

aUToronto Concept Design Report Year 4

Martin Ffrench, Mustafa R. A. Khan, Shichen Lu, Jingxing Qian, Ankur Samanta, Gorden Tseng, Skye Wu, Haowei Zhang, Sherry Zuo, Angela P. Schoellig, Timothy D. Barfoot
University of Toronto, May 13, 2021

1 INTRODUCTION

The GM/SAE AutoDrive Challenge is a four-year collegiate competition that aims to develop a SAE Level 4 (L4) autonomous vehicle capable of operating on an urban closed course. As part of the University of Toronto's initiative in the competition, our team, aUToronto, consists of more than 70 undergraduate and graduate students from a diverse educational background. During the previous four years, our team has developed an automated driving system (ADS) for our vehicle, Zeus, with the ability to perform lane-keeping, navigate using address points, avoid both static and dynamic objects, recognize traffic signals, and handle intersections with dynamic pedestrians [1][2]. The Year 4 competition focuses on achieving SAE Level 4 of Driving Automation and features two types of Dynamic Challenges.

The 99% Buy Off Ride (BOR) Challenge builds on top of the previous years' challenges by integrating all the capabilities into a much longer route. The challenge is designed to simulate a ride-sharing scenario where competing vehicles are required to autonomously navigate to a series of predetermined addresses and perform pick-up and drop-off operations. This year's route involves a diverse set of scenarios including construction zones, traffic lights, traffic signs, static and dynamic objects, and both controlled and uncontrolled intersections. The Level 4 Challenge takes it one step further by simulating a critical loss-of-GPS failure scenario, where the vehicles must automatically perform *fallback* or *failure mitigation* strategies and achieve a *minimum risk condition* in order to demonstrate SAE Level 4 capability.

In order to score full marks in the Dynamic Challenges, Zeus must respond appropriately to the aforementioned scenarios while providing a smooth riding experience for the passengers, as well as actively monitor the ADS performance and engage in fallback or failure mitigation strategies when a failure occurs. With this in mind, it is crucial to design a system that demonstrates responsive and robust functionality under various environment and system health conditions. Developing such a system given minimal resources and a constrained time frame required versatile design choices, retrospective reflection and a markedly innovative approach.

The report first summarizes the major changes to our hardware and software systems and the lessons we

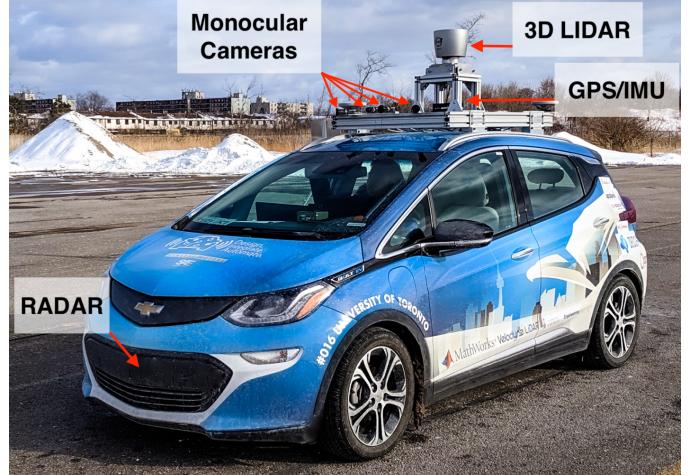


Figure 1: aUToronto's self-driving car Zeus at a test track in Oshawa.

learned while updating Zeus to its Year 4 configuration. It also outlines potential advancements to achieve SAE Level 4 in the future. The report then describes the desired operational design domain (ODD) for a hypothetical SAE Level 4 ADS, followed by a design failure mode and effect analysis (DFMEA) that identifies the four most critical failure modes for the system. Finally, report discusses the significance and nature of each failure mode while proposing relevant ADS fallback strategies that could help mitigate them.

This report is organized as follows:

1. Introduction
2. Design Review: System Design Review
3. Design Failure Mode and Effect Analysis
4. Failure Mode 1: Loss of GPS
5. Failure Mode 2: Semantic Map Inaccuracy
6. Failure Mode 3: Compromised Perception Sensors
7. Failure Mode 4: Dynamic Object Motion Prediction Error
8. SAE Level 4 Conditions
9. Papers, Patents and Conference
10. Conclusion
11. References

2 DESIGN REVIEW

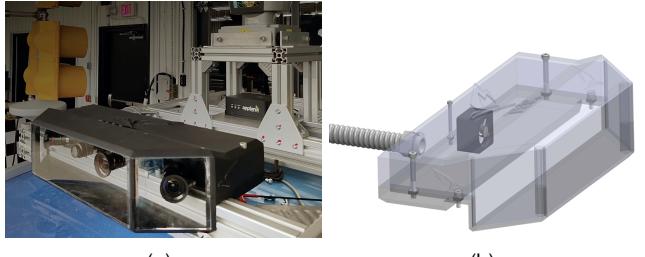
This section extends the Year 3 Report [3] and reviews the major changes to our mechanical, computing, and software systems, as well as the lessons learned along the journey. Potential updates in methodology and approach to achieve SAE Level 4 are also discussed.

2.1 MECHANICAL AND SENSOR INTEGRATION

For the past four years, our design for sensor mounting and placement have focused on robustness to disturbance and versatility in accommodating various sensor configurations. As such, we currently employ a sensor placement strategy that provides maximum perception coverage and avoids occlusion from other sensors and structures.

Year 1 Design and Lessons Learned The roof rack, shown in Figure 2, consists of aluminum extrusions that are mounted to Zeus' sport rails. This structure provides a base for reliable and configurable sensor attachments where most sensors can be mounted with a simple mounting block. The mounting system has not shown any signs of wear or deformation since Year 1. Our Year 1 sensor suite included a Velodyne HDL64 LiDAR, a Bumblebee stereo camera, and a forward facing monocular camera. However, we learned that the aforementioned sensor suite was not optimal, since the Velodyne LiDAR provided better depth measurement and so diminished the purpose of the Bumblebee stereo camera, while the field of view (FOV) of the monocular camera was not sufficient for complex tasks such as pedestrian handling.

Year 2 Design and Lessons Learned Since the competition did not require any backward movements or handling of overtaking vehicles, we focused on maximizing the FOV in front of Zeus. Accordingly, we added four Blackfly S



(a)

(b)

Figure 3: Left: assembled enclosure on the vehicle's roof rack. Right: transparent view.

monocular cameras: two facing forward and two angled at 45 degrees. One of those forward-facing cameras was equipped with a long-range lens to boost the visual detection range. Additionally, an ARS430 radar was mounted directly to the impact bar. We learned that the use of multiple radar units resulted in receiver crosstalk and diminished their performance; thus, we opted to use a single radar unit. We also observed that our sensor suite performance degraded drastically under harsh weather conditions due to condensation and dust accumulation on the cameras and LiDAR.

Year 3 and Year 4 Design and Lessons Learned In Year 3 and Year 4, the Novatel GNSS system was replaced with an Applanix dual-antenna solution to improve heading and positioning accuracy. The 4 USB cameras were replaced with Gigabit Ethernet versions and are now connected through a single 10-gigabit Ethernet switch. The cameras are also synchronized such that a single camera acts as the master trigger for the others. Finally, The cameras and the Velodyne LiDAR are now time-stamped using GNSS time. Figure 2 illustrates the placement of the current sensor suite on the roof rack. However, through

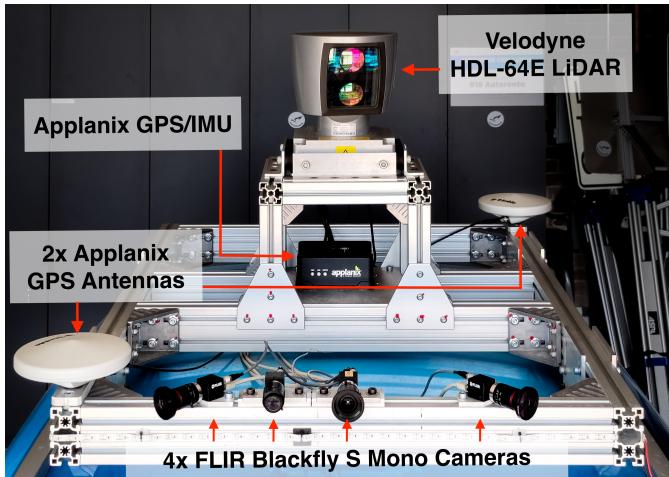


Figure 2: Vehicle's roof rack mounting structure and mounted sensing components. Weatherproof enclosure is not installed.

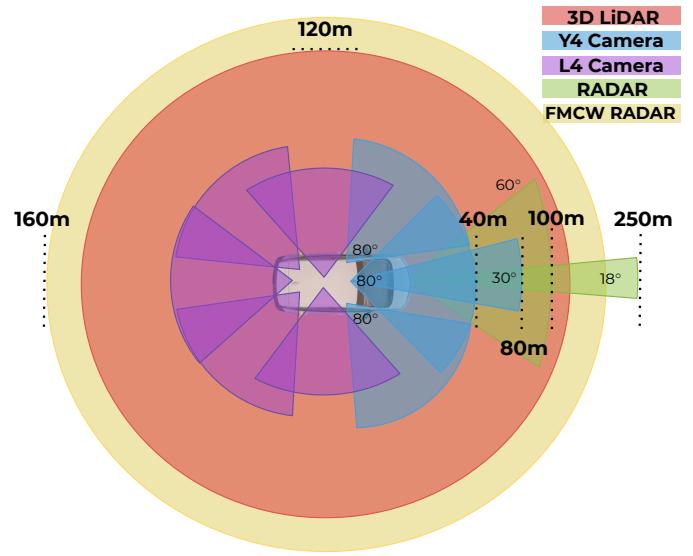


Figure 4: Field of view of the Year 4 sensor suite and hypothetical L4 sensors (L4 cameras and FMCW radar)

further experimentation, we noticed that the current sensor layout produced many blind spots close to the vehicle, which could limit our performance when executing delicate maneuvers such as pulling over to a road shoulder.

To achieve all-weather operational capability, our team designed a weather-proof, temperature-regulated enclosure for the cameras, as shown in Figure 3. The enclosure shields the camera lens from water and dust, while also preventing fogging through the use of internal heaters. Unfortunately, the team is still unable to fully test the new sensor enclosure’s performance in harsh weather conditions such as heavy rain and snowfall, since such circumstances can often corrupt the measurements of the currently unprotected Velodyne LiDAR.

Towards Level 4 While the current mounting system and sensor suite provide satisfactory performance for the competition, we have identified several potential improvements in order to achieve SAE Level 4 in more realistic driving scenarios, including 1) adding additional cameras to cover visual blind spots around the vehicle and create backup systems; 2) upgrading to a better LiDAR, such as the Velodyne VLS-128, to achieve better detection performance for distant objects; and 3) adding an additional Frequency-Modulated Continuous Wave (FMWC) scanning radar to improve perception under inclement weather conditions. Figure 4 also shows the placement and coverage of the hypothetical Level 4 sensors.

2.1.1 Computing Platform Several of our software components, such as object detection and traffic light detection, present a heavy computational load. Additionally, the design criterion for the computing platform has been focused on reducing end-to-end latency between sensor inputs and control actions.

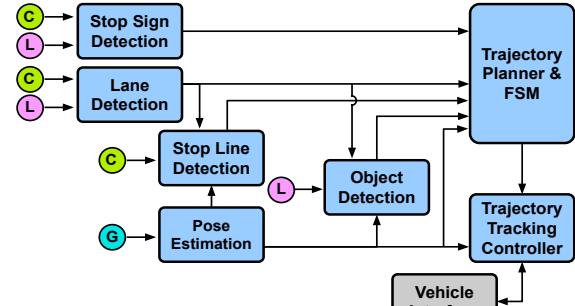
Year 1 and Year 2 Design In Year 1 and Year 2, the Intel Crystal Rugged server was selected as the main computing platform. The server has two Intel Xeon E5-2699v4 processors that together contain 44 physical cores operating at 3.6 GHz. The peak floating-point operation throughput is 1.8 TFLOPS. We also utilize one Intel Arria 10 FPGA as a deep neural network (DNN) accelerator which delivers a peak floating-point operation of up to 1.5 TFLOPS.

Year 3 and Year 4 Design and Lessons Learned To better support the computational load for the Year 3 and Year 4 competitions, we have added two more acceleration cards to the server for a total of three. For compatibility with the recent Intel OpenVINO 2020.301 release, we chose Intel Arria 10 GX1150 programmable acceleration cards (PACs) over the previous Arria 10 development kit. However, through experimentation we noticed that the current setup (Ubuntu 16.04, OpenVINO 2020.0301, Aocl 19.2, PAC Stack 1.2) only supports 2 FPGA devices. We plan to update our run-time environment after the Year 4 competition to fully utilize all 3 FPGAs.

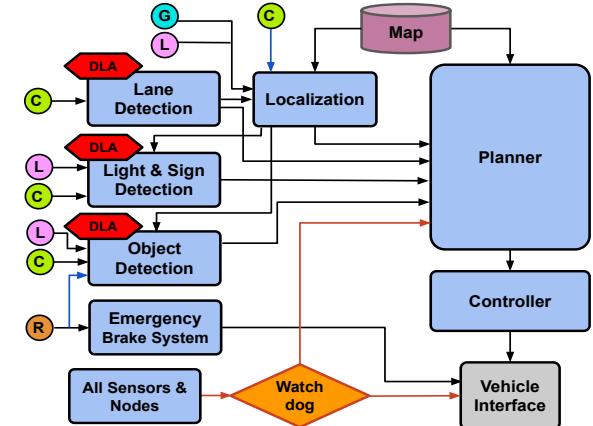
Towards Level 4 There are several shortcomings in our current system that could pose performance and safety concerns. First, the three Intel PACs only provide up to 4.5 TFLOPS, which is not sufficient to support real-time performance of more sophisticated deep learning models. As such, we must seek more powerful hardware acceleration devices for safer and faster driving. Second, the current system lacks backup computing and power. In Year 4, the team experienced a power distribution failure which incapacitated the entire ADS system, and the safety driver had to intervene. To achieve the Level 4 fallback requirements, we must add an independent computing platform that runs in parallel to the main sever, while being powered by a separate power source.

2.2 ALGORITHM SOFTWARE This section presents design changes of the overall software architecture and major autonomy algorithms of Zeus - Mapping and Localization, Perception, Navigation and Control - as well as lessons learned along the way.

