2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Anomaly Detection For Robust Autonomous Navigation

Kefan Jin, Fan Mu, Xingyao Han, Guangming Wang and Zhe Liu

***Abstract*— Human drivers are remarkably robust against various unexpected occurring variations and corruptions by understanding temporal changes and traffic scenes. In contrast, the neural network based autonomous navigation system can be easily affected by sensor data anomaly, like occlusion, sensor noise, challenging weather and illumination conditions. Such external disturbances are inevitable in practical driving applications. In this paper, we develop a semi-supervised anomaly detection module to detect the corrupted data while extracting the traffic scenario features. We further introduce an end-to-end robust autonomous navigation framework based on the idea that the consecutive frames of clean data depict a similar traffic scenario and the differences among the sequential data imply the dynamic state changes. By taking into consideration both spatial traffic scenario and temporal environmental variation, the model is able to achieve robust navigation against sensor data corruptions. We conduct experiments in CARLA platform and the evaluation results show the effectiveness of the proposed method.**

I. INTRODUCTION

Autonomous driving, which attracts a lot of works in both academia and industry, is still an under-explored task. Traditional methods decomposes the navigation task into multiple subsystems, for example, perception [1], lane following [2], path planning [5] and action prediction [3]. These sub-systems will then be integrated with some fine tuning to realize autonomous driving. However, the modular fashion is liable to cause accumulating errors and also requires a lot of human annotation. Recently, end-to-end autonomous driving [5], [11] arises with the development of deep learning techniques to solve these problems. By learning a policy agent, the end-to-end approaches are able to map the raw sensor data to control signals directly, which greatly reduces modules' internal dependencies and improves model scalability.

A lot of existing end-to-end navigation approaches take multi-modal sensor data as model input, including camera data and LiDAR point cloud data [12], [13], which have shown satisfactory results in the navigation task. These methods assume that the sensor data is always valid and clean, and do not consider the potential data interference. However, the practical driving scenario is complex, containing various disturbances. For example, during the real driving process, vehicles are likely to experience perception restricted situations, like dark tunnels and adverse weather conditions, as well as sensor disturbances, which may lead to partial sensor data failure.

The authors are with the MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, China. Corresponding author: Z. Liu (liuzhesjtu@sjtu.edu.cn).

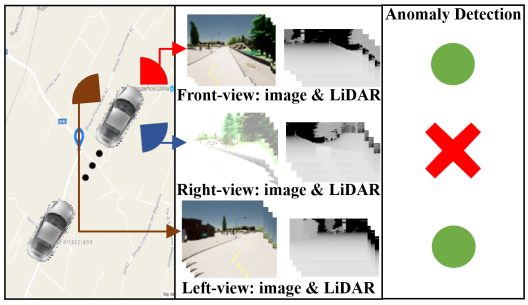


Fig. 1. The anomaly detection task considered in this paper for autonomous vehicle navigation.

To solve this problem, a lot of approaches are proposed to detect the corrupted data. The authors in [4] introduce an information-theoretic framework, Deep-SVDD, for deep anomaly detection. It assumes that the entropy of the latent distribution for normal data should be lower than the entropy of the anomalous distribution. They train a neural network to learn a transformation that minimizes the volume of a data-enclosing hypersphere in output space centered on a predetermined point. Inspired by Deep-SVDD, we propose an anomaly detection module based on the idea that the sequential sensor data frames depict similar navigation conditions and the corresponding features should be close in output feature space. However, a single-center feature distribution cannot depict various navigation scenes. To extract the scenario feature, in this work, we do not pre-determine a single-center point like [4], instead, we train the features of the clean data samples under the same condition to distribute as close as possible, thus naturally generating a center point for each navigation scenario, which represents the traffic scenario feature.

Many approaches are proposed to achieve the robust control [6][7][8]. And for robust navigation, it has been observed in [9] that adversarial training can greatly improve the navigation performance under sensor data degradation. While adversarial training may be a promising strategy to improve the robustness to sensor failure, however, human driver can easily handle external disturbances without adversarial training. This is due to our understanding of the whole traffic scenario and temporal environmental changes. For example, human drivers can understanding the general driving condition, which will not change much in a short time, and make control actions according to any slight changes in the environment. In this way, when the driver encounters unseen disturbances, they can still do the appropriate action. What's more, even when drivers' vision are disrupted, they remember the driving situation prior to the disruption, and therefore are still able to make reasonable

driving decisions. In this paper, as show in Fig. 1, our method is capable of utilizing historical sensor data for anomaly detection through temporal-consistency learning. In addition, this approach enables the retention of information necessary for navigation in the representation of abnormal detection, thus achieving the restoration of abnormal information and robust navigation. The main contributions of this paper are listed as follows.

