Dynamic Semantic Occupancy Mapping Using 3D Scene Flow and Closed-Form Bayesian Inference

AISHWARYA UNNIKRISHNAN1, JOEY WILSON1, (Graduate Student Member, IEEE), LU GAN1, ANDREW CAPODIECI2, PARAMSOTHY JAYAKUMAR3, KIRA BARTON1, (Senior Member, IEEE), AND MAANI GHAFFARI1, (Member, IEEE)

1Department of Robotics, University of Michigan, Ann Arbor, MI 48109, USA
2Neya Systems Division, Applied Research Associates, Pittsburgh, PA 15086, USA
3U.S. Army CCDC Ground Vehicle Systems Center, Warren, MI 48397, USA

Corresponding author: Joey Wilson (wilsoniv@umich.edu)

ABSTRACT This paper reports on a dynamic semantic mapping framework that incorporates 3D scene flow measurements into a closed-form Bayesian inference model. Existence of dynamic objects in the environment can cause artifacts and traces in current mapping algorithms, leading to an inconsistent map posterior. We leverage state-of-the-art semantic segmentation and 3D flow estimation using deep learning to provide measurements for map inference. We develop a Bayesian model that propagates the scene with flow and infers a 3D continuous (i.e., can be queried at arbitrary resolution) semantic occupancy map outperforming its static counterpart. Extensive experiments using publicly available data sets show that the proposed framework improves over its predecessors and input measurements from deep neural networks consistently.

INDEX TERMS Bayesian inference, computer vision, mapping, semantic scene understanding.

I. INTRODUCTION

Mapping, localization and navigation are among the key capabilities of autonomous systems. For robots to navigate safely in complex and evolving environments, mapping can act as a unified framework that addresses multiple perception sub-tasks required for a higher-level scene understanding, such as occupancy/traversability estimation, object detection and tracking. While some research streams employ end-to-end deep neural networks for mapless navigation via imitation [1], [2], reinforcement [3], [4] or self-supervised learning [5], maps are still widely used for explicit reliability, interpretability, and predictability. In this work, we focus on the map inference problem instead of Simultaneous Localization and Mapping (SLAM), and aim at improving the inference performance in dynamic environments.

Map inference can aid robots in reasoning about areas that are currently occluded but previously observed (occlusion-awareness), or inferring the geometry and semantics of an unseen landmark near those that were previously observed (smoothing). In complex environments (e.g., driving scenarios), robots can recognize stationary cars and people, while consistently tracking moving vehicles and pedestrians.

Semantic mapping complements geometric modelling of a robot’s surroundings with semantic concepts, i.e., an understanding of what the environment means to the robot. With semantic mapping, these semantic concepts manifest as a representation of the environment, thus lending robots more resources for task planning and execution. The emergence of semantic mapping can be attributed to (i) the limitations of purely geometric maps, and (ii) the advancements in deep neural networks that allow semantic interpretation of raw sensory data [6].

In early semantic mapping works, semantics and geometries are modeled independently, where semantic labels are added on top of an existing geometric representation, such as point cloud model [7], surfel-based map [8], and voxel-based map [9]. As this field progresses, semantics and geometry have been modeled jointly and inferred in a unified framework [10], [11]. Gan et al. [11] proposed a unified semantic
mapping framework for closed-form Bayesian inference of the semantic map posterior. However, its underlying static world assumption limits its applications in real-world dynamic environments. In scenarios with dynamic objects, static mapping may provide less detailed or even inconsistent reconstruction due to obscured views. The novelty of this work is thus to propose a unified closed-form Bayesian inference framework which extends semantic mapping to dynamic environments.

To this end, we develop a scalable dynamic semantic mapping framework that combines motion and semantic information through closed-form Bayesian inference in a single pipeline as shown in Fig. 1. Spatio-temporal motion data and semantic labels are aggregated over past frames and neighbouring voxels. The aggregated motion is then used in a proposed Bayesian model to propagate the current scene and its semantic labels.

In particular, this work has the following contributions.

1) We propose a kernel method for scene flow aggregation and an efficient auto-regressive Bayesian model for scene propagation.
2) We extend the Bayesian Kernel Inference (BKI) semantic mapping framework [11] to dynamic scenes by incorporating motion information.
3) We introduce an evaluation methodology for dynamic semantic mapping using single and multi-view data.
4) The open-source software is publicly available at: https://github.com/UMich-CURLY/BKIDynamicSemanticMapping

The remaining sections are organized as follows: A comprehensive literature review is presented in Section II. Section III presents the problem setup and preliminaries. The methodology is discussed in Section IV. Section V presents quantitative evaluation methods for dynamic mapping. Results and discussion are given in Section VI. Finally, Section VII concludes the paper and provides ideas for future work.

II. RELATED WORK

In this section, we review works on semantic and dynamic mapping. While our work is focused specifically on mapping, the improved mapping algorithm can lead to improvements in downstream tasks such as localization when integrated in a SLAM system. Therefore, we provide background and perform comparisons of both mapping and SLAM systems to highlight the differences in how the environment is perceived, and subsequently represented. A taxonomy of the state-of-the-art dynamic mapping works is given in Table 1, based on the presence (✓) or absence (×) of semantics and scene dynamics usage, the type of sensors they operate on, and the type of flow measurements incorporated.

A. SEMANTIC MAPPING

Semantics are important to robot perception for better scene understanding and interaction [6]. Whereas many semantic mapping works have explored learning-based local mapping [12], [13], [14], [15], [16], [17], our work is a mathematically-derived 3D global mapping algorithm. Additionally, our method builds upon existing deep learning research by directly taking the output from neural networks as input, instead of attempting to embed all information within a latent space. With explicit intermediate steps, our model is diagnosable and reliable, as failures at each stage of the pipeline may be identified and observed. For this reason, we consider works that incorporate semantic measurements into maps given poses and estimated semantic labels.

Among a large body of semantic mapping studies, SemanticFusion [8] can be regarded as a classic approach where the semantic probabilities of single-frame 2D images are obtained from a Convolutional Neural Network (CNN) and re-projected into 3D, after which a Bayesian update scheme fuses multi-scan probabilities into a semantic surfel map. Other works differ in the deep neural network used (e.g., recurrent neural networks on consecutive frames [18], [19], 3D CNN for point clouds [20]), the map representation employed (point-cloud maps [7], [19] and voxel-based maps [9], [18], [21]), or the type of semantics (instance-level [22], object-level [7], [23], [24], [25] and place-level [26]). More recently, distributed semantic mapping for multi-robots [27], [28] and 3D scene graphs [29] are also trending research topics.

Another line of research concerns continuous semantic mapping with uncertainty [30], [31], [32], [33], that allows one to query maps at arbitrary resolutions. Kernel methods such as Gaussian Processes (GPs) are well-established for predicting a continuous non-parametric function to represent the semantic map [34], [35], [36]. BKI is an efficient approximation of GPs which yields fast computation and accurate
TABLE 1. Comparison of properties of DynamicSemanticBKI with respect to other dynamic SLAM and dynamic mapping systems. Although we compare D-BKI only to other mapping baselines, we elaborate on properties of both SLAM and mapping systems to highlight the dynamic mapping taxonomy. In the table, C = (mono, stereo, RGB-D) and 3DC = (stereo, RGB-D).

| Paper                        | Semantic | Scene Dynamics Retention | Sensor | Flow |
|------------------------------|----------|--------------------------|--------|------|
| DynaSLAM [37]               | ✓        | ✓                        | C      |      |
| Alcantarilla et. al. [38]    | ×        | ✓                        | stereo | Scene |
| DSOD [39]                   | ✓        |                         | mono   |      |
| SOF-SLAM [40]               | ✓        | ✓                        | RGB-D  | Optical |
| DS-SLAM [41]                | ✓        | ✓                        | RGB-D  | Optical |
| Brusch. et. al. [42]        | ✓        | ✓                        | mono   |      |
| SLAM++ [43]                 | ✓        | ✓                        | RGB-D  |      |
| DOS-SLAM [44]               | ✓        | ✓                        | RGB-D  | Scene |
| Detect-SLAM [25]            | ✓        | ✓                        | mono, stereo |      |
| SLAMANTIC [45]              | ✓        | ✓                        | RGB-D  |      |
| Fusion++ [21]               | ✓        | ✓                        | RGB-D  |      |
| MaskFusion [24]             | ✓        | ✓                        | RGB-D  |      |
| MID-Fusion [46]             | ✓        | ✓                        | RGB-D  |      |
| ClusterSLAM [47]            | ✓        | ✓                        | stereo | Scene |
| DynaSLAM [48]               | ✓        | ✓                        | stereo | Scene |
| EM-Fusion [49]              | ✓        | ✓                        | RGB-D  |      |
| Rosinol et. al. [29]        | ✓        | ✓                        | stereo |      |
| Vespa et. al. [50]          | ✓        | ✓                        | RGB-D  |      |
| Henet et. al. [51]          | ✓        | ✓                        | RGB-D  |      |

| Mapping                      | Semantic | Scene Dynamics Retention | Sensor | Flow |
|------------------------------|----------|--------------------------|--------|------|
| Sun et. al. [52]             | ✓        | ✓                        | LIDAR  |      |
| Kashoap et. al. [53]         | ✓        | ✓                        | LIDAR  | Scene |
| Sun et. al. [54]             | ✓        | ✓                        | LIDAR, 3DC |      |
| Ours                        | ✓        | ✓                        | LIDAR, 3DC | Scene |

inference for semantic mapping [11]. This work extends [11] to dynamic scenes.

