Stochastic Occupancy Grid Map Prediction in Dynamic Scenes

Anonymous Author(s)
Affiliation
Address
email

Abstract: This paper presents two variations of a novel stochastic prediction algorithm that enables mobile robots to accurately and robustly predict the future state of complex dynamic scenes. The proposed algorithm uses a variational autoencoder to predict a range of possible future states of the environment. The algorithm takes full advantage of the motion of the robot itself, the motion of dynamic objects, and the geometry of static objects in the scene to improve prediction accuracy. Three simulated and real-world datasets collected by different robot models are used to demonstrate that the proposed algorithm is able to achieve more accurate and robust prediction performance than other prediction algorithms. Furthermore, a predictive uncertainty-aware planner is proposed to demonstrate the effectiveness of the proposed predictor in simulation and real-world navigation experiments. Implementations are open source at https://github.com/xxx.

Keywords: Environment Prediction, Probabilistic Inference, Robot Learning

1 Introduction

Autonomous mobile robots are beginning to enter people's lives and try to help us provide different last-mile delivery services, such as moving goods in warehouses or hospitals and assisting grocery shoppers [1–3]. To realize this vision, robots are required to safely and efficiently navigate through complex and dynamic environments that are full of people and/or other robots. The first prerequisite for robots to navigate and perform tasks is to use their sensors to perceive the surrounding environment. This work focuses on the next step, which is to accurately and reliably predict how the surrounding environment will change based on these sensor data. This will allow mobile robots to proactively act to avoid potential future collisions, a key part of autonomous robot navigation.

Environment prediction remains an open problem as the future state of the environment is unknown, complex, and stochastic. Many interesting works have focused on this prediction problem. Traditional object detection and tracking methods [4, 5] use multi-stage procedures, hand-designed features, and explicitly detect and track objects. More recently, deep learning (DL)-based methods that are detection and tracking-free have been able to obtain more accurate predictions [6–11]. Occupancy grid maps (OGMs), the most widely successful spatial representation in robotics, are the most common environment representation in these DL-based methods. This transforms the complex environment prediction problem into an OGM prediction problem, outlined in Figure 1. Since OGMs can be treated as images (both are 2D arrays of data), the multi-step OGM prediction problem can be thought of as a video prediction task, a well-studied problem in machine learning.

The most common technique for OGM prediction uses recurrent neural networks (RNNs), which are widely used in video prediction [12–14]. For example, Ondruska and Posner [6] first propose an RNN-based deep tracking framework to directly track and predict unoccluded OGM states from raw sensor data. Itkina et al. [7] directly adapt PredNet [13] to predict the dynamic OGMs (DOGM) in urban scenes. Following this line of thought, Toyungyernsub et al. [8] decouple the

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

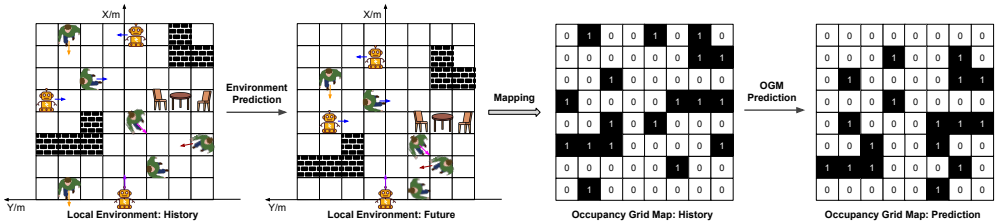


Figure 1: A simple illustration of the OGM prediction problem.

static and dynamic OGMs and propose a double-prong PredNet to predict occupancy states of the environment. Schreiber et al. [9, 10] embed the ConvLSTM units in the U-Net to capture spatiotemporal information of DOGMs and predict them in the stationary vehicle setting. Lange et al. [11] propose two attention-augmented ConvLSTM networks to capture long-range dependencies and predict future OGMs in the moving vehicle setting. However, these image-based works only focus on improving network architectures and just treat the OGMs as images, ignoring much useful information from the kinematics and dynamics behind the environment.

There are other DL-based approaches that do exploit the ego-motion and motion flow of the environment to improve the OGM prediction accuracy. By using input placement and recurrent states shifting to compensate for ego-motion, Schreiber et al. [15] extend their previous image-based works [9, 10] to predict DOGMs in moving vehicle scenarios. By extending the deep tracking framework [6] with a spatial transformer module to compensate for ego-motion, Dequaire et al. [16] propose a gated recurrent unit (GRU)-based network to predict future states in moving vehicle settings. Song et al. [17] propose a GRU-based LiDAR-FlowNet to estimate the forward/backward motion flow between two OGMs and predict future OGMs. By considering the ego-motion and motion flow together, Mohajerin and Rohani [18] first use the geometric image transformation to compensate for ego-motion, and then propose a ConvLSTM-based difference learning architecture to extract the motion difference between consecutive OGMs. However, most of these motion-based works are designed for autonomous vehicles and cannot be directly deployed on mobile robots with limited resources. Furthermore, all the above-described works assume that the environmental state is deterministic and cannot estimate the uncertainty of future states, which we believe is key to helping robots robustly navigate in dynamic environments.

