Visual Anomaly Detection and Multi-Agent Coordination in Autonomous Driving Settings

Ajith Senthil

***Abstract*—This project investigates anomaly detection for autonomous driving using both pixel-level semantic segmentation and binary classification approaches. The CARLA simulator environment is utilized with road hazards including weather, obstructing objects, and pedestrians. Anomaly detection is formulated as a 13-class semantic segmentation task using max logit adaptation. This Object Out of Distribution (OOD) method trains on the StreetHazards dataset with an anomaly class. In addition, Deep Active Inference (DAI) and a binary classifier are implemented for comparison on the binary anomaly detection task. Multi-agent experiments are conducted with Soft Actor-Critic (SAC) [6] [7] to enable collaborative responses. A decentralized SAC is utilized to allow efficient anomaly information sharing between agents. While full end-to-end results are still pending, preliminary anomaly detection capabilities have been demonstrated with the OOD segmentation approach. Early experiments in the multi-agent CARLA environment have also shown successful integration with the multi-agent soft actor-critic algorithm for the vehicles, however, the hazard detection is the only thing besides information about the vehicle. Only tests on the binary classification of anomalies have been conducted as a result optimal learning of the mSAC algorithm has not been achieved yet. Next steps include finalizing the training and integration of the DAI and SAC components. Additional benchmarking on the Combined Anomalous Object Segmentation (CAOS) dataset will evaluate generalization capabilities. The integrated system will be validated on the CARLA simulator prior to deployment on physical autonomous vehicles. In summary, this work has the potential to make multiple contributions including multi-class OOD anomaly detection using a computational neuroscientific learning algorithm, comparison of binary classifiers, and integration of decentralized multi-agent reinforcement learning for collaborative anomalous driving scenario responses. The code base can be found at the readme provides instructions to set up the system using docker and CARLA:**

https://github.com/ajithksenthil/MultiAgentRoadHazardAnomalyDetection

I. Introduction

In this project, I investigate multi-agent anomaly detection and response for autonomous driving simulations using novel deep learning and reinforcement learning techniques. Reliable anomaly detection is critical for ensuring effective and safe autonomous vehicle operation in complex urban environments. However, detecting anomalous objects or events in dynamic driving scenarios remains challenging.

I formulate anomaly detection both as a multi-class semantic segmentation task and as a binary classification task. The segmentation approach classifies each pixel as one of 13 classes including a dedicated anomaly class using an Object Out of Distribution detection method. For binary classification, I implement and compare Deep Active Inference (DAI) and a standard binary classifier. DAI provides an efficient anomaly detection approach based on information theory, but has not been explored yet for visual anomaly detection. For collaborative anomaly response, I utilize a decentralized multi-agent Soft Actor-Critic algorithm to enable agents to efficiently share information about detected anomalies. While multi-agent Soft Actor-Critic exists, my implementation is new for this anomaly detection application.

My experiments are conducted in the CARLA autonomous driving simulator using road hazards like weather, obstruction objects, and pedestrians. I use the StreetHazards an anomaly segmentation urban driving dataset for training and benchmarking the anomaly detection models. Initial results show successful anomaly detection with the multi-class semantic segmentation approach. However, I am still finalizing the training and integration of the DAI and multi-agent Soft Actor-Critic components before generating full end-to-end results.

Only tests on the binary classification of anomalies has been conducted as a result, optimal learning of the mSAC algorithm has not been achieved yet.

Results are at their early stages so it is too early to say if it worked or not. The results obtained so far have not demonstrated learning in the mSAC because testing has only been done on the binary anomaly detection integrated into the mSAC implementation. The max logit implementation shows decent pixel accuracy but the false positive rate is too high and it is likely the model does not generalize well. The binary anomaly detection have not shown success and are essentially just chance.

In summary, more experiments should be done and this project has the potential to make multiple key contributions to anomaly handling for autonomous driving, including novel applications of multi-class segmentation, efficient binary classifiers, and multi-agent reinforcement learning. The techniques have the potential to provide a robust anomaly detection and response pipeline suitable for complex urban environments.

II. Related Works

Existing Approaches to Visual Anomaly Detection:

Current techniques for visual anomaly detection in autonomous driving, including methods based on reconstruction errors, novelty detection, and maximum softmax probability (Hendrycks et al., 2017). However, these approaches have limitations when applied to large-scale, complex domains.

A recent paper by Hendrycks et al. (2022) introduces new large-scale benchmarks and proposes using the maximum

logit for substantially improved out-of-distribution detection in multi-class, multi-label, and segmentation settings. Their work pushes anomaly detection towards more realistic settings.

