2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

HabitatDyn 2.0: Dataset for Spatial Anticipation and Dynamic Object Localization

Zhengcheng Shen[1], Linh Kästner[1], Yi Gao and Jens Lambrecht[1]

***Abstract*—The ability of a robot to perceive and understand its environment is crucial for its actions and behavior. Humans are adept at using semantic information for object localization and path planning, a skill that robots need to emulate for intelligent adaptation in dynamic settings. Training of the spatial anticipation ability, which can enhance spatial perception through semantic understanding, necessitates the availability of appropriate data. Although extensive research has been conducted on datasets for outdoor environments, especially in the context of autonomous driving, there is still a notable lack of datasets specifically designed for indoor environments, with a focus on dynamic object localization. This paper introduces HabitatDyn 2.0, a dataset specifically designed for enhancing object localization capabilities with semantic information from a robot's perspective. Besides RGB videos, semantic annotations, and depth information, HabitatDyn 2.0 also features top-down view labels for dynamic objects, which is required for training the spatial anticipation ability based on semantic information. Additionally, an algorithm that leverages spatial anticipation for dynamic object localization is presented, trained, and evaluated on the dataset.**

***Index Terms*—Spatial Anticipation, Occupied Mapping, Object Detection, Object Localization, Navigation, Semantic,**

I. Introduction

Cognitive abilities of mobile robots are critical for seamless and safe navigation [1], [2], [3]. Understanding the kinetic information of objects, particularly dynamic objects, holds immense value in mobile robot navigation [4], [5]. The task of detecting and localizing the dynamic objects mostly relies on the combination of depth information and RGB observations from the sensors and involves a machine learning-based pipeline [6], [7], [8]. The challenges in conducting experiments for dynamic object detection and localization stem from several factors. First, there is the issue of inaccurate ground truth locations for the dynamic objects, which complicates accurate detection and tracking. Additionally, the experiments face limitations due to a lack of variety in backgrounds and dynamic object combinations, which are essential for robust testing. Observations from robot perspectives introduce another layer of complexity, as this viewpoint can differ significantly from standard observation angles. Lastly, the process of semantically labeling dynamic objects is both time-consuming and expensive, further hindering the development and testing of effective detection and localization systems [9], [10], [11], [12].

For performing the research in a dynamic environment, the paper introduces a specially designed dataset, HabitatDyn 2.0. Additionally, we propose a pipeline that employs a spatial anticipation algorithm to improve the localization accuracy of dynamic objects, serving as a practical demonstration of the dataset's utility. The algorithm enhances the spatial understanding through the prior knowledge learned from the dataset by spatial anticipation, which is shown in Fig. 1. The various dynamic objects and scenes combinations ensure the stability of the training and reliability of the validation. With different robot perspectives from HabitatDyn 2.0, we also investigated the influence of the height and perspective of the camera on dynamic object localization. Besides, the proposed methods can be used with any other first-person view segmentation methods without the influence on the detection rate. The improvement of localization accuracy through the incorporation of machine learning techniques further underscores the dataset's potential. The paper has the following contribution:

- HabitatDyn 2.0, a synthetic dataset specifically crafted for the training and evaluation of dynamic object localization tasks.
- An algorithm to leverage the semantic information in the RGB observation to extend the observation of the depth sensors or laser sensors for trained object categories.

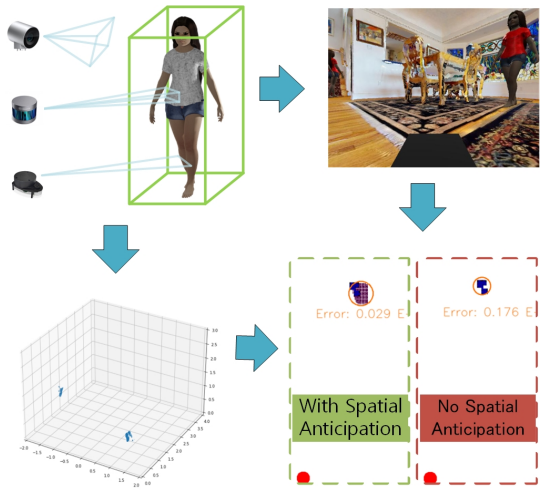

Fig. 1: The laser scanners can only gather limited information about the object's location due to self-occlusion. The computation cost for extracting semantic information from the laser points is also high. Therefore, the work gathers the semantic information from the RGB camera and uses it to enhance the perception of the mobile robots. With the proposed methods, we could improve the localization accuracy and annotate more occupied areas of an object.