B. MAPPING IN DYNAMIC ENVIRONMENTS

Dynamic objects can break the assumption of scene rigidity in most mapping algorithms and cause failure. When combined with localization in a SLAM pipeline, artifacts left by dynamic objects can introduce errors for downstream tasks such as pose estimation and loop closure. Thus, some SLAM systems treat dynamic objects in a scene as spurious data or outliers, excluding them entirely from pose estimation and mapping to achieve better accuracy and robustness [37], [38], [39]. However, discounting dynamic objects ultimately relies upon the ability to reject dynamic objects and decreases the level of scene understanding embedded within the map. In this section, we provide background on rejection-based approaches as well as mapping algorithms which jointly model the static world and dynamic objects.

Discarding dynamic objects may be performed through probabilistic outlier rejection [42], moving consistency check [41], feature-based filtering [37], measurement-map semantic inconsistency check [54], culling out with object-camera relative poses [43], semantic and geometric information coupling [40], or residual motion likelihood calculation [38], [44]. These methods can partially reduce localization error brought by dynamic objects, but still have limitations. For instance, discarding information based on semantic labels completely depends on the prediction accuracy. Moreover, the discarded motion information, if modeled correctly, could be further leveraged to predict the scene dynamics.

There are two primary approaches to integrating motion within maps. In the first method, scene dynamics are incorporated into a single reconstruction volume. This could be done by maintaining an observation point cloud with a moving probability [25], calculating a dynamics factor for classes that could be mis-classified as “dynamic” (e.g., parked cars) and incorporating those into pose estimation [45], propagating feature points by sampling from scene flow measurements [53], or fusing semantic features by recurrent observation average pooling in an OctoMap cell [52]. Instead of analyzing the motion properties of map cell or feature from a single scan, we combine spatio-temporal motion data over multiple scans and neighbouring voxels.

The second approach is characterized by its underlying object-oriented map representation. These approaches track local objects using Iterative Closest Point (ICP) and semantic segmentation-aided fusion [21], [24], [46], or by clustering on the basis of motion estimation [47]. Object tracking is also done via sparse scene flow estimation [48], frame-to-model data association [55] or Signed Distance Function (SDF)-based data association [49]. In this area, deep learning-based instance segmentation is often the bottleneck for computational efficiency [25], [37], [41], [44]. Although we model the scene as a unified reconstruction volume as in the former approaches, we can incorporate flow during environmental perception as in the latter approaches.

III. SEMANTIC BAYESIAN KERNEL INFERENCE

Semantic-Bayesian Kernel Inference (BKI) [11] is a probabilistic method for 3D semantic mapping with quantifiable uncertainty. The Semantic-BKI framework assumes that the j-th map cell (voxel in 3D, indexed by $j \in \mathbb{Z}^+$) with semantic probability $\theta_j = (\theta_j^1, \ldots, \theta_j^K)$, where $\theta_j^k$ is the probability of the j-th cell belonging to the k-th category, has the Categorical likelihood $p(y_j | \theta_j) = \prod_{k=1}^K \theta_j^k | y_j = k \}$. Here, $y_j \in \{1, \ldots, K\}$ is the semantic measurement at position $x_j \in \mathbb{R}^3$ in or around the j-th cell, and $y_j = k$ evaluates to 1 if $y_j = k$, 0 otherwise. The semantic measurement $y_j$ is usually the semantic label output by a neural network. Given training data with N measurement points $D = \{(x_i, y_i)\}_{i=1}^N$, semantic mapping seeks the posterior distribution $p(\theta_j | D)$ for each map cell j.

For a closed-form solution, BKI semantic mapping adopts a conjugate prior over $\theta_j$ given by a Dirichlet distribution $\text{Dir}(\theta_j, \alpha_0)$, $\alpha_0 = (\alpha_0^1, \ldots, \alpha_0^K)$, where $K \geq 2$ is the number of categories, and $\alpha_0^k \in \mathbb{R}^+$ are concentration parameters for each category. Applying Bayes’ rule and Bayesian kernel inference, the posterior is another Dirichlet distribution, given by $\text{Dir}(\theta_j, \alpha_j)$, $\alpha_j = (\alpha_j^1, \ldots, \alpha_j^K)$, and:

$$\alpha_j^k = \alpha_0^k + \sum_{i=1}^N K_k(x_i, x_j) | y_i = k$$
where \( K_x : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow [0, 1] \) is a spatial kernel function defined on 3D Euclidean space to capture the spatial correlation of two 3D positions, and \( \alpha^k_j \) is the \( k \)-th concentration parameter of the query voxel \( j \) centered at \( x_j \in \mathbb{R}^3 \).

Given \( \alpha_j \), the maximum a posteriori (MAP) estimate of \( \theta_j \) can then be computed in closed-form:

\[
\hat{\theta}^j_k = \frac{\alpha^k_j - 1}{\sum_{k=1}^K \alpha^k_j - K}, \quad \alpha^k_j > 1.
\]

In BKI semantic mapping, the prior distribution of the map at time stamp \( t \) is directly set to be the posterior at time stamp \( t - 1 \) (assuming that the environment does not change between two time stamps), i.e.,

\[
p(\theta_{j,t} | D_{1:t-1}) = p(\theta_{j,t-1} | D_{1:t-1}),
\]

to allow recursive Bayesian updates using sequential training data:

\[
p(\theta_{j,t} | D_{1:t}) \propto p(D_t | \theta_{j,t}, D_{1:t-1})p(\theta_{j,t} | D_{1:t-1})
\]

\[
= p(D_t | \theta_{j,t})p(\theta_{j,t-1} | D_{1:t-1}). \quad (3)
\]

However, this assumption is easily violated by moving objects in the environment or environmental changes. As such, we model the transition from \( p(\theta_{j,t-1} | D_{1:t-1}) \) to \( p(\theta_{j,t} | D_{1:t-1}) \) using spatial and temporal information.

**IV. METHOD: DYNAMIC-BKI**

In this section, we introduce a method to extend Semantic-BKI to dynamic environments. We first formulate an auto-regressive temporal transition model which propagates the map posterior according to the scene dynamics. Next, we show how we aggregate motion information from the training data for incorporation into the map voxels. Finally, we consolidate and summarize the algorithm for dynamic semantic mapping.

**A. TEMPORAL TRANSITION MODEL**

When dynamic objects move in and out of a voxel \( j \), the samples observed in it across time are not independently and identically distributed (i.i.d.). Samples drawn from the map posterior at different time stamps will come from independent but not identically distributed Dirichlet distributions. In Figure 2, we illustrate the motion of an object and a corresponding visualization of the Dirichlet probability density function (PDF) over the 2-simplex when there are just three classes — “robot”, “free space”, and “other”. The static world assumption in (3) solely relies on the frequency of observations in a voxel. This property makes the Dirichlet distribution ignore scene dynamics and become overconfident about classes that contribute more observations over all time stamps rather than the current time stamp. Therefore, for correct classification, the hyperparameters of the Dirichlet distribution under a static world assumption have to evolve with the scene dynamics.

We first introduce notations used in our formulation. Let the set of all classes be \( \mathcal{P} \), the set of moving classes be \( Q \) (\( q \in Q \)) and the free voxel category be denoted as “free.” Additionally, let the set of all classes excluding a class \( r \) be \( \mathcal{P} \setminus r \). We define a voxel flow vector for each voxel as \( v_j = (v^1_j, \ldots, v^K_j) \), which is the motion information captured per semantic category within a voxel \( j \). Details on computing voxel flow is presented in Section IV-D.

