Weighted Supervision Learning for the Automation of Railway Simulation

Josh Hunter[1] [0000−0003−4745−4272], John McDermid[1] [0000−0003−4745−4272], Simon Burton[1], and Mia Dempster[2] [0009−0009−3035−7303]

[1] Centre for Assuring Autonomy, York, United Kingdom josh.hunter@york.ac.uk
[2] Leeds Law School Leeds, United Kingdom

Abstract. The railway industry faces significant challenges in adopting machine learning (ML) and automation technologies due to limitations in available simulation tools, high data acquisition costs, and a lack of comprehensive open-source solutions. OpenBVE, an open-source railway simulator, offers potential as a platform for addressing these gaps. This paper explores the applicability of OpenBVE for academic research, focusing on its modification and adaptation to support ML and safety metric development. Key contributions include introduction of a novel scoring system designed for machine learning, the Observed Weight of Operation Bayesian framework for supervised learning and the assessment of OpenBVE as a vision network. These modifications enable OpenBVE to provide consistent, scalable environments for ML training without compromising operational realism. The work adopts a multi-faceted approach, testing OpenBVE across rules-based, supervised, reinforcement, and vision-based methodologies using a short UK rail route. Results demonstrate the simulator's potential to serve as a proof-of-concept platform for railway research, bridging the gap between simulation and real-world operation. However, this study highlights limitations in OpenBVE's native structure, emphasizing the need for further development to fully realize its potential in advancing railway automation and ML integration.

1 Introduction

Railway as an industry faces significant challenges in simulation, where available options are often limited by cost, fragmentation, or gamification. A study by Moloney highlights that high-end commercial railway simulators typically cover only 30 kilometers of track, with data acquisition costs averaging €900 per kilometer [24]. Fragmentation exacerbates the issue, as Medeossi and Fabris note that while various aspects of the railway ecosystem—such as signaling, timetabling, and stock analysis—have their own specialized open-source tools, no comprehensive "all-in-one" simulator exists at sufficient popularity or usability for academic research [22]. In contrast, the gamification of certain simulators has led researchers, like Mauri *et al.*., to resort to using video game data, such as that extracted from Grand Theft Auto V, to supplement the lack of railway-specific datasets [21].

This scarcity of high-quality, accessible data also constrains the development of machine learning (ML) technologies in railway research. For instance, a study by Aleksandar Petrović *et al.*. on convolutional neural network integration collected datasets of just 150, 500, and 1,000 images, relying heavily on manual curation or extraction from publicly available videos. Despite achieving promising results, the study cites dataset size as a key limitation [33]. Similarly, the RailSem19 dataset developed by Oliver Zendel *et al.*., consisting of 8,500 annotated ego-perspective railway images, is limited in scope. It lacks sequential data and fails to incorporate critical information like signaling or timetabling, thereby restricting its applicability to semantic rail scene understanding [13]. Further attempts to generate datasets, such as those by Faghih-Roohi *et al.*. and Gilbert *et al.* remain hindered by size and contextual limitations, leading to underwhelming results [7, 8, 23].

Compare this to the automotive domain wherein data for the purpose of ML training can be collected using open-source simulators such as CARLA, a simulator which advertises itself as being developed *'from the ground up to support development, training, and validation of autonomous driving systems.'* [5]. CARLA's flexibility and extensibility have enabled its use in thousands of ML research studies, catalyzing advancements in autonomous vehicle technologies. Suggesting that development of open-source railway simulators may assist the railway industry in achieving similar results.

1.1 An open source railway simulator

The OpenBVE project [3] is a ferroequinologist specialty software which began development in 2001, and is still actively maintained. This open-source simulator allows users to operate various user-created trains, often modeled with high fidelity to real-world locomotive specifications, on equally realistic, user-generated routes. OpenBVE prioritizes realism over user-friendliness, offering detailed route timing, an advanced physics engine, and precise train control and cab environments. Its design philosophy, as stated by the developers, emphasizes "realism, not necessarily user-friendliness" compared to other simulators in its genre, especially commercial alternatives [2]. While OpenBVE has yet to see widespread academic application, this is likely a result of its niche audience rather than limitations in its academic utility.

Designed for realism rather than gamification, OpenBVE incorporates detailed physics, accurate route timing, and customizable environments. While not originally intended for academic use, its open-source architecture makes it a candidate for adaptation to ML research. This paper does not evaluate OpenBVE in its native state but instead explores its potential after substantial modifications designed to make it suitable for academic research. Within previous work, we explored the critical information required by cab drivers for safe railway operation and found that railway safety is largely defined by the interaction between drivers and signal operators. This aligns with Tetsuo Uzuka's 'system of systems' model, which we use to contextualize the necessary components for safety-relevant simulations [32] which we show within figure 1. For OpenBVE to qualify as an academic tool, it must facilitate communication with signal operations and dispatch, providing data that mirrors real-world railway interactions.