[1] Zhengcheng Shen , Linh Kästner, Yi Gao and Jens Lambrecht are with the Chair Industry Grade Networks and Clouds, Faculty of Electrical Engineering, and Computer Science, Berlin Institute of Technology, Berlin, Germany zhengcheng.shen@tu-berlin.de

II. Related Works

In recent research, object location estimation and 3D object detection have emerged as popular areas of focus, both in outdoor and indoor environments [14], [15], [16], [17]. While many studies perform detection in a 3D space using RGB data and LIDAR/depth information, there are also approaches that leverage 2D detection results for 3D object detection and localization, mainly due to the flexibility and efficiency of 2D detection methods [18], [19]. Eppenberger et al. [20] introduced an algorithm in their work to reduce point cloud noise for dynamic object localization.

The 3D dynamic object detection ability is a desired extension for mobile robotics. The proposed approaches in [20], [21], [22] first estimated the absolute pose of the objects for motion detection, which also rely on an odometry system for self-localization. To detect the dynamic object without an odometry system, we have to leverage the semantic information in the first-person view camera. The well-developed salient object detection[23] can serve the purpose.

Spatial anticipation has been used for exploration in [9] and semantic mapping[24]. The methods are able to extract the semantic and geometry information for the different tasks with a simple network structure and moderate computation power. There exist a number of currently available dynamic object datasets such as [13], [11], [25], [26], or [27]. However, they fall short for tasks requiring kinematic analysis and evaluation, including pose and orientation estimation, velocity calculation, prediction of future movements, and distance measurement. Furthermore, the development of the spatial anticipation ability requires the bird-eye-view labeling of the dynamic objects for training.

III. Dataset Description

This section introduces and details the dataset, which is collected using the Habitat simulator and thoughtfully crafted to address the requirements arising from a blend of machine learning-driven segmentation algorithms and mobile robotics applications.

A. Data Amount and Diversity

The dataset contains 3780 short clips, each lasting 10 seconds and captured at 24 frames per second (fps) by an embodied agent in a dynamic environment using Habitat-sim, resulting in 241 frames per clip. This compiles into, 910980 annotated frames, offering an extensive dataset for training and evaluating the algorithm's proof of concept. Since it is created by the simulator, there is extremely limited error for the location record of the objects, which distinguishes itself from other real-world datasets. There are 108 clips for each scene, which has the three different objects' maximum speed categories, six different dynamic object combinations, various moving paths, and two different sensor configurations.

B. Data Splits

The dataset has three major splits, the training split, the validation split, and the evaluation split. The training split contains six different dynamic objects, which are depicted in Fig. 2. There are 25 different scenes in the training split, which results in 1080 clips for each high and low perspective. The moving paths of the agent and the object for the different perspectives are exactly matched, which enables fair cooperation between those two different perspectives. Besides the six trained dynamic objects, the validation dataset and evaluation dataset contain three extra dynamic objects, which have been shown in Fig. 3. The validation split can be used to adjust the hyperparameter for the machine learning-based pipeline. Furthermore, the evaluation data splits created from another four untrained scenes provide a detailed insight into the performance evaluation of the trained model.

Fig. 2: The trained objects in the dataset. The training dataset contains six salient objects, namely angry girl, toy cars, miniature cat, shiba, robot 2020, and ferbibliotecario

Fig. 3: The evaluation objects that not be trained. The evaluation dataset includes three untrained objects, namely large robot, human A, human B.

Further information about the dataset HabidtatDyn 2.0 can be viewed in the following repository: https://github.com/ignc-research/HabitatDyn-2.0.

C. Kinematics

Furthermore, the dataset contains the top-down view annotation for each frame, which is essential for training the spatial anticipation ability for dynamic object localization. Examples of the top-down view annotation are depicted in Fig. 4. The annotation is calculated based on the object model size, the location of the center point of the object, and the object orientation. With the extra labeling, the dataset can be used for applications like spatial anticipation training, object tracking and movement predictions.