Anomaly Detection Data Sets:

The anomaly detection methods I have surveyed rely on "closed world" assumptions and may fail to generalize to novel objects and situations (Hendrycks et al., 2022). While synthetic datasets enable more open world modeling, they can introduce domain gaps compared to real sensor data that limit transferability (Hendrycks et al., 2022). This motivates my use of real-world urban driving datasets, however acknowledging that this too uses the "closed world" assumption and is therefore limited.

Advancements in Multi-Agent Coordination:

Recent work by Buckman et al. (2023) collects vehicle trajectory data with induced failures in a miniature city to train external failure detection. Their approach shows promise for detecting trajectory anomalies at intersections. I am interested in extending this principle to multi-agent scenarios by coordinating failures across multiple agents.

Multi-agent Soft Actor-Critic has been implemented in a centralized and decentralized manner.

My approach is related as it uses multi agent coordination and instead of detecting failure cases it is detecting visual anomalies in the camera sensor feed of each vehicle. The multi-agent soft actor critic implementation is built to train the agents (each vehicle) to avoid collisions.

III. Methods

Overview

The cornerstone of this project involves implementing multiple anomaly detection and segmentation models, focusing on integrating advanced visual feature extraction and object segmentation techniques. Key components include ResNet32 and ResNet101, pretrained on the ImageNet dataset, which are then integrated into PSPNet and UNet++ for effective segmentation. This setup is crucial for both binary anomaly detection by image and pixel-wise multi-class anomaly detection, adhering to the CAOS benchmark using the StreetHazards dataset, with the inclusion of an anomaly class as the 13th category.

Anomaly Detection Techniques

1. **Max Logit Technique for Anomaly Detection**: This method is applied to identify anomalies and predict class probabilities in both binary and multi-class OoD pixel-wise anomaly detection. It leverages the object detection capabilities of segmentation masks and features extracted through ResNet models.

2. **Deep Active Inference (DAI)**: DAI is tested as an approach for anomaly detection in both the binary and multi-class scenarios. Its application is primarily aimed at enhancing the predictive capabilities of the system in identifying anomalies.

Integration in mSAC Algorithm

The binary anomaly detection and the OoD anomaly detection methods are integrated into the multi-agent soft actor-critic (mSAC) algorithm. This integration is specifically tested within the CARLA Simulator environment. A key aspect of the state representation in this implementation is the inclusion of vehicle information such as telemetry, position, velocity, acceleration, and hazard detection data from the camera sensor. This approach is adopted due to the computational intensiveness of incorporating the entire visual feed into the state representation. The primary goal is to determine if hazard detection alone can suffice for effective agent training.

Visual Anomaly Detection and Street Hazard Detection Algorithms

Multiple visual anomaly detection and street hazard detection algorithms have been implemented to facilitate this process. These algorithms are crucial for enabling the system to identify potential hazards and irregularities in the environment, thereby informing the decision-making processes of the mSAC agents.

Overview of mSAC + Hazard Avoidance Implementation in GitHub

environment.py

- Role: Essential for setting up a realistic CARLA simulation environment, including vehicles, sensors, and the hazard detection model.

- Key Focus: Accurately simulate dynamic environments for agent interaction and decision-making, processing sensor data and translating vehicle actions within CARLA.

mSACModels.py

- Role: Hosts the neural network architectures for the mSAC algorithm, including Actor, Critic, and Mixing Network models.

- Key Focus: The Actor model selects actions, the Critic evaluates these actions, and the Mixing Network combines individual value functions for coordinated multi-agent decision-making.

replayBuffer.py

- Role: Manages the ReplayBuffer for storing and retrieving agent experiences.

- Key Focus: Efficiently handle past experiences to diversify training data, crucial for learning stability in complex environments.

trafficmanagerapi.py

- Role: Interfaces with CARLA's Traffic Manager to control NPC and traffic behavior.

- Key Focus: Create realistic traffic scenarios and unexpected events for testing agents' hazard avoidance strategies.

experiments.py

- Role: Conducts training, testing, and evaluation of mSAC agents within CARLA.

- Key Focus: Test agents' effectiveness in varied environments and hazard scenarios, assessing performance under different conditions.

mSACtrain.py

- Role: Contains the training loop for agent-environment interaction and policyunction updates.

- Key Focus: Optimize the learning process, balancing exploration and exploitation, and updating neural networks.

mSACagent.py

- Role: Defines the Agent class for decision-making and learning.

- Key Focus: Enable agents to make independent, informed decisions and collaboratively achieve hazard avoidance.