In this paper, we propose two versions of a variational autoencoder (VAE)-based stochastic OGM predictor for resource-limited mobile robots, namely SOGMP and SOGMP++, both of which predict a distribution of possible future states of dynamic scenes. The primary contribution of our approach is that we fully exploit the kinematics and dynamics of the robot and its surrounding objects to improve prediction performance. Specifically, we first develop a simple and effective ego-motion compensation mechanism for robot motion, then utilize a ConvLSTM unit to extract the spatiotemporal motion information of dynamic objects, and finally generate a local environment map to interpret static objects. Another key contribution is that by relaxing the deterministic environment assumption, our proposed approaches are able to provide uncertainty estimates and predict a distribution over future states of the environment with the help of variational inference techniques. We demonstrate the effectiveness of our approaches by using both computer vision metrics and multi-target tracking metrics on three simulated and real-world datasets with different robot models, showing that our proposed predictors have a smaller absolute error, higher structural similarity, higher tracking accuracy than other state-of-the-art approaches, and can provide robust and diverse uncertainty estimates for future OGMs. Note that while all other published works only evaluate the image quality performance of the predicted OGMs, to the best of our knowledge, we are the first to employ a multi-object tracking metric, optimal subpattern assignment metric (OSPA) [19], to more fully evaluate OGM prediction performance in terms of tracking accuracy. In addition, we propose a predictive uncertainty-aware planner by integrating the predicted and uncertain costmaps from our predictor, and demonstrate its superior navigation performance in dynamic simulated and real-world environments.

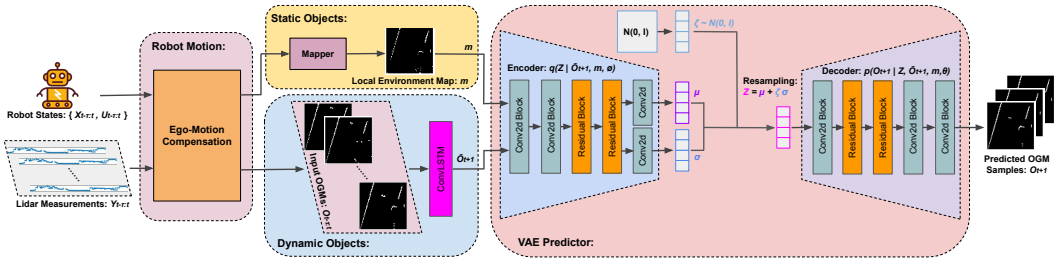


Figure 2: System architecture of the SOGMP++ predictor (SOGMP omits the Static Objects block).

2 Stochastic Occupancy Grip Map Predictor

2.1 Problem Formulation

In a complex dynamic environment with both static and dynamic obstacles, the robot is equipped with a lidar sensor to sense its surroundings and uses the OGM to represent the environment. We assume each grid cell in the OGM is either occupied or free, *i.e.,* a binary OGM. We assume that the robot is able to obtain relatively accurate estimates of its own pose and velocities from its odometry sensors or other localization algorithms over short periods of time (on the order of 1 s). We denote the pose and velocity of the robot at time t by $\mathbf{x}_t = [x_t \; y_t \; \theta_t]^T$ and $\mathbf{u}_t = [v_t \; w_t]^T$ respectively. Let $\mathbf{y}_t = [r_t \; b_t]^T$ denote the lidar measurements (range r and bearing b) at time t, and $\mathbf{o}_t$ denote the OGM at time t. Giving a history of τ lidar measurements $\mathbf{y}_{t-\tau:t}$ and robot states $\{\mathbf{x}_{t-\tau:t}, \mathbf{u}_{t-\tau:t}\}$, the robot needs to predict the future state (*i.e.,* OGM) of the environment $\mathbf{o}_{t+1}$. Then, this environment prediction problem can be formulated as a prediction model:

$$p_\theta(\mathbf{o}_{t+1} \mid \mathbf{y}_{t-\tau:t}, \mathbf{x}_{t-\tau:t}, \mathbf{u}_{t-\tau:t}), \tag{1}$$

where θ are the model parameters. The goal is to find the optimal θ to maximize (1).

Note that in this paper, we set $\tau = 10$ and limit the physical size of OGMs to $[0, 6.4]$ m along the x axis (forward) and $[-3.2, 3.2]$ m along the y axis (left). We use a cell size of 0.1 m, resulting in 64×64 OGMs. All data $\mathbf{o}, \mathbf{u}, \mathbf{y}$ are represented in the local coordinate frame of the robot.