2.2.1 Software Architecture Since Year 1, Zeus’ software architecture follows the perception-computation-



(a) Year 1 architecture



(b) Year 2 to Year 4 and Level 4 architecture

Figure 5: Zeus’ software architecture Year 1 to Year 4 and Level 4. C: Cameras, L: LiDAR, G: GNSS, R: Radar, DLA: Deep Learning Acceleration. Blue indicates new changes in Year 4 and Red indicates hypothetical Level 4 updates.

action paradigm using a modular design. Different functional components, in the form of one or more ROS nodes written in C++, are grouped as independent modules with pre-specified I/O interfaces. This design enables easy testing and modification to the modules and the overall system. Communication between modules are facilitated by ROS, sensor drivers, and custom CAN messages.

Year 1 Design and Lessons Learned The Year 1 design, shown in Figure 5(a) Raw outputs of the sensors are consumed by the perception nodes. Computed high-level understandings of the environment are passed to the Planner to generate a trajectory for the Controller. The Controller then executes commands through the Vehicle Interface. Zeus completed all of the required dynamic tasks in Year 1. However, since Year 1 tasks only required the vehicle to follow painted lane markings, the system lacked global information required for more complex autonomy tasks.

Year 2 Design and Lessons Learned Year 2 competition required the vehicle to navigate through a mock urban environment. A custom Mapping service, Zeus Map, was introduced to provide global semantic map information for both perception and planning [4][3]. A custom Intel FPGA-based inference engine (DLA) was added to accelerate the perception modules. Finally, watchdog and emergency braking systems were proposed to perform *failure mitigation*. The modular design adopted since Year 1 was crucial in making the system more versatile and generalizable: the entire software architecture was agnostic to the actual algorithms implemented by the modules, allowing for rapid prototyping and performance evaluation.

Year 3 and Year 4 Design The overall design was deemed sufficient for the Year 3 and Year 4 competitions; hence, we focused on improving each of the individual modules as discussed below.

Towards Level 4 The most distinct difference between SAE Level 3 and Level 4 is that a Level 4 ADS must monitor its own performance and automatically perform *fallback* or *failure mitigation* strategies when a failure occurs. The current design, which features a watchdog and E-stop system that directly interfaces to the low level controller, is only able to prevent imminent collisions (*failure mitigation*). This is not sufficient for Level 4 *fallback* handling. To achieve Level 4, the watchdog system must also report performance status back to the ADS such that the ADS can execute suitable *fallback* decisions (e.g. slow down and pull over) when performance is degraded. Figure 5(b) shows the current system design and proposed Level 4 extension.

2.2.2 Mapping and Localization We have developed and evaluated many different state estimation strategies including lane detection, GNSS/IMU (with an RTK ground station), visual odometry, LiDAR localization and SLAM, and semantic localization.

Year 1 Design and Lessons Learned Our successful completion of the Year 1 lane following tasks heavily relied on reliable and real-time lane detection. We developed fast image- and LiDAR-based lane detection pipelines; GNSS and wheel odometry and visual odometry was also evaluated. Several lessons were learned: 1) wheel odometry is not reliable even at low speeds; 2) visual odometry with a stereo camera suffers from scaling and drifting errors; and 3) GNSS may experience frequent dropouts in wooded areas. As a result, we opted for GNSS and lane detection as the primary localization strategies for Year 1.

Year 2 Design and Lessons Learned In Year 2, the vehicle was required to navigate on unpainted roads and therefore lane detection could not be used and a global localization strategy was required. Figure 6 illustrates our custom mapping service, Zeus Map, that imports third-party HD maps and provides semantic information to other autonomy modules. To handle GNSS dropouts and GNSS-denied areas, LiDAR localization was adopted. Lessons learned from these attempts were: 1) The accuracy of LiDAR localization is comparable to GNSS with a RTK station in well-mapped environments; 2) LiDAR localization may suffer in open environments and areas where the reference pointcloud map does not match observations; and 3) GNSS positioning is often biased with respect to third-party semantic maps and this offset must be calibrated.

Year 3 and Year 4 Design and Lessons Learned To achieve more robust GNSS-free navigation, a semantic localization system that looks for correspondences between perception node outputs and Zeus Map semantic cues was developed to provide global positioning when GNSS is unreliable [3]. Similar to LiDAR localization, a prior reference map is required for semantic localization to work properly. Moreover, we noticed that the actual painted lane markings usually do not match the markings from reference semantic map. In order to navigate in unmapped areas, we further investigated the possibility of using LiDAR SLAM to simultaneously map the environment and localize the vehicle. More details are discussed in Section 4.

Towards Level 4 Through experiments, we demonstrated that it is crucial to have reliable global localization (GNSS or pre-installed anchors). If the vehicle ever needed to navigate without GNSS for extended period of time,

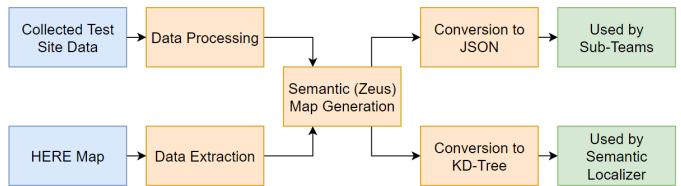


Figure 6: Pipeline to convert third-party semantic map to Zeus Map format

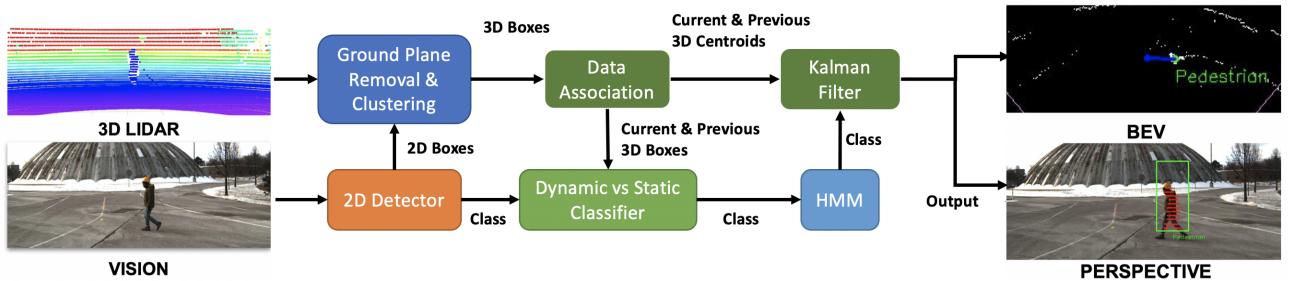


Figure 7: Improved aUToTrack that features object class-based motion models.

then accurate, up-to-date reference maps must be readily available to the ADS. Section 6 presents some potential projects that can improve the robustness of the localization and mapping system.

2.2.3 Perception Detecting lane lines, traffic signals and objects around the vehicle is critical for safe navigation. Over the last 4 years, we have developed various perception algorithms, both classical and learning-based.

Year 1 Design and Lessons Learned In Year 1, Zeus was required to actively detect lane lines, stop signs and static obstacles that blocked lanes. The lane detection pipeline consisted of steerable filters, convolutional networks, and a LiDAR-based gradient filter to extract lane marking pixels in a bird’s eye view (BEV) image. A Haar cascade classifier [5] was used to detect stop signs and an occupancy grid was used to detect obstacles [6]. The lessons learned here were: 1) accurate detection relies on high image resolution and fast shutter speed; 2) classical vision methods such as Haar cascades can perform on par with modern machine learning methods for simple objects such as traffic signs and are much easier to train; and 3) learning-based methods perform better under changing illumination conditions.

Year 2 Design and Lessons Learned Zeus was challenged to detect more types of traffic signs, traffic lights, pedestrian dummies and mock vehicles in Year 2. More Haar cascade classifiers were trained for signs, and SqueezeDet-based [7] detectors were deployed for traffic light and pedestrian detection. A light-weight object tracking pipeline, aUToTrack, was proposed to estimate object motion [8]. To achieve better traffic light detection accuracy, we associate detected traffic lights to their expected position in the image frame as stored in Zeus Map [2]. All training data was either collected in Toronto or extracted from YouTube videos. There were many lessons learned in Year 2: 1) quality and variety of training data has a substantial impact on performance; 2) false negatives can be more detrimental than false positives for safety concerns; and 3) LiDAR point clustering performs well for pose estimation of pedestrians but not of large objects such as vehicles; to retrieve accurate vehicles poses, a modern 3D object detector is required.

Year 3 Design and Lessons Learned In Year 3, the drawbacks of the Year 2 design were addressed by exploring different algorithms for object and traffic light detection, and further developing aUToTrack. We implemented a PointPillar-based [9] 3D detector for pedestrians to estimate poses. Additionally, an improved aUToTrack allowed us to achieve a higher performance, reliability, and frame rate, all while running only on CPUs [3]. For traffic lights detection, SqueezeDet and YOLOv3 architectures were applied; while SqueezeDet suffered from Off-State Detection, YOLOv3 achieved desirable performance. However, due to the large backbone of YOLOv3, it had a slower inference rate than SqueezeDet.

There were many lessons learned in Year 3: 1) Models trained using a dataset with a lower LiDAR beam count cannot simply be applied to a LiDAR of a higher beam count. Additional strategies such as downsampling and augmentation must be applied. 2) SqueezeDet was the primary object detector, but its lightweight nature proved insufficient for accurate detections in low contrast and distant regions. A detector with more expressive power, such as YOLOv3, would be needed for successful detections in these scenarios. 3) Larger models like YOLOv3 could result in better detection performance compared to lightweight models like SqueezeDet, but we must strike a balance between inference speed and input resolution to maintain real-time detection performance on CPUs and FPGAs. 4) It can be difficult to minimize the spurious detections from the LiDAR point clustering detector while being able to capture objects of various sizes and distances. Separate modules may be needed to more effectively detect different types of obstacles at different ranges.

Year 4 Design and Lessons Learned In order to achieve more robust vision-based object detection in low-light and distant regions, YOLOv3 is deployed. With our optimized DLA pipeline, the YOLOv3 model can process (416x416) camera images at 20Hz on the Intel Arria 10 PAC. Point-Pillars struggled during domain transfer between training on the nuScenes Data set (collected using a 32-beam LiDAR), and inference using Zeus’ LiDAR input (64-beam LiDAR). By employing strategies like down-sampling, performance was slightly improved, yet still not comparable to the performance of 2D Object Detectors. The aUToTrack pipeline is further upgraded with better ground plane segmentation to boost effective perception range and ob-

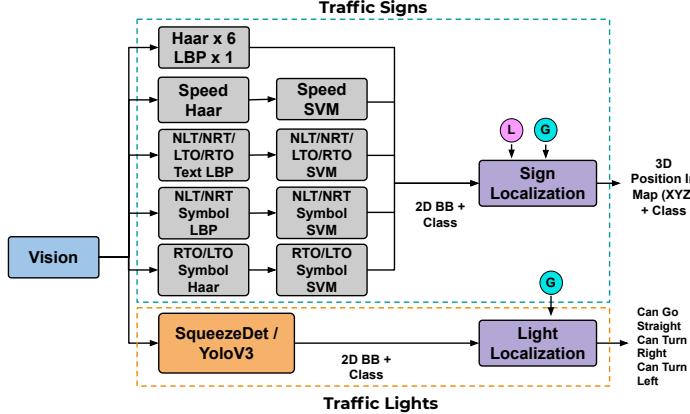


Figure 8: Year 4 object detection and traffic signal detection pipelines. L: LiDAR, G: GNSS

ject type-based motion models to improve tracking performance for both static and dynamic objects. The new object detection and tracking pipeline is shown in Figure 7. We are also tasked with detecting new traffic signs. Again, we utilize the existing traffic signs detection pipeline where the signs are first detected and classified with 2D cascaded models and SVMs, and then localized in 3D by finding the closest LiDAR clusters that projects onto the 2D detection bounding boxes. The complete traffic signal detection pipeline is shown in Figure 8.

Towards Level 4 The current system is optimized for the competition requirements. Nevertheless, the strategies and design choices for object and traffic sign detection and tracking can be extended to more complex environments. For instance, we can deploy more expressive DNNs and train them on larger data sets to support more types of traffic signage and object classes; we can use more sophisticated, adaptive motion models to improve the object tracking performance; and we can utilize watchdog systems to monitor the health of both the software and camera hardware systems. More details are discussed in Section 5 and Section 7.

2.2.4 Navigation and Planning The navigation and planning pipeline generates trajectories for the vehicle. We have developed three different planning strategies with growing complexity and capability over the years.

Year 1 Design and Lessons Learned The Year 1 competition required the vehicle to track its current lane. Hence, only local paths ahead of the vehicle were necessary. A robust quadratic path generation pipeline was implemented. The pipeline first ran RANSAC on the detected lane lines to reject outliers. Linear Least Squares was then run on the inliers to yield the instantaneous quadratic path parameters. To further improve the smoothness and robustness of the paths generated, path parameters were filtered by a linear Kalman filter. One lesson learned

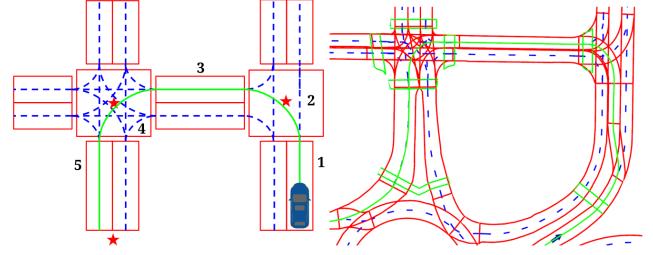


Figure 9: Left - Toy example of path planning utilizing the semantic map. Here the the lane lines are used to determine the optimal path to a target location. Right - Path planning with semantic map from MCity.

was that lane detection is prone to visual distractors and changing lighting conditions.

Year 2 Design and Lessons Learned A hierarchical design that contains 1) a Global (Route) Planner that finds a route with connected semantic road segments, 2) a Local (Path) Planner that generates a path based on the road geometry as defined in the semantic map, and 3) a Velocity (Trajectory) Planner that calculates the desired speed at along the path, was proposed to handle multi-destination navigation. Figure 9 illustrates this process. The lessons learned here were: 1) stitching together road centerlines and performing local smoothing worked well in constrained environments but may not work reliably in more complex scenarios; 2) Zeus collided with a deer dummy in the Year 2 competition as the system did not take the possible motion of surrounding objects into account; this emphasized the necessity of a framework to predict the future behaviours of traffic participants.