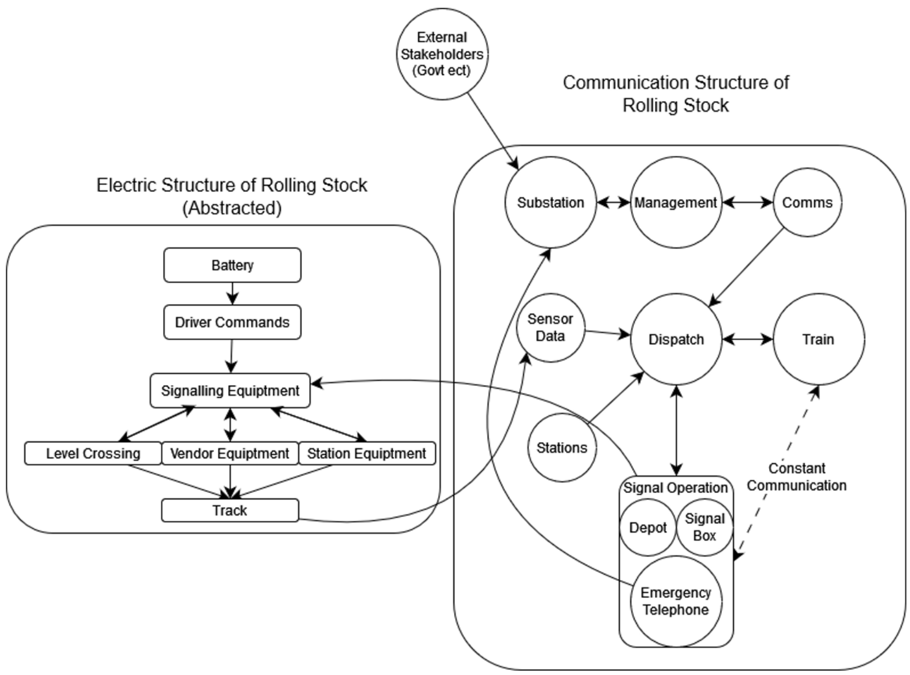

Fig. 1. Japanese System of Systems model as defined by Uzuka [32].

To evaluate OpenBVE's suitability, we test three distinct approaches to data collection and ML application within the simulator:

1. Rules-based operation
   OpenBVE must demonstrate its ability to replicate Automatic Train Operation (ATO), classified as "low-autonomy" in the Grades of Automation (GoA) framework [30]. ATO involves automated starting, stopping, and routine non-emergency operations, driven by predefined rules such as reducing speed at specified track sections to adhere to speed limits.
2. Naive Classification Naive Classification uses a probabilistic approach to predict train control inputs based on past data. It applies a Naive Bayes (NB) model, assuming each observed variable (such as time, speed, and acceleration) contributes independently to the prediction of the next control action. This method processes large amounts of simulation data to classify inputs based on probability distributions.
3. Weighted Supervision Learning Weighted Supervised Learning builds on Naive Classification by introducing context-based weighting for observed

variables. Instead of treating all observations equally, it adjusts the influence of each variable depending on the train's operational state. This approach refines input predictions by prioritizing the most relevant factors for each specific driving condition.

1.2 Modifications to OpenBVE

OpenBVE's original scoring system was designed to replicate real-world train operation, emphasizing safety and precision in a predominantly low-reward environment. As shown in Table 1, this framework imposes strict penalties for significant operational failures and moderate penalties for lesser infractions while rewards for correct actions remain modest. Such a penalty-focused scheme aligns with real-world priorities in which avoiding accidents and major mistakes is paramount. However, for the sake of machine learning, this strict, low-volatility scoring system makes tracking progress difficult, for the purposes of this experiment, a fork of OpenBVE was made so that any modifications made did not interfere with the non-academic uses of the software. for purposes of clarity and replication, the specific modifications made were:

Table 1. Original OpenBVE Scoring System

Event	**Score**
Overspeed	-1 per second
Passed Red Signal	-100
Toppling	-10
Derailed	-1000
Passenger Discomfort	-20
Doors Opened in Transit	-10
Arrived at Station	+100
Perfect Time Bonus	+15
Late to Stop	-0.33 per second
Perfect Stop Bonus	+15

Scoring Modification To make automation more traceable, we introduced additional reward signals and adjusted penalty magnitudes, striving for more frequent and interpretable feedback without sacrificing the simulator's commitment to safety. Table 2 outlines these modifications, including a new *NoDiscomfortAtSignal* reward and a dynamic scaling of overspeeding and derailment penalties. The purpose of these changes was to verify where exactly imperfections were occurring.