IV. Algorithm Description

With the HabitatDyn 2.0 Dataset, the following algorithm demonstrated in Fig. 5 can be trained and evaluated. The

TABLE I: Comparison Between Different Datasets

Dataset	Depth	Semantic Mask	Scenes	Perspective	Dynamic	Agent Loc./Orient.	Top-down View Label	Data Size
KITTI-Optical Flow Eval. [10]	3D LiDAR	Yes	Road	Top of a moving car	Car, Bicycle, Pedestrian	Yes	No	8000 frames
Davis.[13]	No	Yes	Various	Human	Various	No	No	3455 frames
YouTube-VOS [11]	No	Yes	Various	Human	Various	No	No	133590 frames
FlyingThings3D [12]	Yes	Yes	Various	Human	Various	No	No	39000 frames
HabitatDyn 2.0 (Ours)	Yes	Yes	30 real apart-ments	Moving Mobile Robot	Nine Artificial Models	Yes	Yes	910980 frames

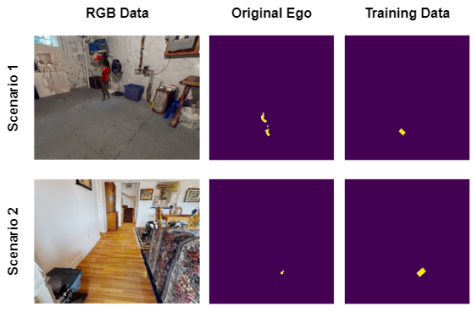

Fig. 4: The top-down view ground truth from HabitatDyn 2.0. The first column of the images is RGB data. The second column is the egocentric map for dynamic objects created from depth images. The last column shows the ground truth egocentric map provided by the dataset

entire algorithm takes RGBD information or RGB+LiDAR as input and gives the location information of the interested objects as the output. The first-person view segmentation algorithm can be any segmentation algorithm based on RGBD information, and it will label the interested objects with a first-person view mask[28], [29], [30]. In the paper, we use 3DC-Seg[30] as the object segmentation algorithm to demonstrate a salient object localization use case. On the other hand, the spatial anticipation layer, which is decoupled from the segmentation backend, can be trained and deployed separately and flexibly as long as the objects are included in its training data.

A. Spatial Anticipation Module

We treat the spatial anticipation task as a segmentation task for a top-down view image. The network should have the ability to classify the pixel if it is occupied by a salient object or not. Any network framework that serves the purpose can be used for the task. In the real use case, the module can use the same visual encoder with the first-person view segmentation methods, which may further reduce the potential computation overhead. Here we use a network introduced in [9], [24], which has a relatively small size and is efficient on the task compared to other segmentation approaches based on vision transformers [31]. The network consists of three different parts and is designed according to an Unet framework. Further detailed information about the network structure is described in [9], [24].

B. Loss Function

The loss function of the training contains two parts, one is the error from the difference between the ground truth object probability map and the predicted one. Another part is the error for the location estimation. In the training dataset, we also have a high data unbalance because most pixels in the training data are black. To avoid the neural network having the intention to predict fewer pixels, the weighted matrix will be used to balance the pixel distribution. All the balance factors are calculated in a batch base, and all the predictions in the same batch use the same parameters.

1) Parameter For Data Balance: There are three parameters we utilized to balance the occupied pixels and the unoccupied pixels. α is used to balance the occupied area, which is calculated by the following formula:

$$\alpha = \min(\frac{number\ of\ pixels}{No.\ of\ occu.\ pixels + \varepsilon}, \alpha_{max}) \tag{1}$$

γ is used for the edge area of the ground truth objects. It asks the spatial anticipation layer to predict the object with a clean edge. Before the calculation of γ, we dilate the egocentric map including all the ground truth objects map_{gt} with a 5x5 kernel and iteration of 3 and mark it as map_{gt_ext}. It is calculated according to:

$$\gamma = \min(\frac{number\ of\ pixels}{No.\ of\ occu.\ pixels\ of\ map_{gt\_ext} + \varepsilon}, \gamma_{max}) \tag{2}$$

β is used for the area with unexpected objects. We add extra weight to award the network when it reduces unexpected objects. To calculate the coefficient, we need to first calculate the image for all the unexpected predictions with the prediction map_{pre}. The egocentric map of the false positive prediction map_{fp} will be calculated as follows:

$$map_{fp}[i,j] = \begin{cases} 1 & map_{pre}[i,j] > 0.1\ and\ map_{gt}[i,j] < 0 \\ 0 & else \end{cases} \tag{3}$$

And β is calculated according to:

$$\beta = \min(\frac{number\ of\ pixels}{No.\ of\ occu.\ pixels\ of\ map_{fp} + \varepsilon}, \beta max) \tag{4}$$

In the equations, ε is set as 1e-4 to avoid overflow, α_{max} is 600, γ_{max} is 300, and β_{max} is 300.