2.2 System Overview

Before describing our proposed SOGMP and SOGMP++ methods, we first briefly discuss image-based prediction methods. The prediction model (1) of image-based approaches is rewritten as

$$p_\theta(\mathbf{o}_{t+1} \mid \mathbf{y}_{t-\tau:t}, \mathbf{x}_{t-\tau:t}, \mathbf{u}_{t-\tau:t}) = f_\theta(\mathbf{o}_{t-\tau:t}), \qquad \mathbf{o}_{t-\tau:t} = c(\mathbf{y}_{t-\tau:t}), \tag{2}$$

where $c(\cdot)$ is the conversion function to convert the lidar measurements to the binary OGMs in the robot's local coordinate frame. From this image-based model (2), we can easily see that image-based methods ignore the kinematics and dynamics of the robot and surrounding objects, assume a deterministic future, and fail to provide a range of possible and reliable OGM predictions.

Based on these limitations, we argue that: 1) the future state of the environment depends on the motion of the robot itself, the motion of dynamic objects, and the state of static objects within the environment; 2) the future state of the environment is stochastic and unknown, and a range of possible future states helps provide robust predictions. With these two assumptions, we fully exploit the kinematic and dynamic information of these three different types of objects, utilize the VAE-based network to provide stochastic predictions, and finally propose two novel stochastic OGM predictors (*i.e.,* SOGMP and SOGMP++, shown in Figure 2) to predict the future state of the environment. Since the only difference between the SOGMP and SOGMP++ is that the SOGMP++ considers the static objects and the SOGMP does not, we mainly describe the SOGMP++ model. The prediction

model (1) of our SOGMP++, outlined in Figure 2, can be rewritten as

$$p_\theta(\mathbf{o}_{t+1} \mid \mathbf{y}_{t-\tau:t}, \mathbf{x}_{t-\tau:t}, \mathbf{u}_{t-\tau:t}) = p_\theta(\mathbf{o}_{t+1} \mid \hat{\mathbf{o}}_{t+1}, \mathbf{m}), \tag{3a}$$

$$\hat{\mathbf{o}}_{t+1} = h(\mathbf{o}_{t-\tau:t}),\ \mathbf{o}_{t-\tau:t} = c(\mathbf{y}^R_{t-\tau:t}), \tag{3b}$$

$$\mathbf{m} = g(\mathbf{y}^R_{t-\tau:t}), \tag{3c}$$

$$\mathbf{y}^R_{t-\tau:t} = k(\mathbf{y}_{t-\tau:t}, \mathbf{x}_{t-\tau:t}, \mathbf{u}_{t-\tau:t}). \tag{3d}$$

The VAE predictor module corresponds to (3a). The dynamic object's module corresponds to (3b), where $h(\cdot)$ is the time series data processing function for dynamic objects, $c(\cdot)$ is the conversion function like (2), and R denotes the local coordinate frame of the robot at predicted time $t+n$. The static object's module corresponds to (3c), where $g(\cdot)$ is the occupancy grid mapping function for static objects. Finally, the robot motion module corresponds to (3d), where $k(\cdot)$ is the transformation function for robot motion compensation. Note that this prediction model (3) only predicts the future state at the next time step $t+1$. To predict a multi-step future state at time $t+n$, we can easily utilize the autoregressive mechanism and feed the next-step prediction back $\mathbf{o}_{t+1}$ to our SOGMP/SOGMP++ network (3a) for $n-1$ time steps to predict the future state at time step $t+n$.

2.3 Robot Motion

To account for the robot motion in dynamic scenarios, we propose a simple and effective ego-motion compensation mechanism $k(\cdot)$ to mitigate its dynamic effects and allow the environment dynamics to be consistent in the robot's local coordinate frame. To predict the future OGM state in the robot's local future view, we consider the robot's future ego-motion and transform the observed OGMs $\mathbf{o}_{t-\tau:t}$ to the robot's local coordinate frame at prediction time step $t+n$. So, our proposed ego-motion compensation mechanism can be divided into two steps: robot pose prediction and coordinate transformation. To account for the robot's future ego motion, we first use a constant velocity motion model to predict the robot's future pose. Then, we use a homogeneous transformation matrix to transform the robot poses $\mathbf{x}_{t-\tau:t}$ and lidar measurements $\mathbf{y}_{t-\tau:t}$ to the robot' local future coordinate frame R and compensate for its ego-motion (see Appendix A.1 and Appendix A.2 for details).

2.4 Dynamic Objects

Since we use the OGM to represent the environmental state, we first need to implement a conversion function $c(\cdot)$ to convert the compensated lidar measurements $\mathbf{y}^R_{t-\tau:t}$ to the corresponding OGMs $\mathbf{o}_{t-\tau:t}$ by using coordinates to subscripts conversion equations (see Appendix A.3 for details).