Year 3 and Year 4 Design and Lessons Learned In order to support more complex maneuvers, a lattice-graph based approach is developed to replace the Global and Local planner. The new planner generates a large set of feasible maneuvers (edges) offline based on road con-

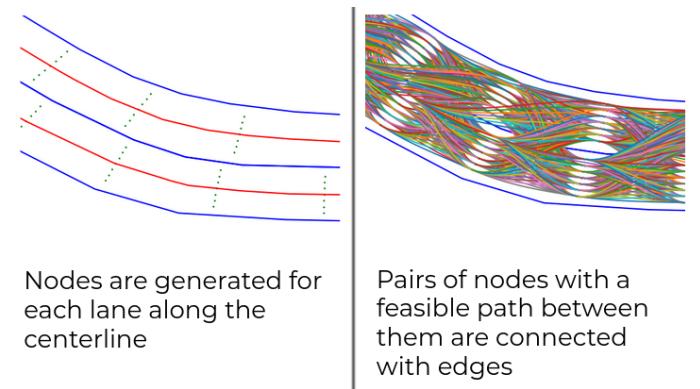


Figure 10: Steps in lattice graph generation and path finding. Lane boundaries are shown in blue, lane centerlines are shown in red, and lattice edges are multicolored.

nectivity and geometry stored in the semantic map (Zeus Map). During online operation, A* is applied to find the most efficient path over the generated edges. Figure 10 shows this process. When obstacles are detected, edges that may coincide with on-road obstacles are pruned such that the generated path will be collision-free. Figure 11 shows this edge pruning process. Many sub-routines run parallel to guide direction of path search (e.g. Turn Only Sign handling) and set speed limits (e.g. traffic light and pedestrian handling). Through our development process, we learned that: 1) lattice-based planning can be very effective when the environment is small and well-known but is difficult to extend to larger areas due to storage and runtime complexity; 2) lattice-based planning can be fragile when the vehicle needs to perform complex maneuvers or interact with other agents since the vehicle’s motion is restricted to the offline generated edges; 3) the planner must take perception uncertainties as well as the current situation into account to make efficient but safe driving decisions.

Towards Level 4 The current system is heavily optimized with respect to the competition requirements. For example, we only consider objects in front of us; we assume that dynamic agents only consist of pedestrians and deer; and we do not have the capability to handle vehicle-vehicle interactions. With this in mind, there is a huge room for improvement before we could deploy the vehicle in the real world. Some examples are 1) develop more sophisticated motion planning algorithms that can take temporal information into account and provide better state-space coverage during path search; 2) develop better motion prediction algorithms to handle complex, interactive traffic scenarios; 3) develop a comprehensive watchdog system to monitor ADS health and detect potential high-risk situations; and 4) define and implement a set of fallback strategies to bring the car to a minimum risk state when the ADS is compromised.

2.2.5 Control A significant portion of the dynamic challenge points are allocated to staying within lane boundaries, stopping accurately, driving at desired speeds, and obeying kinematic constraints. For this reason, having a high performance controller is critical to securing high scores in the competition.

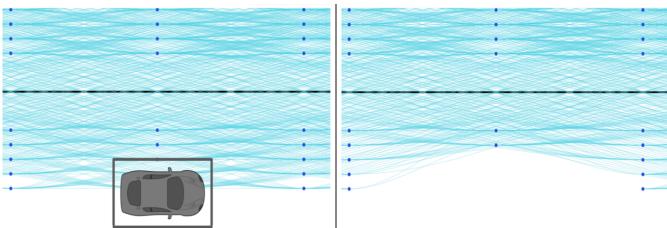


Figure 11: Before (left) and after (right) of lattice pruning. Edges inside the grey obstacle bounding box (obstacle dimension inflated with a safety buffer) are pruned.

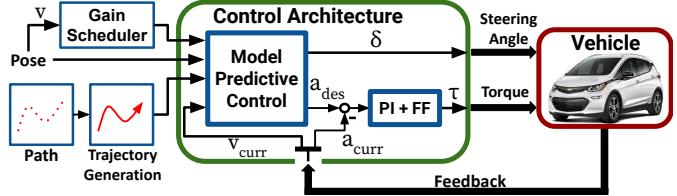


Figure 12: Year 4 controller architecture.

Year 1 Design and Lessons Learned A decoupled control system was developed for Year 1. The longitudinal cascade PID controller computed the motor torque whereas a lateral feedback-linearized (FBL) controller calculated the steering angle. Although the design was simple and effective, it was impossible to impose hard motion constraints into the PID controllers, resulting in aggressive motions and a loss of points in the Year 1 competition. Furthermore, the PI-based longitudinal controller was not able to reject disturbances due to gravity, causing inaccurate speed tracking on slopes.

Year 2 Design and Lessons Learned In order to achieve optimal tracking performance and systematically manage all motion constraints, the Year 1 system was upgraded to a Nonlinear Model Predictive Control (NMPC) design. NMPC achieved superior path and speed tracking performance; however, the team learned that it was time consuming to find one suitable parameter set for all the required driving scenarios. For instance, the controller should be more aggressive in correcting lateral error under lower speeds but be more gentle when driving fast. Using a simulator greatly accelerated the parameter tuning process.

Year 3 and Year 4 Design and Lessons Learned The year 2 NMPC design is updated with the addition of a gain scheduler that automatically selects one set of pre-tuned control parameters based on the vehicle’s current state. We experimented with low pass filters to smooth out control signs but they did not provide a noticeable improvement in comfort at the cost of introducing additional latency. After careful tuning, the controller now achieves good tracking performance from low speeds to 25Km/h on dry ground. Figure 12 shows the new controller design.

Toward Level 4 The current NMPC uses a kinematic vehicle model to predict the vehicle motion. It does not take tire friction into consideration and thus cannot account for abnormal road conditions. During testing, we have observed that Zeus could experience larger-than-normal turning radius, longer stopping distance, slippery and even become unstable during rain and snow. To achieve safe all-weather operation, one potential improvement is to develop a dynamic vehicle model that can automatically adapt to changing road conditions.

2.3 PREPARING FOR THE YEAR 4 CHALLENGES
Over the past year, we have made many exciting improvements to Zeus' ADS. Despite the difficult circumstances surrounding the COVID-19 pandemic, the team has formed a small but dedicated task force to perform frequent testing on our private test track at UofT's Institute for Aerospace Studies (UTIAS). These real world experiments cover a wide range of Year 4 competition scenarios, including intersection (traffic light and pedestrian) handling, traffic sign handling, blocked roads, GNSS-free navigation, pulling over maneuvers, parking and deer crossing, and help us identify many shortcomings that could have gone undiscovered. Due to pandemic-related restrictions, our test track testing time was significantly reduced. To compensate for the loss in testing time, we introduced the Zeus Auto-Evaluator Pipeline. The pipeline leverages the MathWorks Virtual MCity scene and Jenkins' automated testing capability to provide standardized, fully automated, close-loop system performance assessment. With this pipeline, we unveiled several edge cases that only occur in complex traffic scenarios. To tackle the Level 4 Challenge, the proposed GNSS-denied localization strategies, as discussed in Section 4, were first evaluated in ideal conditions in the simulation and then evaluated on real world driving data collected on the public roads around UTIAS. Having utilized both real-world and simulation testing, we are confident that Zeus is well prepared to tackle the Year 4 challenges.

3 DESIGN FAILURE MODE AND EFFECT ANALYSIS AND FALBACK STRATEGY

One of the main objectives of Year 4 is to understand the limitations of the current system and design a hypothetical vehicle that aligns with the SAE J3016 Level 4 standard. We adopt the following procedure to identify the gaps between the Year 4 ADS and the SAE Level 4 standard over the desired Operational Design Domain (ODD) and prioritize potential failures via a Design Failure Mode and Effect Analysis (DFMEA):

1. **Review Current ODD:** The current ODD defines the situations the Year 4 ADS is designed to handle.

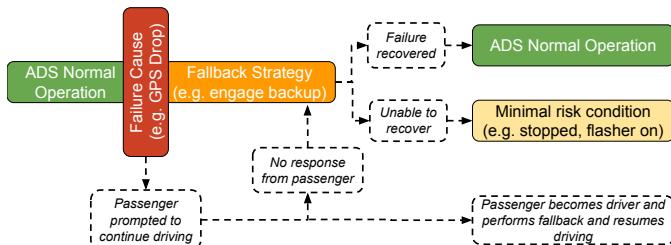


Figure 13: Use case sequence of the hypothetical L4 ADS. When a failure occurs, we must engage an appropriate fallback strategy and either automatically recover from the failure or bring the car into a minimum risk condition.

Thus, it contains many assumptions that we have built into the system. For example, the Year 4 ADS has no specialized handling mechanism for other moving vehicles as this scenario is not required by the competition.

2. **Define Target L4 ODD:** A key component of the SAE Level 4 automation specification is the limited set of environmental conditions that it operates within; as such, a well-defined target ODD is critical to determining the requisite capabilities needed for our hypothetical SAE Level 4 ADS. Section 8 outlines our target SAE Level 4 conditions.
3. **Design Adversarial Scenarios:** We then design various adversarial traffic and failure scenarios based on the capability of the Year 4 ADS and the target ODD. Using these scenarios, we can examine how the Year 4 ADS reacts to failure situations and what potential failure modes could occur.
4. **Analyze Failure Modes:** For each identified failure mode, we follow the SAE J1739 standard to determine its priority rating. We first assign a Severity rating from 1-10 based on the worst case outcome. Next, we identify the failure causes (i.e. how a scenario breaks our assumptions) for each failure mode and estimate their occurrence rate, error detectability and current prevention methods (if any). Finally, a Risk Priority Number (RPN) is calculated based on the worst case estimates. Note that it could be difficult to quantitatively determine the exact Occurrence and Detectability ratings for software systems. In this case we qualitatively assign a rating based on past experience and empirical reasoning.

With each failure mode and its associated failure causes, we conduct a literature review and design new systems and Dynamic Driving Task (DDT) fallback/failure mitigation strategies to achieve a minimum risk condition. The new system is then analyzed with respect to the flow diagram shown in Figure 13 to ensure it aligns with the SAE J3016 Level 4 standards.

4 FAILURE MODE: LOSS OF GNSS

As discussed in Section 2, many components in the Year 4 ADS require global positioning to achieve full functionality. Since Year 1, we have been using the GNSS unit installed on the roof rack to provide accurate and low-latency positioning information. However, there are several failure causes that can lead to catastrophic GNSS failure. It is well known that the GNSS signal can suffer from blockage and multi-path reflection around trees and buildings. During testing in wooded areas, we have observed frequent precision drops and large localization jumps, which, in most cases, required the safety driver to intervene and stop the vehicle. Additionally, both the GNSS unit itself and the power supply and cables connecting it to the ADS' computing server may experience mechanical and electrical failures. According to SAE J3016, a SAE Level 4 ADS

must actively monitor the performance of all ADS functionalities and perform fallback or failure mitigation strategies when a failure occurs. With this in mind, we have developed three backup localization systems that do not depend on GNSS: LiDAR Localization using a LiDAR map, LiDAR SLAM (Simultaneous Localization and Mapping) and Semantic Localization with a semantic map. We expect these backup systems to provide sufficiently accurate global positioning to sustain ADS functionalities until GNSS has recovered or the vehicle has reached a minimum risk condition.

This section is organized as follows: 1) we discuss how to detect GNSS drops; 2) we review the LiDAR Localization deployed in Year 2 and a major issue we encountered at MCity; 3) we review the semantic localization developed in Year 3 and discuss its advantages and shortcomings; 4) we review a LiDAR SLAM method developed in Year 4; 5) we demonstrate the effectiveness of these localization methods on a data set collected on Toronto public roads; and finally 6) we discuss how these methods enable the ADS to continue its current DDT or perform safe fallback strategies when GNSS is not available.

4.1 GNSS FAILURE DETECTION

The Year 4 ADS can detect GNSS failure in two ways. First, our GNSS driver provides diagnostic information about GNSS unit's operation mode and accuracy. For safety, we require the GNSS unit to operate under the Precise Point Positioning (PPP) mode, which delivers a localization accuracy of up to 20 centimetres [10]. When this high accuracy service is lost (e.g. due to blockage), the ADS will be notified. Second, the ADS monitors a heartbeat signal from the GNSS driver. This allows the ADS to automatically detect any driver or hardware failures. We plan to investigate more sophisticated failure detection methods in the future. For example, a residual chi-square test [11] is a popular statistical method used to detect GNSS localization jumps. Once the ADS detects a GNSS failure, the localization module will leverage the recorded GNSS localization history to initiate one of the backup methods and support ADS fallback or recovery strategies.

4.2 LIDAR LOCALIZATION

LiDAR-based localization has been widely used in the robotics community in GPS-denied environments. LiDAR localization can be considered as a pointcloud registration problem where the optimal rigid transformation between the latest LiDAR pointcloud and the reference pointcloud map is estimated at each timestep. Thanks to our Velodyne LiDAR's large FOV, we have shown that LiDAR Localization is robust to smaller scene changes, such as the movement of other vehicles, as long as there are persistent structures, such as buildings, in the scene. Since LiDAR Localization provides global position estimates in the LiDAR map, it has the potential to enable seamless operation when the GNSS signal is lost.

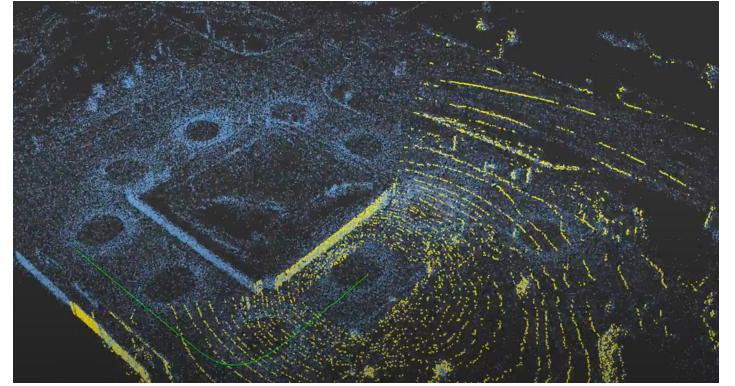


Figure 14: Visualization of LiDAR localization at MCity. Yellow points correspond to the current LiDAR scan and blue points correspond to the reference LiDAR map.

4.2.1 Year 2 Review

In Year 2, we integrated Applanix's newly developed LiDAR mapping and localization solution into our system. We successfully performed LiDAR-only localization with sub-decimeter accuracy at UTIAS and MCity with respect to our custom-built LiDAR maps and maps provided by Carmera. Figure 14 shows a snapshot of LiDAR Localization in action during the 1-hr practice at MCity.

4.2.2 MCity Failure

We initially planned to use LiDAR localization at the Year 2 competition. Unfortunately the vehicle experienced localization jumps during the 1-hr practice in some regions of MCity. One possibility is that our process of converting the Carmera map into the required Applanix format resulted in deformations which in turn led to large point cloud registration errors.

4.3 SEMANTIC LOCALIZATION

LiDAR Localization is a dense method and as a result its performance heavily relies on the accuracy of the reference map. Long-term environment changes, such as trees growing along roads, may cause data association errors, resulting in deteriorating performance over time. This motivated us to develop a sparse, semantics-based localization method to achieve more robust localization.