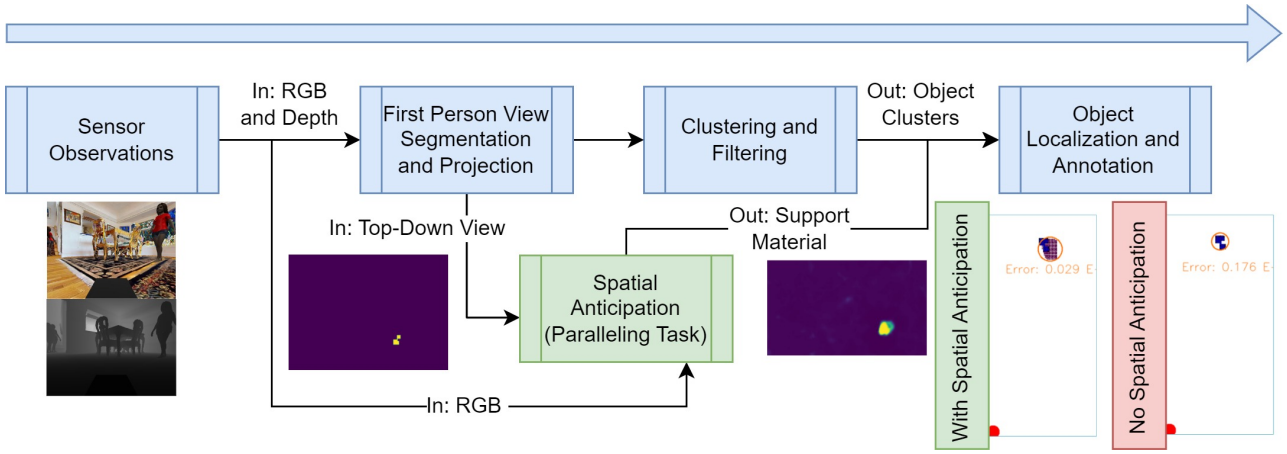

Fig. 5: System design of the algorithm. The system acquires the RGBD information as input. It follows a standard procedure for object segmentation clustering. In the meantime, a parallel task is performed by a spatial anticipation module, which provides the support information for further calculation of the objects' location. The extra module will not introduce a significant time overhead and substantially reduce the error of the object localization which is caused by self-occlusion.

2) Loss Function of Egocentric Map: After the calculation of the balance factors, we could use them for the first part of the loss function, which awards the network if it predicts the ground truth egocentric map right. We use the weighted binary cross entropy which contains a weight tensor W, the weight tensor is different from each prediction (even in the same batch) and is calculated according to:

$$W = \alpha map_{gt} + \gamma map_{gt\_ext} + \beta map_{fp} + 1 \tag{5}$$

The loss function will then be:

$$loss_{map} = Binary\ Cross\ Entropy(map_{pre},\ map_{gt},\ W) \tag{6}$$

3) Loss Function of Position Estimation: The other part of the loss function is designed for position estimation. To calculate the position of the object, we first have to define 2 position matrices X and Z. Because the maximum number of objects in the dataset is six, we use it for the tensor construction. It is worth mentioning that there is no object number limitation in the real deployment. Both X and Z have the same dimension with the egocentric map, and they are defined as follows:

$$X[i,j] = j - (map\ size + 1)/2 \tag{7}$$

$$Z[i,j] = i - map\ size + 1 \tag{8}$$

On the other hand, we need to match the prediction with the ground truth object if we want to calculate the distance error. The idea is to define an object-relevant area, where it may be predicted. Inside the area, the values of the pixel are treated as the probability distribution of the object. The object-relevant area will be calculated according to several masks. First is the mask $mask_{in}$ created by the egocentric map map_{in} generated from the depth image, which is also the input of the network. We dilate the map_{in} with a 5x5 kernel and the iterations of 3. The second mask is created by the prediction map_{pre}. We first apply a threshold of 0.4 on map_{pre}:

$$map_{pre\_th}[i,j] = \begin{cases} 1 & map_{pre}[i,j] > thres \\ 0 & else \end{cases} \tag{9}$$

The mask $mask_{pre}$ will be calculated by a dilation of map_{pre_th} with a 5x5 kernel and iteration of 3. The third mask $mask_{obj\_gt}$ is calculated according to the object ground truth map_{obj_gt}, where only the area of the specific object is set as 1 and all the other pixels have a value of 0. Different from the map_{gt} we mentioned earlier, map_{pre_th} has six layers for each prediction and each layer represents one object's ground truth position. After a dilation of map_{obj_gt} with a 7x7 kernel and iterations of 3, we get the third mask $mask_{obj\_gt}$. We take the other objects in the map as map_{others}. The mask $mask_{others}$ is calculated based on it with a 3x3 kernel and iteration of 1. The relevant area of the object is determined by the mask $mask_{obj}$:

$$mask_{obj} = mask_{obj\_gt} * mask_{pre} * mask_{in} * mask_{others} \tag{10}$$