Each script plays a pivotal role in developing agents capable of effectively navigating and avoiding hazards in a dynamic simulation environment, thereby contributing to the overarching goal of sophisticated autonomous vehicle operation.

3.2 Simulation Overview

In the context of my project, I have designed a comprehensive simulation framework within the CARLA simulator to test and validate my anomaly detection and hazard avoidance strategies. This simulation is structured around three main types of hazards: pedestrians, static road blocks, and weather changes. Each episode within the simulation randomly introduces 10 hazards from these categories, creating a dynamic and challenging environment for my agents.

Hazard Types and Vehicle Setup

Hazard Types: The chosen hazards – pedestrians, static road blocks, and weather changes – are representative of common challenges encountered in real-world urban driving scenarios.

Vehicle Equipment: I equipped five vehicles with both RGB and collision sensors to detect and respond to these hazards.

A. Autopilot Functionality and Action Space

Autopilot Testing: Both autopilot-enabled and disabled modes were tested for the agents. This was crucial to understand the agents' performance under different levels of autonomy.

Reward Function Focus: It's important to note that the current reward function is primarily focused on collision avoidance, not on general driving proficiency.

Action Space: When autopilot is disabled, the action space encompasses throttle, steering, and braking, along with traffic manager controls. The use of traffic manager actions, such as changing lanes and speeds, forms the core of the action space when autopilot is enabled.

B. Visual Anomaly Detection and State Representation

Continuous Monitoring: Visual anomaly detection is an ongoing process throughout the simulation. This is facilitated through a camera callback mechanism, ensuring real-time monitoring and response.

State Representation Update: Each time an anomaly is detected, the state representation for the agents is updated with an anomaly detection flag. This flag plays a crucial role in informing the agents' decision-making process.

State Space Composition: The state space for each agent includes location, rotation, velocity, acceleration, environmental conditions, and vehicle telemetry. The inclusion of the hazard detection flag is a vital addition, enhancing the agents' awareness and responsiveness to anomalies.

Simulation Episodes

Each episode in my simulation is designed to run until a predefined maximum number of timesteps is reached. This setup allows us to evaluate the agents' performance over consistent and controlled timeframes, providing a standardized basis for assessment and comparison across different scenarios and configurations.

Through this carefully structured simulation environment, I aim to comprehensively test my agents' ability to detect and avoid hazards, adapt to changing conditions, and make informed decisions based on real-time data. This setup allows us to test autonomous vehicle behavior in complex urban environments and refining the algorithms driving their operation.

3.3 Theory for Deep Active Inference

Overview of Deep Active Inference

Concept

Deep Active Inference (DAI) combines the principles of deep learning with active inference, a theoretical framework originating from neuroscience. This framework is designed to explain the interactions between the brain and its environment. [5]

Key Principle

The central tenet of active inference is that the brain constantly updates its internal world model using sensory data. This update process aims to minimize the discrepancy between predicted and observed states, essentially reducing the 'surprise' or uncertainty associated with sensory inputs.

*1) **Anomaly Detection Problem Setup**:*

Processes and States

I consider 'N' processes, each of which can be in one of two states: normal (0) or anomalous (1). The primary goal in this setup is to accurately estimate the state vector 's' for these processes.

Observation and Cost

At each time step 'k', the agent observes some of these processes, gathering observations 'y' along with associated sensing costs. This dynamic captures the practical constraints and trade-offs involved in real-time anomaly detection.

2) DAI Framework in Context:

Generative Model

DAI employs a generative model, denoted as 'Q', which represents the probability distribution of the environment's states, possible actions, and observations. This model is used for understanding and predicting the dynamics of the environment.

Variational Distribution

The framework optimizes a variational distribution 'q', which is fine-tuned to minimize its divergence from the generative model 'Q'. This optimization aids in effective decision-making under uncertainty.

3) Implementation for Anomaly Detection:

Environment Representation

The approach uses a posterior belief '(k)' about the state vector 's', which reflects the agent's current understanding of the environment's state.

Action Selection

Actions are chosen based on the policy derived from the current belief state. This policy guides the agent's interactions with the environment to gather relevant information and make informed decisions.

Optimization Objective

The key objective is to minimize the Expected Free Energy (EFE) in decision-making processes. EFE can be made into a measure that captures the trade-off between the accuracy of predictions and the cost of gathering information in traditional DAI settings. This was not implemented for this implementation.

Deep Learning Application

In my implementation, neural networks are employed to model both the policy and the estimation of EFE. These networks are continuously updated based on incoming data, enhancing the agent's ability to detect and respond to anomalies.