Tracking and predicting dynamic objects such as pedestrians is the hardest part of environmental prediction in complex dynamic scenes. It requires some technique to process a set of time series data to capture the motion information. While the traditional particle-based methods [5] require explicitly tracking objects and treat each grid cell as an independent state, recent learning-based methods [7–11, 15–18] prefer to use RNNs to directly process the observed time series OGMs. Based on these trends, we choose the most popular ConvLSTM unit to process the spatiotemporal OGM sequences $\mathbf{o}_{t-\tau:t}$. However, while other works [8, 14] explicitly decouple the dynamic and static/unknown objects and use different networks to process them separately, we argue that the motion of dynamic objects is related to their surroundings, and that explicit disentangling may lose some useful contextual information. For example, pedestrians walking through a narrow corridor are less likely to collide with or pass through surrounding walls. To exploit the useful contextual information between dynamic objects and their surrounding, we directly feed the observed OGMs $\mathbf{o}_{t-\tau:t}$ into a ConvLSTM unit $h(\cdot)$ and implicitly predict the future state $\hat{\mathbf{o}}_{t+1}$ of dynamic objects.

2.5 Static Objects

While predicting dynamic objects plays a key role in environmental prediction, paying extra attention to static objects is also important to improve prediction accuracy. The main reason is that the area occupied by static objects is much larger than that of dynamic objects, as shown in Figure 2,

where static objects such as walls are consistently clustered together, while dynamic objects such as pedestrians are sparse and scattered point clusters. Another reason is that static objects contribute to the scene geometry and give a global view of the surroundings, where these static objects maintain their shape and position over time. To account for them, we utilize a local static environment map $\mathbf{m}$ as a prediction for future static objects, which is a key contribution of our work. We generate this local environment map $\mathbf{m}$ using the standard inverse sensor model [20]. This Bayesian approach generates a robust estimate of the local map, where dynamic objects are treated as noise data and removed over time. We speed up this step by implementing a GPU-accelerated OGM mapping algorithm $g(\cdot)$ that parallelizes the independent cell state update operations (see Appendix A.4 for details).

2.6 VAE Predictor

If we treat the dynamic prediction function (3b) and the static prediction function (3c) as the prediction step of Bayes filters, then the VAE predictor (3a) is the update step of Bayes filters. It fuses the predicted features of static objects from (3c) and dynamic objects from (3b), and finally predicts a distribution over future states of the environment. A range of possible future environment states allows the robot to capture the uncertainty of the environment and enable risk-aware operational behavior in complex dynamic scenarios. To represent the stochasticity of environment states, we assume that environment states $\mathbf{o}_{t-\tau:t+n}$ are generated by some unobserved, random, latent variables $\mathbf{z}$ that follow a prior distribution $p_\theta(\mathbf{z})$. Then, our VAE predictor model (3a) can be rewritten as

$$p_\theta(\mathbf{o}_{t+1} \mid \hat{\mathbf{o}}_{t+1}, \mathbf{m}) = \int p_\theta(\mathbf{z}) p_\theta(\mathbf{o}_{t+1} \mid \mathbf{z}, \hat{\mathbf{o}}_{t+1}, \mathbf{m}) dz. \tag{4}$$

Since we are unable to directly optimize this marginal likelihood and obtain optimal parameters θ, we use a VAE network to parameterize our prediction model $p_\theta(\mathbf{o}_{t+1} \mid \hat{\mathbf{o}}_{t+1}, \mathbf{m})$, outlined in Figure 2, where the inference network (encoder) parameterized by ϕ refers to the variational approximation $q_\phi(\mathbf{z} \mid \hat{\mathbf{o}}_{t+1}, \mathbf{m})$, the generative network (decoder) parameterized by θ refers to the likelihood $p_\theta(\mathbf{o}_{t+1} \mid \mathbf{z}, \hat{\mathbf{o}}_{t+1}, \mathbf{m})$, and the standard Gaussian distribution $\mathcal{N}(0, 1)$ refers to the prior $p_\theta(\mathbf{z})$. Then, from work [21], we can simply maximize the evidence lower bound (ELBO) loss $\mathcal{L}(\theta, \phi; \mathbf{o}_{t+1})$ to optimize this marginal likelihood and get optimal θ:

$$\mathcal{L}(\theta, \phi; \mathbf{o}_{t+1}) = \mathbb{E}_{q_\phi(\mathbf{z}|\hat{\mathbf{o}}_{t+1}, \mathbf{m})} \left[\log p_\theta(\mathbf{o}_{t+1} \mid \mathbf{z}, \hat{\mathbf{o}}_{t+1}, \mathbf{m})\right] - KL\big(q_\phi(\mathbf{z} \mid \hat{\mathbf{o}}_{t+1}, \mathbf{m}) \,\|\, p_\theta(\mathbf{z})\big). \tag{5}$$

The first term on the right-hand side (RHS) is the expected generative error, describing how well the future environment states can be generated from the latent variable $\mathbf{z}$. The second RHS term is the Kullback–Leibler (KL) divergence, describing how close the variational approximation is to the prior.