1. Scoring Access Modified scoring to be read/write rather than private/read only, allowing for manual override of score.

Table 2. Modified OpenBVE Scoring System with RL Variables

Event	**Reward** (R)	**Bonus** (B)
Overspeed	-(Speed - Speed Limit × 2)	-10%
Passed Red Signal	-100	0
Minor Topple	-10	0
Derailed	-1000	-1000 per second
Passenger Discomfort	-20	-1 per second
Doors Opened in Transit	-10	0
Premature Departure	-50	0
Late to Stop	0	-0.33 per second
Arrived at Station	$R_{\text{Arrival}} = +100$	0
Perfect Time Bonus	$R_{\text{OT}} = +10$	$B_{\text{OT}} = +10\%$
Perfect Stop Bonus	$R_{\text{PS}} = +15$	$B_{\text{PS}} = +15\%$
No Discomfort At Signal	$R_{\text{Signal}} = +3$	0

2. Failure Handling Introduced code allowing for automatic reset on failure rather than requiring user input.
3. Failure Logging Introduced logging on score changes, explaining why and what changes are needed.
4. Signal Classification Modification to scoring for arrival based multiplicative rewards required re-classification of signals (Originally there was no difference between a station signal and a normal signal.)
5. Operation Limitation Removed timetabling integration as timetabling was overriding modified scoring structure.
6. Obstacle Addition Added collision objects on specified route to force derailment and test for derailment avoidance.
7. Vision Classification Modified BVE so vision can be toggled between mesh and wireframe for purposes of collision verification (testing)

2 Data collection approaches for Machine Learning application.

The successful application of ML in railway operations depends heavily on the quality and quantity of data available for training and evaluation. As outlined in the introduction, the 'critical factors' need to be analysed in order to judge the overall applicability to ML training/development. This section examines the data collection methods employed in OpenBVE to support four distinct ML approaches: rules-based operation, supervised learning, RL, and vision suitability.

Each approach is analyzed within the constraints outlined in the scope, focusing on its ability to capture meaningful data relevant to safe and efficient railway operations. Through communication with the OpenBVE team, a modified version of BVE was created which gives more access to the scoring functionality.

This modified version of BVE is available at a forked version of the OpenBVE main GitHub [3] with each individual change being available on the GitHub belonging to the author of this paper [13]. Initial access and modification to OpenBVE was created through email correspondence [16]. By leveraging the modified OpenBVE platform, this section explores how these approaches enable the simulation to act as a testing ground for ML systems, evaluating their potential to advance automation and safety within railway environments.

2.1 Rules-Based Operation

Current implementations of GoA-2 systems are intended as a transitional stage between driver-operated trains and fully autonomous operations. GoA-2 relies heavily on Automatic Train Operation (ATO) for tasks like acceleration and braking, while still requiring human intervention for more nuanced actions such as monitoring for anomalies, opening and closing doors, and handling emergency situations. Despite its automation, the authors of this paper have previously criticized GoA-2 for its susceptibility to monotony-induced human error [14], as drivers are relegated to passive roles for long durations. This lack of active engagement has been correlated with issues such as microsleep and delayed reaction times during critical moments, as demonstrated by Brandenburger et al in [1]. The summary being that GoA-2 systems can reasonably drive a typical operating scenario, getting from station A to station B while obeying all traffic rules and signal commands, which we refer to as 'the robotics approach.' Despite these challenges, robotics-driven approaches serve as a critical baseline for benchmarking more advanced systems as they provide a "control" against which the efficacy of learning-based models can be measured. By operating within a fixed rule set, they highlight the potential gains achievable by integrating more adaptive strategies, such as machine learning.

2.2 Naive Approach

NB within railway is well-researched, with NB approaches being proposed within station management, timetabling and for the purposes of incident detection. [17, 18, 29]. As identified by Shi et al, in the field of railways, outside of governmental bodies, quantitative safety risk analysis methods such as FMEA, HAZOP and indeed, NB. The area of accident causation in particular is a complicated topic, the railway industry cannot agree on a standard for risk evaluation, a paper by Wei-Ting Hong et al *Railway accident causation analysis: Current approaches, challenges and potential solutions* reviews different novel approaches and discusses the reason why they may not be accepted, with NB accounting for 5% of the literature within the field of accident causation [10]. Hong et al mention simulation as well as discussing this 'systems thinking-based approach' as a potential avenue for the causation analysis, providing ample justification for the implementation of a bayesian approach within railway simulation for the purpose of operation and potentially incident detection.