Semantic Localization uses an Iterative Extended Kalman Filter (IEKF) to fuse wheel odometry measurements with lane markings and traffic light detections. Global positioning can be achieved by comparing observed semantic cues, such as lane markings and traffic light detections, against their expected locations stored in a reference semantic map. The performance of Semantic Localization relies on a few critical components, including a semantic map that accurately reflects the locations of traffic lights and lane markings, and high performance lane and traffic light detectors. This subsection provides an overview of the system but readers may refer to our recent CRV paper [12] for detailed technical explanations.

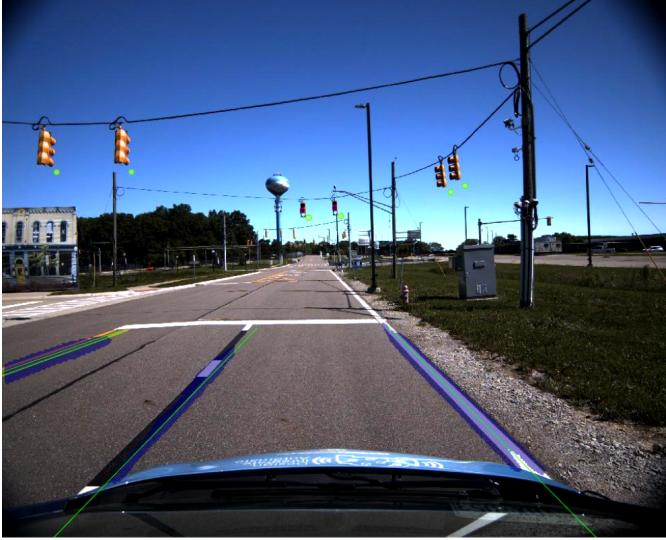


Figure 15: Lane marking detections (blue mask) overlaid with the reference semantic map projection (green lines).

4.3.1 Semantic Map As discussed in Section 2, semantic maps are managed through the Zeus Map pipeline to provide high level, abstract information of the environment. Semantic Localization thus leverages these readily available semantic cues (i.e. traffic lights and lane lines) to localize the vehicle. In the Zeus Map, each traffic light is represented by a point coordinate, while lane lines are represented as line segments. To achieve online run-time performance, the cues are stored as a KD tree such that all traffic lights and lane lines within a distance threshold of any position can be queried efficiently.

4.3.2 Lane Marking Detection Lane markings provide the lateral correction required to keep the vehicle inside a lane, while providing little information regarding the longitudinal position of the vehicle. We treat lane marking detection as a semantic segmentation problem where the objective is to predict whether each individual pixel in the image belongs to a lane line. Given a camera image, the lane marking detector utilizes the Gated Shape CNN (GSCNN) architecture [13] and returns a pixel-wise prediction mask (observations). Next, we project the expected lane lines from the reference semantic map to the image frame based on known camera extrinsics. The observed lane marking pixels are greedily associated to the reference lane lines via Iterated Closest Point (ICP) in the image plane. The distances between the associated pixels (the projection error) serves as the error signal which will be minimized by the Semantic Localization model to reduce lateral positioning error. To speed up the computation, the lane line masks are first down-sampled. Figure 15 illustrates the result of this procedure.

4.3.3 Traffic Light Detection Traffic lights help with longitudinal localization near intersections. Traffic light detec-

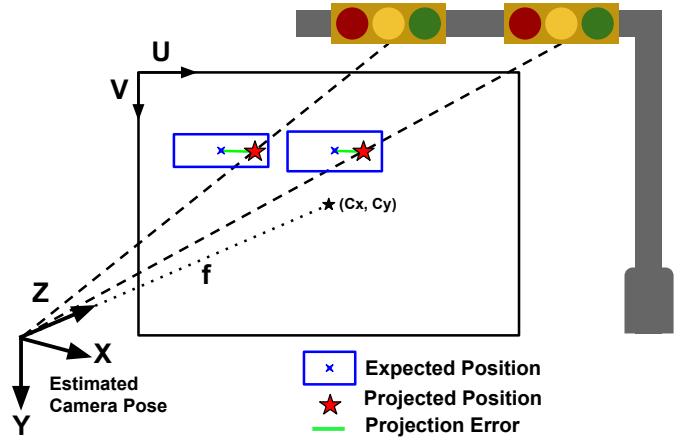


Figure 16: The traffic light projection error calculated as the displacement between the expected light projection and the actual measurement.

tion is already a part of the ADS, so Semantic Localization can take advantage of the detection results without incurring an additional computation cost. The raw output of the traffic light detector is a bounding box for each of the detected traffic lights. The center point of each bounding box is extracted as a point landmark and compared against their expected locations stored in the reference semantic map. Given a set of point detections and projected traffic light positions, point-to-point correspondences between them can be computed greedily using ICP. The pixel coordinate difference of each pair becomes the error signal which will be minimized by the Semantic Localization model to reduce longitudinal positioning error. Figure 16 shows how traffic lights are projected and compared against their expected locations.

4.3.4 Year 3 Review The complete pipeline can be described by Figure 17. As discussed in the Year 3 Report [3], we evaluated Semantic Localization in the MCity dataset collected during the Year 2 competition. Semantic Localization is particularly advantageous for a few notable reasons. First, it leverages existing ADS perception algorithms (traffic light and lane detection) without introducing additional computational costs. Second, it does not require additional maps as the semantic map is al-

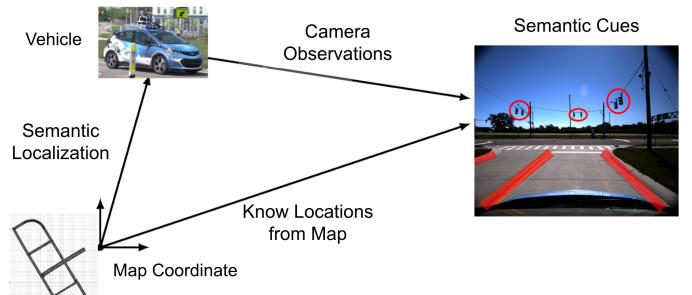


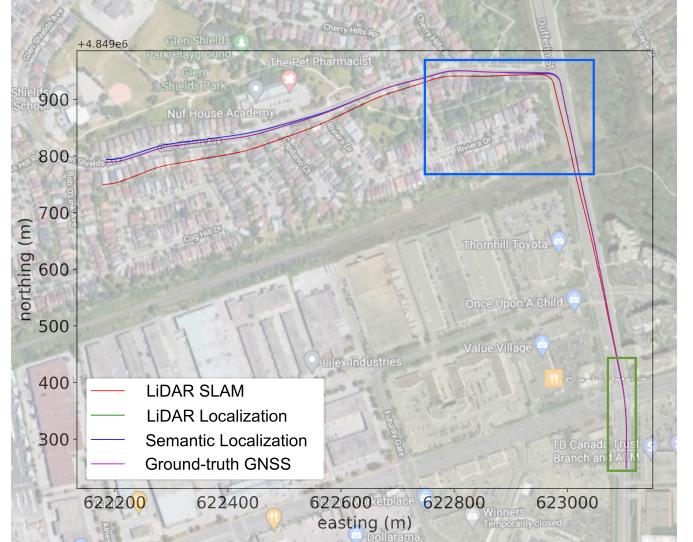
Figure 17: Graphical illustration of Semantic Localization

ready utilized by many other ADS components. Third, its core algorithm is a simple IEKF and can easily achieve real-time performance with little computational burden. Finally, it only relies on sparse semantic cues, which means that it is robust to minor environmental changes. However, Semantic Localization suffers from several shortcomings. We found that semantic maps usually do not reflect the real world environment: even maps provided by professional cartography companies exhibit incorrect/outdated information and local deformations (see Section 6 for examples). Such discrepancies between the observations and the reference map can result in significant localization errors. Moreover, the accuracy of Semantic Localization is much lower than LiDAR Localization. As discussed in [3], Semantic Localization is only accurate to 50cm with a carefully verified reference map.

4.4 LIDAR SLAM Both LiDAR Localization and Semantic Localization require accurate reference maps to perform correctly. However, due to competition restrictions we may not have sufficient time to build and validate a reference map before the scored runs. This motivated us to develop a LiDAR-based Simultaneous Localization and Mapping (SLAM). Unlike the first two methods where current observations (camera, LiDAR and wheel odometry) are compared against a reference map, LiDAR SLAM concurrently builds a map of the environment and estimates the vehicle’s position. Our implementation is based on the Applanix mapping and localization code base, which leverages highly optimized pose-graph optimization and LiDAR pointcloud registration techniques to achieve real-time performance.

4.4.1 Global Accuracy Through extensive rosbag and real world experiments, we noticed that LiDAR SLAM can produce highly accurate odometry information (i.e. local pose change), but global localization accuracy may suffer due to the lack of global corrections. In order to improve our global localization accuracy, we introduce two distinct operation modes. Under the online mapping mode, the LiDAR SLAM system will accept GNSS information whenever available and build a reference map online that is consistent with the GNSS localization information. When the GNSS signal is lost, LiDAR SLAM changes to SLAM mode where it will use pointcloud registration (w.r.t. the reference map and previous LiDAR scan) to localize the vehicle and make updates to the reference map upon receiving new measurements (SLAM mode). This strategy ensures highly accurate global positioning in previously-visited areas and reduces drift in to unknown regions.

4.5 REAL WORLD EXPERIMENTS In Year 3, we evaluated LiDAR Localization and Semantic Localization on a MCity dataset collected during the Year 2 competition. This year we take it one step further and evaluate all three methods discussed above on real world datasets collected on Toronto public roads.



(a) Ground-truth and GNSS-free localization trajectories overlaid on a satellite image of the traversed area



(b) Manually labelled lane markings overlaid on a satellite image. The shown area corresponds to the blue box in (a)

Figure 18: Top down view of the test route and associated semantic map

4.5.1 Data Collection and Map Generation We collected real world driving data with our vehicle Zeus on public roads around UTIAS on a day with clear weather and moderate traffic. The average driving speed during data collection was around 50Km/h. As seen in Figure 18(a), the route consists of both highway style roads and neighborhood areas to mimic the typical operation condition of an autonomous ride-sharing vehicle. The route was repeated twice, once for mapping and once for testing. For LiDAR Localization, we generated the reference LiDAR map by piping the route data into the Applanix mapping software. For Semantic Localization, we simply grabbed a satellite image of the route from Google Maps and manually labelled all the identifiable lane markings and traffic lights. One portion of this manually labelled semantic map is shown in Figure 18(b).

4.5.2 Performance Analysis We evaluate the three methods discussed above on the test data set. Unfortunately, a direct quantitative comparison between the methods is difficult due the lack of a common ground-truth. Here we simply adopt the post-processed GNSS

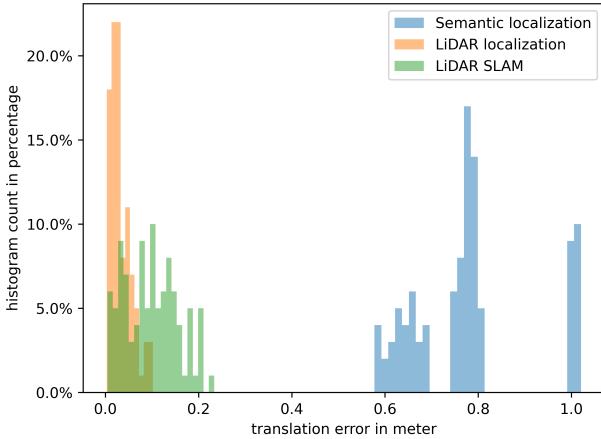


Figure 19: Error histograms of the first 100m of the estimated trajectories

information as the ground-truth. Post-processing is done using the Applanix mapping software where batch optimization using both positioning and IMU data is performed over the entire trajectory. Since this batch optimization incorporates both future and past information, it cannot be used online. However, it provides highly accurate position estimates which can be treated as ground-truth.

We initialize the algorithms by simulating a normal operation (all sensors including GNSS enabled) for 50m. GNSS data is then cut off. Figure 18(a) depicts the estimated trajectories of the three methods after the GNSS cutoff point. Qualitatively, LiDAR Localization best matches the ground-truth but Semantic Localization is also able to stay within lane boundaries. On the other hand, LiDAR SLAM diverges quickly after the first 200m as it lacks global correction.

Ideally, ADS fallback strategies are performed right after the failure point. Thus, we pay special attention to the results in the first 100m of the estimated trajectories. As shown in Figure 19, LiDAR Localization still performs the best and its translation error is always under 10cm. LiDAR SLAM also provides an accurate estimate with a maximum error of approximately 20cm. Semantic Localization, however, suffers in this scenario because the failure point is too far away from the north intersection and thus it cannot utilize traffic lights to correct its longitudinal position estimate.

The results align with our expectation: LiDAR Localization is the best approach if an accurate LiDAR map is available; Semantic Localization provides reliable, lane-level localization at a reduced accuracy; and finally, LiDAR SLAM delivers accurate but short-term positioning without any prior maps.

4.6 LEVEL 4 ADS FALBACK STRATEGIES With the properties of the backup localization systems in mind, we have designed the following SAE Level 4 fallback strategy: 1) during normal operation, the backup localization systems run in the background and re-calibrate themselves with the latest GNSS data; 2) when a loss-of-GNSS failure is detected, the ADS immediately slows down, turns on the flasher, prompts the passengers, and switches its localization source to one of the backup systems; 3) if a recent LiDAR map of the area is available, the ADS will activate LiDAR Localization; 4) if lane markings are detected by the perception system, Semantic Localization will engage (and potentially run as a sanity checker for LiDAR Localization); 5) if GNSS has recovered, the ADS will recover from fallback and resume normal operation; 6) if LiDAR and/or Semantic Localization reports low confidence or inconsistent estimates, LiDAR SLAM will engage and the ADS will pull the vehicle to a road shoulder (if available) or stop right away to achieve a minimum risk condition. 7) once the vehicle is in the minimum risk condition, the ADS will contact remote assistance.

The system discussed above enables the ADS to automatically detect the loss-of-GNSS failure mode, invoke fallback strategies and achieve a minimum risk condition. Thus, we conclude it aligns with the SAE J3016 Level 4 requirements.

5 FAILURE MODE: COMPROMISED PERCEPTION SENSORS

Perception sensor functionality is important for detecting traffic signage, objects, pedestrians, and lane lines on the road. This is essential for understanding the environment the car is driving within and is also necessary for the vehicle's decision making. The DFMEA reveals that the following failures can occur in Zeus' current perception system: spurious detections, slow detection speeds, and compromised perception sensor functionality amongst other failure modes.

Compromised perception sensor functionality during autonomous DDT is found to have the highest Severity, Occurrence, and Detectability score of 10 with a RPN of 1000. A detailed description of this failure mode, Zeus's current perception system, and a plan to achieve a hypothesized SAE Level 4 ADS will therefore be presented in this section. After accounting for the suggested SAE Level 4 fallback strategies and failure mitigation protocols, the presented failure mode has an improved RPN score of 140.