With $mask_{obj}$, we can take the corresponding area of the predicted distribution map to calculate the location according to:

$$\hat{x}[obj\_ID] = \frac{\sum(X * mask_{obj}[obj\_ID] * map_{pre})}{\sum(mask_{obj}[obj\_ID] * map_{pre})} \tag{11}$$

$$\hat{z}[obj\_ID] = \frac{\sum(Z * mask_{obj}[obj\_ID] * map_{pre})}{\sum(mask_{obj}[obj\_ID] * map_{pre})} \tag{12}$$

where obj_ID is from 0 to 5, which indicates the possible existing objects in the map. The x-axis is from left to right, starting with 0 on the robot position. The z-axis is from front to back, which means an object in front of the robot will have a negative z-value. When the predicted location is calculated, the loss function will be:

$$loss_{loc} = MEAN((\frac{x - \hat{x}}{|x| + 1})^2 + (\frac{z - \hat{z}}{|z| + 1})^2) \tag{13}$$

Overall, the final loss function will be:

$$loss = loss_{map} + \tau loss_{loc} \tag{14}$$

where τ is a balance coefficient to scale the loss of localization. The coefficient we used here is 200, which enables a stable convergence.

C. Object Localization with Support Material

We follow the procedure of clustering and object localization as described in [23]. Once we have the center of the object based on the vanilla sensor information, we use the support material created by the spatial anticipation layer to recalculate the location and annotate more areas based on the support material. We dilate the objects' cluster in the map_{in} with a 5x5 kernel and iterations of 3, which is identical to the training procedure. We also calculate the $mask_{others}$ according to the training procedure. Then the $mask_{obj}$ will be calculated according to:

$$mask_{obj} = mask_{in} * mask_{others} \tag{15}$$

The trained spatial anticipation module will generate map_{pre} and the location of the object will be calculated according to Equation. 11 and Equation. 12 with the given $mask_{obj}$. For object annotation of the object, we use the union of the sensor observation of the object and the extended points from the $mask_{obj}$ area, where the pixel value is larger than 0.4. Since we perform the clustering and the spatial anticipation at the same time, the extra network computation will not introduce any additional overhead.

V. Training and Evaluation

We evaluated the three approaches under different sensor configurations, namely a depth camera positioned at a height of 1.2 meters, a depth camera positioned at a height of 0.2 meters, and a 2D laser scanner positioned at a height of 0.2 meters in the simulation. The RGB cameras are also positioned accordingly. For evaluating the localization error, we matched each cluster with the corresponding ground truth object. The matching algorithm calculates the intersection between predicted objects and ground truth objects, selecting the predicted objects with the largest intersections as pairs. Predicted objects with an intersection less than 10% of the cluster size (the number of points in the cluster) and no matching ground truth object were labeled as unexpected objects. These unexpected objects were attributed to label errors in the first-person view segmentation methods.

A. Evaluation Matrices

There are several evaluation matrices used for a detailed comparison of the detection and localization abilities. To evaluate the prediction quality of the egocentric map, we introduce the precision, recall, and intersection over union(IOU) for all the detected objects. To evaluate the distance estimation and location estimation quality, we use the mean absolute error(MAE) and the root mean squared error(RMSE).

B. Results from Trained Objects in Untrained Scenes

Fig. 6 shows the localization results of the four frames for four different objects. The blue areas, which are annotated by the algorithm as the occupied area of the objects, change the shape and orientation according to the objects because it is also considered the semantic information in the RGB images.

Table II shows the evaluation result of the three methods. The original method has the highest precision value in all three configurations because almost all the points from the sensor should be on the object. Although technically the precision should be 100%, the result is significantly below it because the precision will become lower when we treat two objects as one when they are close. An example of the special situation has been shown in Fig. 7. In example A, there are two dogs with very close positions and in the egocentric map, they are treated as a single cluster. Within this situation, the precision will be lower than 1. The error of the location estimation will also be wrongly calculated. To avoid the influence of these evaluation data, we only take the error within 0.25 meters into consideration.

