J Intel Connect Veh

Research Article

Reinforcement Learning Agent Embedded in Roadside Units to Implement an Adaptive Variable Speed Limit System in a Cooperative Intelligent Transportation Systems Environment

Eléonore Fauchet, Pierre-Antoine Laharotte✉, Nour-Eddin El Faouzi

Cite this article: https://doi.org/10.26599/JICV.2026.9210104

ABSTRACT:

Freeway shockwaves remain a major source of recurrent congestion, yet most reinforcement learning (RL) approaches to variable speed limit (VSL) control remain theoretical, relying on fixed detectors, predefined congestion zones and full observability.

This paper introduces an RL agent embedded in a Roadside Unit (RSU) within a Cooperative Intelligent Transportation System. Operating solely through V2I communication, the agent collects speed and position data from Cooperative Awareness Messages, reconstructs traffic state as a space-time matrix built from sparse connected vehicle (CV) observations, and broadcasts speed limits, requiring no additional sensing infrastructure.

Training proceeds in three stages: exhaustive dataset generation in a microscopic simulator, reward selection and hyperparameter calibration on a learned surrogate model, and few-shot refinement in the simulator. The transfer operates along two axes: (i) increasing environment fidelity, and (ii) the release of the simulation seed, so that a policy acquired deterministically is refined under stochastic conditions in 400 episodes, against the 4,000 required from a random policy.

Across 30 seeds, the agent selects a consistent mitigation strategy from 30% CV penetration, where attenuation is partial; the physics-based benchmark performs better above 50%, while at full penetration the two are statistically indistinguishable, with lower inter-seed variance for the RL agent. Two coordinating RSUs with 2 km coverage manage a 10 km corridor.

KEYWORDS: Reinforcement Learning (RL), Variable Speed Limit (VSL), Connected Vehicles (CVs), Roadside Units (RSUs), Deep Q-Learning, Transfer.

Just Accepted

1 Introduction

Freeway shockwaves are among the most persistent and costly forms of recurrent congestion, leading to significant increases in travel time, fuel consumption, and crash risk (Zheng et al., 2010). Variable Speed Limit (VSL) systems have long been recognised as an effective tool for damping shockwave propagation by harmonising speeds upstream of congestion (Hegyi et al., 2008; Papageorgiou et al., 2008). Yet translating VSL strategies into deployable, adaptive systems remains an open challenge, particularly as transportation infrastructure transitions toward connected and cooperative environments.

This paper investigates how Reinforcement Learning (RL) approaches to VSL must be redesigned (and how their performance is challenged) when confronted with two requirements that field deployment imposes: (i) the controller must be a decentralised agent embedded in a connected Roadside Unit (RSU); and (ii) the learnt strategy must transfer gradually toward the field, across a hierarchy of environments of increasing fidelity. We deliberately rely on a standard RL architecture. Consequently, our contribution lies not in a new learning algorithm but (i) in the agent design process composed of state, action, and especially an interpretable reward, and (ii) in a structured training pipeline that transfers knowledge from a lightweight surrogate model to a full microscopic traffic simulator (SUMO, (Lopez et al., 2018)) through few-shot refinement. Since direct field experimentation with exploratory RL controllers is neither safe nor affordable, we address here transferability as a staged, simulation-to-simulation transfer along two axes: (i) increasing model fidelity and (ii) the passage from a deterministic to a stochastic environment when the simulation seed is released. We position this as a prerequisite and a stepping stone toward sim-to-real transfer rather than as a demonstration of it.

1.1 RL-based VSL strategies: state of the art and limitations

Reinforcement Learning has emerged as a promising alternative to physics-based VSL control, and a comprehensive overview is proposed in (Kušić et al., 2020b). Most existing RL-VSL studies, however, remain theoretical in nature. In (Li et al., 2017; Li et al., 2020; Schmidt-Dumont & Van Vuuren, 2015; Vrbanić et al., 2021; Zhu & Ukkusuri, 2014), a single VSL agent controls its section using a finite set of states defined by discrete density values and a finite set of actions corresponding to a limited number of speed limit options. Furthermore, the VSL agent has prior information about congestion: both the congested areas and the section where the VSL will be applied are predefined. Some studies propose an adjustment of the pre-defined zones, but it remains a choice between only two possible configurations (Kušić et al., 2021a). Alternatively, the VSL area can only be shifted incrementally, by one cell per control step, and remains anchored to the detected jam (Han et

EMob-Lab, École Nationale des Travaux Publics de l'État, Université Gustave Eiffel, Université de Lyon, Bron 69675 Cedex, France.