Through collaboration with the BVE team, a version of OpenBVE wherein data can be scraped using the 'iRuntime' API has been developed, using this modified version, a listener add-on has been developed for the purpose of gathering the variables of:
Time (T), Speed (S), Speed Limits (SL, SLS)[3], Rate of Acceleration (RoA), Engine State (ES), Power (P) and Brake (B).

These variables are then classified as controlled Inputs (I) which the simulator's controller can directly manipulate, and observed Outputs (O)[4] which result from these inputs.

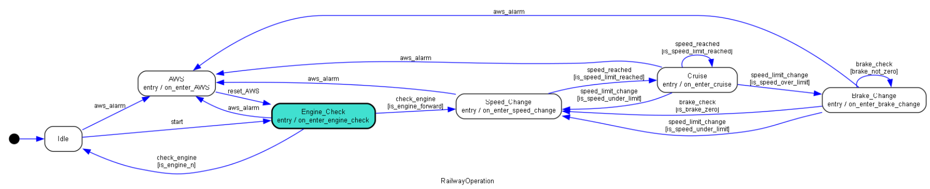

Fig. 2. Finite state machine influencing supervision approach for railway operation.

Building on this foundation, the behavior of the train is further classified into distinct operational "states" to contextualize the relationships between I and O. Each state represents a specific mode of operation within railway operation and determines the relative importance of observed outputs for guiding the next set of controlled inputs. This approach acknowledges that the relevance of outputs is not static but varies depending on the operational context. An abstract representation of these 'classes' is represented within Figure 2.2 and the full representation as well as further documentation can be found at [11, 12].

2.3 Observed Weight of Output

Within different 'states' the system has a different set of given goals, for instance, in the 'cruise' state, the train's primary objective is to maintain a steady speed below the speed limit. In this context, the SL is the most critical observed output, while the RoA becomes less relevant since maintaining speed takes precedence over rapid changes. Conversely, in the 'Brake_Change' state, where a significant reduction in speed is required, RoA gains critical importance as the system needs to decelerate effectively without risking a complete stop or

---

[3] SL refers to the set speed limit of the track, while SLS refers to the Speed Limit of the Signal, the required speed of the locomotive when a signal is to be acknowledged.

[4] Different locomotives use different representations for O with the two standards being B and P as either two separate variables as within our example, or as one variable with negative results representing breaking, positive results representing acceleration. For the purposes of our example, B and P are separate.

violating safety thresholds. These context-specific weights are quantified as the Observed weight of Outputs (OwO) and serve as inputs to a Bayesian model for predicting the next controlled input $\hat{I}$.

The OwO framework dynamically assigns weights to outputs based on their relevance within each state, ensuring that predictions about the next inputs are context-sensitive and optimized for the current operational mode. The context behind each weight of operation has been determined through previous study, communication with railway drivers and overall has been determined through the application of the Safe Autonomy of Complex Railway Environments within a Digital Space (SACRED) methodology [15]. An explanation of each 'state' as well as the specific values chosen for the weighted Outputs (wO) is shown within table 3.

Table 3. Observation Weights Breakdown by State

State	$\boldsymbol{T}$	$\boldsymbol{S}$	$\boldsymbol{SL}$	$\boldsymbol{SLS}$	$\boldsymbol{RoA}$	$\boldsymbol{ES}$	PI	**Description**
Cruise	135%	150%	150%	0%	10%	0%	0%	Maintains speed close to the limit between stations. Adjustments are minimal and handled in Brake_Change or Speed_Change.
AWS	0%	0%	0%	200%	0%	0%	200%	Triggered when passing a signal; train exits its current state to acknowledge AWS. Only valid input: setting both P and B to 0.
Engine_Check	0%	0%	0%	0%	0%	200%	0%	A buffer state at journey start or after AWS exit, ensuring $\hat{I} = 0$ before speed-modifying states.
Brake_Change	10%	150%	150%	50%	150%	0%	50%	Reduces speed to stay within limits. Considers S, SL, and RoA while avoiding counteracting inputs.
Speed_Change	10%	150%	150%	50%	150%	0%	50%	Increases speed after a drop. Similar to Brake_Change but ensures consistency and prevents over-correction.

Given these state-specific weights, we apply the OwO within a modified Bayesian framework to predict the next input, using the formula:

$$\hat{I} = \arg\max_{I} \left(P(I) \prod_{o \in O} P(o \mid I)^{wO} \right)$$

Here, $P(I)$ represents the prior probability of the input, $P(o \mid I)$ represents the conditional probability of observing o given I.

wO is the weight of the output O as determined by the current state. Specifics of each given state have been obtained by previous research, wherein we discussed operation of the railway with current and ex railway professionals (drivers, signal officers and route managers) gaining insight into what is considered important during operation.