Both methods with the spatial anticipation module have higher recall than the original method. The proposed method has the highest recall because we combine the sensor observation with the extended area given by the support material from the spatial anticipation module. While using the end-to-end methods, the algorithm will not consider the sensor observation, which loses a part of valuable information. The detection rate of the proposed methods will always be the same as the vanilla approach because both methods use sensor observation for clustering. The higher IOU of the object annotation of the proposed can be helpful for the robot's local path planning.

Comparing the evaluation data across the sensor setup, we noticed that the perspective of the camera will substantially influence the detection rate, which is attributed to the limitation of the first-person view segmentation. The precision of the vanilla approach in the last configuration is the highest since the laser data are more discrete and have less overlap phenomenon described in Fig. 7. Although the increase of the Recall and the IOU can be observed in all three cases, the increase in the last scenarios is most significant, since the original laser scanner data contribute poorly to object labeling on the egocentric map. The error of the vanilla approach is caused mostly by self-occlusion, so a higher camera positioning can mitigate the error. Our methods substantially increase the accuracy of the localization based on the support material from spatial anticipation.

C. Results from Untrained Objects in Untrained Scenes

Since we only trained and evaluated the spatial anticipation module on 6 different objects, the biggest challenge of the spatial anticipation module and the given approach is the ability to transfer knowledge, which means the network should also work on unseen objects. Therefore we evaluate the approach also with three unseen objects within the untrained scenes to give further validation of the performance.

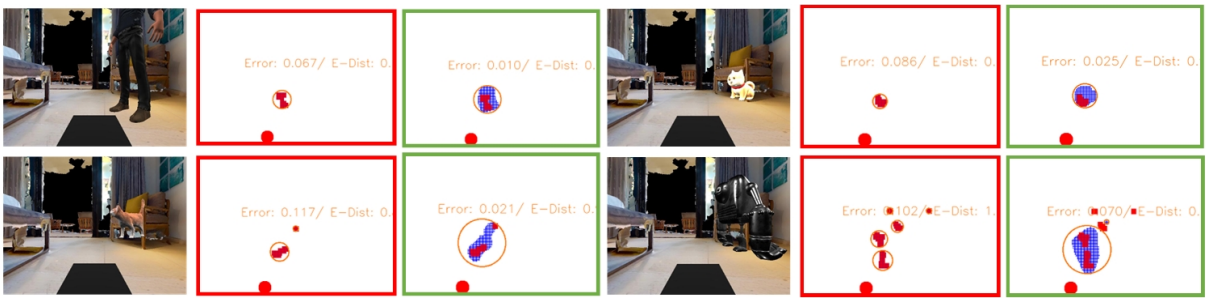

Fig. 6: Quality evaluation of the proposed method. The image contains a set of frames in the dataset for four different objects. The red dot on the bottom is the robot itself. the red areas in the map are the observations from the laser scanner. The blue areas indicate the object annotation from the proposed algorithm. We can notice that the blue area changes shape according to the object type.

TABLE II: Evaluation from Trained Objects in Untrained Scenes and Datasets For Egocentric Map

Metric	**1.2m Depth Camera**			**0.2m Depth Camera**			**0.2m 2D Laser Scanner**		
	Method A	**Method B**	**Vanilla**	**Method A**	**Method B**	**Vanilla**	**Method A**	**Method B**	**Vanilla**
Detection Rate	**0.7407**	0.7220	0.7407	**0.5483**	0.5402	0.5483	**0.4580**	0.4472	0.4580
No. of Objects	**29163**	28425	29163	**11679**	11507	11679	**8827**	8642	8827
Precision	0.7403	0.8259	**0.9167**	0.7883	0.7863	**0.9292**	0.7648	0.7952	**0.9797**
Recall	**0.7595**	0.5504	0.4253	**0.7255**	0.6590	0.5214	**0.5783**	0.4473	0.2077
IOU	**0.5997**	0.4932	0.4095	**0.6072**	0.5589	0.5015	**0.4910**	0.4011	0.2068
RMSE	**0.1262**	0.1266	0.1341	**0.1232**	0.1243	0.2010	**0.1190**	0.1205	0.1897
MAE	**0.0704**	0.0752	0.0892	**0.0622**	0.0630	0.0936	**0.0595**	0.0602	0.0981