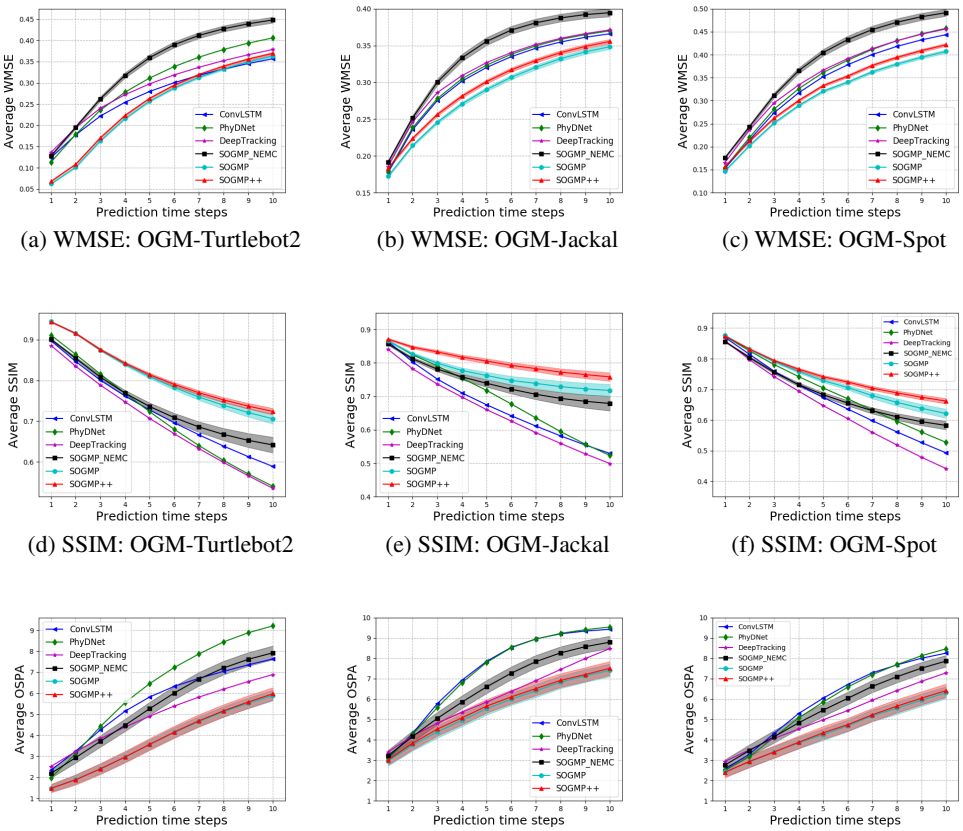
Finally, we use mini-batching, Monte-Carlo estimation, and reparameterization tricks to calculate the stochastic gradients of the ELBO (5) [21], and obtain the optimized model parameters ϕ and θ. Using them, our VAE predictor can integrate the predicted features $\{\hat{\mathbf{o}}_{t+1}, \mathbf{m}\}$ of dynamic and static objects, and output a probabilistic estimate of the future OGM states with uncertainty awareness.

3 Experiments and Results

To demonstrate the prediction performance of our proposed approaches, we first test our algorithms on a self-collected simulated dataset and two public real-world sub-datasets from the socially compliant navigation dataset (SCAND) [22]. Second, we characterize the uncertainty of our proposed predictors across different sample sizes and numbers of objects. Finally, we propose a predictive uncertainty-aware planner by using the prediction and uncertainty information from our predictors to demonstrate how it improves robot navigation performance in crowded dynamic simulated/real-world environments. See also Appendix B.1 for experimental evaluation of the run time of different prediction algorithms.

3.1 Prediction Results

Dataset: To train our networks and baselines, we collected an OGM dataset, called the OGM-Turtlebot2 dataset, using the 3D human-robot interaction simulator with a 0.8 m/s Turtlebot2