This approach ensures that the system dynamically adjusts its predictions based on the context of operation. For example, during 'Cruise,' higher weights for T and SL bias the predictions toward maintaining speed and adhering to speed limits, while reducing the influence of RoA. In contrast, during 'Brake Change,' the prominence of RoA directs the system to prioritize deceleration, accounting for rapid changes in speed limits.

3 Results

To test the hypothesis that the OwO model provides improved predictions over a traditional NB approach, an experiment was conducted using OpenBVE. Three hours of simulator data was collected using the modified version of OpenBVE, which tracks key inputs and outputs as described in Section I.B. This process yielded a dataset of 1,500,000 input-output pairs over nine runs of the selected route. While the dataset serves as a proof of concept, it is not without limitations—none of the authors are professional railway drivers, and we were unable to involve an operator familiar with this specific route. As a result, the dataset reflects typical amateur-level driving behavior rather than optimized professional operation.

To evaluate the hypothesis, the data was processed using three predictive models: Gaussian Naive Bayes, and two versions of the OwO model described within the arg-max formula. One model with previous inputs excluded and one with them included. All models aim to predict the next input sequence $\hat{I}$ based on observed outputs O. The results of this comparison are summarized in Table 4.

The findings indicate that the OwO model demonstrated a ∼5% improvement over the NB approach across most states, with the most significant gains observed in states requiring contextual awareness, such as Cruise, Brake Change, and Speed Change. When the PI is introduced, the improvement further increases [5].

[5] the reason for differentiating between PI and non PI is relevant within the future work section.

Table 4. Comparative Accuracy Results Between Naive Bayes and Observed Weight of Outputs Models

State	NB Acc.	OwO w/o PI.	OwO with PI	Description
Cruise	65-75%	~80%	~90%	Failures in models typically occur due to overspeeding.
AWS	~99%	~99%	~99%	AWS state has a single valid input: setting both P and B to 0. The small error margin (~1%) is due to minor delays leading to slight inaccuracies in frame-perfect inputs.
Engine_Check	~65%	~60%	~85%	NB outperforms OwO in transitioning into E_C. When previous inputs are given alongside the context of transitioning state, OwO's accuracy improves substantially.
Brake_Change	~85%	95-99%	~82-99%	OwOPI struggles due to undercorrection when P and B are low, however OwOPI is quicker than NB when P/B are high.
Speed_Change	~85%	95-99%	~82-99%	Similar to Brake Change, OwO outperforms NB by effectively handling input adjustments, preventing simultaneous or conflicting changes.

Interestingly, certain states showed little to no improvement. For instance, in the AWS state, where only one valid input exists (acknowledging the alarm), both models achieved near-perfect accuracy (99%), indicating that context had minimal impact in this scenario. Conversely, in the Engine Check state, the NB model marginally outperformed the OwO model, likely due to inconsistencies in the transition to and from this state, which are not yet fully modeled in the OwO framework. Furthermore, to properly model the Engine Check so that it would account for every engine state, not just the state described within Section I.B. A further investigation into both 'the station cycle' and 'pre-departure' would be required, as described within the complete version of the state machine which can be found as preliminary work hosted on GitHub [11].

These results suggest that the incorporation of contextual information significantly enhances performance in states where decision-making relies on a nuanced understanding of the operational environment. This aligns with real-world railway training, where operators are taught to adapt to route-specific and situational factors. However, the route-specific nature of *OwO* highlights the need for robust Operational Domain Model (ODM) definitions to generalize the approach across various routes and scenarios.

While further research is required to refine the definition and implementation of contextual weights, this study demonstrates the potential of the OwO model as a foundation for supervised learning in railway simulation. The results are promising and underscore the value of integrating state-dependent observations into predictive frameworks for railway automation.

4 Naive Bayes for Accident Causation.

Within the paper by Hong et al, a common pitfall in systems thinking-based theory is identified: focusing too narrowly on individual incidents leads to overly specific models, while too broad a focus produces vague, unusable outcomes. This challenge reflects some of the unique complexities inherent in railway operations, suggesting that a one-size-fits-all approach to railway safety may be inadequate.

In real railway operations, drivers are trained on individual routes. Hamilton and Clarke [9] demonstrated that even slight environmental changes can affect driver performance, as drivers often rely on memory to recognize signal locations. Similarly, Pandou et al. [25] and Luther et al. [20] emphasize the human factors influencing safe operation. While these studies focus on the human element, Hong's analysis on the pitfalls within currently existing accident causation models indicates that the inherent complexity of railway systems might require more nuanced, route-specific methodologies. In other words, it may be that each railway route necessitates its own investigation and safety analysis, adding justification to the *OwO* framework and at a larger scale, the SACRED methodology [15] to fully capture the operational nuances of the railway space.