4) Application in Project:

ResNet for Feature Extraction

I utilize ResNet to extract visual features from the StreetHazards dataset. This extraction process is key to updating the agent's beliefs about potential anomalies in the environment.

Active Inference for Decision Making

DAI is applied to enhance the efficiency and accuracy of anomaly detection. By leveraging this framework, the agent is able to make more informed decisions based on a comprehensive understanding of the environmental state and the likelihood of anomalies.

Through the integration of these elements, my approach aims to harness the strengths of deep learning and active inference in creating a sophisticated system for effective anomaly detection and response in autonomous vehicle environments.

3.4 Anomaly Detection

Methodology: Integrating Max Logit Approach for Object Out-of-Distribution Anomaly Detection

In this project, I have incorporated the Max Logit approach as a standard for Object Out-of-Distribution (OoD) anomaly detection. This method is a key part of a larger framework involving a multi-agent Soft Actor-Critic (mSAC) system with vehicle agents in the CARLA simulation environment [1]. Adopting the Max Logit approach, as outlined in the CAOS Benchmark, provides a baseline for assessing the effectiveness of anomaly detection in my autonomous vehicle navigation system.

Model Architecture and Implementation

For semantic segmentation and OoD anomaly detection, I chose the PSPNet model integrated with a ResNet-101 encoder. This decision was influenced by PSPNet's proficiency in handling high-resolution imagery and complex segmentation challenges. I adapted this architecture to identify a range of predefined classes, including an anomaly class during testing, which the model had not encountered during its training phase.

Implementation Analysis

In my implementation, I developed several key components: 'PolicyNetwork', 'BootstrappedEFENetwork', and 'BeliefUpdateNetwork'. Each of these networks plays a vital role in the pixel-wise, multi-class anomaly detection task.

PolicyNetwork

- Purpose: Determines actions for each pixel based on the current belief states.
- Implementation:
- Incorporates convolutional layers, essential for preserving spatial information.
- Concludes with a 'softmax' function, ensuring that the output for each pixel is a probabilistic distribution over the classes.

BootstrappedEFENetwork

- Purpose: Estimates the expected free energy (EFE) for each pixel, considering the current actions and belief states.
- Implementation:
- Utilizes convolutional layers, suitable for processing spatial data.
- Produces a single-channel output, representing the EFE for each pixel.

BeliefUpdateNetwork

- Purpose: Updates the belief states for each pixel based on new observational data.
- Implementation:
- Employs convolutional layers to process spatial features effectively.
- Outputs a belief map for each class at every pixel, while preserving spatial dimensions.

Summary

The integration of the Max Logit approach, coupled with the deep learning components, can be used for anomaly detection within an autonomous vehicle system. The construction of each network component allows the system to process visual data, update belief states, and make informed decisions, crucial for navigating complex urban environments.

Dataset and Preprocessing

The training and validation processes were conducted using the StreetHazards dataset from the CAOS benchmark, which provides a variety of simulated anomalous objects in driving scenes. There are 12 classes used for training: background, road, street lines, traffic signs, sidewalk, pedestrian, vehicle, building, wall, pole, fence, and vegetation. The thirteenth class is the anomaly class that is only used at test time. The dataset includes high-resolution images (720 × 1280), each paired with a corresponding semantic segmentation mask. My preprocessing steps involved resizing the images and segmentation masks to 512x512 and applying standard normalization techniques for the binary classification methods and max logit OOD. 256x256 for the deep active inference OOD implementation.

Anomaly Detection Approach

I incorporate Max Logit anomaly detection into my project as a benchmark approach based on (Hendrycks, 2022). The Max Logit method involves analyzing the maximum logit score across the class predictions for each pixel. This approach helps in effectively distinguishing between regular objects and anomalies that are out-of-distribution.

Training and Evaluation Methodology

In my project, the model was trained on the StreetHazards dataset, with an emphasis on accurate segmentation and effective anomaly detection. The evaluation metrics used were cross-entropy loss for training the model and Intersection over Union (IoU) for assessing the segmentation accuracy. These metrics were crucial in quantifying the model's performance in both standard segmentation tasks and in identifying anomalies.