✉ Corresponding author. E-mail: pierre-antoine.laharotte@univ-eiffel.fr

Received: July 24, 2026; Revised: August 28, 2026; Accepted: September 19, 2026

© The Author(s) 2026. This is an open access article under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0, http://creativecommons.org/licenses/by/4.0/).

al., 2022). These approaches either rely on fixed detectors and communication infrastructure or integrate data from Connected Vehicles (CVs), yet provide limited discussion on real-world implementation.

Some studies address more complex problems by considering larger and continuous state spaces (Kušić et al., 2018; Walraven et al., 2016), or by implementing multi-agent formulations (Kušić et al., 2020a, 2021a, 2021b; Wang et al., 2019). However, limited attention is given to the transferability of RL agents to real-world contexts. Agents are developed and tested within traffic simulators; from macroscopic ones such as CTM (Li et al., 2017, 2019) or METANET (Han et al., 2022) to microscopic ones, such as SUMO (Vrbanić et al., 2021). No explicit strategy to bridge the gap to field deployment is provided. (Han et al., 2022) represents a notable step forward by proposing both offline and online training, but a structured sim-to-real transfer methodology remains lacking in the RL-VSL literature.

1.2 Traffic state estimation from connected vehicle data

A key challenge when deploying RL-VSL systems in connected environments is reconstructing traffic state from sparse, incomplete observations. Connected Vehicles generate floating car data (FCD) whose spatial and temporal density depends critically on the Market Penetration Rate (MPR). Several works have addressed traffic state estimation from FCD using model-based approaches (Seo et al., 2017) and deep learning methods (Shi et al., 2021). However, existing RL-VSL formulations do not explicitly address the challenge of building reliable state representations from partial CV observations across varying MPRs. The sensitivity of RL agent performance to data sparsity is rarely quantified, and its implications for action selection are seldom discussed.

1.3 Sim-to-real transfer and surrogate models in RL

Transfer learning and sim-to-real methodologies have received growing attention in the broader RL literature (Zhao et al., 2020), and are increasingly identified as the main bottleneck for RL-based traffic control (Han et al., 2023). Two ingredients of this literature are central here. First, curriculum learning (Bengio et al., 2009): exposing the agent to environments of gradually increasing complexity improves convergence and generalisation. Second, surrogate (or world) models: learning a lightweight approximation of the environment transition dynamics to train the agent at low computational cost, before refining it in the target environment — an idea at the core of model-based RL (Moerland et al., 2023). A specific risk of this staged approach is catastrophic forgetting during fine-tuning in the higher-fidelity environment (Kirkpatrick et al., 2017). To the best of our knowledge, the systematic integration of these ingredients (*i.e.*, surrogate-based pre-training, curriculum-style staging and few-shot refinement) into a reproducible training pipeline for RL-VSL has not been explored, in a literature where training procedures are often insufficiently reported to be reproduced (Pineau et al., 2021).

1.4 Deployment constraints of RSU-embedded control in C-ITS architectures

Embedding controllers in the RSUs of a C-ITS architecture introduces constraints absent from most RL-VSL studies. In a V2I framework, each RSU perceives only its local communication zone, and coordination between RSUs must be achieved through message passing rather than centralised state sharing. Existing multi-agent RL approaches to VSL (Kušić et al., 2020a, 2021b; Wang et al., 2019) generally assume centralised training or shared global state information, which is inconsistent with these constraints. How independent, RSU-embedded agents with limited coverage can coordinate using only locally available and communicated information remains an open question, as does the sensitivity of the resulting system to RSU placement (Bhattacharyya et al., 2024).