(a) WMSE: OGM-Turtlebot2 (b) WMSE: OGM-Jackal (c) WMSE: OGM-Spot

(d) SSIM: OGM-Turtlebot2 (e) SSIM: OGM-Jackal (f) SSIM: OGM-Spot

(g) OSPA: OGM-Turtlebot2 (h) OSPA: OGM-Jackal (i) OSPA: OGM-Spot

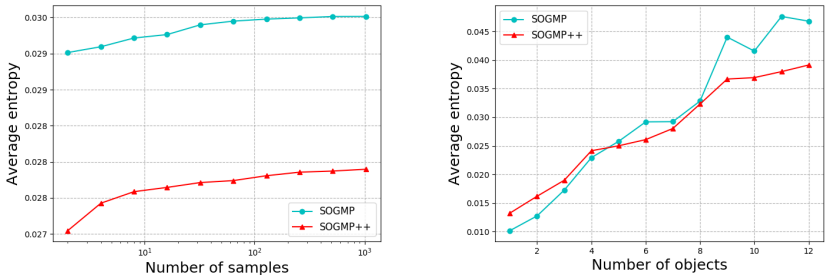
Figure 3: Average WMSE, average SSIM, and average OSPA of 10 different prediction time steps for all tested methods on 3 different datasets. Note that the shadows for SOGMP_NEMC, SOGMP and SOGMP++ approaches are plotted with 95% confidence interval over 32 samples. Note that for all curves for WMSE and OSPA, lower is better. For all curves for SSIM, higher is better.

robot [23]. We collected a total of 94,891 $(\mathbf{x}, \mathbf{u}, \mathbf{y})$ tuples, where 17,000 tuples are used for testing. In addition, to fairly evaluate all networks and examine performance in the real world, we extracted two real-world sub-datasets (*i.e.,* $(\mathbf{x}, \mathbf{u}, \mathbf{y})$ tuples) from the public SCAND [22] dataset: OGM-Jackal dataset (collected by a 2m/s Jackal robot) and OGM-Spot dataset (collected by a 1.6m/s Spot robot). These two real-world datasets are only used for testing. See Appendix A.5 for more details.

Baselines: We compare our proposed SOGMP and SOGMP++ algorithms with four DL-based baselines: ConvLSTM [12], DeepTracking [6], PhyDNet [14], and an ablation baseline without the ego-motion compensation module (*i.e.,* SOGMP_NEMC). Note that all networks were implemented by PyTorch framework [24] and trained using the self-collected OGM-Turtlebot2 dataset.

3.1.1 Quantitative Results

We evaluate all test OGM predictors on the three datasets from above by evaluating the absolute error, structural similarity, and tracking accuracy (see Appendix A.6 for details). Figure 3 shows the detailed results obtained from these tests, from which we can observe four phenomena. First, our proposed SOGMP predictors with ego-motion compensation achieve significantly better average WMSE, SSIM, and OSPA than the SOGMP_NEMC baseline without ego-motion compensation in all test datasets, which illustrates the effectiveness of ego-motion compensation. Second, the average WMSE of our proposed SOGMP predictors at different prediction time steps is lower than the other image-based baselines in all three test datasets collected by different robots. This shows that the



(a) Entropy vs. Number of samples (b) Entropy vs. Number of objects

Figure 4: Average entropy of our SOGMP and SOGMP++ predictors at 5th prediction time step.

proposed SOGMP predictors utilizing kinematic and dynamic information are able to predict more accurate OGMs (*i.e.,* smaller absolute error) than the state-of-art image-based approaches. Third, while our SOGMP predictors achieve the highest average SSIMs in all test datasets, it is interesting that the average SSIM of our SOGMP++ with a local environment map is significantly higher than that of SOGMP without a local environment map in long-term predictions over multiple time steps. This indicates that the local environment map, which accounts for static objects, helps for longer-term prediction. Finally, the average OSPA errors of our proposed SOGMP predictors are significantly lower than that of the other four baselines, especially in longer prediction time steps. This further demonstrates the preferential performance of our proposed motion-based methods in tracking or predicting environmental states (*i.e.,* localization and cardinality) over other image-based methods. However, the average OSPA errors of our proposed SOGMP and SOGMP++ predictors are almost the same. We believe that this is because static objects are more persistent than dynamic objects, so the static map does not provide significant benefits here. See also Appendix B.2 for examples of predicted OGMs and qualitative analysis.

3.1.2 Uncertainty Characterization

One of the biggest differences of our proposed methods compared to other previous baseline works is that they can provide uncertainty estimates. To demonstrate the diversity and consistency of uncertainty estimates, we run two experiments to characterize and analyze the output distribution of our SOGMP/SOGMP++ predictors. One is to show how the entropy of the final probabilistic OGM changes as the number of OGM samples increases, with the hypothesis that it will level off at some value well below that of the entropy of a uniform distribution. The other is to show how the entropy of the final probabilistic OGM changes as the number of objects in the OGM increases, with the hypothesis that it will increase with the number of objects in it. Figure 4 shows the entropy results of these experiments on the OGM-Turtlebot2 dataset, which validates our hypotheses and demonstrates the consistency of our VAE-based predictors. See Appendix A.7 for experimental details.

3.2 Navigation Results

We next test the applicability of our proposed methods to practical robotics applications. Figure 5 shows our planner, where we add two new costmaps to the standard ROS [25] navigation stack using the outputs of our SOGMP predictor: one for the predicted scene and one for the uncertainty of the scene. See Appendix A.8 for additional details about the experimental setup.