Full railway automation cannot be achieved without integrating information from the broader systems [27]. Railway systems should be viewed not as isolated entities but as interconnected parts of a larger whole. The railway landscape would benefit from a comprehensive re-evaluation from a systems-of-systems perspective to assist the redefinition of the field for a digital age. Supervised learning offers a promising alternative to deterministic models such as those in GoA-2, which rely on human intervention to respond to dynamic inputs, machine learning methods can generate dynamic outputs based on historical input data.

This is however, not without drawbacks. The simulation environment in which the *OwO* approach has been tested is a simplified environment, one where incidents do not occur; while signal problems are indeed possible due to the modifications to the SPU, they are rare to the point they were not encountered during experimentation. The domain in which this experiment takes place is one of clear weather, tracks free from debris and one where timetabling issues are nonexistent. As a proof of concept this presents little issue, however it is clear that further work is required to investigate that these improvements brought from the *OwO* approach hold within more complex environments.

4.1 Future Work

In order to test the scalability of the *OwO* approach, the complexity of the simulated environment must increase, in future research, the approach could be applied and extended to analyze and represent areas identified as critical to railway safety within external research.

Vision Application While ML operation on BVE presents an interesting opportunity and the results seem promising, the novelty of the system is called into question if BVE's applicability as a vision agent cannot be verified. As mentioned, the 'state of the art' regarding vision technology within rail is lacking, with existing datasets like RailSem19 limited by their static, non-sequential nature. For BVE to establish itself as a viable alternative or complement to RailSem19, it must demonstrate its ability to generate datasets that not only capture semantic richness but also provide temporal continuity essential for real-time decision-making in safety-critical applications.

Future work will focus on bridging this gap by leveraging BVE's dynamic simulation capabilities to create annotated datasets tailored for tasks such as signal detection, obstacle recognition, and track segmentation. These datasets will be benchmarked against RailSem19 using cutting-edge models, including DE-ViT [34], YOLO, and Roboflow [6], to assess their efficacy in open-set and few-shot object detection scenarios. Additionally, efforts will be directed toward automating the annotation pipeline for BVE-generated data, ensuring scalability while maintaining precision.

Reinforcement Learning As part of the initial experimentation within this paper, a reinforcement agent was developed and tested within OpenBVE to compare the suitability of Reinforcement Learning given both the old scoring system and the new, the initial scoring system was designed to replicate real-world train operation, emphasizing safety and precision in a predominantly low-reward environment. However, for reinforcement learning (RL) applications, this leads to sparse, high volatility rewards that can destabilize agent training [31].

To make RL more traceable, we introduced additional reward signals and adjusted penalty magnitudes, striving for more frequent and interpretable feedback without sacrificing the simulator's commitment to safety. Table 2 outlines these modifications, including a new *NoDiscomfortAtSignal* reward (for smooth signal handling) and a dynamic scaling of overspeeding and derailment penalties. Although *NoDiscomfortAtSignal* partially boosts early rewards, the environment remains low-reward overall. Indeed, our analysis indicates that heavy volatility still manifests when signals cluster at early stations. In principle, one could follow non-exponential stabilization techniques, such as those proposed by Schultheis *et al.* [28], to systematically regulate this variance. However, implementing their framework in full lies beyond the scope of this work.

To mitigate the most extreme early-stage spikes, we hypothesize that the implementation of a buffer score could be used to stabilize RL feedback without completely eliminating volitility. This aligns with the general principle of non-exponential stabilization focusing longer-term performance over immediate gains yet stops short of a full Schultheis-style approach. However, experimentation yielded mixed results and overall a satisfactory RL agent could not be produced at this time. We aim to improve upon this RL system and hopefully produce a reinforcement agent operating on the simulated route sometime within the future.

Failure Management Failure in railway operations is currently managed through the actions of a human driver, who is responsible for securing the train, assessing environmental damage through physical evaluation, and ensuring passenger safety. Autonomous systems, however, lack the capability to perform these tasks, presenting a significant limitation that must be addressed. Furthermore, simulation software, including OpenBVE, inherently terminates operation at the point of failure, precluding any assessment or management of post-failure scenarios. This highlights a critical gap in current approaches to railway automation, underscoring the need for future research into the physical and operational evaluation of failures in autonomous systems.

As with any work regarding the delineation of labor to autonomy, assignment of legal liability must be considered when discussing any possibility of fault. Ultimately the complexity of the topic shows incomplete and misguided reasoning about where responsibility lies, however, as far as the law is currently concerned liability is rarely assigned solely to the driver due to the integrated nature of railway systems, where faults can stem from signalling, track infrastructure, or vehicle defects. As a result, Railway operators or infrastructure managers are usually held accountable under EU and national laws, supported by insurance and indemnity provisions. [19] Additionally, a recent report by the European Commission regarding product liability has stated that autonomous vehicles are not currently treated in Union legislation any differently from non-autonomous vehicles insofar as defects are concerned, [4] suggesting that the current position would be to assign variable liability to every separate legal person involved in an incident, avoiding any assignment of strict liability. The issue of liability remains complex due to the relative infancy of autonomy as a field. Nevertheless, it is a critical topic that requires thorough discussion and analysis, which has been explored in separate research which the authors have contributed to. [26].