We propose a time-series model to account for temporal discrepancies in the Dirichlet distribution caused by moving objects. To forecast \( \bar{\alpha}_{j,t} \) of voxel \( j \) when a moving object passes through at time stamp \( t - 1 \), we apply an auto-regressive (AR) model that leverages the 3D motion information captured from the environment and applies it to the map prior. The class-wise AR model is as follows:

\[
\alpha^k_{j,t} = e^{-v^k_{j,t-1}^2} \alpha^k_{j,t-1}, \quad (4)
\]

where \( e^{-v^k_{j,t-1}^2} \) is the AR model’s parameter, \( \alpha^k_{j,t-1} \) is prior concentration parameter for class \( k \) and \( v^k_{j,t-1} \) is the voxel flow at time stamp \( t - 1 \) which influences the hyperparameter \( \alpha^k_{j,t} \) for semantic class \( k \). We depict a graphical model for this temporal transition model in Figure 3.

The concept behind the transition model is to redistribute the probability mass of the concentration parameters when there is motion observed in the environment. Therefore, we warp the concentration parameters according to the effect that the motion of a dynamic object has on \( j \) its corresponding class and 2) other classes. Keeping these two factors in mind, we introduce two modules to predict the concentration parameters \( \bar{\alpha}_{j,t} \) for voxel \( j \) at time stamp \( t \).

**B. BACKWARD OR EXIT CORRECTION (BACC)**

When a moving object of category \( q \) is detected in voxel \( j \) at time stamp \( t - 1 \) is in motion and could exit in time

**FIGURE 2.** We illustrate the observation of a moving object through the middle voxel in this map and display how that voxel’s semantics are different at every time step. For every time step, we plot the posterior Dirichlet probability density function (PDF) of the voxel on a 2-simplex. The shift in the rainbow gradient demonstrates what the belief about a semantic category (robot, free or other) can be over \( t=0 \) to \( t=2 \) to classify the voxel correctly at that time. This shift can be influenced by changing the concentration parameters (hyperparameters) of the Dirichlet posterior as: \( \alpha_{q,0} \rightarrow \alpha_{q,1} \rightarrow \alpha_{q,2} \).
When a voxel category is updated independently.

Given a voxel $j \in \mathcal{X}_t$ centered at $x_j$, we wish to get a low-level understanding of how an object is moving in or out of it to model $v^q_{j,t-1}$ in (4). Scene flow provides us with the underlying 3D motion field of the points in the scene. Given two incoming point clouds $\mathcal{X}_{t-1}$ and $\mathcal{X}_t$, recorded at time stamp $t - 1$ and $t$, respectively, we require a translational motion vector $u_i \in \mathbb{R}^3$ that conveys how much a point $x_i \in \mathcal{X}_{t-1}$ has displaced to a new location $x'_i \in \mathcal{X}_t$. In practice, this translational motion vector can be obtained from the “scene flow” associated with each point in the point cloud [56], [57], [58].

To capture the voxel flow $v_j = (v^q_j, \ldots, v^K_j)$ pertaining to any semantic category for voxel $j$, we aggregate the flow from training points around voxel centroid $x_j$. Thus, given training points $D = \{(x_i, y_i, u_i)\}_{i=1}^N$, a spatial kernel $K_v : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow [0,1]$ is used to weight the influence of each point $x_i$ on $v_j$ so that the closer a dynamic object of class $q \in \mathcal{Q}$ is to the voxel center, the more influence it has. Mathematically, this becomes a kernel density estimation problem and the per-class voxel flow is calculated as:

$$
 v^q_j = \frac{1}{N} \sum_{i=1}^N K_v(x_i, x_j) ||u_i||_{y_i=q},
$$

where $||\cdot||$ takes the Euclidean norm. We take the vector norm consistent with $v^q_j$’s usage in the exponential AR-model in (4). Therefore, our objective is to get a quantitative estimate of how much motion there is around the voxel, rather than capture the direction of motion.

In Section IV-C, we introduced FORC for free and other static classes. As explained previously, we calculate their per-class voxel flow together by considering the dynamic objects of all categories moving in voxel $j$:

$$
 v^\text{free}_j = \frac{1}{N} \sum_{i=1}^N K^\text{free}_v(x_i, x_j) ||u_i||_{y_i=q},
$$

where $\mathcal{Q}$ is the set of dynamic classes and $K^\text{free}_v$ is a special case of $K_v$ that weights the influence of dynamic training point $x_i$ on $v^\text{free}_j$. Specific details about $K_v$ and $K^\text{free}_v$ will be discussed in Section VI-A.

In Algorithm 1, we summarize how the per-class voxel flow for a query voxel $j$ is estimated using the positional $\mathcal{X}_t$, semantic $Y_t$, and egomotion-compensated $U_t$ information of each point $x_j$ in a point cloud. For BACC, we aggregate the flows of the training points encountered around $x_j$ in line 1.4. For FORC, we aggregate the flows of the training points while weighing the ones in neighbouring voxels more (than in BACC) in line 1.5. Whereas both equations have a similar form, BACC and FORC have separate kernels. Additionally, while FORC considers all neighboring dynamic points when...
Algorithm 2 Dynamic Semantic Mapping

1: Input: Training data: (X_t, Y_t, V_t); Query point: x_j; Previous state estimate $\alpha_{j,t-1}$; Flow estimate $v_{j,t}$
2: procedure Combined Posterior Update
3: for k = 1, ..., K do
4: $\alpha^k_{j,t} \leftarrow \exp(-((v^k_{j,t-1})^2)/\alpha^k_{j,t-1}) \triangleright$ Incorporate motion
5: for each $(x_i, y_i) \in (X_t, Y_t)$ do
6: $w_k \leftarrow K_\alpha(x_i, x_j) \triangleright$ Weight of $x_i$ on observation
7: $\alpha^k_{j,t} \leftarrow \alpha^k_{j,t} + w_k[y_i = k]$
8: end for
9: end for
10: return $\alpha_{j,t}$
11: end procedure

updating $v_{j,t}^t$, BACC only computes $v_{j,t}^q$, from dynamic points with matching semantic label $q$. After calculating, $v_{j,t}^q$, for any class $k$, in line 1.12 and 1.14, we post-process $v_{j,t}^q$ with a filter $f: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ to aggregate information from the voxel flow in the previous time step $v_{j,t-1}^q$ in the final calculation of $v_{j,t}^q$.

E. MAP POSTERIOR UPDATE FOR SCENE PROPAGATION

Section IV-A describes how we account for the change in concentration parameters of the Dirichlet distribution caused by the motion of objects. Using this model and following a Bayesian approach, $\alpha_{j,t}$ in (4) can be substituted as the prior in (1), i.e.,

$$\alpha_{j,t}^k = \alpha_{j,t-1}^k + \sum_{i=1}^N K_\alpha(x_i, x_j)y_i. \quad (7)$$

Algorithm 2 consists of prediction and update steps as in a recursive Bayes filtering. For the prediction step in line 2.4, we apply the temporal transition model with the query point’s flow estimate $v_{j,t-1}$. The prediction step with BACC enables removal of traces left by moving objects “exiting” voxels. For example, if a car was in motion at time stamp $t - 1$ and moving out from a voxel $j$, calculating $v_{j,t}^{car}$ ensures that the map maintains confidence about a static class such as “road” $\alpha_{j,t}^{road}$ and decreases confidence about the car class $\alpha_{j,t}^{car}$. Additionally, the FORC in our algorithm facilitates the “entry” of dynamic objects into previously encountered areas in the map by reducing overconfidence in “static” and “free” classes.

With the update step in line 2.7, incoming spatial and semantic training data $(X_t, Y_t)$ is incorporated.

V. QUANTITATIVE EVALUATION FOR DYNAMIC MAPPING

Typically, dynamic and semantic mapping methods that operate on stereo images re-project the map onto the image plane, and evaluation is done based on pixel-wise semantic segmentation of the image [48], [53]. Suppose one uses the same quantitative metric to evaluate the entire scene’s geometric-semantic reconstruction accuracy, this would fail to capture the “complete” scene — i.e., how well the map can represent portions of the environment that are not reflected on the evaluation image, e.g., free cells in the map. This problem is significant for evaluating dynamic maps as free space in the environment might be mis-classified as occupied due to artifacts. Therefore, we propose a querying framework for dynamic semantic occupancy mapping that considers both the “complete” scene and scene dynamics for the map evaluation.