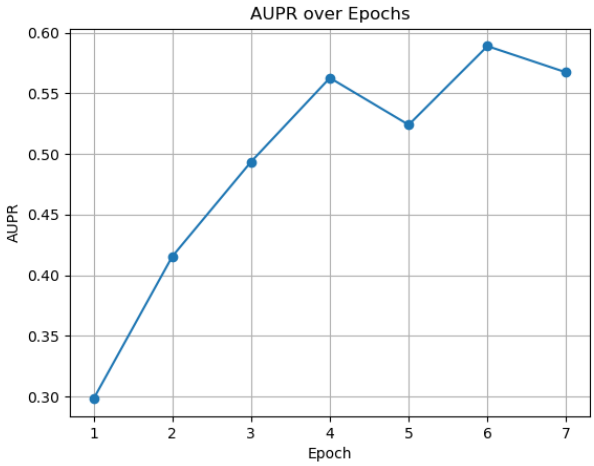

Fig. 1. AUPR over epochs, measuring the model's precision and recall balance in anomaly detection.

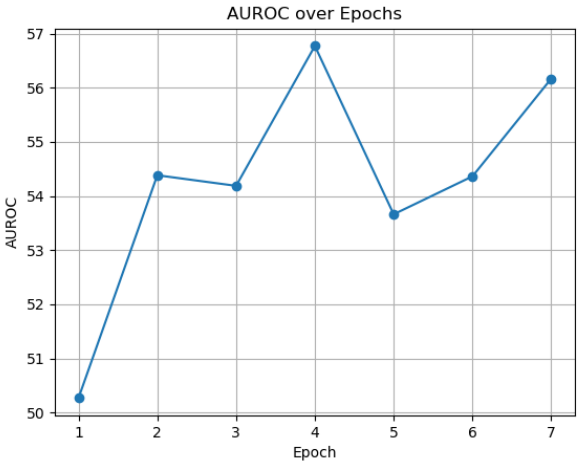

Fig. 2. AUROC trend over epochs, indicating the model's ability to distinguish between normal and anomalous data.

Training and Preliminary Results

The training process was initiated, focusing on both segmentation accuracy and OoD anomaly detection capabilities. Limited to the first seven epochs due to computational constraints, this early phase offered insights into the system's performance and the potential need for further refinement.

The figures below summarizes the results of the Max Logit OoD anomaly detection and semantic segmentation results:

Analysis of Initial Results

The results summarized in fig 1, 2, 3, 4, 5 while preliminary, indicate a start, with the model demonstrating a slight upward trend in pixel accuracy and mean Intersection over Union (mIoU). However, the Area Under the Precision-Recall Curve (AUPR) and the Area Under the Receiver Operating Characteristic Curve (AUROC) metrics suggest there is significant room for improvement, particularly in the model's ability to differentiate between in-distribution and out-of-distribution data. The 100 percent False Positive Rate at 95 percent True Positive Rate (FPR95) indicates that the model currently identifies all pixels as anomalous when aiming for high recall, underscoring a need for calibration in the anomaly scoring mechanism. Results can be seen in Table 1 and 2 for the training and validation respectively.

Forward Plan

Moving forward, I intend to extend the training to more epochs, aiming to solidify the trend of improving accuracy and IoU while addressing the challenges highlighted by the AUPR and AUROC results. The goal is to enhance the model's discriminative power, reducing false positives without compromising the true positive rate.

Integration with Multi-Agent System

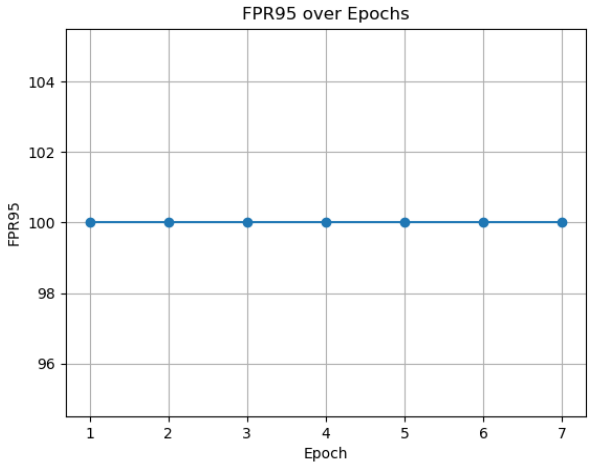

Fig. 3. False Positive Rate at 95 percent True Positive Rate (FPR95): This plot maintains a constant rate over the epochs, representing the proportion of normal data misclassified as anomalies when the true positive rate is 95 percent. The consistent FPR95 across epochs warrants further investigation to enhance model reliability.