TABLE III: Evaluation from untrained Object with 0.2m 2D Laser Scanner

Metric	**Big Robot**			**Woman A**			**Woman B**		
	Method A	**Method B**	**Vanilla**	**Method A**	**Method B**	**Vanilla**	**Method A**	**Method B**	**Vanilla**
Detection Rate	**0.4059**	0.3984	0.4059	**0.5622**	0.5607	0.5622	**0.3750**	0.3709	0.3750
No. of Objects	**3288**	3228	3288	**3432**	3423	3432	**1614**	1598	1614
Precision	0.8901	0.8677	**0.9857**	0.8656	0.8880	**0.9908**	0.8139	0.8457	**0.9879**
Recall	**0.1522**	0.1208	0.0292	**0.4600**	0.3358	0.1672	**0.4827**	0.3570	0.1828
IOU	**0.1494**	0.1187	0.0292	**0.4294**	0.3222	0.1669	**0.4347**	0.3351	0.1824
RMSE	0.2474	**0.2411**	0.3075	**0.0921**	0.0937	0.1407	**0.0902**	0.0932	0.1153
MAE	0.1198	**0.1031**	0.1214	0.0522	**0.0507**	0.1064	0.0519	**0.0511**	0.0937

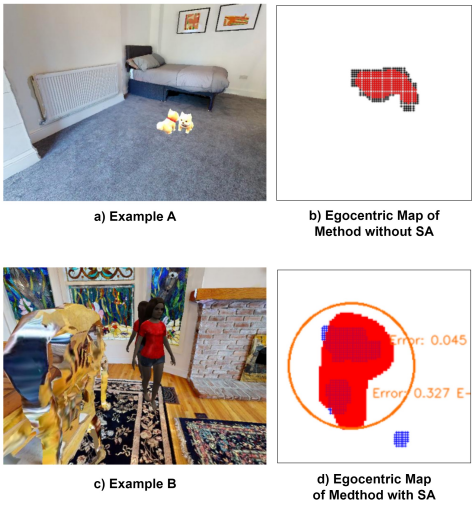

Fig. 7: The images show two special examples in the evaluation. Example A shows a scenario with two dogs, which are close to each other. They form a single cluster with the provided sensor data. Example B shows that the spatial anticipation module will also make two objects become one cluster.

The evaluation results for the three untrained objects are presented in Tab. III. With the equal detection rate with the vanilla approach, our approach improves the Recall, the IOU, and localization accuracy significantly. We believe one of the most important reasons for the successful transfer is to use the pre-trained ResNet as the visual encoder, which has been trained on a larger dataset and is sensitive to common features. With the observation, we are confident that the spatial anticipation module can be generalized for all the common objects if we train it in a larger and more diverse dataset, but it is not been proven in the current work.

VI. Conclusion

The paper introduces an indoor dataset HabitatDyn 2.0 with the top-view label of dynamic objects. The various dynamic patterns, environment settings, and kinetic information enable new algorithm developing like spatial anticipation. The spatial anticipation leverages the semantic information for object localization. With the generated support material from the spatial anticipation module, the algorithm can improve the localization accuracy and extend the annotated occupied area of the objects. The method is evaluated against two baselines within 3 different sensor configurations for both trained objects and untrained objects on HabitatDyn 2.0 and it was shown that the localization improvement of the trained objects is significant across all the scenarios even for untrained objects. For future works, we aspire to extend and optimize the dataset to support the training of a zero-shot spatial anticipation model.

REFERENCES

[1] L. Liu, D. Dugas, G. Cesari, R. Siegwart, and R. Dubé, "Robot navigation in crowded environments using deep reinforcement learning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5671–5677.

[2] H. Andreasson, G. Grisetti, T. Stoyanov, and A. Pretto, "Sensors for mobile robots," 2022.

[3] P. Wenzel, T. Schön, L. Leal-Taixé, and D. Cremers, "Vision-based mobile robotics obstacle avoidance with deep reinforcement learning," in *IEEE International Conference on Robotics and Automation, ICRA 2021, Xi'an, China, May 30 - June 5, 2021*. IEEE, 2021, pp. 14 360–14 366. [Online]. Available: https://doi.org/10.1109/ICRA48506.2021.9560787

[4] K. D. Katyal, Y. Gao, J. Markowitz, S. Pohland, C. G. Rivera, I. Wang, and C. Huang, "Learning a group-aware policy for robot navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2022, Kyoto, Japan, October 23-27, 2022*. IEEE, 2022, pp. 11 328–11 335. [Online]. Available: https://doi.org/10.1109/IROS47612.2022.9981183

[5] M. Everett, Y. F. Chen, and J. P. How, "Collision avoidance in pedestrian-rich environments with deep reinforcement learning," *IEEE Access*, vol. 9, pp. 10 357–10 377, 2021.