Let $\mathcal{M}$ be the map we are building to represent an environment. Let the corresponding ground truth model of the environment be denoted by $\mathcal{G}$. The ground truth model could be sensor data post-processed with correct labels or as a map representation — e.g., a semantically-labelled point cloud, a set of RGB-D images, a heightmap, etc. Let us assume that the rays from our current scan intersect voxels that are “observed” by the robot (marked as “Visible” in Figure 4). We model this “Visible” portion of $\mathcal{G}$ with $\mathcal{M}_v$, i.e., voxels in the map currently being “observed” by the robot. The portion of the environment that is not seen in the current scan could then either be previously explored or still unexplored. If some portion of the environment is “Unexplored” as in Figure 4, there would be no voxel created in $\mathcal{M}$ for that portion. Otherwise, if the voxel was previously explored and is not visible to the robot in the current scan, we consider it “Occluded” as shown in Figure 4. We denote the voxels in $\mathcal{M}$ that represent the occluded portion of the environment as $\mathcal{M}_o$. Our map $\mathcal{M}$ is thus comprised of voxels that are visible in the current scan ($\mathcal{M}_v$) and voxels that are not ($\mathcal{M}_o$).

To perform a quantitative comparison on the semantic scene representation between our model $\mathcal{M}$ and any ground truth $\mathcal{G}$, we build upon two different map query frameworks introduced in [59]: accuracy and completeness. In both metrics, we assess the intersection between $\mathcal{M}$ and $\mathcal{G}$. However, in map accuracy, we evaluate each element in $\mathcal{M}_v$ against $\mathcal{G}$, and in map completeness each element of $\mathcal{M}$ against $\mathcal{M}_v \cup \mathcal{M}_o$.

A. MAP ACCURACY

The accuracy of the map, as the name suggests, quantifies how correct the visible metric-semantic representation $\mathcal{M}$ is when compared with the true value $\mathcal{G}$ of the environment. As this work pertains to semantic occupancy mapping, “correctness” is specifically the semantic classification accuracy.

For each element (in our case, voxel) $m \in \mathcal{M}_v$, we generate the corresponding ground truth $g_m \in \mathcal{G}$ that is the “closest semantic neighbor” to $m$. In practice, $g_m$ will be the nearest element in $\mathcal{G}$ to $m$ in metric space and also, most representative of the semantic category $m$ could belong to. For example, if $\mathcal{G}$ is a point cloud there may be many points residing within $m$. The most representative semantic category is then the majority semantic label of points within voxel $m$.

The semantic prediction for voxel $m$ can then be evaluated against that of $g_m$. Details about how $g_m$ can be generated from sensor data for single- and multi-view data sets are discussed in Section VI-C1. If we are comparing the map accuracies of mapping methods with different map representations (e.g. uniform resolution voxel map versus point cloud map with a non-uniform point distribution), however, the
query elements \( m \in M_v \) could cover metric space differently. Therefore, map accuracy is more suited for comparing maps under the same representation (e.g. both are uniform resolution voxel maps). Consequently, we compare Semantic-BKI (S-BKI) and Dynamic-BKI (D-BKI) using the same map representation, and compute the precision, recall, and Jaccard scores across all classes.

**B. MAP COMPLETENESS**

Completeness of the map pertains to how much of the environment, including both visible and occluded regions \( M_v \cup M_o \), is reconstructed correctly. For instance, if we want to evaluate a portion of the environment that was previously observed, but is not currently visible, “completeness” can be used to ascertain whether the map is able to represent the environment correctly. This is because we sample the environment to query the map, rather than the other way around. If we use multiple views of the environment to obtain ground truth \( G \) for portions of the environment that are currently occluded in the map \( M_v \), it is possible to include \( M_o \) into completeness. Another advantage of using map completeness as a metric is that we can compare the map inference performance across methods with different map representations.

We pick an element \( g \in G \) and find its “closest semantic neighbor” \( m_g \in M \). Again, \( m_g \) would be the nearest element to \( g \) in metric space and most representative semantic category that \( g \) could belong to. In practice, we seek the voxel in which the element \( g \) falls. If the metric distance between \( g \) and \( m_g \) is greater than a certain margin, we consider that \( g \) is currently unexplored by the robot, and exclude these pairs in the evaluation. These kinds of space are shown in the rightmost column in Figure 4.

In addition, to keep the evaluation relevant to dynamic semantic mapping, we treat static and dynamic objects differently when calculating completeness.

**TABLE 2. Parameters for ablation studies.**

| Parameter                        | Value     |
|----------------------------------|-----------|
| Map resolution                  | 0.05      |
| Downsampling resolution         | 0.1       |
| Free space sampling resolution  | 0.5       |
| \( t_s \)                        | 0.15      |
| \( \sigma_s \)                   | 0.2       |
| \( t_1 \)                        | 0.02      |
| \( \sigma_1 \)                   | 50        |

**TABLE 3. Parameters for SemanticKITTI results.**

| Parameter                        | Value     |
|----------------------------------|-----------|
| Map resolution                  | 0.1-0.3   |
| Downsampling resolution         | 0.1-0.3   |
| Free space sampling resolution  | 100       |
| \( t_s \)                        | 0.1       |
| \( t_1 \)                        | 2.5       |
| \( \sigma_1 \)                   | 100       |

1) **If \( g \) is static.** As \( g \) is static, \( g \) could not move irrespective of whether the robot has observed it. As a result, we evaluate semantic accuracy for any nearest neighbor \( m_g \in M \).

2) **If \( g \) is dynamic.** As only dynamic objects in \( M_v \) are currently seen by the robot, we evaluate semantic accuracy for all \( m_g \in M_v \). We do not evaluate on \( M_o \) as these voxels in the map are occluded and could have a different state from the last time observed by the robot.

**C. AUXILIARY TASK: SEMANTIC SEGMENTATION**

Semantic segmentation of a point cloud is another task that can be performed with an existing map model \( M \), and we can consider the map’s performance on this auxiliary task for additional evaluation. With semantic mapping, we can inherently fuse multi-frame measurements and improve semantic classification accuracy as in S-BKI [11]. The querying method for a single scan is simple: we pick each point in \( D_t \) already inserted into the map \( M \) at time stamp \( t \) and check which voxel it falls inside. \( D_t \) contains \( Y_t \), which are the semantic label predictions corresponding to \( X_t \). As discussed in Section III, this is typically obtained from a neural network. The semantic category of the voxel becomes the prediction from our model \( M \). Both of these can then be compared with the ground-truth, semantically-annotated point cloud - that is typically provided in data sets such as SemanticKITTI [60]. This comparison can show whether semantic segmentation predictions can be improved through smoothing [11].

**VI. RESULTS AND DISCUSSION**

In this section, we first describe our experimental setup and the data sets used for evaluation. Then, we demonstrate the performance of the proposed mapping system D-BKI with qualitative results on synthetic and real data sets. Finally, quantitative results on semantic scene understanding sub-tasks using single- and multi-view data sets are presented.
A. EXPERIMENTAL SETUP

We first describe our (i) system design choices, then elaborate on (ii) the data sets used, and lastly discuss (iii) flow estimation for each point cloud data set.

1) SYSTEM DESIGN CHOICES

In the proposed mapping framework, every query voxel has 6 neighbours (one on each facet). For computational efficiency, only the training points within the 6 neighbouring voxels are used in the calculation of both (5) and (6).

We choose a sparse kernel [61] as

\[ \mathcal{K}_v = \mathcal{K}_v^{\text{free}} = \begin{cases} \sigma_1 \left[ \frac{1}{3} (2 + \cos \left( \frac{2\pi d}{l_1} \right) - 1 \frac{d}{l_1}) + \frac{1}{2\pi} \sin \left( \frac{2\pi d}{l_1} \right) \right] & \text{if } d < l_1 \\ 0 & \text{otherwise} \end{cases} \]

where \( d = \|x - x'\|, l_1 > 0 \) is the length scale and \( \sigma_1 \) is the kernel scale parameter.

Typically, the kernel length scale in \( \mathcal{K}_v \) and \( \mathcal{K}_v^{\text{free}} \) is chosen with respect to the resolution of the map being built as it controls how much influence a point in a neighbouring voxel has. In our experiments, \( l_1 \) is greater than the map resolution and \( \sigma_1 \) can be set once in the beginning according to the size of the point set and free-space sampling rate. Lastly, in Algorithm 1, we implement \( f(\ldots) \) as a moving average filter in line 1.12.

2) DATA SETS AND BENCHMARKS

We use point cloud-based data sets that contain only positional information. However, our method is amenable to any point cloud data with intensity, colour or other fields. Additionally, it can also be applied to depth camera data, of which larger data sets for training and evaluation exist.