- We learn a joint feature representation, which can distinguish the abnormal data in a self-supervised manner that implicitly indicates the reliability of the given data sample. The learned feature space contains all the necessary information needed for autonomous driving thus can be used for navigation tasks.
- We further introduce a robust autonomous navigation framework, which reconstructs the disturbed data and makes control signal according to the effective spatial-temporal information. The proposed method is able to achieve robust driving performance against data disturbances and partial sensor failure.
- Experiments and evaluations are conducted on the CARLA platform. Simulation results demonstrate the effectiveness and robustness of the presented approach under various conditions.

II. RELATED WORK

A. End-to-End Navigation

The end-to-end navigation methods [5], [11] train the model to produce the control outputs from raw sensor data inputs. The imitation learning [10], [11], which is a supervised learning method from expert demonstrations, is the most popular strategy. [10] first proposes the conditional imitation learning (CIL) method on high-level command input, which constructs multiple networks for each command, greatly improving the navigation performance. Inspired by CIL, [29] further feeds the semantic segmentation information instead of the raw RGB frames in order to improve the network generalization in different environments. Although the camera image data has been widely applied in self-driving system, it is difficult to capture the 3-D information of driving scenario. To solve this, [12] and [13] utilize both the camera image data and LiDAR point cloud data. Furthermore, instead of producing control commands directly, [16] and [15] predict the future trajectory of the ego-vehicle in the bird-eye-view space, centered at the current coordinate frame of the ego-vehicle to guarantee the driving safety and increase the interpretability. However, these methods only consider the navigation task under valid and clean sensor data.

B. Data Corruption in Self-Driving

In real complex navigation condition, the external disturbances, including natural weather interference and partial sensor failure, are inevitable, which will lead to sensor data corruption. To realize anomaly detection, [18] proposes a hybrid anomaly detection approach, which applies distance-based algorithms on top of deep neural feature maps from pre-trained networks. [20] and [21] realize online anomalous trajectory detection by training an auto-encoder module to learn the trajectory feature, which is then used to identify the abnormal trajectory. Furthermore, [19] develops a graph neural network based relation learning network to detect the abnormal information in vehicle perception results, by learning the spatial-temporal relations among vehicles and the scenario. In [4], an information-theoretic framework, Deep-SVDD, is proposed to detect the data anomaly, based on the idea that the entropy of the latent distribution for normal data should be lower than the entropy of the anomalous distribution. Given a pre-determined central point in the output space, the neural network is trained to minimize the distances between clean data features and the central point, while increasing the distance between abnormal data and the center point. Inspired by Deep-SVDD, [17] proposes an anomaly detection model to quantify LiDAR degradation in dynamic urban environments. However, because there's only one central point in [4], [17], the output features have lost the information needed for navigation. In this paper, we borrow the idea from Deep-SVDD, but we do not predefine a specific central point. We only train the network to minimize the distances among sequential data samples. In this way, the data samples under different driving conditions will have different central points, which is able to contain the scenario information required by navigation task.

C. Robust Navigation

A desired autonomous navigation system is expected to realize robust navigation under data interference. To this end, several driving datasets under various challenging conditions are proposed [22], [23]. In [24], a general notion of adversarial perturbations for image sensor data, which can be created using generative models, is proposed to 'fool' models for steering angle prediction. [25] proposes a behavioral cloning approach to learn a novel representation that combines predicted visual abstractions and scalar confidence value by convolving them in a discrete bird-eye-view grid, which captures perceptual uncertainty across the full scene. In addition, by using adversarial training and data augmentation, [26] enhances the navigation robustness to image corruptions and domain shifts. However, this work only considers the steering control. By learning the features of the driving scenario and dynamic environmental changes, our method not only detects the corrupted data, but also realizes the robust navigation.