5 Acknowledgements

This work is undertaken with support from Siemens Mobility, and in conjunction with the the Fraunhofer Institute for Cognitive Systems (IKS) in Munich. This work was undertaken with the cooperation of the Associated Society of Locomotive Engineers and Firemen (ASLEF). Thank you to the OpenBVE team for their co-operation on the development of the associated plugins and overall modifications to the software.

References

1. Brandenburger, N., Naumann, A.: On Track: A Series of Research about the Effects of Increasing Railway Automation on the Train Driver. IFAC-PapersOnLine **52**(19), 288–293 (2019). https://doi.org/10.1016/j.ifacol.2019.12.115, https://linkinghub.elsevier.com/retrieve/pii/S2405896319319500
2. BVETeam: OpenBVE - Master (Oct 2024), https://github.com/leezer3/OpenBVE
3. BVETeam: OpenBVE Project - Scraper Plugin (Sep 2024), https://github.com/leezer3/OpenBVE/tree/ScoreAdder
4. COMMISSION, E.: REPORT FROM THE COMMISSION TO THE EUROPEAN PARLIAMENT, THE COUNCIL AND THE EUROPEAN ECONOMIC AND SOCIAL COMMITTEE Report on the safety and liability implications of Artificial Intelligence, the Internet of Things and robotics (Feb 2020)
5. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An open urban driving simulator. In: Conference on robot learning. pp. 1–16. PMLR (2017)
6. Dwyer, B., Nelson, J., Hansen, T.: Roboflow Version 1.0 (2024)
7. Faghih-Roohi, S., Hajizadeh, S., Núñez, A., Babuska, R., De Schutter, B.: Deep convolutional neural networks for detection of rail surface defects. In: 2016 International Joint Conference on Neural Networks (IJCNN). pp. 2584–2589 (2016). https://doi.org/10.1109/IJCNN.2016.7727522
8. Gibert, X., Patel, V.M., Chellappa, R.: Deep Multitask Learning for Railway Track Inspection. IEEE Transactions on Intelligent Transportation Systems **18**(1), 153–164 (Jan 2017). https://doi.org/10.1109/TITS.2016.2568758
9. Hamilton, W.I., Clarke, T.: Driver performance modelling and its practical application to railway safety. Applied Ergonomics **36**(6), 661–670 (Nov 2005). https://doi.org/10.1016/j.apergo.2005.07.005, https://linkinghub.elsevier.com/retrieve/pii/S0003687005001122