It is useful, in this respect, to distinguish three paradigms in the RL-VSL literature. In the first, a single centralised agent observes the whole corridor through fixed detectors and controls all VSL gantries. In the second, multi-agent formulations distribute control at execution time but rely on centralised training or on a shared global state, and therefore assume information that an isolated RSU does not have. In the third, control is fully decentralised under communication constraints: each agent perceives only its own communication zone and can exchange information only through standardised ITS messages. The present work belongs to this third category, which is, to the best of our knowledge, the least explored, and which imposes the specific combination of constraints addressed in the remainder of this paper. Table 1 positions the proposed approach against representative studies along five dimensions: sensing source, state representation, definition of the action zone, coordination scheme, and transfer strategy.

Study	Sensing source	State representation	Definition of the action zone	Coordination scheme	Transfer strategy
Zhu & Ukkusuri (2014)	**Connected-vehicle environment assumed** (link-level flow reported through V2I); no detector layout modelled	Four discrete congestion levels per link (density thresholds at 0.25, 0.50 and 0.75 of jam density)	**Predefined**: one speed limit per pre-selected link (371 links), 20–80 mph	**One independent agent per link**; no explicit coordination (localised optimisation, acknowledged by the authors as a limitation)	**None**: learning and implementation phases on the same stochastic network
Walraven et al. (2016)	**Fixed** detectors (section speed and density, METANET)	Continuous: normalised section speeds and densities and the two previous speed limits, optionally augmented with one-step-ahead predictions; tile coding or neural approximation	**Predefined**: a single speed limit applied to a fixed block of sections upstream of the on-ramp	**Single agent** over the whole VSL area	**Zero-shot transfer**: policy learnt on METANET and executed on a different microscopic (SUMO) network, with robustness to measurement noise and partial compliance assessed

Just Accepted

Study	Sensing source	State representation	Definition of the action zone	Coordination scheme	Transfer strategy
Li et al. (2017)	**Fixed** detectors (30 s loop-detector data, PeMS)	Three discretised densities (downstream of the merge, upstream mainline, on-ramp); 4,851 tabular states	**Predefined**: fixed controlled section and acceleration section at a recurrent merge bottleneck	**Single agent**	Continual offline re-learning: policy trained on a calibrated CTM, rewards then re-estimated from field data and the Q-table periodically updated
Wang et al. (2019)	**Connected vehicles**: RSUs aggregate per-vehicle position and speed, limits are pushed back to vehicles, no VMS	Continuous, approximated by k-NN-TD state centres; mobility policy on downstream density, safety policy on the speed difference with the downstream segment	**Predefined**: fixed equidistant segments and controller locations; limits in 30–100 km/h with a 10 km/h step cap	**Distributed Q-learning** with a shared global state (central traffic data centre) and a shared averaged reward; the optimistic assumption replaces explicit messaging	**None**
Kušić et al. (2020a)	**Fixed** detectors, VMS-based	Discretised mean speeds of the two controlled sections and density of the bottleneck section (225 states)	**Predefined** and static: two 1 km zones upstream of the bottleneck; joint action is a pair of limits with a 20 km/h gradient cap	**Multi-agent W-learning**: implicit competition among policies through W-values, no explicit communication	Cross-demand reuse only: policy learnt on dynamic demand and executed on static-demand scenarios
Kušić et al. (2021a)	**Fixed** detectors and tightly spaced VMS; V2I (ISA) assumed for deployment	Speeds (4 intervals) and densities (12 intervals) of the local and neighbouring sections, and the previous action	**Predefined** but incrementally adjustable: the agent selects the speed limit jointly with one of two admissible zone configurations per section	**Distributed W-learning**: local and remote policies fed with neighbouring states and rewards (message passing), weighted by a cooperation coefficient	**None**: separate learning and exploitation phases
Kušić et al. (2021b)	**Fixed** detectors and tightly spaced VMS; CAV-oriented deployment	Idem (Kušić et al. (2021a)), with an extended analysis of the state discretisation	**Predefined** but incrementally adjustable (idem - Kušić et al. (2021a))	**Distributed W-learning** with a sweep over cooperation levels; centralised training assumed	**None**: a broader set of non-stationary demand scenarios is evaluated without re-learning
Vrbanić et al. (2021)	**Connected and automated vehicles** used as mobile actuators, in mixed flow at varying penetration rates	Discretised segment density and speed, tabular Q-learning	**Predefined** and static zone	**Single agent**	**None**: the policy is re-learnt for each penetration rate
Han et al. (2022)	**Fixed detectors** (loop detectors delimiting the cells) and VMS	Inflow to the VSL area, mean density of the VSL area, jam length, mean jam speed and index of the jam head cell (discretised)	**Predefined** but incrementally adjustable: the upstream boundary of the VSL area moves by at most one cell per control step and remains anchored to the detected jam	**Single agent** over a homogeneous stretch	**Iterative offline–online training:** synthetic transitions from an extended CTM are progressively replaced by process data, with SPECIALIST records as the initial training set
Our study	**Connected vehicles only**: CAMs received by the RSU within its coverage radius; no loop detector, radar or additional roadside sensing	**Continuous space–time** matrix of per-cell average CV speeds (5 × 24 cells over 6 km and 300 s) used directly as DQN input; unobserved cells are sentinel-encoded and, under reduced coverage, completed by a learnt reconstruction model	**Free**: one limit per subsection drawn from {50, 70, 90, 110} km/h, with no prior assumption on the location of congestion and the option of imposing no restriction; the only constraint is a 20 km/h gradient between adjacent subsections (68 of 256 combinations retained)	**Fully decentralised**: independent RSU agents, no centralised training and no global state; cooperation through standardised ITS message passing, with exchange of CAM-aggregated per-cell speeds in overlapping cells and of the selected limits, the most restrictive value being applied in shared subsections	**Staged simulation-to-simulation transfer**: SUMO dataset generation for reward selection, pre-training and hyperparameter calibration on a surrogate transition model, then few-shot refinement in SUMO at a reduced learning rate