3.2.1 Simulation Results

Using the Turtlebot2 robot in a simulated lobby environment with different crowd densities, as in [23, 26], we compare our proposed predictive uncertainty-aware planner (*i.e.,* DWA-PU) with four navigation baselines: supervised-learning-based CNN [26], deep reinforcement learning-based

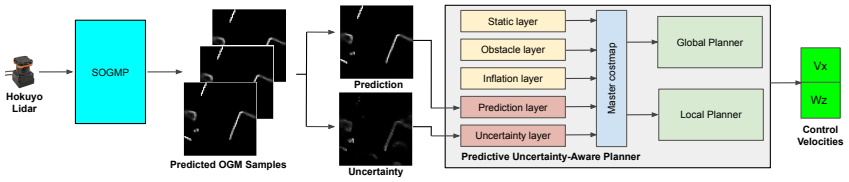


Figure 5: System architecture of the predictive uncertainty-aware navigation planner.

Table 1: Navigation results at different crowd densities

Environment	Method	Success Rate	Average Time (s)	Average Length (m)	Average Speed (m/s)
Lobby world, 35 pedestrians	CNN [26]	0.81	14.30	5.40	0.38
	A1-RC [27]	0.77	16.81	6.89	**0.41**
	DWA [28]	0.82	14.18	5.15	0.36
	DWA-P	0.86	**13.79**	5.12	0.34
	DWA-PU	**0.89**	14.90	**5.12**	0.34
Lobby world, 45 pedestrians	CNN [26]	0.79	16.65	5.62	0.34
	A1-RC [27]	0.77	**14.65**	6.28	**0.43**
	DWA [28]	0.77	15.39	5.16	0.34
	DWA-P	0.79	14.84	**5.14**	0.35
	DWA-PU	**0.82**	15.96	5.17	0.32

A1-RC [27], model-based DWA [28], and an ablation baseline without the uncertainty costmap (*i.e.,* DWA-P). Table 1 summarizes these navigation results. As can be seen, our DWA-PU policy has the highest success rate in each crowd size, while having almost the shortest path length. This shows that the prediction costmap from our SOGMP predictor is able to help the traditional DWA planner to provide safer and shorter paths, and combining it with its associated uncertainty costmap can achieve a much better navigation performance than other baselines. It demonstrates the effectiveness of our proposed predictors in helping design safe robot navigation policies in crowded dynamic scenes. See Appendix B.3 for examples of paths planned by each approach and qualitative analysis.

3.2.2 Hardware Results

Besides the simulated experiments, we also conduct a real-world experiment to demonstrate the applicability of our SOGMP-based DWA-PU policy in the real world. Specifically, we deploy our SOGMP predictor and DWA-PU control policy to a real Turtlebot2 robot with a 2D Hokuyo lidar and an Nvidia Jetson Xavier computer, and let it navigate through a crowded dynamic corridor in a natural condition. From the attached Multimedia, it can be seen that the robot can safely navigate to its goal points among crowded pedestrians without any collision, which demonstrates the real-world effectiveness of our proposed SOGMP predictors and DWA-PU planner.

4 Limitations and Future Work

In this paper, we propose two versions of a novel VAE-based OGM prediction algorithm that provides mobile robots with the ability to accurately and robustly predict the future state of crowded dynamic scenes. In addition, we integrate our predictors with ROS navigation stack to propose a predictive uncertainty-aware planner and demonstrate its effectiveness on the problem of robot crowded navigation.

We found that the prediction accuracy of SOGMP was poor when: 1) the robot moves erratically or rotates rapidly; 2) the robot's field of view is occluded by a large obstacle. We also were not able to compare against of any of the DL-based ego-motion compensation methods [15–18] as they do not have open-source implementations and our attempts to recreate the results were not successful.

Our future work will leverage more accurate robot motion models and correct jump steering angles to overcome these limitations. We will continue to explore how to integrate our uncertainty-aware predictors into learning-based control policies to further improve robot navigation performance in crowded dynamic scenes.

References

[1] R. ED. Types and applications of autonomous mobile robots (amrs). https://www.conveyco.com/blog/types-and-applications-of-amrs, July 2022. (Accessed on 08/20/2022).

[2] J.-u. Kim. Keimyung hospital demonstrates smart autonomous mobile robot. https://www.koreabiomed.com/news/articleView.html?idxno=10585, Mar 2021. (Accessed on 08/20/2022).

[3] SICK. Revolutionizing grocery shopping with mobile robots. https://sickusablog.com/revolutionizing-grocery-shopping-mobile-robots, Mar 2021. (Accessed on 08/20/2022).