10. Hong, W.T., Clifton, G., Nelson, J.D.: Railway accident causation analysis: Current approaches, challenges and potential solutions. Accident Analysis & Prevention **186**, 107049 (Jun 2023). https://doi.org/10.1016/j.aap.2023.107049, https://linkinghub.elsevier.com/retrieve/pii/S0001457523000969
11. Hunter, J.: Railway Simulator, Bayesian DOT file (Nov 2024), https://github.com/TortoiseFather/SACRED/blob/main/Step-6/TrainSM.dot
12. Hunter, J.: Railway Simulator, State machines (Nov 2024), https://github.com/TortoiseFather/SACRED/tree/main/Step-6/img
13. Hunter, J., BVETeam: OpenBVE Data Scraper (Feb 2024), https://github.com/TortoiseFather/OpenBVE-ML-Plugins
14. Hunter, J., McDermid, J.: Investigating Human Error Within GoA-2 Metro Lines. In: Collart-Dutilleul, S., Haxthausen, A.E., Lecomte, T. (eds.) Reliability, Safety, and Security of Railway Systems. Modelling, Analysis, Verification, and Certification, vol. 13294, pp. 179–191. Springer International Publishing, Cham (2022). https://doi.org/10.1007/978-3-031-05814-1_13, https://link.springer.com/10.1007/978-3-031-05814-1_13, series Title: Lecture Notes in Computer Science
15. Hunter, J., McDermid, J., Burton, S.: Safety Analysis of Autonomous Railway Systems: An Introduction to the SACRED Methodology (Mar 2024), http://arxiv.org/abs/2403.12114, arXiv:2403.12114 [cs]
16. Lees, C., Hunter, J.: RE: BVE Cornwall Contact: BVE metric generation (Sep 2024)
17. Li, J., Li, Z., Wen, C., Peng, Q., Huang, P.: Train operation conflict detection for high-speed railways: a naïve Bayes approach. International Journal of Rail Transportation **11**(2), 188–206 (Mar 2023). https://doi.org/10.1080/23248378.2022.2071346, https://www.tandfonline.com/doi/full/10.1080/23248378.2022.2071346
18. Liu, B., Ma, X., Tan, E., Ma, Z.: Passenger flow anomaly detection in urban rail transit networks with graph convolution network–informer and Gaussian Bayes models. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences **381**(2254), 20220253 (Sep 2023). https://doi.org/10.1098/rsta.2022.0253, https://royalsocietypublishing.org/doi/10.1098/rsta.2022.0253
19. lntergovernmental Organisation for, International Carriage by Rail: COTIF Appendix E (Jan 2015)
20. Luther, R., Livingstone, H., Gipson, T., Grimes, E.: Understanding driver route knowledge. In: People and Rail Systems, pp. 95–102. CRC Press (2016)
21. Mauri, A., Khemmar, R., Decoux, B., Haddad, M., Boutteau, R.: Lightweight convolutional neural network for real-time 3D object detection in road and railway environments. Journal of Real-Time Image Processing **19**(3), 499–516 (Jun 2022). https://doi.org/10.1007/s11554-022-01202-6, https://link.springer.com/10.1007/s11554-022-01202-6
22. Medossi, G.: Handbook of optimization in the railway industry. Springer Science+Business Media, New York, NY (2018)
23. Mittal, S., Rao, D.: Vision Based Railway Track Monitoring using Deep Learning (Nov 2018), arXiv:1711.06423 [cs]
24. Moloney, W.: Applying Simulation Techniques to Train Railway Traction Drivers, Section 6.5. Ph.D. thesis, Birmingham, COLLEGE OF ENGINEERING AND PHYSICAL SCIENCES SCHOOL OF ENGINEERING DEPARTMENT OF CIVIL ENGINEERING (Sep 2017), https://etheses.bham.ac.uk/id/eprint/8370/1/Moloney18PhD.pdf

25. Panou, K., Tzieropoulos, P., Emery, D.: Railway driver advice systems: Evaluation of methods, tools and systems. Journal of Rail Transport Planning & Management **3**(4), 150–162 (Nov 2013). https://doi.org/10.1016/j.jrtpm.2013.10.005, https://linkinghub.elsevier.com/retrieve/pii/S2210970613000218
26. Porter, Z., Ryan, P., Morgan, P., Al-Qaddoumi, J., Twomey, B., McDermid, J., Habli, I.: Unravelling Responsibility for Ai (2024). https://doi.org/10.2139/ssrn.4871675, https://www.ssrn.com/abstract=4871675
27. Read, G., Naweed, A., Salmon, P.: Complexity on the rails: A systems-based approach to understanding safety management in rail transport. Reliability Engineering & System Safety **188**, 352–365 (Aug 2019). https://doi.org/10.1016/j.ress.2019.03.038, https://linkinghub.elsevier.com/retrieve/pii/S0951832018311773
28. Schultheis, M., Rothkopf, C.A., Koeppl, H.: Reinforcement Learning with Non-Exponential Discounting (Dec 2022), http://arxiv.org/abs/2209.13413, arXiv:2209.13413 [cs]
29. Shi, L., Liu, Y., Zhang, Y., Liang, J.: Data-Driven Bayesian Network Analysis of Railway Accident Risk. IEEE Access **12**, 38631–38645 (2024). https://doi.org/10.1109/ACCESS.2024.3376590, https://ieeexplore.ieee.org/document/10468590/
30. Staino, A., Suwalka, A., Mitra, P., Basu, B.: Real-Time Detection and Recognition of Railway Traffic Signals Using Deep Learning. Journal of Big Data Analytics in Transportation **4**(1), 57–71 (Apr 2022). https://doi.org/10.1007/s42421-022-00054-7, https://link.springer.com/10.1007/s42421-022-00054-7
31. Sutton, R., Barto, A.: Reinforcement Learning; An introduction. MiT Press (2015)
32. Uzuka, T.: System of Systems in Railway. The Society of Instrument and Control Engineers **59**(12), 953–956 (2020)
33. Zendel, O., Murschitz, M., Zeilinger, M., Steininger, D., Abbasi, S., Beleznai, C.: RailSem19: A Dataset for Semantic Rail Scene Understanding. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 1221–1229. IEEE, Long Beach, CA, USA (Jun 2019). https://doi.org/10.1109/CVPRW.2019.00161, https://ieeexplore.ieee.org/document/9025646/
34. Zhang, X., Liu, Y., Wang, Y., Boularias, A.: Detect Everything with Few Examples (2024), https://arxiv.org/abs/2309.12969, _eprint: 2309.12969