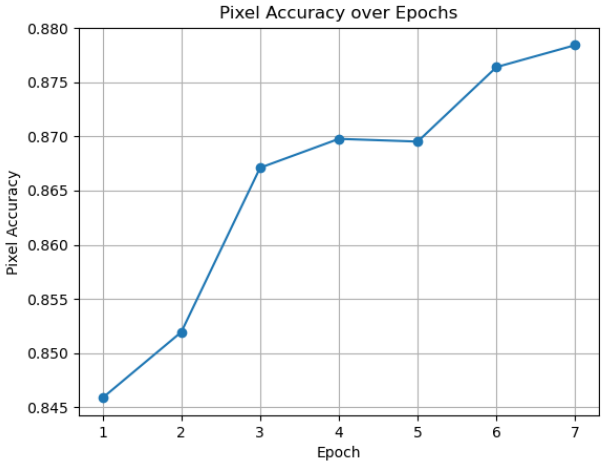

Fig. 4. Pixel Accuracy across Epochs: The plot illustrates the model's increasing pixel accuracy over epochs, reflecting enhanced precision in identifying correct pixels for both anomaly detection and semantic segmentation tasks.

The OoD anomaly detection model was integrated into a larger system where vehicle agents, operating in the CARLA simulator, were equipped to detect anomalies using this approach. The agents, trained on mSAC, intended to respond dynamically to the detected anomalies, in order to demonstrate the real-time application of this technology in autonomous vehicle navigation.

3.5 Methodology: Deep Active Inference for Object Out-of-Distribution Anomaly Detection

In my project, I have developed a novel Deep Active Inference (DAI) approach for Object Out-of-Distribution (OoD) anomaly detection. This approach is intended to complement

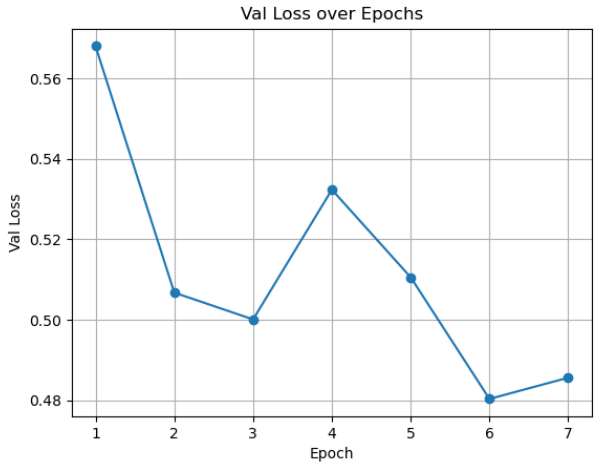

Fig. 5. Decreasing validation loss over the first 7 epochs, indicating improving model performance.

and be compared against the baseline established by the Max Logit method. It forms a part of a broader system utilizing a multi-agent framework in an autonomous driving simulator, where accurately detecting anomalies is vital for the safe operation of autonomous vehicles.

Deep Active Inference Framework The deep active inference framework is grounded in the principle of minimizing variational free energy, which represents a drive to reduce uncertainty or 'surprise' within the system's predictions. In the context of semantic segmentation and OoD detection, the model learns to infer the normality or abnormality of objects in the driving scene, based on the prediction of segmentation classes and the associated uncertainty.

Feature Extraction with PSPNet For high-resolution input and detailed segmentation requirements, I have employed the Pyramid Scene Parsing Network (PSPNet) with a ResNet-101 encoder as the backbone for feature extraction. The PSPNet is chosen for its ability to capture contextual information at multiple scales, crucial for the precise delineation of objects in a scene and identification of anomalies.

Model Architecture Adaptation

Within the DAI framework, the following components have been adapted to handle the rich, multi-scale feature maps produced by PSPNet:

5) ***PSPNetFeatureExtractor::*** Functions as the feature extractor, providing comprehensive feature maps from input images. The classification layer is modified with an identity mapping to focus on feature extraction.

6) ***Belief Update Network::*** Processes the PSPNet feature maps to update the belief states, signifying the model's confidence in the presence of various object classes.

7) ***Policy Network::*** Based on updated beliefs, this network makes decisions aimed at minimizing the expected free energy, effectively choosing actions that reduce uncertainty.

8) ***Bootstrapped EFE Network::*** Predicts the expected free energy using updated beliefs and decisions from the policy network, guiding the model toward more certain predictions.

9) ***Training and Inference****:* The training involves updating the belief state based on the segmentation output, where the model learns to predict the presence of anomalies across different object classes. At inference time, the model uses the learned policies and beliefs to detect OoD objects in new scenes.

10) ***Comparison with Baseline****:* The DAI methodology will be evaluated against the Max Logit baseline. This comparison focuses on each method's ability to accurately segment known objects and detect anomalies, particularly those subtle or rare in the training data.