III. MAIN APPROACH

A. System Framework

In this paper, we assume that the autonomous vehicle drives in a complex outdoor scenario with lots of surrounding vehicles under external disturbances. The ego-vehicle is equipped with multi-view camera sensors and a LiDAR sensor. The overall framework is shown in Fig. 2, which consists of three main components:

- *Joint Representation Learning:* Based on Variational Auto-Encoder (VAE), we build a multi-modal data fusion module to learn joint representation from camera

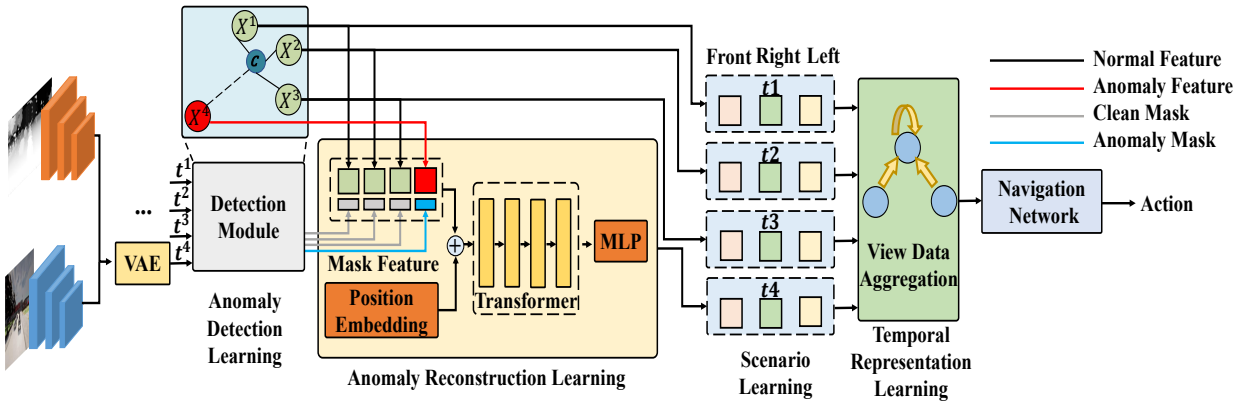


Fig. 2. The overall system framework.

image data and LiDAR point cloud data. The input data of each sensor can be re-constructed from the learned joint representation, which means it contains all the important information in the original data.

- *Anomaly Detection Learning:* By means of self-supervised learning, the network aggregates clean joint representations of the same perspective from various moments, while distancing itself as much as possible from abnormal information to achieve anomaly detection. In addition, the module is also trained to reserve the important information, which is required for navigation task. In this way, the center of the learned features contains the overall scenario information.
- *Anomaly Reconstruction and Navigation Policy Learning:* Given the anomaly detection results, the detected disturbed node will be excluded to reduce the influence of abnormal information and increase the navigation robustness. The recovery module then utilizes the learned clean confidence features to reconstruct the anomalous information by learning the temporal consistency. After that, time-series multi-view features are aggregated by graph attention (GAT) network to generate control commands, achieving robust navigation.

B. Joint Representation Learning

We propose the joint representation learning module to learn a joint feature from camera image data and LiDAR point cloud data. Based on the idea of [27], we elaborate a multi-input-multi-output learning module. First, the LiDAR point cloud data is pre-processed into a virtual 2D image, with the image length corresponding to the horizontal LiDAR angle range and the image height corresponding to the longitudinal LiDAR angle range. After that, both images of LiDAR and camera are input into their corresponding networks respectively to generate the LiDAR feature f_I and image feature f_L. The encoder then takes as input the concatenation of LiDAR feature and image feature $[f_I; f_L]$ and outputs the joint feature f. Given the joint feature f, the MLP-based decoder is trained to recover the image data of camera and LiDAR. In this way, the original VAE loss is extended into both modal data. In addition, we further use MSE losses to train data reconstruction ability.

C. Anomaly Detection Learning

In this section, we introduce the anomaly detection module and our scenario learning method, which is extended from [4]. We assume that the labeled anomaly samples $(\bar{f}^i, \bar{y}^i)$ of joint feature is accessible, where $(\bar{y}^i = +1)$ denotes clean feature and $(\bar{y}^i = -1)$ denotes disturbed feature. The neural network of the detection module is denoted by ϕ with weights W. The network is trained to cluster together the normal features, while the disturbed features will be far apart as shown in Fig. 3. Then, the anomaly detecting objective can be defined as follows:

$$\begin{aligned}\min_W \frac{1}{u}\sum_{j=1}^{u}\Big(\frac{1}{n+m}\sum_{i=1}^{n}(||\phi(f_j^i;W)-c_j||^2)\\ +\frac{\eta}{m+n}\sum_{i=1}^{m}((||\phi(\bar{f}_j^i;W)-c_j||^2)^{\bar{y}_j})\Big)\\ +\frac{\lambda}{2}\sum_{l=1}^{L}||W^l||_F^2, \lambda>0\end{aligned} \tag{1}$$