[6] Y. Gao and C.-M. Huang, "Evaluation of socially-aware robot navigation," *Frontiers in Robotics and AI*, vol. 8, 2022. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frobt.2021.721317

[7] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *Robotics & Automation Magazine, IEEE*, vol. 4, pp. 23 – 33, 04 1997.

[8] T. Raj, F. Hashim, A. Huddin, M. F. Ibrahim, and A. Hussain, "A survey on lidar scanning mechanisms," *Electronics*, vol. 9, p. 741, 04 2020.

[9] S. K. Ramakrishnan, Z. Al-Halah, and K. Grauman, "Occupancy anticipation for efficient exploration and navigation," 2020.

[10] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[11] N. Xu, L. Yang, Y. Fan, J. Yang, D. Yue, Y. Liang, B. L. Price, S. Cohen, and T. S. Huang, "Youtube-vos: Sequence-to-sequence video object segmentation," in *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part V*, ser. Lecture Notes in Computer Science, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., vol. 11209. Springer, 2018, pp. 603–619. [Online]. Available: https://doi.org/10.1007/978-3-030-01228-1_36

[12] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, arXiv:1512.02134. [Online]. Available: http://lmb.informatik.uni-freiburg.de/Publications/2016/MIFDB16

[13] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 724–732.

[14] M. Vajgl, P. Hurtik, and T. Nejezchleba, "Dist-yolo: Fast object detection with distance estimation," *Applied Sciences*, vol. 12, no. 3, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/3/1354

[15] P. Agand, M. Chang, and M. Chen, "DMODE: differential monocular object distance estimation module without class specific information," *CoRR*, vol. abs/2210.12596, 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2210.12596

[16] M. Ahishali, M. yamaç, S. Kiranyaz, and M. Gabbouj, "Representation based regression for object distance estimation," *Neural Networks*, vol. 158, 11 2022.

[17] H. Yang, C. Shi, Y. Chen, and L. Wang, "Boosting 3d object detection via object-focused image fusion," 2022.

[18] J. Lahoud and B. Ghanem, "2d-driven 3d object detection in rgb-d images," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 4632–4640.

[19] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from RGB-D data," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. Computer Vision Foundation / IEEE Computer Society, 2018, pp. 918–927.

[20] T. Eppenberger, G. Cesari, M. Dymczyk, R. Siegwart, and R. Dubé, "Leveraging stereo-camera data for real-time dynamic obstacle detection and tracking," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2020, Las Vegas, NV, USA, October 24, 2020 - January 24, 2021*. IEEE, 2020, pp. 10 528–10 535. [Online]. Available: https://doi.org/10.1109/IROS45743.2020.9340699

[21] Z. Xu, X. Zhan, Y. Xiu, C. Suzuki, and K. Shimada, "Onboard dynamic-object detection and tracking for autonomous robot navigation with rgb-d camera," 2023.

[22] I. Ballester, A. Fontán, J. Civera, K. H. Strobl, and R. Triebel, "DOT: dynamic object tracking for visual SLAM," in *IEEE International Conference on Robotics and Automation, ICRA 2021, Xi'an, China, May 30 - June 5, 2021*. IEEE, 2021, pp. 11 705–11 711. [Online]. Available: https://doi.org/10.1109/ICRA48506.2021.9561452

[23] Z. Shen, Y. Gao, L. Kästner, and J. Lambrecht, "Habitatdyn dataset: Dynamic object detection to kinematics estimation," 2023.

[24] Z. Shen, L. Kästner, and J. Lambrecht, "Spatial imagination with semantic cognition for mobile robots," *CoRR*, vol. abs/2104.03638, 2021. [Online]. Available: https://arxiv.org/abs/2104.03638

[25] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3061–3070.

[26] X. Liu, C. R. Qi, and L. J. Guibas, "Flownet3d: Learning scene flow in 3d point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 529–537.

[27] M. Zhai, X. Xiang, N. Lv, and X. Kong, "Optical flow and scene flow estimation: A survey," *Pattern Recognition*, vol. 114, p. 107861, 2021.

[28] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017. [Online]. Available: http://arxiv.org/abs/1703.06870

[29] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," 2023.

[30] S. Mahadevan, A. Athar, A. Ošep, S. Hennen, L. Leal-Taixé, and B. Leibe, "Making a case for 3d convolutions for object segmentation in videos," in *BMVC*, 2020.

[31] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy