A message passing graph neural architecture for multi-object tracking," *arXiv preprint arXiv:2101.04206*, 2021.

[20] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, "Towards real-time multi-object tracking," in *Proc. ECCV-20*, Apr. 2020, pp. 107–122.

[21] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "FairMOT: On the fairness of detection and re-identification in multiple object tracking," *arXiv preprint arXiv:2004.01888*, 2021.

[22] M. Liang, B. Yang, W. Zeng, Y. Chen, R. Hu, S. Casas, and R. Urtasun, "PnPNet: End-to-end perception and prediction with tracking in the loop," in *Proc. CVPR-20*, Jun. 2020, pp. 11 553–11 562.

[23] J. L. Williams, "Marginal multi-Bernoulli filters: RFS derivation of MHT, JIPDA and association-based MeMBer," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 3, pp. 1664–1687, Jul. 2015.

[24] F. Meyer, P. Braca, P. Willett, and F. Hlawatsch, "A scalable algorithm for tracking an unknown number of targets using multiple sensors," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3478–3493, Jul. 2017.

[25] F. Meyer, T. Kropfreiter, J. L. Williams, R. A. Lau, F. Hlawatsch, P. Braca, and M. Z. Win, "Message passing algorithms for scalable multitarget tracking," *Proc. IEEE*, vol. 106, no. 2, pp. 221–259, Feb. 2018.

[26] Á. F. García-Fernández, J. L. Williams, K. Granström, and L. Svensson, "Poisson multi-Bernoulli mixture filter: Direct derivation and implementation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 4, pp. 1883–1901, Feb. 2018.

[27] S. Scheidegger, J. Benjaminsson, E. Rosenberg, A. Krishnan, and K. Granstrm, "Mono-camera 3D multi-object tracking using deep learning detections and PMBM filtering," in *Proc. IEEE IV-18*, Jun. 2018, pp. 433–440.

[28] S. Pang, D. Morris, and H. Radha, "3D multi-object tracking using random finite set-based multiple measurement models filtering (RFS-M^3) for autonomous vehicles," in *Proc. ICRA-21*, Jun. 2021, pp. 13 701–13 707.

[29] W. Zhang and F. Meyer, "Graph-based multiobject tracking with embedded particle flow," in *Proc. IEEE RadarConf-21*, May 2021, pp. 1–6.

[30] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[31] F. Meyer and M. Z. Win, "Scalable data association for extended object tracking," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 6, pp. 491–507, May 2020.

[32] F. Meyer and J. L. Williams, "Scalable detection and tracking of geometric extended objects," *IEEE Trans. Signal Process.*, vol. 69, pp. 6283–6298, 2021.

[33] V. G. Satorras and M. Welling, "Neural enhanced belief propagation on factor graphs," in *Proc. AISTATS-21*, Apr. 2021, pp. 685–693.

[34] M. Liang and F. Meyer, "Neural enhanced belief propagation for cooperative localization," in *Proc. IEEE SSP-21*, Jul. 2021, pp. 326–330.

[35] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuScenes: A multimodal dataset for autonomous driving," in *Proc. CVPR-20*, Jun. 2020, pp. 11 621–11 631.

[36] K. Granström, M. Baum, and S. Reuter, "Extended object tracking: Introduction, overview and applications," *J. Adv. Inf. Fusion*, vol. 12, no. 2, pp. 139–174, Dec. 2017.

[37] K. Granström, M. Fatemi, and L. Svensson, "Poisson multi-Bernoulli mixture conjugate prior for multiple extended target filtering," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 1, pp. 208–225, Feb. 2020.

[38] Ángel F. García-Fernández, J. L. Williams, L. Svensson, and Y. Xia, "A Poisson multi-Bernoulli mixture filter for coexisting point and extended targets," *IEEE Trans. Signal Process.*, vol. 69, pp. 2600–2610, 2021.

[39] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proc. INNS/IEEE IJCNN-05*, vol. 2, Aug. 2005, pp. 729–734.

[40] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. ICML-17*, Aug. 2017, pp. 1263–1272.

[41] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. ICLR-17*, Apr. 2017, pp. 1–14.

[42] J. L. Williams and R. Lau, "Approximate evaluation of marginal association probabilities with belief propagation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, no. 4, pp. 2942–2959, Oct. 2014.

[43] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[44] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas, "Imbalance problems in object detection: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3388–3415, Mar. 2020.

[45] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *Proc. ICPR-06*, vol. 3, Aug. 2006, pp. 850–855.

[46] Y. Bar-Shalom, T. Kirubarajan, and X.-R. Li, *Estimation with Applications to Tracking and Navigation*. New York, NY: Wiley, 2002.

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

[48] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the CLEAR MOT metrics," *EURASIP J. Image Video Proc.*, vol. 2008, pp. 1–10, May 2008.