It should be noted that each data sample consists of 4 sequential features under the similar traffic scenario. $x^i = \phi(f^i)$ denotes the output of detection network, W^l denotes the parameters of the lth layer of W, j denotes the number of training samples, $i = 1, 2, 3, 4$ denotes the data time steps and c_j denotes the center point of clean detection features of the jth view, which is defined as follows:

$$c_j = \frac{1}{n}\sum_{i=1}^{n}\phi(f_j^i) \tag{2}$$

In this way, the framework penalizes the mean squared distance of the mapped clean samples to the hypershpere center c, thus extracting the common factors of the variation in this data sample, which is the driving scenario.

After that, we calculate the distance between the ith feature of the jth view and the center feature:

$$d_j^i = ||x_j^i - c_j||^2 \tag{3}$$

In each sample, the feature with the largest distance is considered to have been disturbed. In this way, the abnormal data is detected effectively. However, the simple anomaly

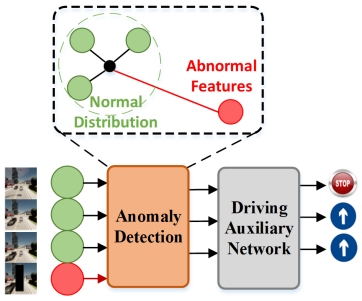


Fig. 3. Anomaly detection and driving auxiliary network.

detection task cannot retain the essential information required to complete the navigation task in the input data. In order to better perform the detection task, only the information related to the disturbance will be extracted. To avoid this problem, we introduce an auxiliary task to guarantee that the learned representation is effective for the navigation task as shown in Fig. 3.

More specifically, for the feature x^i at each view, a driving task auxiliary network is constructed. This network takes the information of corresponding viewpoint at a single moment as input, and then predicts the control command. It should be noted that the we do not use the predicting results from this module directly. Actually, the single moment and single perspective information are insufficient to realize navigation task effectively, especially for the right-view and left-view. However, even though the information of each view is incomplete, it can still provide certain support for the navigation task. For example, when there is a vehicle in the left perspective data, the network can learn that the vehicle should not turn left at this time step. In this way, we ensure that the learned confidence representation still contains the information required to complete navigation tasks.

D. Anomaly Reconstruction and Robust Navigation

In this section, we describe the end-to-end robust navigation policy module. We borrow the idea from human driving behavior: make rough driving actions according to the temporal data and image the current condition with historical memory when vision is blocked. As analysed above, the learned confidence feature contains the information required for navigation task, which has the characteristics of temporal consistency. Therefore, we introduce the transformer [32] approach to make use of the clean feature to re-construct the abnormal feature, as shown in Fig. 2. On the one hand, we take the confidence features of all time steps at each view as inputs of the reconstruction module. On the other hand, the detection module generates the mask features for each confidence feature, which represent the anomaly degree. The confidence feature is then combined with mask features. To be more specifically, if the mask feature shows that the data is clean, the input confidence feature still remains the same. Otherwise, the abnormal input confidence feature will be converted to a fixed vector to eliminate the impact of abnormal information. After that, the position embedding is

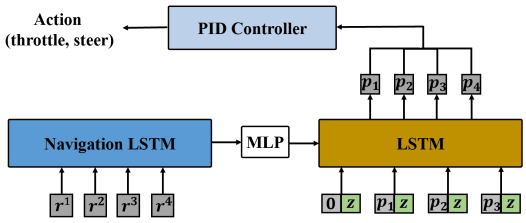


Fig. 4. Navigation module.

introduced to indicate the timestamp information of the representation. Intuitively, the timestamp information implicitly indicates the relevance of the representation to other time information. After that, features from all the time steps are input into the transformer module. The network learns to recover the fixed vector with other temporal information. The network is trained in supervised learning. The reconstructed feature will replace the origin corresponding confidence feature and be noted as new x_j^i.