5.1 ASSUMPTIONS OF SENSOR PERFORMANCE As shown in Figure 8, we utilize DNNs to detect traffic lights and cascade detectors to detect traffic signs on camera images. This part of the system assumes that the cameras are not obstructed by foreign artifacts and that the traffic lights and signs are clearly captured by the cameras.

As shown in Figure 7, the aUToTrack pipeline then fuses the 2D visual detections with clustered LiDAR point cloud to determine the position of the objects in 3D space. The assumption made in this section of the system is that the cameras and the LiDAR are not obstructed and the camera-LiDAR extrinsics is accurate.

5.2 FAILURE MODE DESCRIPTION The Year 4 ADS perception system is heavily reliant on the outputs of the Blackfly cameras and the Velodyne LiDAR. The current ADS is built on the assumption that these sensors will not be compromised. We define compromised perception sensors as the sensors failing to meaningfully and accurately capture the driving environment due to external factors.

Compromised sensors incapacitate the ADS Object and Event Detection and Response (OEDR) capabilities by providing frozen, corrupted, or misaligned data to the ADS’s perception system. As a result, Zeus may fail to detect traffic signs, traffic signals and pedestrians without warning. This may give rise to unsafe driving and also make the vehicle non-compliant with driving laws. Failing to detect a pedestrian or another vehicle can lead to fatal collisions. Such catastrophic outcomes motivate the high Severity score of 10 assigned to this failure mode.

5.3 FAILURE MODE CAUSES The DFMEA reveals a range of causes behind compromised perception sensor functionality. These include the obstruction of the sensors, the operational limits of the sensors being exceeded, and corrupt sensor extrinsics. We present these causes and discuss the extent to which the current and hypothetical Level 4 ADS detect and prevent them.

5.3.1 Obstructed Cameras The cameras can be obstructed by foreign artifacts due to sub-optimal driving conditions and inclement weather.

Example Scenario: Water And Snow Covered Sensors
On rainy days we observed that water accumulates on the sensor lenses. Additionally, in wintertime, sheets of snow can completely block off the cameras from perceiving the driving environment.

Current Occurrence In Toronto, during an entire year, the rain falls for 136.8 days and collects up to 420 mm of precipitation while snowfalls for 45.6 days and aggregates up to 223 mm of snow [14]. This corresponds to roughly 50% of the days in a year experiencing unfavourable weather conditions. As a result of such harsh weather conditions, the team was often unable to perform testing because precipitation, dust, and residue can quickly accumulate on the camera lens. Although studies such as [15] have demonstrated that there is approximately an 80% drop in the detected distance using LiDAR sensors in rain and snow. Therefore, the system would suffer from this

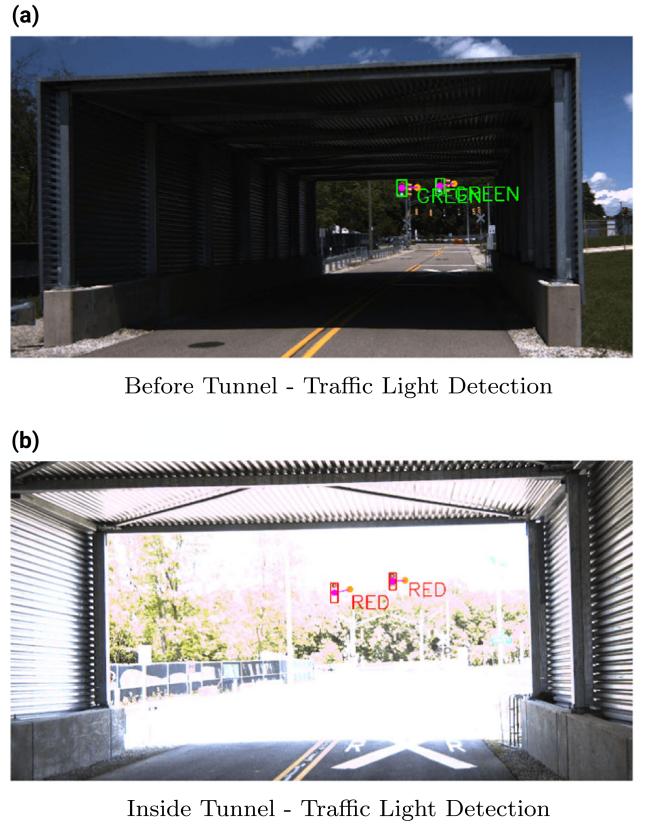


Figure 20: At the Y2 intersection challenge Zeus was stalled at an intersection. (a) Before entering the tunnel, the traffic light was correctly classified as green. (b) The area outside the tunnel became overexposed and caused the traffic light to be incorrectly classified as red.

same problem if it was dispatched on rainy and snowy days. This failure cause was therefore given an Occurrence score of 10 as it exceeds the 100 per thousand items standard, which would correspond to 10% of the days in a year having poor weather or a 10% reduction in the detected LiDAR distance.

Current Prevention and Detectability For the current Year 4 ADS, before engaging the ADS, the dispatcher examines the lens of the cameras before engaging in autonomous mode. Additionally, a passenger in the vehicle monitors what the cameras are perceiving using an in-vehicle display. The passenger intervenes when abnormalities are detected.

Although relying on the passenger or safety driver is sufficient for SAE Level 3 specifications, a defining characteristic of a SAE Level 4 ADS is the ability to identify failure causes and respond to them automatically. This failure cause is given a Detectability score of 10 for the Year 4 ADS as physical obstructions are currently not detected by Zeus. In our hypothetical Level 4 ADS, we plan to improve the watchdog system to detect obstructed sensors.

5.3.2 Camera Limitations The cameras may not capture meaningful information when lighting conditions exceeds their CMOS limits, resulting in underexposure or overexposure. In this case, the ADS perception system will not be able to correctly detect and classify objects, traffic lights, and signs.

Example Scenario: Y2 MCity Challenge Failure An example of this failure cause is depicted in Figure 20. When Zeus approached the end of the tunnel during the Y2 MCity Challenge, the bright sunlight outside caused high contrast. Such lighting conditions exceeded the dynamic range of the cameras and resulted in an incorrect traffic light detection.

Current Occurrence This failure cause transpires when there is a change from underexposed to overexposed lighting, such as when passing under a tunnel or bridge as shown in Figure 20. It also occurs when driving facing the sun or at night. This was given an Occurrence score of 9 for the very high number of times we estimate that vehicles experience these driving conditions. However, this was not awarded a score of 10 since our perception system was trained on data that accounted for diverse lighting conditions.

Current Prevention And Detectability We selected the Blackfly S Monocular camera for its high dynamic range of 71.95 dB [16]. Histogram equalization is conducted on these camera outputs to improve their contrast; however, overexposed images may still result in reduced performance of the perception system. Furthermore, although radar and LiDAR detections act as Zeus’s backup detection systems, they lack the semantic information to classify objects in the driving scene. Figure 21 depicts a night driving scenario where the ADS is still able to detect an object in front of the vehicle but cannot categorize the detection as being a pedestrian. The lack of such semantic information makes it difficult for the planner to interact with the driving environment.

The dispatcher also checks the cameras’ auto-exposure

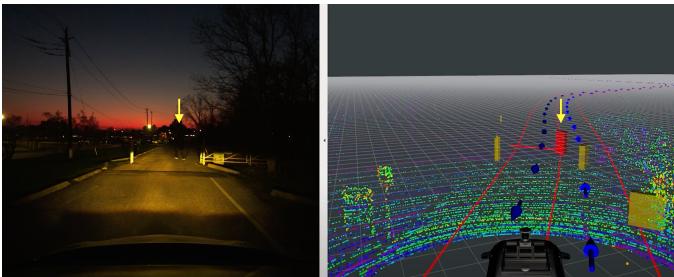


Figure 21: The ADS successfully detects an object under low lighting conditions but cannot recognize the detection as being a pedestrian. In this case the passenger or safety driver must monitor the situation to prevent potential safety hazards, effectively reducing the ADS to SAE Level 3.



Figure 22: Camera and LiDAR sensor calibration using a checkerboard and MATLAB functions.

function before autonomous operations. The passenger can also monitor the camera outputs and associated detections in order to act as a final layer of failure detection. Despite these prevention strategies, the current ADS has no method of conducting autonomous detections and so was given a Detectability score of 10. To improve to a hypothetical Level 4 ADS, these detections should be conducted by a watchdog system within the vehicle.

5.3.3 Corrupted Camera-LIDAR Calibration Zeus’s sensors are mounted on a metal rack on top of the vehicle. Vibrations and impacts to the sensor mounts can corrupt the relative positions between the sensors.

Example Scenario: Failed Stop Sign Localization As discussed in Section 2, the traffic sign detection system projects the LiDAR point cloud onto the image plane to localized the traffic signs in 3D. One failure scenario is shown in Figure 24(a) where the camera-LiDAR extrinsic calibration is corrupted. In this case, the point could from the far away barriers are projected onto the detected Stop sign, causing the vehicle to stop at an incorrect location.

Current Occurrence In the past few years, we have observed errors arising from misaligned extrinsic sensor calibration during testing. This is because the mounts and screws of the sensor rack get loose frequently and so has an Occurrence score of 10.

Current Prevention And Detectability In order to prevent this failure from occurring, sensor extrinsics is periodically calibrated using a MATLAB calibration toolbox. However the Year 4 ADS does not have the capability to detect corrupted calibrations in run-time. The Detectability score of this failure cause is a 10 as no system exists to monitor the extrinsics during a drive.

5.4 HYPOTHETICAL LEVEL 4 ADS This section describes the planned improvements that address the aforementioned failure causes and also describes how these improvements achieve a SAE Level 4 ADS.

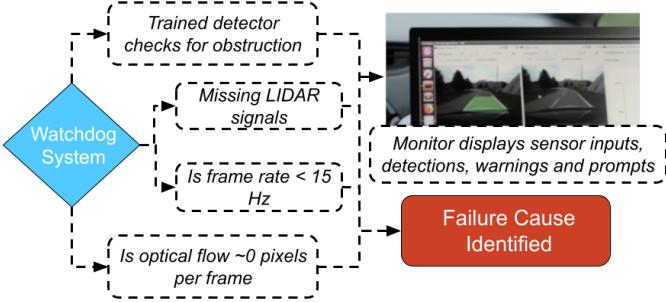


Figure 23: The proposed watchdog system.

5.4.1 Watchdog System For Compromised Sensors

A SAE Level 4 ADS monitors its hardware and software systems to identify and respond to failures. Our current system has a watchdog system that monitors the software and the plan is to extend this to monitoring hardware failures. The planned watchdog runs diagnostic tests to monitor software and hardware components of the perception system to identify key failure causes and the resulting failure modes. The architecture of the planned watchdog is described in Figure 23.

In particular, the watchdog system is responsible for identifying obstructed sensors and inconsistent detections in the world frame of the car. Firstly, the watchdog monitors the frame rate of the cameras to ensure to determine if a camera is offline. Secondly, it checks the optical flow of the video stream. If one patch on the images always has 0 flow then part of the lens might have been covered. Finally, a train DNN detector is used to detect potential fog and rain drops on the camera. For the LiDAR, the watchdog will check for the return signal. If there is one region that consistently does not get a return signal or the same depth estimate is received repeatedly, then this points to the possibility that some laser beams are blocked.

This watchdog reduces the Detectability score of the failure causes described in Sections 5.3.1 and 5.3.2 to a 3 which corresponds to a moderately high chance of detecting the failure cause and resulting failure mode. At the same time however, the Occurrence score is unaffected and remains at a 10 since this improvement does not reduce the occurrence of the failure cause.

If a flag is raised, a possible obstruction in the field-of-view of the sensors or a sensor failure has occurred. The watchdog notifies the ADS to initiate the Dynamic Driving Task (DDT) fallback protocol.

DDT Fallback The DDT fallback shown in Figure 13 begins by slowing down the vehicle and turn on the flasher in order to signal to other drivers in the ODD that a failure has arisen. The passengers are also informed of the issue through the monitor and LED lights inside the vehicle.

At the same time, the ADS attempts to recover the vehicle's functionality by restarting the sensor drivers. Re-

dundant systems such as the LiDAR and RADAR based detectors are given a priority within the planner (as shown in Figure 5) to avoid collisions. If the issue is recoverable, then the ADS resumes normal operation.

The ADS is brought to a minimal risk condition, as shown in Figure 13, when recovery of autonomous functionality is no longer possible. The ADS will pull the vehicle to a road shoulder(if available) or stop right away to achieve a minimum risk condition. A remote dispatcher or vehicle roadside assistance will also be contacted. Finally, the passenger can request manual takeover and become the driver to resume driving. This reduces the Severity score of the failure mode to a 8 since the vehicle can automatically handle the failure mode, however, may become inoperable. Therefore, the improved RPN for this failure cause is a 240.

5.4.2 Online Sensor Extrinsic Calibration Currently, our calibration is conducted offline using static targets. However, online extrinsic camera calibration may also be applied to minimize reprojection errors of the features detected by the cameras and the LiDAR. This is done by estimating the relative pose between the two sensors in real time. The resulting system will no longer rely on a static preset camera frame. A possible method by which online sensor extrinsic calibration can be conducted is presented in [17] which uses lane detection and extended Kalman filtering to update the extrinsic camera parameters. Since our system has lane detection, we can readily apply this method. This improves the occurrence score of the failure cause from 10 to 3 as online calibration would lead to fewer corrupted calibrations. The Detectability score also decreases from a 10 to a 4 as online calibration is an automated system that constantly monitors the system to detect when calibration is required. An example of online extrinsic calibration being used to achieve better localization can be seen in Figure 24.

DDT Fallback When the online extrinsic calibration system identifies that the sensor extrinsics needs to be re-calibrated, it enters the DDT fallback strategy presented in Figure 13. After slowing down, the ADS attempts to re-calibrate the sensors. At the same time, the passengers and dispatcher is informed of the issue. If calibration is successful, normal operation resumes. On the other hand, if the reprojection error remains large after calibration, The ADS will pull the vehicle to a road shoulder(if available) or stop right away to achieve a minimum risk condition. A remote dispatcher or vehicle roadside assistance will also be contacted. Finally, the passenger can request manual takeover and become the driver to resume driving. This reduces the Severity score of the failure mode to a 8 since the vehicle can automatically handle the failure mode, however, may become inoperable. Therefore, the improved RPN for this failure cause is a 320.