Table 1. Positioning of the proposed approach against representative RL-VSL studies, along five dimensions: sensing source, state representation, definition of the action zone, coordination scheme, and transfer strategy.

1.5 Contributions of this paper

Our work introduces an RL agent embedded within a Cooperative ITS architecture as part of a Roadside Unit (RSU). Equipped with hardware

Just Accepted

for computation and RL execution, the RSU-agent interacts exclusively through ITS communication: (i) it collects CV data via Cooperative Awareness Messages (CAMs) within its perception radius, (ii) it computes traffic indicators, (iii) it makes decisions, and (iv) it disseminates speed limits to CVs through ITS messages. We do not claim a theoretical contribution to RL methodologies but focus on the agent's modelling process and deployment-oriented design. To appropriately model the continuous state space, we use Deep Q-Learning (DQL), which approximates Q-values using a deep neural network.

Our contributions are fivefold:

(i) **State construction from sparse CV data**: traffic states are reconstructed exclusively from CV average speed and location measurements via a space-time diagram representation. No additional fixed sensing infrastructure is required.

(ii) **Location-independent action-space construction**: the agent is free to apply speed limits across any subsection without predefined congestion zone assumptions, including the choice to impose no restriction. The only constraint retained is a safety constraint on the speed-limit gradient between adjacent subsections, so the freedom concerns the location of the intervention rather than the values themselves.

(iii) **Systematic reward formulation**: an iterative, dataset-driven methodology for selecting and calibrating the reward function, validated against the full combinatorial strategy space.

(iv) **Gradual knowledge transfer via a surrogate model**: pre-training in a simplified, deterministic environment followed by few-shot refinement in a microscopic simulator with a released seed, so that the transfer spans both environment fidelity and environment stochasticity, establishing a structured pathway toward field deployment.

(v) **Multi-RSU evaluation under realistic C-ITS constraints**: assessment of decentralised, communicating RSUs with reduced coverage areas, partial state reconstruction using a deep learning model, and coordinated action sharing.