3) GAZEBO SIMULATION ENVIRONMENT

To create an indoor synthetic data set, a Gazebo simulation environment was set up with multiple Turtlebots exploring a house. We mounted one robot (the ego-robot) with an omni-directional block laser scanner for data collection in the form of point clouds with positional information only. To simulate dynamic objects in the environment, we have other three Turtlebots exploring the same house. Using a reactive planner, the robots avoid each other and obstacles in the environment. The collected data is processed using Point Cloud Library (PCL) [62] and annotated based on height into three semantic classes - floor, robot and miscellaneous objects including walls and cabinets. The scene flow for each scan is computed with FlowNet3D [56].

4) SemanticKITTI DATA SET

The SemanticKITTI data set [60] is a large-scale real driving data set based on the KITTI Vision Benchmark [63] where semantically-annotated LiDAR scans and camera poses are provided for all sequences. Camera poses are
To evaluate the map completeness of the proposed dynamic semantic occupancy mapping, a reliable ground truth model of the environment including free space is needed. As real data sets collected using a single-view sensor (such as SemanticKITTI) usually do not have sufficient measurement coverage to fully recover the underlying environment model, we leverage a simulation environment CARLA [66].

We generate a synthetic multi-view scene completion data set sequence from the CARLA [66] simulator. The methodology for its creation is available publicly in [17]. We generate ground-truth environment models by uniformly distributing multiple LiDAR sensors around the ego vehicle, effectively obtaining a 3D Monte Carlo sampling of the world which is i.i.d. with respect to time. The simulation environment also provides ground-truth scene flow (velocity) and semantic labels for each point. Free space observations are obtained by linearly interpolating along all points at a fixed interval of 1.5 meters. Ground truth point clouds with semantic labels are then fused into a semantically annotated ground truth voxel model \( G \) with 0.3 meter resolution. The voxel centers in the ground truth model act as query points in the completeness experiments. This approach is similar to the SemanticKITTI [60] scene completion data set; however, it has no traces from dynamic objects and fewer occlusions due to sampling from multiple sensors at the same time.

6) FLOW ESTIMATION

To obtain the corresponding flows \( U_t \) of a point cloud \( X_t \), we choose a state-of-the-art deep learning architecture based on PointNet++ [67] - FlowNet3D [56]: a supervised method based on PointNet++ which estimates scene flow between two successive point clouds \( X_{t-1} \) and \( X_t \).

Typically, implementations for scene flow estimation train on the XYZRGB fields, i.e., the point cloud includes both the position and colour information. We trained an adapted version of FlowNet3D on the KITTI 2015 Scene Flow data set [68] and the FlyingThings driving data set [69] by only including position information for training. After obtaining \( U_t \) from the networks, we perform egomotion-compensation by subtracting the mean flow of static classes from \( U_t \).

As we only need flow for moving objects, our mapping method can be applied when any flow information for \( \{x_t^q \in X_t \mid \forall q \in Q \} \) is available. To demonstrate the performance of the mapping framework independent of flow estimation error, we obtain the ground truth velocity of dynamic objects from the CARLA simulator.

B. QUALITATIVE RESULTS

The goal of this section is to (i) demonstrate improvements of the temporal transition model qualitatively through ablation studies, and (ii) compare the real-time map construction by semantic and dynamic BKI with the data sets described in Section VI-A2.

1) ABLATION STUDIES

We perform two ablation studies to demonstrate the function and efficacy of each component of the method. These studies are aimed to qualitatively show how the global map inference estimated with SuMa [64], and semantic annotations for each LiDAR scan are generated by RangeNet++ [65]. Additional labels are provided to distinguish static objects from dynamic objects, such as person and moving-person. There are 22 sequences, out of which 11 sequences are provided with ground truth labels for training (00-07), validation (08) and testing (09-10). Sequence 11-21 do not come with ground truth semantic labels, but can be evaluated in a public leader-board over the mean Intersection-over-Union (mIoU) metric. Since the most reliable ground truth model in this single-view data set is the semantically-labeled point cloud, we use them to generate a ground truth model \( G \).

5) CARLA DATA SET

To evaluate the map completeness of the proposed dynamic semantic mapping, a reliable ground truth model of the

FIGURE 6. Qualitative results on Sequences 01 (top) and 04 (bottom) of SemanticKITTI. Four images are shown for both frames, which include (Top Left) Right stereo image corresponding to one of the scans. Note that the image is included for validation, however we strictly perform map update from LiDAR when generating results. (Top Right) S-BKI mapping without any free space sampling. Trails are left where cars passed over. (Bottom Right) S-BKI with free space sampling. After a few scans, the map becomes overconfident about the presence of free cells and fails to incorporate dynamic objects in the map. (Bottom Left) D-BKI with free space sampling. The cars are tracked with minimal traces.

97962
FIGURE 7. Qualitative results on S-BKI (top left), Kochanov et al. (top-right) and D-BKI (bottom-right) of the CARLA data set. This scene (bottom-left) showcases a static ego-car (in dark-black) parked on the road while pedestrians walk around within its sensing range. In the map images, the red blobs are pedestrians walking on the street and the black rectangles are other cars (whether parked or moving). One can see that D-BKI "fills" in the shape of the car more than Kochanov et al. (as in the car on the top left) and does not leave trails of the walking pedestrians as with S-BKI. For Kochanov et al., there seems to be a more sparse representation for pedestrians due to their unique shape than D-BKI.

performs without backward (BACC) or forward (FORC) correction. Note that our global map is colored with a gray floor, mustard walls and red Turtlebots. We annotate our ego-robot building the map with a white box around it. Holes on the floor are typically spaces where the sensor has not scanned yet. We only tune parameters (in Table 2) for $K_v$, $l_s$ and $\sigma_s$ are the spatial kernel length scale and scale parameters respectively from [11].

a: WITHOUT BACC
To conduct this study, we remove scene flow aggregation for all dynamic classes by setting $v_{\text{free},j,t-1}^q = 0 \mid \forall q \in Q$ and observe the map as it is being built. Results are shown in the top row of Fig. 5. In the simulation snapshot, we highlight the 3 Turtlebots in the environment that are in motion. Without BACC, trails are visible behind each Turtlebot due to their motion not being considered during map-building. With BACC, no trails are left behind and each robot has consistently the same size due to the incorporation of the temporal transition model.

b: WITHOUT FORC
Results for this experiment are shown in the bottom row of Fig. 5. The simulation snapshot shows 2 moving Turtlebots in the environment. Without FORC, the motion of these Turtlebots around free voxels are not considered to compute $v_{\text{free},j,t-1}^q$ in (6). As a result, the prediction step for $\alpha_{\text{free},j,t}^q$ in 2.4 becomes obsolete. If $\alpha_{\text{free},j,t}^q > \alpha_{\text{robot},j,t}^q$, then voxel $j$ will be (incorrectly) classified as a free cell. Note that the other two robots do not get incorporated into the map as $\alpha_{\text{free},j,t}^q > \alpha_{\text{robot},j,t}^q$ for the voxels. With FORC, the map successfully represents the two Turtlebots.
2) SemanticKITTI DATA SET
We include images from sequence 1 and 4 of the SemanticKITTI data set to highlight the differences between static (S-BKI) and dynamic (D-BKI) mapping, as this is not easily captured in the semantic segmentation competition. These results are shown in Fig. 6, where S-BKI either discards dynamic objects over time completely or leaves them in the map depending on parameter choice. In contrast, D-BKI is able to accurately represent the moving objects without leaving long trails. Some of the parameters used to run the experiments are specified in Table 3.

3) CARLA DATA SET
We show qualitative results on five different scenarios described in Fig. 7 and the appendix section comparing our approach with various baselines in Fig. 8 and 9.

C. QUANTITATIVE EVALUATION
1) SEMANTIC MAPPING
For this sub-task, we compare the estimated map with respect to the ground truth world models $G$ of the SemanticKITTI and CARLA simulator data set (described in Sec. V). We conduct our experiments with two querying and evaluation metrics - map completeness and map accuracy. Map accuracy is measured at the intersection of the visible estimated map $\hat{M}_v$ with the $G$, and evaluated at each voxel in the estimated map. Map completeness includes both visible ($M_o$) and occluded voxels ($M_o$), and is evaluated at each ground truth element. Note that there are some considerations about the ground truth world model generation, which we will discuss in detail next.

We compare our algorithm, D-BKI, against the static semantic mapping baseline S-BKI [11], and a scene-propagation-based dynamic semantic mapping algorithm by Kochanov et al. [53]. Although [53] presents results on building voxel maps with stereo images, the approach is general and performs semantic segmentation and scene flow estimation to incorporate into the mapping pipeline later. We re-implemented their approach and tuned it to generate results on LiDAR point clouds. As their approach updates semantic and occupancy probability separately, we performed free space sampling to provide this extra information.