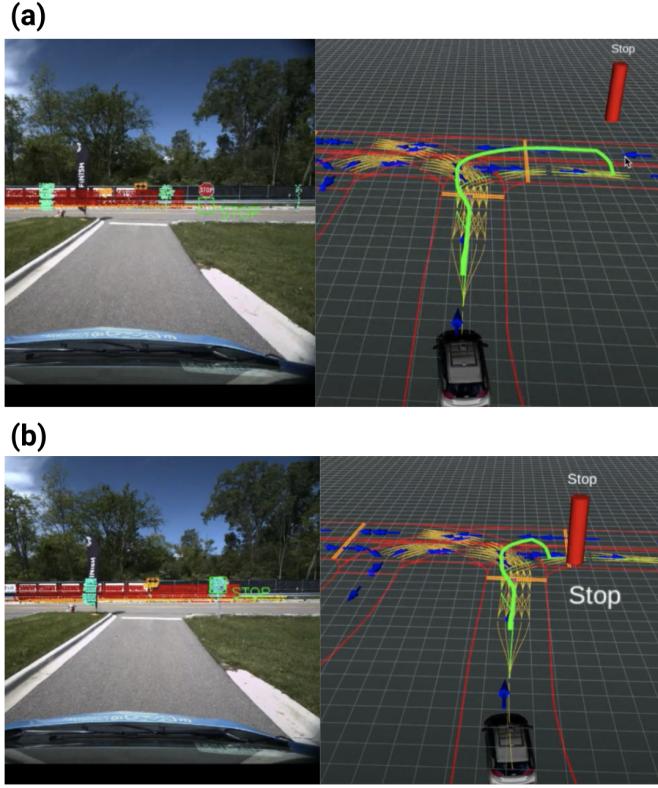


Figure 24: (a) Corrupted extrinsic calibration causes a stop sign to be localized incorrectly. Point clouds from the red barriers on the road as opposed to point clouds from the stop sign are clustered and then associated with the camera based detection. (b) Online extrinsics calibration corrects extrinsics and localizes the stop sign successfully.

5.5 IMPROVED RPN SCORE Compromised perception sensor functionality during autonomous driving was found to have the highest Severity, Occurrence and Detectability score through the DFMEA. If all the planned Level 4 ADS improvements and fallback strategies described in this section are applied, the RPN is improved from 1000 to 320.

The improvements enable the ADS to automatically detect compromised perception sensors, invoke fallback strategies and achieve a minimum risk condition. Thus, we conclude it aligns with the SAE J3016 Level 4 requirements.

6 FAILURE MODE: SEMANTIC MAP INACCURACY

The semantic map is used in traffic light detection, behavior planning, and vehicle path planning as described in Section 2.2.2. The DFMEA revealed the following two failure modes for this system: load failures or errors resulting from corruptions, deletions, and accessibility issues, and semantic map inaccuracies. Semantic map inaccuracies had the highest RPN score of 720, in part due to their high rate of occurrence and low detectability. The implementation of our planned fallback and failure mitigation strate-

gies for a hypothetical SAE Level 4 ADS is projected to reduce the RPN to 240.

The following sections describe the assumptions about the semantic map’s use within the ODD, the nature of the failure mode, as well as the proposed fallback and failure mitigation strategies needed to achieve a Level 4 ADS.

6.1 ASSUMPTIONS ON THE SEMANTIC MAP The semantic map is used in conjunction with perception to localize road elements such as traffic lights as seen in Figure 16. Recorded intersection geometry is also used to infer pedestrian intention/behaviour. Road signs and lane boundaries are utilized to make traffic decisions (i.e., when to stop, or consider alternate paths). Finally, the semantic map is also vital for path planning as detailed in Section 2.2.2. Figure 9 and Figure 10 illustrates this further.

In order to facilitate the use of the map, we need to make a number of assumptions. For the system to remain within the ODD one must first assume that the semantic map is accurate and that it sufficiently represents the current environment. Next, the geo-referencing technology utilized to develop the map is calibrated to work with the GNSS utilized by Zeus.

6.2 FAILURE MODE DESCRIPTION Semantic Map Inaccuracy failures are defined as ADS failures induced by inconsistency between perceived semantic cues and the reference semantic map. This failure can cause the ADS to drive on the wrong side of the road, fail to detect traffic lights, and enter or stop inside intersections. Without proper failure detection we may violate government road regulations without any warning. Hence, the Severity rating for this failure mode is 10.

6.3 FAILURE MODE CAUSES The DFMEA revealed there are two main causes for this failure mode. The following section will discuss how our design choices help detect and prevent these failure causes.

6.3.1 Global Shift in Semantic Map In the ideal case, the ADS would utilize a semantic map generated from the onboard GNSS system. However, it would be impractical for a single vehicle to create a map for a large city. Hence, in practice, the Semantic Map utilized by the ADS is provided by designated cartography companies. As such, the data format and geo-referencing systems used for data collection in the generation of the semantic map will vary. During online operation, we must convert the provided map using our Zeus Map pipeline so it can be utilized for perception or planning tasks. Due to this, global offset errors may be introduced during map conversion. This may be from numerical error, differences in mapping standards or due to differences between geo-referencing



Figure 25: Uncorrected MCity Map from third-party provider. Here we see that the center lane lines of the right hand side lanes appear within the shoulder of the road.

systems. The effect of such an offset was seen within the Year 2 MCity map shown in Figure 25.

Example Scenario: Year 2 Competition at MCity During the Year 2 competition, the semantic map provided by a third party company was developed utilizing different geo-referencing systems than the Novatel system on Zeus. During testing, Zeus was not driving in the center of the road and would often hit the curb when turning. To address this behaviour during the Year 2 Competition, we aligned the semantic map using the recorded GNSS measurements, while correcting the offsets manually. The exact offsets were determined to be 70 cm in Easting, 90 cm in Northing, and 120 cm in altitude [2].

Within Figure 25 we see a noticeable offset between the semantic map and overlayed satellite imagery.

Current Occurrence of Failure Cause Based on our qualitative experience, the occurrence of this failure cause is relatively high. GNSS reference frames and projection standards can vary heavily between the third party maps and the GNSS technology utilized the ADS. Often, the global offset must be corrected by applying various transformations to the semantic map or through GNSS calibration. Further, creation and updates to handmade semantic maps is generally expensive leading to slow update times. Thus, the ADS becomes more susceptible to this failure over longer periods of time. As such, the Occurrence score of this failure cause is 8.

Current Prevention and Detectability of Failure Cause Current detection of the failure cause is done through comparison of the semantic map to the latest satellite imaging data prior to deployment. Once an inconsistency between the semantic map and satellite imagery is detected, the semantic map is manually modified to better align with the imagery. This only accounts for global shifts due to errors in map generation rather than those due to reference frame conversion. In some cases, satellite imagery may not appear in a timely manner. Thus, to detect this failure mode, one would need to check the integrity of the map through a survey of the driving area with high precision geo-referencing technology, which can be onerous.

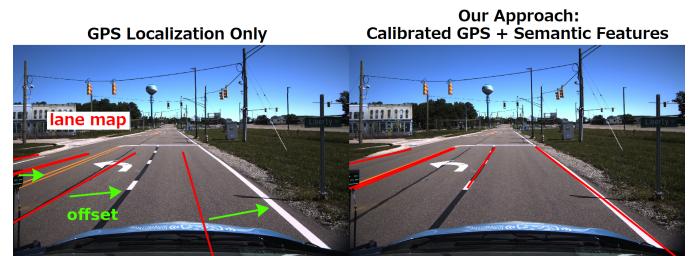


Figure 26: Vehicle localization using uncalibrated GPS (left) and our approach presented in [12] which utilizes Semantic feature calibrated GPS.

Prevention of this failure cause is not possible as the error present within the map is intrinsic to the map. Thus, it can only be detected then corrected but not directly prevented. Given this, the current Year 4 system does not have a robust method to detect such errors during operation. Hence, based on our previous experience, we assign a qualitative Detectability score of 9 to this failure cause.

6.3.2 Local Changes in Environment Frequent road maintenance and construction introduces local changes, which are not accounted for within the semantic map until it is updated much later. Key objects for path planning, such as traffic lights and lane markings, might appear in incorrect locations, leading to inappropriate vehicle behaviour. This failure can result in problems with the perception stack as the stack relies on association of perceived road elements to the reference map.

Example Scenario: Road Line Repainting During the Year 2 competition, the road lines at MCity had been recently repainted. Since the frequency in which the semantic map is updated is relatively slow, various lane boundary elements had additional offsets. Due to these shifts Zeus had erroneously entered a pedestrian crosswalk during a red light as seen in Figure 27. This is a highly severe failure mode as there was no warning for when Zeus attempts to enter the pedestrian crosswalk.



Figure 27: Zeus stopping past stop line due to offset error from repainting of road lines.

Occurrence of Failure Cause Road maintenance and construction frequently occurs within urban environments. For instance, within ward - 11 in the city of Toronto there

are currently 27 active construction notices [18]. Due to this, a number of road elements may change at a higher frequency than that which the semantic map is updated. Hence, this failure cause has an Occurrence rating of 10 due to the high frequency at which road construction can occur within urban environments.

6.3.3 Current Prevention and Detectability of Failure Cause

Unfortunately, we currently do not have an effective method to detect when elements within a given semantic map are outdated. The ability to detect these changes within the semantic map is extremely difficult. Furthermore, developing semantic maps is an extremely costly process. The typical rate for mapping a kilometre of road is \$5K [19]. Further, this process can take up a large amount of time. Within the timeframe of map creation, the landscape can change due to construction, maintenance, and/or other phenomenon (For example, stop light power outages, collisions, road closures, etc). One method used for semantic map error detection is to utilize satellite imagery. However, such satellite imagery tends to be outdated.

Total prevention of this failure cause is also not possible. Any error present within the semantic map is intrinsic to the map and can only be detected then corrected once a survey of the driving area is conducted. Furthermore, road construction is inevitable for cities and urban environments.

Thus, the ability to detect this failure mode is very poor, and so the failure mode receives a qualitative Detectability score of 9.

6.4 HYPOTHETICAL LEVEL 4 SYSTEM Given the previous analysis, we introduce our hypothetical Level 4 ADS, which applies a number of fallback strategies designed to alleviate the failure causes.

6.4.1 Global Offset Correction To alleviate global offset errors, the Level 4 ADS will implement our self-calibrating algorithm (presented in [12]) to correct the GNSS to semantic frame offset. The algorithm functions by finding an optimal offset which minimizes the projection error between perceived semantic cues and the reference map. This process corrects the on-board GNSS so that it coincides more closely with the reference semantic map.

Elaborating further on our recent publication [12], we had evaluated the effectiveness of this strategy within the Carla simulation environment and from data recorded from the Year 2 competition. It was found that this method was able to reduce the GPS offset error to only a few centimeters despite large shifts present within the semantic map. As shown in Figure 26, the lane boundaries of the semantic map projected into the image plane appear to be have an offset of around a meter. On the right side of

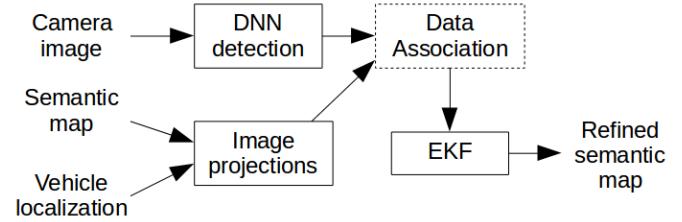


Figure 28: Proposed Change Detection and Correction Pipeline.

the figure, after correcting with our proposed method we see that the painted lane lines and those projected from the map are very similar.

Considering the previous failure scenario, had this self correcting localization algorithm been implemented, the ADS would have been unaffected.

DDT Fall back With the use of this system, the ADS is able to detect global offset errors and attempt to correct them through the re-calibration of the on-board GPS. When the algorithm detects a large discrepancy between the semantic cues and the map, the passenger will be notified and will be given the option to intervene if they so choose. During this time, the ADS will enter the semantic map DDT Fallback. Within this mode the ADS will reduce its speed and attempt to re-calibrate the on-board GPS via the algorithm presented in the previous section. If successful, the ADS will resume normal operation. If the ADS fails to correct the error, it will enter one of two minimum risk conditions. Ideally, it will attempt to park at the shoulder of a road way, engage warning lights and call road side assistance. If a shoulder lane is not available, then the ADS will engage failure mitigation, engaging warning lights, stopping in place, and calling road side assistance.

Given that the ADS is able to execute such a fallback strategy, the severity of the failure mode decreases from 10 to 8. Here we see that the ADS safely stops autonomous functionality in the worst case scenario and provides ample warning. In addition, the Detectability score of this failure cause now receives a rating of 4 as the system will actively look for global shift errors between semantic cues and the reference map. This results in an RPN of 288 for the global offset failure cause.

6.4.2 Local Change Detection and Correction Prior to dispatching, the system will subscribe to road construction announcements to determine and avoid such problematic areas during path planning.

To ameliorate potential inconsistencies between the semantic map and the collected perception data, we can propose a local change correction algorithm that realigns local regions of a map to reflect the actual lane positioning

as detected by vehicle sensors. Furthermore, we can provide updated locations of traffic lights to reflect potential changes in the intersection and new traffic light positions. This framework can be seen in Figure 28. The associations between mapped road elements and those detected via perception will be used to further refine the semantic map during operation through a Kalman filter. The perceived semantic cues will be used to provide a probabilistic update to the GNSS coordinates of the associated element within the semantic map. This local refinement will only affect semantic elements within a small region near the vehicle.

DDT Fallback During normal operation, if a local change to the environment is detected then the driver will be informed and will be given the option to intervene if they so choose. Should they ignore the warning, the Level 4 ADS will enter the fallback mode, reducing speed. During this time the system will attempt to locally correct the map. Should this be successful, the system will resume typical operation. If the map remains uncorrelated, despite the corrections then the system will attempt to enter a minimal risk condition. Similar to the DDT Fallback presented in Section 6.4.1 the system will attempt to park on the shoulder of the road, engage emergency lights and call road side assistance. If such a shoulder is not available, the ADS will engage emergency lights, stop in place and call road side assistance.

As the ADS is able to engage in such failure handling measures, the Severity of the failure mode is once again reduced from a 10 to 8. Similarly the Detectability now receives a rating of 4, as the system has a method to determine if the environment has changed. Thus, the RPN for this remediated failure cause becomes 32.

6.4.3 Improved RPN Score From the previous sections, we see that the global offset failure cause, after correction, retains the highest RPN of 240, down from the original RPN of 720. The proposed SAE Level 4 ADS appears to be more robust to map inaccuracy. In the worst case the system will cease autonomous functionality and enter a minimal risk condition. In the best case scenario, the system will be able to correct the respective failure cause and resume normal operation. Thus, this hypothesized system satisfies the requirements of a SAE Level 4 ADS as outline in SAE J3016.