[4] A. Ess, K. Schindler, B. Leibe, and L. Van Gool. Object detection and tracking for autonomous navigation in dynamic environments. *The International Journal of Robotics Research*, 29(14): 1707–1725, 2010.

[5] D. Nuss, S. Reuter, M. Thom, T. Yuan, G. Krehl, M. Maile, A. Gern, and K. Dietmayer. A random finite set approach for dynamic occupancy grid maps with real-time application. *The International Journal of Robotics Research*, 37(8):841–866, 2018.

[6] P. Ondruska and I. Posner. Deep tracking: Seeing beyond seeing using recurrent neural networks. In *Thirtieth AAAI conference on artificial intelligence*, 2016.

[7] M. Itkina, K. Driggs-Campbell, and M. J. Kochenderfer. Dynamic environment prediction in urban scenes using recurrent representation learning. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 2052–2059. IEEE, 2019.

[8] M. Toyungyernsub, M. Itkina, R. Senanayake, and M. J. Kochenderfer. Double-prong convlstm for spatiotemporal occupancy prediction in dynamic environments. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13931–13937. IEEE, 2021.

[9] M. Schreiber, S. Hoermann, and K. Dietmayer. Long-term occupancy grid prediction using recurrent neural networks. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9299–9305. IEEE, 2019.

[10] M. Schreiber, V. Belagiannis, C. Gläser, and K. Dietmayer. Motion estimation in occupancy grid maps in stationary settings using recurrent neural networks. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8587–8593. IEEE, 2020.

[11] B. Lange, M. Itkina, and M. J. Kochenderfer. Attention augmented convlstm for environment prediction. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1346–1353. IEEE, 2021.

[12] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.

[13] W. Lotter, G. Kreiman, and D. Cox. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*, 2016.

[14] V. L. Guen and N. Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11474–11484, 2020.

[15] M. Schreiber, V. Belagiannis, C. Gläser, and K. Dietmayer. Dynamic occupancy grid mapping with recurrent neural networks. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6717–6724. IEEE, 2021.

[16] J. Dequaire, P. Ondrúška, D. Rao, D. Wang, and I. Posner. Deep tracking in the wild: End-to-end tracking using recurrent neural networks. *The International Journal of Robotics Research*, 37 (4-5):492–512, 2018.

[17] Y. Song, Y. Tian, G. Wang, and M. Li. 2d lidar map prediction via estimating motion flow with gru. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6617–6623. IEEE, 2019.

[18] N. Mohajerin and M. Rohani. Multi-step prediction of occupancy grid maps with recurrent neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10600–10608, 2019.

[19] D. Schuhmacher, B.-T. Vo, and B.-N. Vo. A consistent metric for performance evaluation of multi-object filters. *IEEE Transactions on Signal Processing*, 56(8):3447–3457, 2008. doi:10.1109/TSP.2008.920469.

[20] S. Thrun. Learning occupancy grid maps with forward sensor models. *Autonomous robots*, 15 (2):111–127, 2003.

[21] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[22] H. Karnan, A. Nair, X. Xiao, G. Warnell, S. Pirk, A. Toshev, J. Hart, J. Biswas, and P. Stone. Socially compliant navigation dataset (scand): A large-scale dataset of demonstrations for social navigation. *arXiv preprint arXiv:2203.15041*, 2022.

[23] Z. Xie and P. Dames. Drl-vo: Learning to navigate through crowded dynamic scenes using velocity obstacles. *IEEE Transactions on Robotics*, pages 1–20, 2023. doi:10.1109/TRO.2023.3257549.

[24] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8026–8037, 2019.

[25] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[26] Z. Xie, P. Xin, and P. Dames. Towards safe navigation through crowded dynamic environments. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4934–4940. IEEE, 2021.

[27] R. Guldenring, M. Görner, N. Hendrich, N. J. Jacobsen, and J. Zhang. Learning local planners for human-aware navigation in indoor environments. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6053–6060. IEEE, 2020.

[28] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1):23–33, 1997.

[29] N. L. Baisa. Derivation of a constant velocity motion model for visual tracking. *arXiv preprint arXiv:2005.00844*, 2020.

[30] D. Helbing and P. Molnar. Social force model for pedestrian dynamics. *Physical Review E*, 51 (5):4282, 1995.

[31] M. Moussaïd, D. Helbing, S. Garnier, A. Johansson, M. Combe, and G. Theraulaz. Experimental study of the behavioural mechanisms underlying self-organization in human crowds. *Proceedings of the Royal Society B: Biological Sciences*, 276(1668):2755–2762, 2009.

[32] N. Ponomarenko, S. Krivenko, K. Egiazarian, V. Lukin, and J. Astola. Weighted mean square error for estimation of visual quality of image denoising methods. In *CD ROM Proceedings of VPQM*, volume 5. Scottsdale USA, 2010.

[33] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015.

[34] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[35] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[36] C. E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55, 2001.