The remainder of this paper is organised as follows. Section 2 details the RL-VSL agent formulation. Section 3 describes the calibration and training methodology. Sections 4 and 5 present performance results for single- and multi-RSU configurations, respectively. Section 6 concludes with perspectives for future work.

2 Modelling Formulation of our RL-VSL Agent

This section details the formulation of the RL-VSL problem in a Cooperative ITS (C-ITS) environment. The proposed system relies exclusively on a single type of infrastructure, namely the Roadside Unit (RSU), which serves as a sensing node, a computation unit, and a communication gateway. This architectural choice is motivated by the objective of minimising deployment costs and infrastructure heterogeneity, a key consideration for real-world C-ITS rollout.

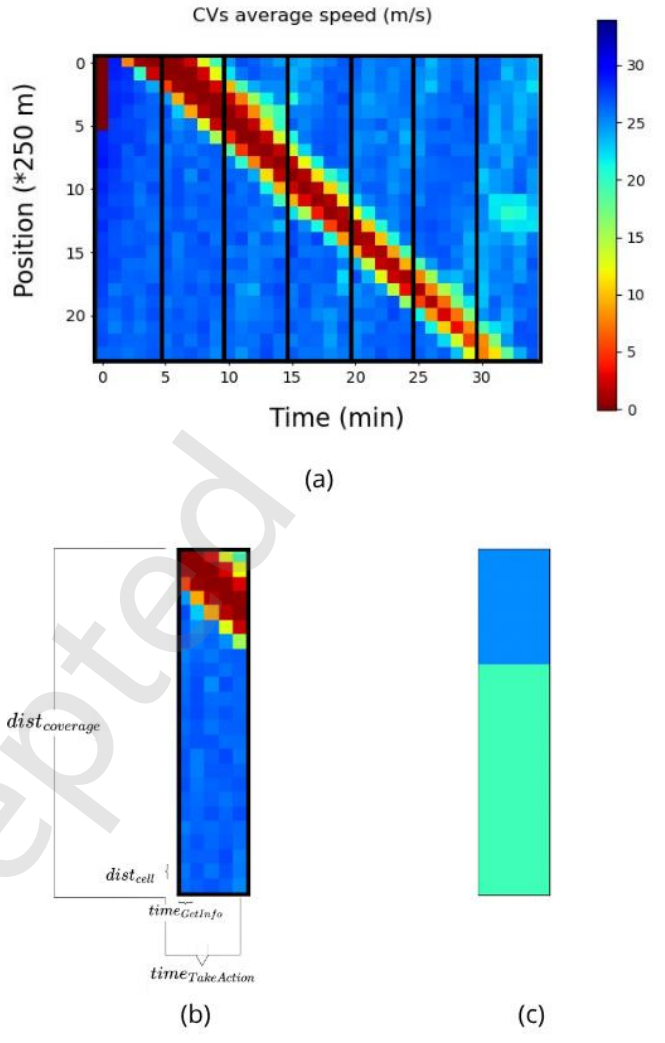


Fig. 1. Illustration of one RL agent under training: (a) baseline environment; (b) state illustration, (c) action illustration

2.1 System architecture and communication model

The RSU agent operates within a V2I communication framework compliant with the ETSI ITS-G5 standard (European Telecommunications Standards Institute (ETSI), 2019a). Within its coverage area, which spans a road length $dist_{coverage}$, the RSU passively receives Cooperative Awareness Messages (CAMs) broadcast periodically by equipped Connected Vehicles (CVs). Each CAM stores the vehicle's instantaneous speed and position, which serve as the sole input to the agent's state estimation process. No fixed loop detectors, radar sensors, or additional infrastructure are required.

At regular intervals (every 20 s in the released implementation), the RSU updates its internal representation of the traffic state using all CAMs received since the last update. Every $time_{GetInfo}$ seconds, these observations are consolidated into a new column of the state matrix and, every $time_{TakeAction}$ seconds, the RSU computes a new action and broadcasts the resulting speed limit instructions to CVs within its coverage area via In-Vehicle Information Messages (IVIMs) (ETSI, 2020) or Decentralised Environmental Notification Messages (DENMs) (ETSI, 2019b). The relationship between the sub-step and

Just Accepted

the decision step is given by Eq. (1).