α: SINGLE-VIEW DATA SET
For a single-view data set, we select the well-known SemanticKITTI data set [60]. Semantic labels $Y$ for training are obtained from the Cylinder3D-multiscan model [70] and the ground truth labels in [60] are used to generate a ground truth world model $G$. As evaluation data for multi-scan dynamic semantic mapping with free space labels is not available, we generate it ourselves by keeping the semantically-labeled point cloud intact; but adding free space samples onto it. For evaluation of both map accuracy and map completeness, we create a point set $D_f^{\text{free}}$ containing only free space labels, by sampling free space every 1.5m from the sensor origin to each point in the point cloud. $D_f^{\text{free}}$ is then downsampled by a voxel-grid filter and added to the ground truth semantically labeled point cloud $D_t$ to generate $G$.

For map accuracy, to compute the “closest semantic neighbor” $g_m$ discussed in Section V, we consider the semantics of all points in $G$ that fall within each visible voxel $m \in M_v$. The semantic category that has the most points in $m$ is chosen to be the ground truth semantic category of $g_m$. Occupied space samples ($D_t$) are given priority over free space samples, i.e., $g_m$ is considered to be “free” only if $m$ exclusively contains free space samples.

We show results of this approach on the entire SemanticKITTI data set in Table 4 and demonstrate how we improve map accuracy over S-BKI with D-BKI. The average IoU over each scan is computed for each sequence and aggregated per class. The IoUs of a particular class are highlighted if there is a >0.01 difference between the methods. One can see that S-BKI and D-BKI perform similarly for 20 of the static classes. However, for all the 6 dynamic classes, D-BKI consistently outperforms S-BKI by a significant margin. This result also shows the importance of free space consideration for evaluating dynamic maps as the artifacts introduced by S-BKI seen in Fig. 6a and Fig. 6b would not be evaluated if we restrict our evaluation methodology only to occupied space.

For map completeness, the “closest semantic neighbor” $m_g$ for each $g$ in the ground truth $G$ is the voxel in $M$ that $g$ falls within. To compare with other dynamic semantic mapping methods, we picked four sequences that are representative of the challenges faced while driving in dynamic environments — highways (sequence 01) at high speed, countrysides (sequence 03) and cities (sequence 06) at normal speed, and residential areas in city (sequence 10) at slow speed. On this subset, we showcase in Table 4 how D-BKI and Kochanov et al. [53] perform in the map completeness metric. We average the Jaccard scores (i.e., mean IoU) across each scan in these four sequences for both methods at the same resolution of 0.3 m. D-BKI performs better or similarly to Kochanov et al. [53] in dynamic classes. As the map resolution is low, D-BKI’s mIoU drops slightly for pedestrians (smaller objects) but remains high for larger dynamic objects. The mIoU is observed to be significantly higher in static classes than [53].

β: MULTI-VIEW DATA SET
For the CARLA data set, we picked five scenarios often encountered in a dynamic urban environment — (i) a static car in the presence of moving pedestrians, (ii) a car having to stop in the presence of a jaywalking individual, (iii) a car driving at fast and (iv) slow speeds in dense traffic conditions and lastly, (v) a car driving in light traffic conditions. This data is acquired over an 1800 scan sequence and each of these scenarios is 100 scans long, but presents different challenges. The semantic segmentation labels input into the map are obtained from the simulator.
TABLE 4. Quantitative results for Dynamic-BKI and Kochanov et al.
[53] using the map evaluation method Map Accuracy on the CARLA data set.
The data collected is evaluated with two performance metrics - Precision
and Recall.

| Metric               | Mapping Method         | Prec | Rec  |
|----------------------|------------------------|------|------|
|                      | Dynamic-BKI (Out)      | 89.53| 95.99|
|                      | S-BKI                  | 85.50| 91.76|
|                      | D-BKI                  | 86.13| 91.76|
|                      | Mean                   | 88.37| 93.33|
|                      | Carola                 | 90.23| 92.23|
|                      | Kochanov et al. [53]   | 85.50| 91.76|
|                      | Map Completeness       | 88.37| 93.33|
|                      | Carola                 | 90.23| 92.23|

Precision is modeled as the proportion of ground truth measurements
that were correctly identified. The difference in recall between static and dynamic mapping
is most evident in the free class. If traces from dynamic objects
are not removed, free space voxels will be incorrectly labeled
occupied, and thus recall for the free category will be lower.
This is also evident in Table 5.

To evaluate how much of the ground truth $G$ is modeled
correctly by the maps, we report map completeness using
both the visible and occluded portions of the environment
(shown in Fig. 4) that correspond to $M_v$ and $M_o$ in the
map.

TABLE 5. Quantitative results for Dynamic-BKI and Kochanov et al.
[53] using the map evaluation method Map Completeness on the CARLA data set.
The data collected for the visible map $M_v$ is evaluated with two performance metrics - Precision
and Recall. The data collected for the occluded map $M_o$ is evaluated with mean IoU (mIoU).

| Indicator            | Mapping Method         | Mean  |
|----------------------|------------------------|-------|
|                      | Dynamic-BKI (Out)      | 89.64|
|                      | S-BKI                  | 85.50|
|                      | D-BKI                  | 90.01|
|                      | Kochanov et al. [53]   | 85.50|
|                      | Map Completeness       | 90.01|
|                      | Carola                 | 89.98|
|                      | Kochanov et al. [53]   | 85.50|

TABLE 6. Quantitative results for Dynamic-BKI and Kochanov et al.
[53] using the map evaluation method Map Completeness on the CARLA data set.
The data collected for the visible map $M_v$ is evaluated with two performance metrics - Precision
and Recall. The data collected for the occluded map $M_o$ is evaluated with mean IoU (mIoU).

We compare the map accuracy between S-BKI and D-BKI
in Table 5. Note that in accuracy, each visible voxel ($m \in M_v$)
is queried against the ground truth model $G$. Since S-BKI and
D-BKI share the same map representation, the maps share the
same origin and have overlapping voxels.

Precision is an indicator of how many predictions made by
the map match the ground truth, and is calculated per semantic
class $k$ as the number of voxels $m$ correctly labeled $k$, divided by the total number of voxels $m$ labeled $k$. Therefore,
precision will be lower for dynamic classes if residual traces
are not removed during map propagation. For example, the
trails seen in Fig. 6 for S-BKI lead to a low precision for
dynamic classes. This pattern may be seen in Table 5, where
D-BKI has improved precision on the vehicle and pedestrian
classes.

Recall is another useful metric for evaluating maps in
dynamic environments. In contrast to precision, recall is
calculated as the proportion of ground truth measurements
$g_m$ with semantic label $k$ that were correctly identified. The difference in recall between static and dynamic mapping
is most evident in the free class. If traces from dynamic objects
are not removed, free space voxels will be incorrectly labeled
occupied, and thus recall for the free category will be lower.
This is also evident in Table 5.

To evaluate how much of the ground truth $G$ is modeled
correctly by the maps, we report map completeness using
both the visible and occluded portions of the environment
(shown in Fig. 4) that correspond to $M_v$ and $M_o$ in the
map.

D. DISCUSSION
We showed that a simple auto-regressive transition model
enables dynamic scene propagation and rectifies the pitfalls
TABLE 7. Quantitative results on Dynamic-BKI and Cylinder3D [70] for evaluation of the Semantic Segmentation Sub-Task on the SemanticKITTI data set sequence 00-10 (Training) and 11-21 (Testing) [60] for 26 semantic classes. The collected data is evaluated with the performance metric mean IoU (mIoU).