As analysed above, all detection feature x_j^i contains effective information for view j at the time step t. After that, we construct a scenario learning modules, which consists of an one-layer GAT network, to aggregate the confidence features at 4 time steps respectively. A key function, a query function and a value function are constructed to produce a basic value representation v_i, a key value k_i, and a query value q_i of input x_j^i respectively. For each view j, its attention to other view k is formulated as:

$$att_{j,k} = softmax(q_j * k_k) \tag{4}$$

Nodes of all views are connected to the node of front-view, and the output of the GAT is formulated as following:

$$y^i = \sum_j v_i * att_{i,j} \tag{5}$$

In addition, we construct an LSTM-based navigation policy module, which takes the scenario feature c as initial hidden state and takes y^i as sequential input, as shown in Fig. 4. The output of navigation LSTM module is then input into MLP. Finally, the LSTM-based position prediction module generates the predicted position with the concatenation of target position z and current predicted position $p_i, i = 0, 1, 2, 3$ and $p_0 = 0$ being sequential inputs. The navigation policy is trained with imitation learning. The VAE module and anomaly detection module are pre-trained firstly. Before training the navigation module, the pre-trained VAE module and anomaly detection module are pre-loaded. During the navigation training process, all the parameters are updated.

IV. IMPLEMENTATION

In this work, we use Pytorch framework to construct the neural networks. The model is trained on a computer with GTX1070ti GPU, i7-10400 CPU and 32G RAM. We use the Adam optimization optimizer to train the neural networks. The learning rate is set to be 0.0001 and the batch size is 32. We set the maximum training epoch number as 200, the fix vector is zero vector. During the training process, one of

the 4 sequential data sample of each view will be disturbed randomly.

A. Data Collection

The data is collected on the CARLA platform. We collect 37000 data samples in town02 and town03 scenario. We take 32000 data samples as training dataset and 5000 data samples as test dataset. Each data sample contains camera image data of 3 views, LiDAR point cloud data, the target destination position and the 4 future positions. During the collection process, the CARLA autopilot agent is used to control the surrounding vehicles and the privileged agent in [28] is utilized as the ego vehicle. The frequency is 10 frames per second.

B. Data Disturbances

In order to make the simulation closer to the real scene, we mainly consider 3 types of disturbances:

- *Semantic Disturbance*: It is known to all that perceptual neural networks sometimes misdetect the presence of vehicles around. In this work, we design a kind of disturbance that will generate a "ghost" vehicle, which actually does not exist, in the camera image data and LiDAR point cloud data.
- *Occlusion*: Data occlusion is very common in real scenarios. To simulate this situation, we consider the occlusion, which can cause a part area of the data (camera image and LiDAR point cloud) to be blocked.
- *Luminance Contrast Interference*: A lot of practical factors can lead to the luminance contrast disturbances for camera data. In this work, the luminance contrast disturbance will affect the brightness and contrast of the image data randomly.

C. Metrics

In this work, we consider 3 metrics for navigation:

- *Route Completion (RC)*: The percentage of completed route distance:

$$RC = \frac{1}{N}\sum_{i=1}^{N} R_i \tag{6}$$

- *Driving Score (DS)*: The weighted average of the route completion with infraction multiplier P_i as described in [16]:

$$DS = \frac{1}{N}\sum_{i=1}^{N} R_i P_i \tag{7}$$

D. Comparison Models

In this work, we compare the following models:

- *LateFusion*: The model proposed in [29], which does not consider the robustness issue explicitly.
- *CILRS*: The model proposed in [33], which does not consider the robustness issue explicitly.
- *Our_Previous_Model*: Our previous work proposed in [34], which considers the robustness issue explicitly.
- *Transfuser*: The model proposed in [31], which does not consider the robustness issue explicitly.

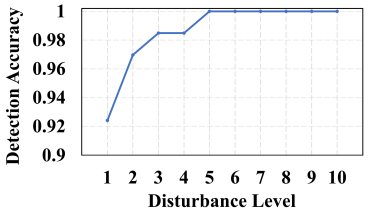
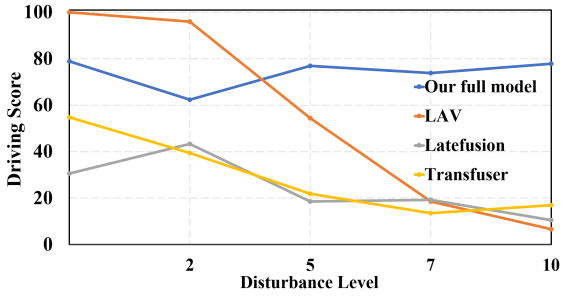
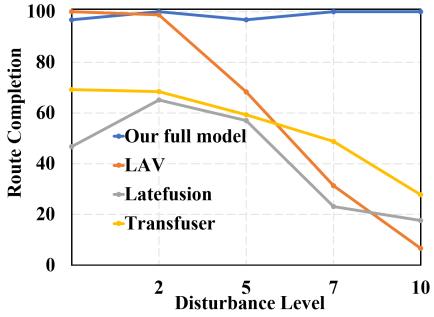