$time_{TakeAction} = k \cdot time_{GetInfo}$, with $k \in \mathbb{N}^*$ (Eq. (1))

A decision step of 5 minutes is chosen, consistent with field implementation standards and allowing drivers sufficient time to adjust their speed before the state is reassessed and a new reward computed. In the released implementation, the RSU aggregates the CAMs it has received every 20 s, consolidates them into a state row every $time_{GetInfo}$ = 60 s, and issues a new action every time_TakeAction = 300 s, i.e., k = 5 sub-steps per decision step.

2.2 State space construction from sparse CV observations

At each decision step t, the RSU constructs a state matrix S_t of shape: S_t_shape (see Eq. (2)).

$$S_t\text{_shape} = \frac{dist_{coverage}}{dist_{cell}} * \frac{time_{TakeAction}}{time_{GetInfo}} \quad \text{(Eq. (2))}$$

Each cell [i, j] of S_t stores the average speed of CVs observed in spatial cell i during temporal sub-step j. This space-time matrix representation provides the agent with a structured, two-dimensional view of traffic dynamics within its coverage area, capturing both the spatial propagation and temporal evolution of congestion. With dist_coverage = 6 km, $dist_{cell}$ = 250 m, time_TakeAction = 300 s and $time_{GetInfo}$ = 60 s, the state matrix has 5 rows and 24 columns, i.e., 120 cells, which is the input dimension of the network described in Section 2.4.

A critical feature of this formulation is its robustness to data sparsity. Since state values are derived exclusively from CV observations, cell occupancy depends directly on the CV Market Penetration Rate (MPR). At low MPRs, many cells may be empty or statistically unreliable. Rather than imputing missing values with a fixed assumption (e.g. free-flow speed), the agent is trained to make decisions under this uncertainty, i.e., a design choice that contributes to the system's resilience across a wide range of MPRs, as evaluated in Section 4. For reproducibility, the numerical encoding is stated explicitly: cells for which no CAM has been received during the corresponding sub-step are set to a fixed sentinel value of -1, and no imputation from neighbouring cells or from previous decision steps is performed. In particular, this value is not the free-flow speed, so that an unobserved cell is not presented to the network as an uncongested one.

This approach departs from most existing RL-VSL formulations, which either assume full observability via fixed detectors or rely on density-based state representations that require calibrated sensor networks (Kušić et al., 2020b; Li et al., 2017). It is instead closer in spirit to floating car data (FCD) based traffic state estimation (Seo et al., 2017), extended here into the RL decision-making loop.

2.3 Action space construction

An action a_t is defined as a vector of speed limit values, one per RSU subsection, whose shape is detailed in Eq. (3).

$$a_t\text{_shape} = \frac{dist_{coverage}}{length_{subsection}} \quad \text{(Eq. (3))}$$

Each element of a_t is drawn from a discrete set of admissible speed limits {50, 70, 90, 110} km/h in the full training phase, reduced to {70, 90} km/h during calibration. Crucially, no prior assumption is made about the location of congestion relative to the RSU's subsections: the agent is free to apply any speed limit (including the free-flow speed, *i.e.,* no restriction) to any subsection. This contrasts with most prior work, where the spatial extent of the VSL intervention is predefined based on known congestion locations (Han et al., 2022; Kušić et al., 2021a).

To ensure road safety and driver compliance, a constraint is imposed: two spatially adjacent subsections cannot be assigned speed limits differing by more than 20 km/h. This rule is consistent with the graduated speed limit transitions applied in operational VSL schemes such as SPECIALIST (Hegyi et al., 2008), and mirrors the 20 km/h step-size cap prescribed for VSL on French motorways (Glasson et al., 2021). With four subsections and four admissible speed limit values, the unconstrained action space contains $4^4 = 256$ combinations, of which 68 satisfy this constraint and are retained as the admissible action set in the complete training phase.