| Seq. | Method      | Car | Person | Motorcycle | Truck | Van | Bus | Train | Road | Sidewalk | Sign | Bicycle | Pedestrian | Wall | Floor | Stairs | Roof | Sign board | Door | Window | Chair | Table | Chair | Table | Average |
|------|-------------|-----|--------|------------|-------|-----|-----|-------|------|----------|------|---------|------------|------|-------|--------|------|------------|------|--------|-------|-------|-------|-------|---------|
|      | Cylinder3D  | 0.93  | 0.86  | 0.69  | 0.96  | 0.63  | 0.45 | 0.32  | 0.75  | 0.72  | 0.74  | 0.72  | 0.49  | 0.68  | 0.55  | 0.59  | 0.64  | 0.65  | 0.60  | 0.62  | 0.62  | 0.62  | 0.62  | 0.62  |
|      | D-BKI (ours) | 0.96  | 0.89  | 0.73  | 0.96  | 0.71  | 0.43 | 0.38  | 0.78  | 0.75  | 0.79  | 0.75  | 0.56  | 0.67  | 0.59  | 0.63  | 0.69  | 0.65  | 0.63  | 0.66  | 0.65  | 0.65  | 0.65  | 0.65  |
| Training | Cylinder3D  | 0.94  | 0.87  | 0.71  | 0.95  | 0.62  | 0.41 | 0.35  | 0.77  | 0.74  | 0.76  | 0.74  | 0.52  | 0.66  | 0.55  | 0.59  | 0.63  | 0.64  | 0.62  | 0.65  | 0.62  | 0.62  | 0.62  | 0.62  |
|      | D-BKI (ours) | 0.96  | 0.89  | 0.73  | 0.96  | 0.71  | 0.43 | 0.38  | 0.78  | 0.75  | 0.79  | 0.75  | 0.56  | 0.67  | 0.59  | 0.63  | 0.69  | 0.65  | 0.63  | 0.66  | 0.65  | 0.65  | 0.65  | 0.65  |
| Testing | Cylinder3D  | 0.94  | 0.87  | 0.71  | 0.95  | 0.62  | 0.41 | 0.35  | 0.77  | 0.74  | 0.76  | 0.74  | 0.52  | 0.66  | 0.55  | 0.59  | 0.63  | 0.64  | 0.62  | 0.65  | 0.62  | 0.62  | 0.62  | 0.62  |
|      | D-BKI (ours) | 0.96  | 0.89  | 0.73  | 0.96  | 0.71  | 0.43 | 0.38  | 0.78  | 0.75  | 0.79  | 0.75  | 0.56  | 0.67  | 0.59  | 0.63  | 0.69  | 0.65  | 0.63  | 0.66  | 0.65  | 0.65  | 0.65  | 0.65  |

FIGURE 8. Qualitative results on S-BKI (top left), Kochanov et al. (top-right) and D-BKI (bottom-right) of the CARLA dataset. In this scene, there is a jaywalking pedestrian (bottom-left) and the ego-car (in dark-blue) stops for them. In all the map images, the red blob is the jaywalking pedestrian and the black rectangle is the car. One can see that D-BKI "fills" in the general area of the pedestrian’s existence (more than Kochanov et al) and does not leave trails of the walking pedestrians as with S-BKI.

of the static world assumption in the Semantic-BKI mapping algorithm - either by reducing traces in the map or by preventing overconfidence in free space. Metrics proposed to evaluate dynamic mapping quantitatively can aid in
A. Unnikrishnan et al.: Dynamic Semantic Occupancy Mapping Using 3D Scene Flow and Closed-Form Bayesian Inference

Providing more perspective on the appearance of the global metric-semantic map rather than the local view. This can be helpful in checking whether unoccupied space is erroneously classified as occupied or vice versa. The work can be applied to any sensor data that can be represented in the form of an XYZ point cloud. Given acquiring scene flow data for a full point cloud is more challenging than acquiring it for camera data, we anticipate that the performance is transferable to other 3D sensors.

Setting the map to build at finer resolutions achieves significantly better performance, but at the expense of higher computational burden and memory usage. A garbage collection process when a sequence runs too long for dynamic-mapping application may be useful. Having adaptive kernel lengths according to object size (e.g., for vehicular and human classes) may improve results. Future work includes investigating methods to compress and streamline data acquisition, demonstrating results on data sets in unstructured environments, and investigating memory-based alternatives to the autoregressive model.

VII. CONCLUSION

We developed a dynamic mapping algorithm based on Bayesian kernel inference that models the motion of dynamic...
objects using scene flow. Our map may be built from any 3D sensor and uses deep neural networks to obtain semantic labels and scene flow from raw point cloud data. For the evaluation of dynamic semantic maps we build, we utilize a quantitative metric that assesses semantics and unoccupied space in a unified manner. We demonstrated the efficacy of the mapping system on simulated data, a single-view real data set and multi-view synthetic data set. In particular, the proposed method, D-BKI, can reconstruct a dynamic scene more precisely than its static counterpart while maintaining high recall. D-BKI can also perform on par with state-of-the-art methods in dynamic semantic mapping in the presence of noisy as well as more accurate labels.

Future work includes investigating methods to compress and streamline data acquisition, working on data sets in unstructured environments and investigating memory-based alternatives to the autoregressive model. Exploiting the uncertainty estimated by D-BKI in scene understanding and robot navigation tasks [71] is also an attractive future work direction.

APPENDIX A
ADDITIONAL QUALITATIVE RESULTS
This section provides additional qualitative results shown in Fig. 8 and 9.

ACKNOWLEDGMENT
DISTRIBUTION A. Approved for public release; distribution unlimited. OPSEC #5575.

REFERENCES
[1] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Montfort, U. Müller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, “End to end learning for self-driving cars,” 2016, arXiv:1604.07316.
[2] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, “End-to-end driving via conditional imitation learning,” in Proc. IEEE Int. Conf. Robot. Automat., May 2018, pp. 4693–4700.
[3] L. Tai, G. Paolo, and M. Liu, “Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Sep. 2017, pp. 31–36.
[4] H.-T. L. Chiang, A. Faust, M. Fiser, and A. Francis, “Learning navigation behaviors end-to-end with AutoRL,” IEEE Robot. Autom. Lett., vol. 4, no. 2, pp. 2007–2014, Apr. 2019.
[5] G. Kahn, P. Abbeel, and S. Levine, “BADGR: An autonomous self-supervised learning-based navigation system,” IEEE Robot. Autom. Lett., vol. 6, no. 2, pp. 1312–1319, Apr. 2021.
[6] S. Garg, N. Sündenrath, F. Dayoub, D. Morrison, A. Cosgun, G. Carneiro, Q. Wu, T.-J. Chin, I. Reid, S. Gould, P. Corke, and M. Milford, “Semantics for robotic mapping, perception and interaction: A survey,” 2021, arXiv:2101.00443.
[7] N. Sündenrath, T. T. Pham, Y. Latif, M. Milford, and I. Reid, “Meaningful maps with object-oriented semantic mapping,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Sep. 2017, pp. 5079–5085.
[8] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, “SemanticFusion: Dense 3D semantic mapping with convolutional neural networks,” in Proc. IEEE Int. Conf. Robot. Automat., May 2017, pp. 4628–4635.
[9] S. Yang, Y. Huang, and S. Scherer, “Semantic 3D occupancy mapping through efficient high order CRFs,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Sep. 2017, pp. 590–597.
[10] I. Cherubier, J. L. Schönberger, M. R. Oswald, M. Pollefeys, and A. Geiger, “Learning priors for semantic 3D reconstruction,” in Proc. Eur. Conf. Comput. Vis., Sep. 2018, pp. 314–330.
[11] L. Gan, R. Zhang, J. W. Grizzle, R. M. Eustice, and M. Ghaffari, “Bayesian spatial kernel smoothing for scalable dense semantic mapping,” IEEE Robot. Autom. Lett., vol. 5, no. 2, pp. 790–797, Apr. 2020.
[12] J. Philion and S. Fidler, “Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3D,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 194–210.
[13] J. Dequaire, P. Ondréška, D. Rao, D. Wang, and I. Posner, “Deep tracking in the wild: End-to-end tracking using recurrent neural networks,” Int. J. Robot. Res., vol. 37, nos. 4–5, pp. 492–512, 2017, doi: 10.1177/0278364917701543.
[14] T. Roddick and R. Cipolla, “Predicting semantic map representations from images using pyramid occupancy networks,” 2020, arXiv:2003.13402.
[15] K. Peng, J. Fei, K. Yang, A. Roitberg, J. Zhang, F. Bieder, P. Heidenreich, C. Stiller, and R. Stiefelhagen, “MASS: Multi-attentional semantic segmentation of LiDAR data for dense top-view understanding,” IEEE Trans. Intell. Transp. Syst., early access, Feb. 1, 2022, doi: 10.1109/TITS.2022.3145588.
[16] J. Fei, K. Peng, P. Heidenreich, F. Bieder, and C. Stiller, “PillarSegNet: Pillar-based semantic grid map estimation using sparse LiDAR data,” in Proc. IEEE/RSJ Int. Conf. Robot. Automat., Jul. 2021, pp. 838–844.
[17] J. Wilson, J. Song, Y. Fu, A. Zhang, A. Capodici, P. Jayakumar, K. Barton, and M. Ghaffari, “Motion3C: Data set and network for real-time semantic mapping in dynamic environments,” IEEE Robot. Autom. Lett., vol. 7, no. 3, pp. 8439–8446, Mar. 2022.
[18] Y. Xiang and D. Fox, “DA-RNN: Semantic mapping with data associated recurrent neural networks,” 2017, arXiv:1703.03908.
[19] J. Cheng, Y. Sun, and M. Q.-H. Meng, “Robust semantic mapping in challenging environments,” Robotics, vol. 38, no. 2, pp. 256–270, 2020.
[20] R. Dubé, A. Cramariuc, D. Dugas, H. Sommer, M. Dymczyk, J. Nieto, R. Siegwart, and C. Cadena, “SegMap: Segment-based mapping and localization using data-driven descriptors,” Int. J. Robot. Res., vol. 39, nos. 2–3, pp. 339–355, 2020.
[21] J. McCormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger, “Fusion++: Volumetric object-level slam,” in Proc. Int. Conf. 3D Vis. (3DV), Sep. 2018, pp. 32–41.
[22] M. Grinvald, F. Furrer, T. Novkovic, J. J. Chung, C. Cadena, R. Siegwart, and J. Nieto, “Volumetric instance-aware semantic mapping and 3D object discovery,” IEEE Robot. Autom. Lett., vol. 4, no. 3, pp. 3037–3044, Jun. 2019.
[23] Z. Zeng, Y. Zhou, O. C. Jenkins, and K. Desingh, “Semantic mapping with simultaneous object detection and localization,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Oct. 2018, pp. 911–918.
[24] M. Rüns, M. Buffier, and L. Agapito, “Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects,” in Proc. IEEE Int. Symp. Mixed Augmented Reality, D. Chu, J. L. Gabbard, J. Grubert, and H. Regenbrecht, Eds. Munich, Germany, Oct. 2018, pp. 10–20.
[25] F. Zhong, S. Wang, Z. Zhang, and Y. Wang, “Detect-SLAM: Making object detection and SLAM mutually beneficial,” in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), Mar. 2018, pp. 1001–1010.
[26] N. Sündenrath, F. Dayoub, S. McMahon, B. Talbot, R. Schulz, P. Corke, G. Wyeth, B. Upcroft, and M. Milford, “Place categorization and semantic mapping on a mobile robot,” in Proc. IEEE Int. Conf. Robot. Automat., May 2016, pp. 5729–5736.
[27] Y. Yue, C. Zhao, R. Li, C. Yang, J. Zhang, M. Wen, Y. Wang, and D. Wang, “A hierarchical framework for probabilistic probabilistic semantic mapping,” in Proc. IEEE Int. Conf. Robot. Automat., May 2020, pp. 9659–9665.
[28] S. Jamieson, K. Fathian, K. Khosoussi, J. P. How, and Y. Girdhar, “Multi-robot distributed semantic mapping in unfamiliar environments through online matching of learned representations,” 2021, arXiv:2103.14805.
[29] A. Rosinol, A. Gupta, M. Abate, J. Shi, and L. Carlone, “3D dynamic scene graphs: Actionable spatial perception with places, objects, and humans,” 2020, arXiv:2002.06289.
[30] F. Wang and B. Englot, “Fast, accurate Gaussian process occupancy maps via test-data octrees and nested Bayesian fusion,” in Proc. IEEE Int. Conf. Robot. Automat., May 2016, pp. 1003–1010.
[31] M. G. Jaddidi, J. V. Miro, and G. Dissanayake, “Warped Gaussian processes occupancy mapping with uncertain inputs,” IEEE Robot. Autom. Lett., vol. 2, no. 2, pp. 680–687, Jan. 2017.
[32] K. Doherty, T. Shan, J. Wang, and B. Englot, “Learning-aided 3D occupancy mapping with Bayesian generalized kernel inference,” IEEE Trans. Robot., vol. 35, no. 4, pp. 953–966, Aug. 2019.
1065