Fig. 5. Anomaly detection evaluations



(a) DS Scores



(b) RC Scores

Fig. 6. Navigation performance under different disturbance levels

- *LAV*: The model proposed in [30], which does not consider the robustness issue explicitly.
- *Our_Model*: Our full model described in this paper.
- *Our_Privileged_Model*: The proposed model described in Section III without the anomaly detection learning and anomaly reconstruction learning. However, this model is able to eliminate the abnormal data directly according to the privileged information (it knows the ground-truth information of the anomaly detection).

V. EXPERIMENTS

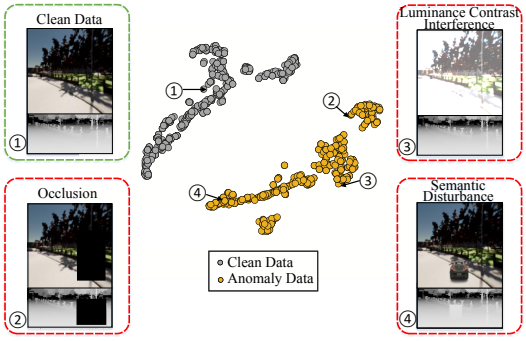
A. Anomaly Detection Evaluation

We first demonstrate the anomaly detection performance. We divide the disturbance strength into 10 levels. The disturbance strength 1 indicates the least data disturbance. As the strength increases from 1 to 10, for semantic disturbance, the occlusion area gradually increases to 50 percents, for luminance contrast disturbance, the brightness increases to 50 percents and for semantic disturbance, the size of "ghost" vehicle gradually increases to 50 percents of the width of the data. It should be noted that during the training process, the number of disturbed view is always set to 2 and the disturbance strength is 7. The disturbances on the camera image data and the LiDAR point cloud data are synchronized.

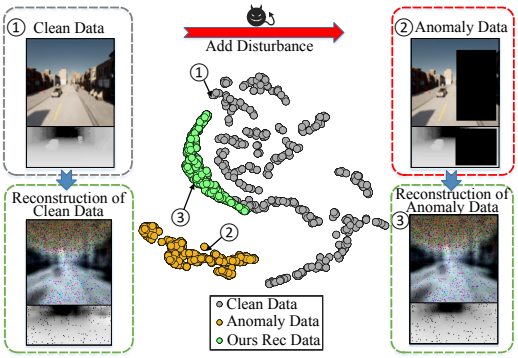
TABLE I
ROBUSTNESS SCALABILITY EVALUATION

	Our_Model		[30]		[31]		[33]		[34]		[29]		Our_Privileged_Model	
	RC	DS	RC	DS	RC	DS	RC	DS	RC	DS	RC	DS	RC	DS
clean	96.7	78.8	**100**	**100**	69.2	54.8	85.7	50.0	93.9	78.2	46.7	30.5	100	77.4
disturb 1 view	**100**	**76.0**	57.6	52.9	45.9	35.6	59.2	40.1	89.2	50.8	45.4	23.6	100	78.3
disturb 2 views	**96.7**	**72.1**	42.9	36.8	49.6	32.4	40.2	23.5	100	51.7	39.9	25.0	100	76.3
disturb 3 views	**100**	**73.8**	33.5	20.8	35.7	17.0	31.8	14.1	100	51.22	24.5	16.0	100	67.3

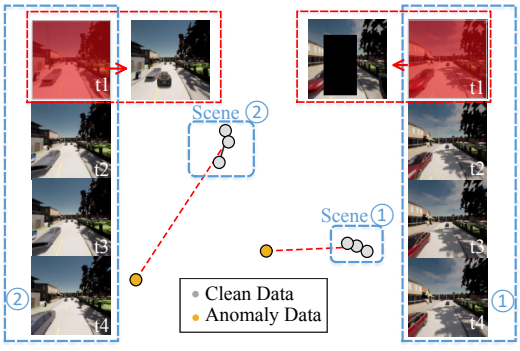
As shown in Fig. 5, when the disturbance intensity is 1, the accuracy rate is 92.4 percents. And the accuracy reaches 100 percents at level 5 and remained constant as the interference grew stronger.