7 FAILURE MODE: DYNAMIC OBJECT MOTION PREDICTION ERROR

For safe and efficient operation, Zeus must consider the observed states of other vehicles, pedestrians on or near the road, and other objects such as wildlife, and accurately predict the future trajectories of each dynamic object in the scene. A failure in the motion prediction pipeline occurs when the future trajectory of an object diverges from the predicted trajectories and can result in highly severe consequences, such as vehicle collisions. The pre-

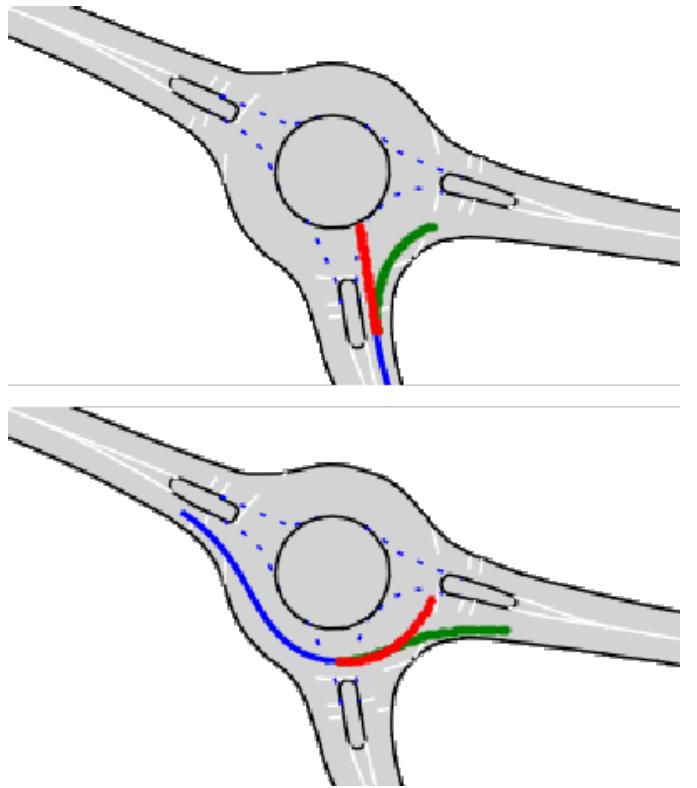


Figure 29: (Top) An example failed prediction using the CV model at a roundabout. (Bot) An example failed prediction using a toy NN model (not our current system) at the same roundabout. Past vehicle trajectories are shown in blue, ground truth future trajectories are shown in green, and predicted trajectories are shown in red.

diction failure rate and the lack of failure detection in our current motion prediction pipeline result in an evaluated RPN rating of 900 for this failure mode.

This section will first go over a brief overview of the planner's functionality related to motion prediction and path planning. We will discuss the severity of this failure mode, followed by a detailed description of two causes for the overall failure mode. Finally, we will lay out a proposed fix that addresses the relevant points of failure and provides a high-level blueprint for enabling Level 4 ADS functionality. Our final proposed system is able to achieve an improved RPN score of 405.

7.1 ASSUMPTIONS OF MOTION PREDICTION SYSTEM Our current motion prediction system is split into two steps. We first perform detection and classification on all objects on the scene via a simple LiDAR-based algorithm. Detection is performed by LiDAR point clustering to generate 3D bounding boxes for each object. Then, to classify each object, we use a Hidden Markov Model (HMM) conditioned on the past detected bounding box dimensions for the object, where associations are made to previous detections using the greedy (Hungarian) method. Here, we assume that the deer is the only

dynamic object and thus only consider bounding box dimensions close to the shape of a deer as dynamic objects. Finally, we use a Kalman filter to process the past observed bounding boxes to produce a position, heading, and velocity estimate for each object.

After we have obtained the classification, position estimate, and velocity estimate for each object, we move on to the motion prediction step. Here, we use a Constant Velocity (CV) model for dynamic objects and estimate the object's future trajectory by propagating the current estimated position of the object forward using the currently estimated velocity up to a certain prediction horizon.

7.2 FAILURE MODE DESCRIPTION AND SEVERITY

We follow common practice for motion prediction algorithms in academia and define a failure in dynamic motion prediction (i.e., a "miss") to have occurred when the difference between the predicted position of the object and its actual position at the final prediction timestep is greater than 100% of the object's maximum dimension or 2 meters, whichever is smaller.

Such a failure could significantly impact the safety of the vehicle and its passengers, given that an inaccurate trajectory prediction for a nearby dynamic object could potentially lead to dangerous maneuvers and potential collisions with the AV. For instance, if the vehicle is in a dense, high-speed driving environment and fails to predict an accurate trajectory for an adjacent vehicle's lane-change maneuver, while the system may be able to provide a warning of an impending collision, it may be unable to engage an evasive maneuver in time, and the vehicle could find itself in a fatal high-speed collision. Therefore, we determine the severity of this failure mode as a 9.

7.3 FAILURE MODE CAUSES We identify two main causes behind this failure mode. The first is an unpredicted behaviour of dynamic objects, which occurs when the motion prediction algorithm fails to predict the future trajectory of the object correctly. Second is the inaccurate class and object state detection, which occurs when perception algorithms fail to discern the type correctly (i.e., vehicle, pedestrian, other) or motion state (i.e., bounding box dimensions, position, velocity) of an object. We discuss both in the following section.

7.3.1 Unpredicted Behaviour of Dynamic Objects Our current assumed CV model and linear Kalman filtering approach utilizes a Maximum a Posteriori (MAP) estimation to find the most probable linear velocity and trajectory. However, this assumes that the dynamic object moves in a straight line, which works well for deers but is often an invalid assumption for vehicles and pedestrians. Additionally, our current approach only predicts a single future trajectory for each agent. In real-world scenarios, the future motion of a vehicle or pedestrian is often

ambiguous when only conditioned on their past trajectory (i.e., without extraneous information such as the driver's intended destination). By only predicting one mode of future motion, we have a high likelihood of motion prediction failure in such cases when a dynamic agent may take multiple future trajectories.

Example Scenario: Roundabout Motion Prediction

Examples of the mentioned problems can be seen in Figure 29, where a roundabout results in the predicted trajectory for the vehicle significantly deviating from the ground truth future trajectory. In the first case, the CV model only predicts linear motion and fails to predict the turning motion of a vehicle entering the roundabout. In the second case, we trained a simple LSTM network to output unimodal predictions for the shown roundabout. However, the network incorrectly estimates that the vehicle will continue in the roundabout rather than exiting the roundabout. Due to the unimodal output of the network, it is not possible to capture both possible future trajectories. Other scenarios where we may observe these failures include lane changes, merging, and intersection navigation.

Current Occurrence Given the issues with both assumed linear motion and unimodal trajectory predictions, it is reasonable to assume that motion prediction failures will frequently occur for vehicles and pedestrians in nearly all situations. When testing our current approach on sample roundabout and uncontrolled intersection situations from the INTERACTION dataset [20], we observed a well over 50% chance for motion prediction failure in every tested scenario. This failure rate is well over a 10% failure occurrence rate, and thus we this failure cause an Occurrence rating of 10.

Current Prevention and Detectability For our one-shot, deterministic prediction model, it is impossible to reason about whether or not a trajectory prediction may be incorrect. Additionally, the only way to know that a trajectory prediction is incorrect is to observe the ground truth future trajectory of the vehicle and compare it with the prediction. This limitation means that our current system cannot detect a motion prediction failure before it occurs. Thus, we cannot detect or prevent this failure cause and give it a Detectability rating of 10.

7.3.2 Inaccurate Object Class and State Detection

Because our system only performs motion prediction on dynamic objects, a misclassification of dynamic objects as static (or vice versa) will result in a motion prediction failure. This misclassification is caused by the noise in our LiDAR data affecting the detected bounding boxes for each object.

Example Scenario: Deer Collision An example of this failure can be seen in Figure 30. This scenario was taken from our run in the Year 2 Autodrive Challenge, where our system incorrectly classified a deer on the road as a static object. As such, the planner operated under an assump-

tion of static behaviour for the deer and thus could not react in time when the deer suddenly started moving and collided with the vehicle. This misclassification of the deer led to an incorrect estimation of the deer’s possible motion and resulted in a situation that would be quite dangerous in the real world.

Current Occurrence As our current system is designed for only differentiating a deer from other static objects, it is reasonable to assume that we will almost always fail in classifying vehicles or pedestrians correctly. However, we have also tested our current system’s performance on sample scenarios at UTIAS. We observe that we incorrectly classify a deer as a static object 0.3% of the time, but we incorrectly classify various static objects as dynamic around 14.3% of the time. This accuracy discrepancy is due to our currently conservative tuning, as we are much more concerned with missing a deer detection than missing detections of stationary objects. Overall, however, we still see a >10% failure rate in our current system and therefore assign this failure cause an Occurrence rating of 10.

Current Prevention and Detectability For obstacle detection/classification, because we do not have access to ground truth information, we may only detect a failure in this system through a low confidence rating on the detection. However, while our system does output a confidence level for each detection, our confidence level is not meaningful for our classification. The confidence level will tell us whether or not the detection is correct, but will not give us information on whether or not the detection is correctly classified as static or dynamic. Therefore, our system cannot detect or prevent this failure cause ahead of time, so we assign this failure cause a Detectability rating of 10.

7.4 HYPOTHETICAL LEVEL 4 ADS Our proposed Level 4 ADS system attempts to mitigate this failure mode via a new motion detection/prediction pipeline with two main components: improved object classification and improved class-based motion prediction algorithms. We wish to develop a final motion prediction pipeline that can

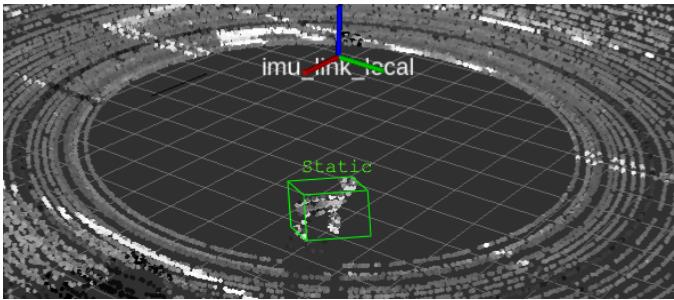


Figure 30: A sample LiDAR detection from our Y2 competition at Mcity. Our detection and classification pipeline mistakenly detects a deer as a static obstacle.

produce much more accurate predictions and generate multimodal predictions to allow for the planner to be robust to all potential future trajectories of dynamic agents. Additionally, to meet Level 4 ADS requirements, we also extend our motion prediction to other vehicles and pedestrians in the proposed pipeline. A diagram of our proposed model can be seen in Figure 31.

7.4.1 Improved Object Detection/Classification Different classes of dynamic objects may have considerably different motion behaviours. For example, in traffic situations, vehicles have completely different motion patterns compared to pedestrians or deer. Our current object classification system uses a very simple LiDAR point clustering algorithm that determines an object to be static or dynamic based on the size and shape of the detected bounding box. While this works well for specifically detecting a dynamic deer object, it is hard to extend to multiple object classes and performs poorly at longer ranges (>30m) where LiDAR points are sparse.

To improve our object detection and classification performance, we propose to use SE-SSD [21], which is a LiDAR-based 3D object detection and classification network based on a teacher-student framework. It is able to be lightweight enough for real-time processing while achieving top performance on the KITTI object detection benchmark [22].

Additionally, we propose to augment our sensor suite with a front-facing Velarray H800 to get better LiDAR data at long ranges and improve our detection performance in areas where our current Velodyne HDL64 would not provide dense enough point clouds. Here, we choose to improve our LiDAR sensor suite over fusing RGB data into detections because the recent SOTA has shown pure LiDAR based object detection techniques to outperform LiDAR+RGB sensor fusion techniques (likely due to the difficulty of training models based on multiple data sources).

DDT Fallback With a single Velodyne HDL64, SE-SSD reports an Average Precision (AP) of 91.84% on BEV bounding box detection on the moderate difficulty KITTI dataset slice (which is representative of typical driving scenarios). Here, we use AP to estimate for our failure occurrence rate to arrive at a final Occurrence rating of 9. However, we note that this is an upper bound on the Occurrence rating, as our proposed addition of a Velarray H800 will improve forward-facing detection results, especially at longer ranges.

Additionally, SE-SSD is able to output a confidence rating for both detected bounding box and object classification. This confidence rating allows us to choose fallback protocols and detect failure cases. In particular, if the distribution of the confidence levels for each class is too flat or if the bounding box confidence level is too low, we may detect an object detection/classification failure and switch to a fallback strategy based on the proximity of the uncer-

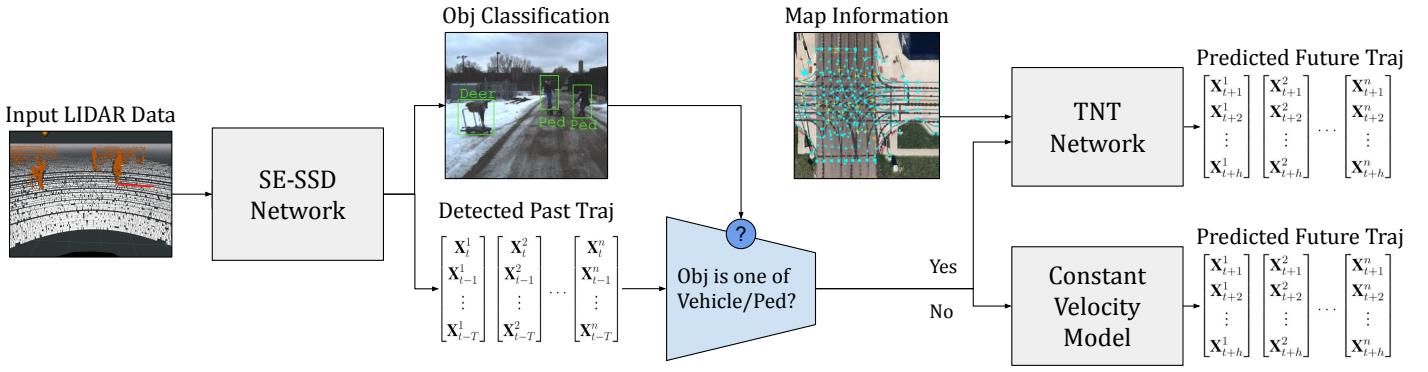


Figure 31: Proposed motion prediction model. The model takes in both RGB and LiDAR data to produce object classifications and past trajectory information, based on the detected bounding boxes. Then, for each object, we check if it is classified as a vehicle/pedestrian or a miscellaneous dynamic object and correspondingly use either the TNT network with input semantic map information of the environment or a CV model to predict its future trajectory.

tain detection to our vehicle. For example, if we receive multiple uncertain detections close to the vehicle, we will achieve a minimal risk condition by stopping the vehicle. Overall, we qualitatively estimate a new Detectability rating for this failure case of 5.

7.4.2 Improved Motion Prediction Algorithms With the development of an improved object detection/classification pipeline, we can condition our output future trajectory prediction on the object’s class in question. In particular, we propose to classify dynamic objects into three distinct classes of vehicles (including bikes), pedestrians, and other objects (e.g., animals) and use a separate motion prediction model for each class. The classification allows us to tailor each motion prediction model to the specific motion tendencies of each class.