Summary

This methodological framework, based on active inference principles, presents an approach to semantic segmentation and OoD anomaly detection. Deploying this within a multi-agent autonomous system, I aim to explore the benefits of an active inference-based model in complex, dynamic environments, contributing valuable insights to autonomous navigation technologies.

3.6 Decentralized Communication in Multi-Agent System

In the implemented system, vehicles do not communicate directly with each other, operating in a decentralized manner. However, they can achieve a level of coordination through indirect communication and shared environment feedback:

Indirect Communication

- Agents share their states through the ReplayBuffer, including position, velocity, sensor data, and other details. This shared information leads to indirect coordination.
- All agents use the same mSAC algorithm and neural network architecture, promoting similar behavior and understanding of the environment.
- The reward function, which incentivizes collision and hazard avoidance, encourages cooperative behavior among agents.

Shared Environment Feedback

- Environmental changes caused by other agents' actions provide feedback, influencing each agent's decisions.
- The Traffic Manager, which controls NPCs and traffic, indirectly influences the agents, providing coordination through centralized control.

Direct communication is absent which is a reason why the agents would have a hard time coordinating. Future experiments with a method of communication are necessary.

Note: Additional Experimental results will be added as part of the supplemental information

C. Experimental Results on Binary Anomaly Detection

The accuracy on both Deep Active Inference and the simple ResNet Binary classifier trained on the StreetHazards data set, sweeped through relevant hyperparameters, show around 50 percent accuracy or basically chance likely due to the nature of the problem. Anomaly detection by image, where the anomaly can be located anywhere in the image is likely a problem a binary classification task would struggle with at least the way I implemented it. Accuracy of the sweep on deep active inference binary image wise anomaly detection can be found in figure 6.

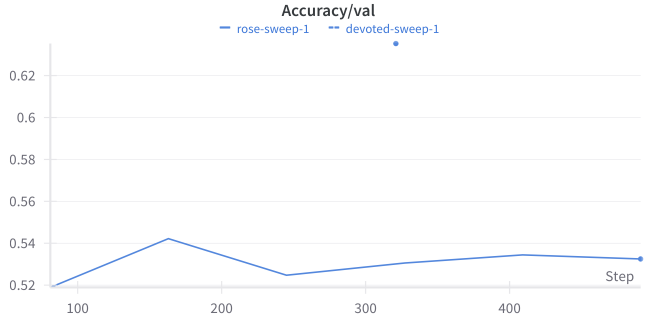

Fig. 6. Deep Active Inference: Binary Image Wise Anomaly Detection Accuracy Sweeped over Hyperparameters

D. Experimental Results on Collision Avoidance with Binary Anomaly Detection and multi agent Soft Actor-Critic

Figure 7, 8, 9, 10, 11 summarize the results of the mSAC algorithm trained using the binary anomaly detection hazard detection system. 5 agents each vehicles with RGB and collision sensor controlled by the traffic manager with an action space of left or right lane change. 5 hazards were introduced each episode randomly from the 3 hazard types. Learning likely did not occur because of the bottle neck with the anomaly detection. Future tests using a more robust anomaly detection system are required.

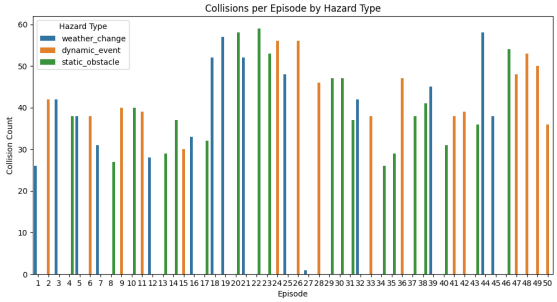

Fig. 7. Collisions per episode for 50 episodes using binary anomaly detection and mSAC

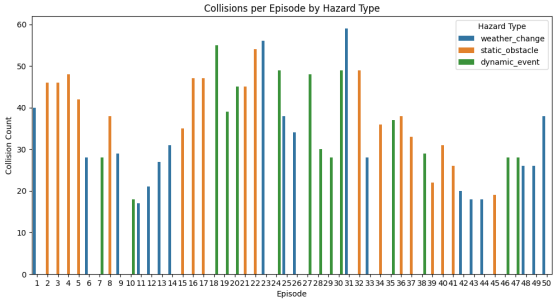

Fig. 8. Collisions per episode for 50 episodes with 5 hazards randomly added each episode using binary anomaly detection and mSAC