1066

1063

1062

1060

1049

1048

1045

1043

1002

1006

1009

1008

1007

1004

1003

1002

1031

1030

1035

1034

1033

1032

1039

1038

1036

1061

1047

1060

1052

1051

1050

1049

1048

1045

1043

1042

1040

1039

1038

1036

1035

1034

1033

1032

1031

1030

1029

1028

1027

1026

1025

1024

1023

1022

1021

1020

1019

1018

1017

1016

1015

1014

1013

1012

1011

1010

1009

1008

1007

1006

1005

1004

1003

1002

1001

1000

999

998

997

996

995

994

993

992

991

990

989

988

987

986

985

984

983

982

981

980

979

978

977

976

975

974

973

972

971

970

969

968

967

966

965

964

963

962

961

960

959

958

957

956

955

954

953

952

951

950

949

948

947

946

945

944

943

942

941

940

939

938

937

936

935

934

933

932

931

930

929

928

927

926

925

924

923

922

921

920

919

918

917

916

915

914

913

912

911

910

909

908

907

906

905

904

903

902

901

900

899

898

897

896

895

894

893

892

891

890

889

888

887

886

885

884

883

882

881

880

879

878

877

876

875

874

873

872

871

870

869

868

867

866

865

864

863

862

861

860

859

858

857

856

855

854

853

852

851

850

849

848

847

846

845

844

843

842

841

840

839

838

837

836

835

834

833

832

831

830

829

828

827

826

825

824

823

822

821

820

819

818

817

816

815

814

813

812

811

810

809

808

807

806

805

804

803

802

801

800

799

798

797

796

795

794

793

792

791

790

789

788

787

786

785

784

783

782

781

780

779

778

777

776

775

774

773

772

771

770

769

768

767

766

765

764

763

762

761

760

759

758

757

756

755

754

753

752

751

750

749

748

747

746

745

744

743

742

741

740

739

738

737

736

735

734

733

732

731

730
JOEY WILSON (Graduate Student Member, IEEE) received the B.S. degree in computer engineering from California Polytechnic State University San Luis Obispo (Cal Poly), CA, USA, in 2019. He is currently pursuing the Ph.D. degree with the Robotics Institute, University of Michigan, Ann Arbor, MI, USA. His current research interest includes scene understanding in dynamic environments for autonomous systems.

LU GAN received the M.S. and Ph.D. degrees in robotics from the University of Michigan (UM), Ann Arbor, MI, USA, in 2021 and 2022, respectively. She is currently a Postdoctoral Scholar with the Graduate Aerospace Laboratories, California Institute of Technology (GALCIT), Pasadena, CA, USA. Her current research interests include computer vision and perception and navigation for autonomous systems.

ANDREW CAPODIECI is with Neya Systems for nine years, where he is currently the Leader of the Robotics Integration Group. During this time, he has managed and performed applied research in all areas of the robotics stack including, kinodynamically-feasible path planning in congested spaces, traversability estimation in off-road terrain, and object detection and classification. He is focused on transitioning Neya’s state-of-the-art autonomy research into fieldable, robust autonomy capabilities that deliver value to the warfighter. He has led numerous multi-million-dollar programs, including Neya’s work on GVSC’s Combat Vehicle Robotics Program, and programs to develop autonomous construction vehicles for the commercial sector.

PARAMSOTHY JAYAKUMAR received the B.Sc.Eng. degree (Hons.) from the University of Peradeniya, Sri Lanka, and the M.S. and Ph.D. degrees from Caltech. He worked at Ford Motor Company and BAE Systems. He is a Senior Technical Expert of analytics with U.S. Army DEVCOM Ground Vehicle Systems Center (GVSC). He has published over 200 papers in peer-reviewed literatures. He is a fellow of the Society of Automotive Engineers and the American Society of Mechanical Engineers. He received the DoD Laboratory Scientist of the Quarter Award, the NATO Applied Vehicle Technology Panel Excellence Awards, the SAE Arch T. Colwell Cooperative Engineering Medal, the SAE James M. Crawford Technical Standards Board Outstanding Achievement Award, the BAE Systems Chairman’s Award, and the NDIA GVSETS Best Paper Awards. He is also an Associate Editor of the ASME Journal of Autonomous Vehicles and Systems, and the Editorial Board Member of the International Journal of Vehicle Performance and the Journal of Terramechanics.

KIRA BARTON (Senior Member, IEEE) received the Ph.D. degree in mechanical engineering from the University of Illinois, Urbana–Champaign, IL, USA, in 2010. She is currently an Associate Professor with the Department of Mechanical Engineering, Robotics Institute, University of Michigan, Ann Arbor, MI, USA. Her research interests include control theory and applications including high precision motion control, iterative learning control, and control for autonomous vehicles.

MAANI GHAFFARI (Member, IEEE) received the Ph.D. degree from the Centre for Autonomous Systems (CAS), University of Technology Sydney, NSW, Australia, in 2017. He is currently an Assistant Professor with the Department of Naval Architecture and Marine Engineering, Robotics Institute, University of Michigan, Ann Arbor, MI, USA. He recently established the Computational Autonomy and Robotics Laboratory. His research interests include theory and applications of robotics and autonomous systems.