For vehicle and pedestrian motion prediction, we propose to use Target Driven Trajectory Prediction (TNT) [23], which is a neural network based approach for vehicle and pedestrian motion prediction recently published by Waymo. TNT learns a model for both destination prediction and future trajectory generation for each agent in the scene and is one of the top-performing techniques on the Argoverse Motion Forecasting benchmark [24]. A key feature of this method is its ability to produce feasible multimodal predictions with corresponding confidence levels, which is essential for situations in which the future intention of surrounding agents may be ambiguous and allow us to account for multiple futures. TNT can also condition its trajectory predictions on a semantic map of the scene, which has been shown in multiple works to increase prediction accuracy greatly [25][26].

We will stick with the currently implemented constant velocity model for motion prediction of other objects. This motion prediction model works well for motion prediction of miscellaneous objects, particularly road agents that do not typically exhibit complex motion (e.g., animals cross-

ing onto the road).

DDT Fallback With the implementation of a multi-modal prediction model for dynamic obstacles, we can significantly reduce the number of incorrect predictions by sampling and considering multiple future trajectories. In terms of vehicles and pedestrians, TNT reports a 9% miss rate (i.e., incorrect prediction) for trajectory predictions on the Argoverse validation set when predicting six possible futures. From our testing, we notice the CV model to produce accurate predictions in an overwhelming majority of cases in terms of other dynamic objects. However, due to the negligible number of encounters with other dynamical objects in a typical AV driving scenario (compared to the number of vehicles/pedestrians encountered), we do not consider this scenario in our final Occurrence rating. Overall, this translates to a new Occurrence rating of 9.

Additionally, because our pipeline now outputs confidence levels for each predicted trajectory, we are able to condition on these confidence levels for fallback protocols and potential detection of motion prediction failures. When the model outputs either low confidence levels for trajectory predictions or the distribution of confidence levels is too flat, we will notify the driver and initiate a DDT fallback to either slow down or stop the AV based on the degree of uncertainty of the trajectory predictions. These low confidence levels may also indicate an impending motion prediction failure, so we qualitatively estimate a new detection rating of 5.

7.5 IMPROVED RPN SCORE As we do not consider improvements related to conservative trajectory planning methods for this failure mode, we are not able to reduce the severity of a motion prediction failure. Additionally, while the proposed system may significantly increase the accuracy of object detections and trajectory predictions, we were only able to slightly decrease the Occurrence ratings due to the difficulty of this failure mode and the high

requirements of the Occurrence ratings. However, by incorporating confidence levels in our models, we can dramatically decrease the Detectability ratings for both failure causes. The final RPN ratings of both failure causes decrease to 405, giving the overall failure mode an RPN of 405. Finally, The proposed pipeline also introduces a concrete metric for evaluating when to perform a DDT fallback and is theoretically able to switch the vehicle to a minimum risk condition in the event of a complete motion prediction failure. From this, the proposed pipeline is able to satisfy the requirements of a Level 4 ADS as outlined in SAE J3016.

8 SAE LEVEL 4 CONDITIONS

SAE J3016 defines the Operational Design Domain (ODD) as "[the] operating conditions under which a given ADS [...] is specifically designed to function". Examples of such conditions include restrictions on geographical areas, traffic scenarios (road participants and traffic signals), road types, environment lighting, weather, and maximum driving speed. This section discusses the relevant operation conditions for our hypothetical SAE Level 4 ADS.

8.1 ENVIRONMENT LIGHTING We require the ADS to only operate at full SAE Level 4 when the environment can be clearly captured by the mounted cameras. This is because our ADS heavily relies on the semantic information extracted from the camera images to make safe decisions. As discussed in Section 5, when the environment is too bright (e.g. when driving facing the sun), too dark (e.g. when driving after sunset) or too high contrast (e.g. when leaving a tunnel), the lighting condition may exceed the camera sensors' physical limitations and result in overexposed or underexposed images. However, complex lighting conditions will not incapacitate the entire ADS as long as appropriate DDT fallback and recovery protocols are in place. As discussed, the vehicle is equipped with additional lighting-agnostic sensors (LiDAR on the Year 4 ADS and additional scanning RADAR on the Level 4 ADS) to provide geometric information of the en-

vironment. Figure 32 depicts a scenario of a yield sign in snowy weather. While we are able to detect and localize an object in front of the vehicle under low-light condition, the ADS is not able to identify the type of the sign and must inquire the driver for high-level instructions.

8.2 WEATHER CONDITION We require the ADS to only operate at full SAE Level 4 under sunny or overcast conditions. As a Canadian team, we have been combatting inclement weather conditions since Year 1: we train the perception DNNs on datasets collected on Toronto public roads in various weather and illumination conditions, tune the controller in both dry and snowy road conditions, and perform closed-loop experiments as long as the weather is not too inclement. Nevertheless, as previously mentioned, we have observed reduced effective detection range and increased braking distance in rain and snow which could lead to longer-than-usual response time and even catastrophic collisions. While it is possible to develop weather-specific driving strategies for an SAE Level 4 ADS, we believe safety driver supervision is still required to ensure safe operation under inclement weather conditions.

8.3 GEOGRAPHICAL AREA We require the ADS to only operate at full SAE Level 4 in areas where a semantic map is available, traffic signage on the road is included in training and testing data, and thorough deployment tests have been done. As discussed in the previous section, the semantic map touches upon almost every component of the ADS and plays a critical role to ensure safe and robust operation. Traffic signage, such as traffic lights and traffic signs, may have vastly different appearances across regions. Thus the perception algorithms may not be able to recognize critical signals if they have not been trained on relevant samples. Finally, different areas may present special road or environment characteristics that could be easily overlooked without a complete deployment test.

8.4 ROADWAY TYPE We require the ADS to only operate at full SAE Level 4 capabilities when lane markings are clearly visible. While the ADS only requires a semantic map to navigate when GNSS information is available, the Semantic Localization system discussed in Section 4 leverages visible lane markings to localize the vehicle when GNSS drops. Thus, driving on unmarked roads may incapacitate the GNSS failure DDT fallback system. Moreover, vulnerable road users such as pedestrians and cyclists have a higher chance of unintentionally venturing onto the vehicle's current lane on unmarked roads. Compared to vehicles, pedestrians and cyclists can more easily change their motion randomly and thus the ADS object motion prediction system may fail to correctly predict their intentions. Therefore, we require safety driver supervision on unmarked roads to ensure safe operation.



Figure 32: An example of a failed sign detection. (Left) Original image. (Right) Darkened version of the image.

8.5 MAXIMUM SPEED We only allow the ADS to engage in autonomous operation at speeds that do not exceed an empirically verified speed limit. For example, we limit the Year 4 vehicle to a maximum speed of 25Km/h based on real world driving experiments. There are two primary reasons. Firstly, vehicle dynamics can vary drastically at different speeds. At low speeds (less than 15Km/h), we usually do not need to consider air drag and tire slippage. On the other hand, the effect of external forces becomes more significant at higher speeds and in extreme cases, could lead to controller instability. Unfortunately, the only way to validate the vehicle's performance is via extensive dynamic testing at designated test tracks. Secondly, the vehicle should not move too fast in order to give the ADS sufficient time to respond to external events. Assuming a perception-reaction time of 0.5s, an effective perception range of 100m, and a harsh braking deceleration of 3m/s, the vehicle should always drive slower than 85Km/h to ensure safety. Again, the exact speed upper bound must be verified through extensive closed-loop driving experiments.

9 PATENTS, PAPERS AND CONFERENCES

This section lists all publications and conferences attendance by aUToronto throughout all 3 years of the competition.

Conference and journal papers published:

1. K. Burnett, A. Schimpe, S. Samavi, M. Gridseth, C. W. Liu, Q. Li, Z. Kroese, and A. P. Schoellig, "Building a winning self-driving car in six months," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 9583–9589.
2. K. Burnett, S. Samavi, S. Waslander, T. D. Barfoot, and A. P. Schoellig, "aUToTrack: a lightweight object detection and tracking system for the SAE AutoDrive challenge," in *Proc. of the Conference on Computer and Robot Vision (CRV)*, Best poster presentation award, 2019, pp. 209–216.
3. K. Burnett, J. Qian, X. Du, L. Liu, D. J. Yoon, T. Shen, S. Sun, S. Samavi, M. J. Sorocky, M. Bianchi, et al., "Zeus: A system description of the two-time winner of the collegiate sae autodrive competition," *Journal of Field Robotics*, 2020.
4. W.-K. Tseng, A. P. Schoellig, and T. D. Barfoot, "Self-Calibration of the Offset Between GPS and Semantic Map Frames for Robust Localization," in *Proc. of the Conference on Computer and Robot Vision (CRV)*, 2021.

Conferences attendance:

- IEEE International Conference on Robotics and Automation (ICRA), Montreal, QC, 2019
- Conference on Computer and Robot Vision(CRV), Kingston, ON, 2019
- Conference on Computer and Robot Vision(CRV), Burnaby, BC, 2021

10 CONCLUSION

This report summarizes Zeus' major hardware and software design changes and lessons learned over the past four years and outlines potential extensions to reach a hypothetical SAE Level 4 automated driving system. Through a whole-system Design Failure Mode and Effect Analysis, the top 4 failure modes are identified: loss of GNSS, inaccurate semantic map, compromised perception sensors, and incorrect dynamic object motion prediction. For each failure mode, the report justifies its significance and walks through the Level 4 DDT fallback strategies. Finally, the report discusses the operational design domain of a hypothetical Level 4 system.

References

- [1] K. Burnett, A. Schimpe, S. Samavi, M. Gridseth, C. W. Liu, Q. Li, Z. Kroeze, and A. P. Schoellig, "Building a winning self-driving car in six months," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 9583–9589.
- [2] K. Burnett, J. Qian, X. Du, L. Liu, D. J. Yoon, T. Shen, S. Sun, S. Samavi, M. J. Sorocky, M. Bianchi, *et al.*, "Zeus: A system description of the two-time winner of the collegiate sae autodrive competition," *Journal of Field Robotics*, 2020.
- [3] J. Qian, K. Burnett, O. Rasheed, P. Malik, L. L. S. Yu, S. Lu, J. Li, C. Rosic, J. Liao, W. Tseng, Q. Li, Z. Edher, A. Seifeldin, Y. Huang, Y. Chen, A. Schoellig, and T. D. Barfoot, *Autoronto concept design report year 3*, 2020.
- [4] K. Burnett, R. Adragna, A. Arkhangorodsky, X. D. M. Bianchi, K. He, Z. Huang, Z. Huang, L. Liu, S. Lu, M. Pham-Hung, O. Rasheed, J. Qian, T. Silva, Q. Sykora, D. J. Yoon, K. Zhang, and T. D. Barfoot, *Autoronto concept design report year 2*, 2019.
- [5] M. J. Paul Viola, "Rapid object detection using a boosted cascade of simple features," in *COMPUTER VISION AND PATTERN RECOGNITION 2001*.
- [6] P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, and R. Siegwart, "Robot-centric elevation mapping with uncertainty estimates," in *International Conference on Climbing and Walking Robots (CLAWAR)*, 2014.
- [7] B. Wu, F. Iandola, P. H. Jin, and K. Keutzer, "Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 129–137.
- [8] K. Burnett, S. Samavi, S. Waslander, T. D. Barfoot, and A. P. Schoellig, "aUToTrack: a lightweight object detection and tracking system for the SAE AutoDrive challenge," in *Proc. of the Conference on Computer and Robot Vision (CRV)*, Best poster presentation award, 2019, pp. 209–216.
- [9] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 697–12 705.
- [10] Novatel. (2020). "Precise point positioning (ppp)," [Online]. Available: <https://www.novatel.com/an-introduction-to-gnss/chapter-5-resolving-errors/precise-point-positioning-ppp/>.
- [11] X. Wen, L. Ji, X. Zhang, and J. Zhao, "Fault detection and diagnosis in the ins/gps navigation system," in *2014 World Automation Congress (WAC)*, 2014, pp. 27–32. DOI: 10.1109/WAC.2014.6935645.
- [12] W.-K. Tseng, A. P. Schoellig, and T. D. Barfoot, "Self-calibration of the offset between gps and semantic map frames for robust localization," in *Proc. of the Conference on Computer and Robot Vision (CRV)*, 2021.
- [13] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, "Gated-scnn: Gated shape cnns for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 5229–5238.
- [14] Y. M. G. d.o.o., *Toronto, canada - detailed climate information and monthly weather forecast*. [Online]. Available: <https://www.weather-atlas.com/en/canada/toronto-climate>.
- [15] M. Kutila, P. Pyykonen, M. Jokela, T. Gruber, M. Bijelic, and W. Ritter, "Benchmarking automotive lidar performance in arctic conditions," *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020. DOI: 10.1109/itsc45102.2020.9294367.
- [16] D. Scimeca, *Blackfly s camera*, Sep. 2019. [Online]. Available: <https://www.vision-systems.com/cameras-accessories/article/14040570/flir-releases-board-version-of-blackfly-s-camera#:~:text=The%5C%20cameras%5C%20have%5C%20an%5C%20operating,require%5C%20require%5C%20%5C%20V%5C%20power>.
- [17] G. R. Mueller and H.-J. Wuensche, "Continuous stereo camera calibration in urban scenarios," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–6. DOI: 10.1109/ITSC.2017.8317675.
- [18] C. of Toronto, *Ward 11 - university-rosedale*, May 2021. [Online]. Available: <https://www.toronto.ca/city-government/data-research-maps/neighbourhoods-communities/ward-profiles/ward-11-university-rosedale/>.
- [19] Y. Charugundla and Y. Swaminathan, "Mapping an autonomous driving future," p. 6, 2020.
- [20] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kümmeler, H. Königshof, C. Stiller, A. de La Fortelle, and M. Tomizuka, "INTERACTION Dataset: An INTERnational, Adversarial and Cooperative moTION Dataset in Interactive Driving Scenarios with Semantic Maps," *arXiv:1910.03088 [cs, eess]*, Sep. 2019.
- [21] W. Zheng, W. Tang, L. Jiang, and C.-W. Fu, *Se-ssd: Self-ensembling single-stage object detector from point cloud*, 2021. *arXiv: 2104.09804 [cs.CV]*.
- [22] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.
- [23] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid, *et al.*, "Tnt: Target-driven trajectory prediction," *arXiv preprint arXiv:2008.08294*, 2020.
- [24] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, *et al.*, "Argoverse: 3d tracking and forecasting with rich maps," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8748–8757.
- [25] K. Messaoud, N. Deo, M. M. Trivedi, and F. Nashashibi, "Multi-head attention with joint agent-map representation for trajectory prediction in autonomous driving," *arXiv preprint arXiv:2005.02545*, 2020.
- [26] N. Djuric, V. Radosavljevic, H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, N. Singh, and J. Schneider, "Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 2095–2104.