2.4 Deep Q-Learning formulation

To handle the continuous and high-dimensional state space described above, we adopt the Deep Q-Learning (DQL) algorithm (Mnih et al., 2015), which approximates the action-value function Q(S, a) using a deep neural network. DQL is chosen over actor-critic methods (e.g., PPO, SAC) because the action space, although large, remains discrete and finite after safety filtering. This is a setting where value-based methods have demonstrated highly competitive performance (Hessel et al., 2018).

The network architecture, adapted from (Vidali et al., 2019) for traffic signal control and retained here after ablation during the calibration phase (Section 3), comprises:

- an input layer of N = 120 neurons, with $N = S_t$_shape;
- 5 hidden layers of 400 neurons each, with ReLU activation;
- an output layer of M neurons with linear activation, where M = size of the admissible action set, each neuron encoding the Q-value of one action given the current state.

Training follows an epsilon-greedy exploration strategy with experience replay. The replay buffer stores transition tuples (S_t, a_t, r_t, S_{t+1}), with buffer size and batch size calibrated as described in Section 3. The network is optimised with Adam on the mean squared temporal-difference error. Q-targets are computed with the same online network rather than with a separate target network, which departs from the original DQN formulation (Mnih et al., 2015); the stabilising role usually played by a target network is here fulfilled by the size of the replay buffer relative to the number of transitions collected per iteration, and by the single gradient pass performed at each replay step.

3 Calibration and training of the RL-VSL agent based on a gradual knowledge transfer

3.1 Overview and motivation

The calibration and training of an RL agent for VSL control

Just Accepted

raises two intertwined challenges that are frequently underreported in the literature: (i) the formulation of a reward function that faithfully encodes the operational objective, and (ii) the selection of hyperparameters that ensure stable convergence. These two tasks are typically addressed sequentially in existing works, often with minimal methodological detail (Kušić et al., 2020b). This lack of reproducibility is a recognised limitation of applied RL research (Pineau et al., 2021).

We address both challenges through a structured three-stage methodology (Fig. 2) that progressively increases environment complexity, inspired by curriculum learning principles (Bengio et al., 2009). Learning begins in a computationally lightweight surrogate environment and is progressively transferred to a full microscopic simulator. This gradual knowledge transfer reduces the risk of premature convergence to suboptimal policies during early exploration, and establishes a pathway toward real-world deployment, where direct exploration would be unsafe and costly (Zhao et al., 2020).

The simulation scenario used throughout is a 3-lane homogeneous freeway where a downstream boundary blockage of 280 seconds, applied after a 7-minute warm-up period at 70% capacity (2130 veh/h/lane), generates a persistent shockwave propagating upstream at approximately 20 km/h (Fig. 1a). This scenario is deliberately kept constant across all training stages. The freeway is modelled as a 10 km three-lane link with a lane speed of 130 km/h. Vehicles follow the Krauss car-following model with a driver imperfection of 0.99, a minimum gap of 2 m and a desired speed of 108 km/h. The demand consists of 3200 vehicles released over the first 1800 s, *i.e.*, approximately 2130 veh/h/lane. The bottleneck is caused by capping vehicle speeds on the downstream link after a 7-minute warm-up period. Speed limits are transmitted only to connected vehicles, and equipped vehicles are assumed to comply fully with the instructions they receive. Since the object of the study focuses on the effects of environment fidelity (surrogate model versus microscopic simulator) and of observation sparsity on the learnt policy, holding the scenario fixed allows isolation from scenario variability. The counterpart of this choice, discussed in Section 6, is that the generalisation of the agent to other geometries, incident types and demand profiles is not established here.

3.2 Step 1 — SUMO dataset generation

To build the surrogate model and evaluate reward functions exhaustively, a simplified action space is first defined: speed limit values are restricted to {70, 90} km/h, subsection length is set to 2 km over a 6 km coverage area, yielding actions of the form $[v_1, v_2, v_3] \in \{70, 90\}^3$. The action set is further restricted to 4 combinations: {[70,70,90], [70,70,70], [90,90,90], [90,70,70]}, chosen to represent qualitatively distinct strategies (full restriction, partial upstream restriction, no restriction, downstream restriction). These yield $4^6 = 4096$ possible strategies for the 6 decision steps in one simulation episode.