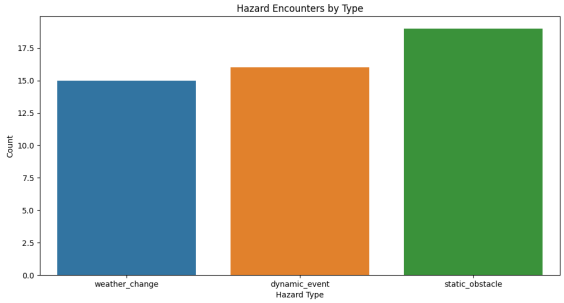

Fig. 9. How many times the visual anomaly detection system detected a hazard throughout the simulation

E. *Future Directions*

Future experiments need to be conducted to verify the efficacy of the set up and algorithm. The system is set up to allow for a variety of anomaly detection algorithms to be tested on a multi agent autonomous driving environment using the Soft Actor-Critic decomposed for multi agent settings allowing for decentralized learning for collision avoidance.

REFERENCES

[1] CARLA Simulator. *CARLA: An Open Urban Driving Simulator*. Available at: http://carla.org/

[2] Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joe Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song, *Scaling Out-of-Distribution Detection for Real-World Settings*, May 15, 2022.

[3] Bogdoll, D., Uhlemeyer, S., Kowol, K., & Zöllner, J. M. *Perception Datasets for Anomaly Detection in Autonomous Driving: A Survey*. 2023 IEEE Intelligent Vehicles Symposium (IV), pages 1–8, 2023. https://doi.org/10.1109/IV55152.2023.10186609

[4] Jumman Hossain, *Autonomous Driving with Deep Reinforcement Learning in CARLA Simulation*, Department of Information Systems, University of Maryland, Baltimore County (UMBC), Email: jumman.hossain@umbc.edu, Year of Publication.

[5] G. Joseph, C. Zhong, M. C. Gursoy, S. Velipasalar, and P. K. Varshney, "Anomaly Detection via Controlled Sensing and Deep Active Inference," *arXiv preprint arXiv:2105.06288*, 2021. [Online]. Available: https://doi.org/10.48550/arXiv.2105.06288

[6] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine, "Soft Actor-Critic Algorithms and Applications," *arXiv preprint arXiv:1812.05905*, 2019. [Online]. Available: https://arxiv.org/abs/1812.05905

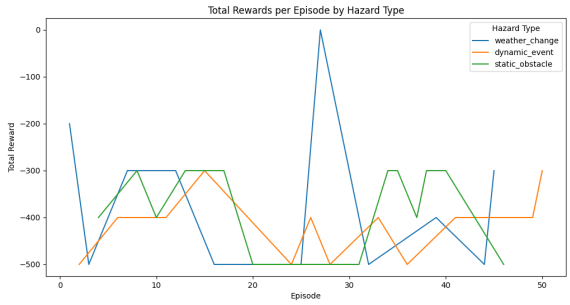

Fig. 10. The total rewards for all agents summed per episode in the simulation

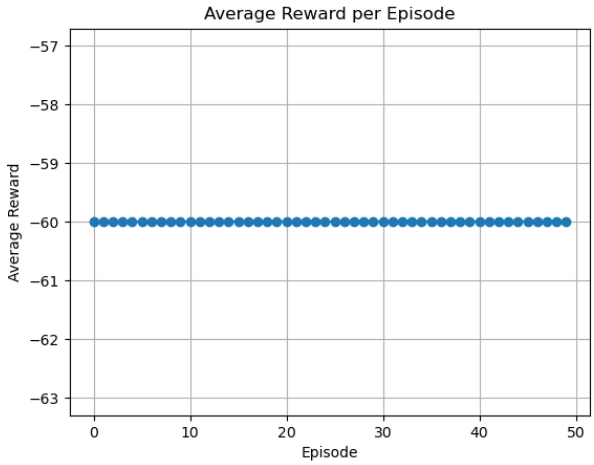

Fig. 11. Avg rewards per training episode for binary anomaly detection and mSAC response

[7] Pu, Y., Wang, S., Yang, R., Yao, X., & Li, B. *Decomposed Soft Actor-Critic Method for Cooperative Multi-Agent Reinforcement Learning*. arXiv preprint arXiv:2104.06655, 2021. https://doi.org/10.48550/arXiv.2104.06655

[8] Z. Zhang, Y. Sun, F. Huang, and F. Miao, *Safe and Robust Multi-Agent Reinforcement Learning for Connected Autonomous Vehicles under State Perturbations.* arXiv, Sep. 20, 2023. Accessed: Dec. 10, 2023. [Online]. Available: http://arxiv.org/abs/2309.11057