(a) Feature space visualization of clean data and disturbed data



(b) Visualization of reconstructed data



(c) Feature space visualization of different scenes

Fig. 7. Visualization results

B. Robustness and Scalability Evaluation

We test the driving performance of each model under the disturbance level 0, 2, 5, 7 and 10. In this test, we disturb the front-view data and another randomly selected left and right view data. For `latefusion` and `cilrs`, they only uses one view information, so we also only disturb one view. As shown in Fig. 6, when the interference intensity is 0, which means that the information is completely clean, the driving score (DS) and route completion (RC) of `Our_Model` is the second best. This is reasonable because these methods are designed to improve navigation performance under clean data. As the disturbance level increases, the DS and RC of the other three models rapidly decreases, while the navigation performance of our model remains relatively stable. When the interference intensity reaches 10, the other three models almost completely fail. `Our_Model` outperforms other models by a large margin. Furthermore, we test the navigation performance of each model as the number of perturbed viewpoints increases from 0 to 3. For `latefusion` and `cilrs`, they only uses one view information, so we just increase the disturbing frequency. The results in the Tab. I demonstrate the scalability of our robustness.

C. Visualization

We visualize our learned feature space distribution and the corresponding data in Fig. 7. Fig. 7 (a) shows that our model has learned the difference between clean features and disturbance features. In Fig. 7 (b) we can see that, although the features of the perturbed data are far away from the features of the clean data, however, the reconstructed features are much close to the clean features. In addition, Fig. 7 (b) also shows that our reconstructed data from anomaly input is very similar to that from the corresponding clean data. Fig. 7 (c) depicts the figure distribution of the front-view data at four different time steps in different routes. In each route, the data of one random time step is disturbed. It can be observed that under the same scenario, 3 clean features cluster together while the disturbed feature is far away, indicating that the network has learned the ability of anomaly detection. Besides, the clean features of different scenarios are aggregated in different areas, indicating that the network has learned scene-related information.

VI. CONCLUSION

In this work, we present an end-to-end navigation framework which learns the anomaly detection ability to achieve robust navigation against disturbances. Experimental results on CARLA simulator demonstrate the our robustness and scalability under various disturbance types and strength. In future work, we will further investigate the robustness to unseen disturbances and evaluate our model in real scenarios.

REFERENCES

[1] Z. Liu, C. Suo, Y. Liu, Y. Shen, Z. Qiao, H. Wei, S. Zhou, H. Li, X. Liang, H. Wang, Y.-H. Liu, "Deep Learning-Based Localization and Perception Systems: Approaches for Autonomous Cargo Transportation Vehicles in Large-Scale, Semiclosed Environments", *IEEE Robotics and Automation Magazine*, vol. 27, no. 2, pp. 139–150, 2020.

[2] C. Chen, A. Seff, A. Kornhauser, J. Xiao, "DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving", *IEEE International Conference on Computer Vision*, pp, 2722–2730, 2015.

[3] Y. Hou, S. Hornauer, K. Zipser, "Fast recurrent fully convolutional networks for direct perception in autonomous driving", *arXiv:1711.06459*, 2017.

[4] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K. R. Müller, M. Kloft, "Deep semi-supervised anomaly detection", *arXiv:1906.02694*, 2019.

[5] J. Leonard, et al, "A perception-driven autonomous urban vehicle", *Journal of Field Robotics* 25.10, pp:727-774, 2008.

[6] F. Zhong, P. Sun, W. Luo, T. Yan, Y. Wang, "Towards distraction-robust active visual tracking", *International Conference on Machine Learning. PMLR*, pp:12782-12792, 2021.

[7] F. Zhong, P. Sun, W. Luo, T. Yan, Y. Wang, "AD-VAT: An asymmetric dueling mechanism for learning visual active tracking", *International Conference on Learning Representations*, 2019.

[8] F. Zhong, P. Sun, W. Luo, T. Yan, Y. Wang, "Ad-vat+: An asymmetric dueling mechanism for learning and understanding visual active tracking", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp:1467–1482, 2019.

[9] M. Shu, Y. Shen, M. C. Lin, T. Goldstein, "Adversarial differentiable data augmentation for autonomous systems", *JIEEE International Conference on Robotics and Automation*. 2021: 14069-14075.

[10] F. Codevilla, M. Müller, A. López, V. Koltun, A. Dosovitskiy, "End-to-end driving via conditional imitation learning", *IEEE international conference on robotics and automation*. IEEE, 2018: 4693-4700.

[11] Q. Wang, L. Chen, B. Tian, W. Tian, L. Li, D. Cao, "End-to-end autonomous driving: An angle branched network approach." *IEEE Transactions on Vehicular Technology* 68.12 (2019): 11599-11610.

[12] Y. Xiao; F. Codevilla; A. Gurram; O. Urfalioglu; A. M. López, "Multimodal end-to-end autonomous driving." *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[13] P. Cai, S. Wang, Y. Sun, M. Liu, "Probabilistic end-to-end vehicle navigation in complex dynamic environments with multimodal sensor fusion." *IEEE Robotics and Automation Letters* 5.3 pp: 4218-4224, 2020.

[14] P. Cai, H. Wang, Y. Sun, M. Liu, "Learning scalable self-driving policies for generic traffic scenarios", arXiv preprint arXiv:2011.06775, 2020.

[15] W. Zeng, W. Luo, S. Suo, A. Sadat, B. Yang, S. Casas, R. Urtasun, "End-to-end interpretable neural motion planner", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp: 8660-8669, 2019.

[16] A. Prakash, K. Chitta, A. Geiger, "Multi-modal fusion transformer for end-to-end autonomous driving", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp: 7077-7087, 2021.

[17] C. Zhang, Z. Huang, M. H. Ang, Da. Rus, "LiDAR Degradation Quantification for Autonomous Driving in Rain", *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp: 3458-3464, 2021.

[18] L. Bergman, N. Cohen, Y. Hoshen, "Deep nearest neighbor anomaly detection", *arXiv:2002.10445*, 2020.

[19] K. Jin; H. Wang; C. Liu; Y. Zhai; L. Tang, "Graph Neural Network Based Relation Learning for Abnormal Perception Information Detection in Self-Driving Scenarios", *2022 International Conference on Robotics and Automation*, pp: 8943-8949, 2022

[20] Y. Liu, K. Zhao, G. Cong, Z. Bao, "Online anomalous trajectory detection with deep generative sequence modeling", *2020 IEEE 36th International Conference on Data Engineering*, pp: 949-960, 2020.

[21] J. James, "Sybil attack identification for crowdsourced navigation: A self-supervised deep learning approach", *IEEE Transactions on Intelligent Transportation Systems*, 22(7), pp: 4622-4634, 2020.

[22] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning", *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp: 2636-2645, 2020.

[23] W. Maddern, G. Pascoe, C. Linegar and P. Newman, "1 year, 1000 km: The Oxford RobotCar dataset", *The International Journal of Robotics Research*, 36(1) pp: 3-15, 2017.

[24] H. Machiraju, V. N. Balasubramanian, "A little fog for a large turn", *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp: 2902-2911, 2020.

[25] A. Bühler, A. Gaidon; A. Cramariuc, R. Ambrus, G. Rosman, W. Burgard, "Driving through ghosts: Behavioral cloning with false positives", *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp: 5431-5437, 2020.

[26] M. Shu, Y. Shen, M. C. Lin and T. Goldstein, "Adversarial differentiable data augmentation for autonomous systems", *2021 IEEE International Conference on Robotics and Automation*, pp: 14069-14075, 2021.

[27] D. Kingma, M. Welling, "Auto-encoding variational bayes", *arXiv:1312.6114*, 2013.

[28] D. Chen, B. Zhou, V. Koltun, P. Krahenbuhl, "Learning by cheating", *Proceedings of the Conference on Robot Learning*, pp: 66-75, 2020.

[29] I. Sobh, L. Amin, S. Abdelkarim, K. Elmadawy, M. Saeed, et al, "End-to-end multi-modal sensors fusion system for urban automated driving", *Conference and Workshop on Neural Information Processing Systems (NIPS)*, 2018.

[30] D. Chen, P. Krahenb, "Learning from all vehicles", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17222-17231,2022.

[31] K. Chitta, A. Prakash, "TransFuser: Imitation with Transformer-Based Sensor Fusion for Autonomous Driving", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[32] A. Vaswani, N. Shazeer, et al, "Attention Is All You Need", *Conference and Workshop on Neural Information Processing Systems (NIPS)*, 2017.

[33] F. Codevilla, E. Santana, A. Lopez and A. Gaidon, "Exploring the Limitations of Behavior Cloning for Autonomous Driving", *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp: 9329-9338, 2019.

[34] Z. Liu, K. Jin, Y. Zhai and Y. Miao, "Learning Robust Vehicle Navigation Policy Under Interference and Partial Sensor Failures", *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp: 15-21, 2022.