All 4096 strategies are evaluated exhaustively in SUMO using a fixed random seed to eliminate stochasticity. Each simulation generates a sequence of transition triplets (S_t, a_t, S_{t+1}), assembling a dataset of $4096 \times 6 = 24\,576$ transitions. The fixed seed ensures that differences in outcomes are attributable solely to the action sequence, rather than to stochastic vehicle distributions, which is essential for reliable reward-function comparison in Step 2. The fixed seed therefore acts as a control for the paired comparison of the 4096 action sequences and not as a measure of performance; replicating this exhaustive comparison over n seeds would have required 4096 × n SUMO runs.

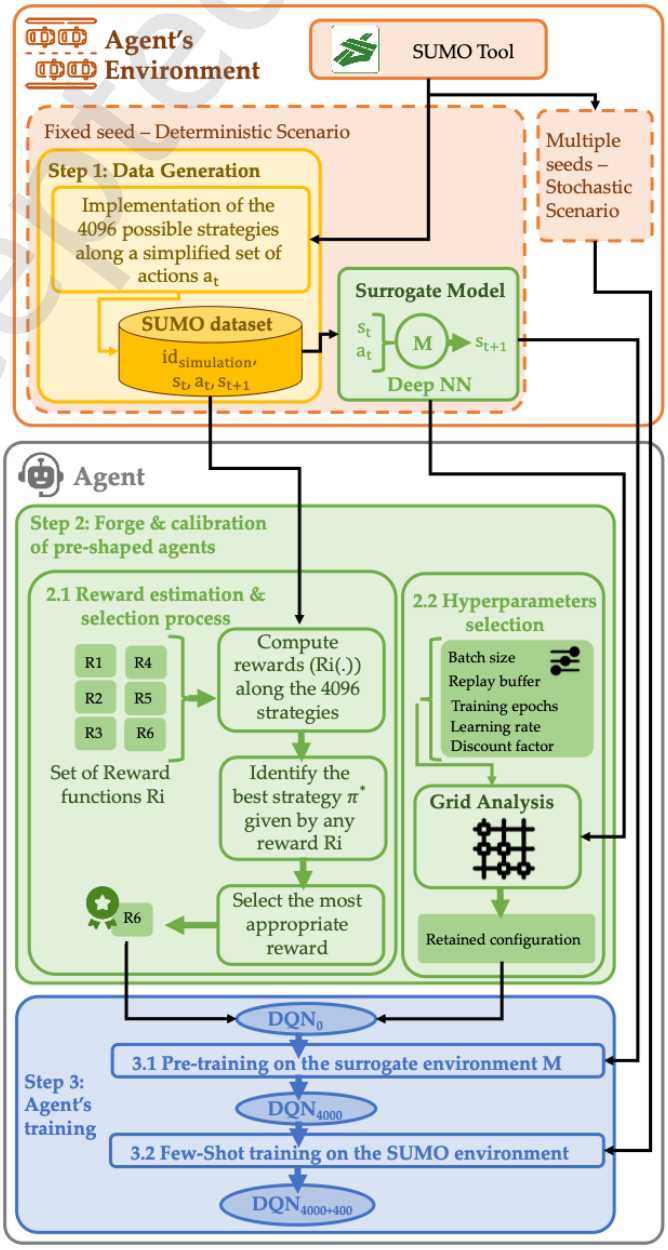


Fig. 2. Illustration of the calibration and training steps of the RL-VSL agent

3.3 Step 2.1 — Reward function selection

Just Accepted

A well-designed reward function must simultaneously encourage shockwave dissipation and penalise unnecessary speed restrictions downstream of congestion (two objectives that are easy to conflate). Using the exhaustive SUMO dataset, we evaluate six reward formulations (Fig. 3) by identifying, for each, the action sequence that achieves the

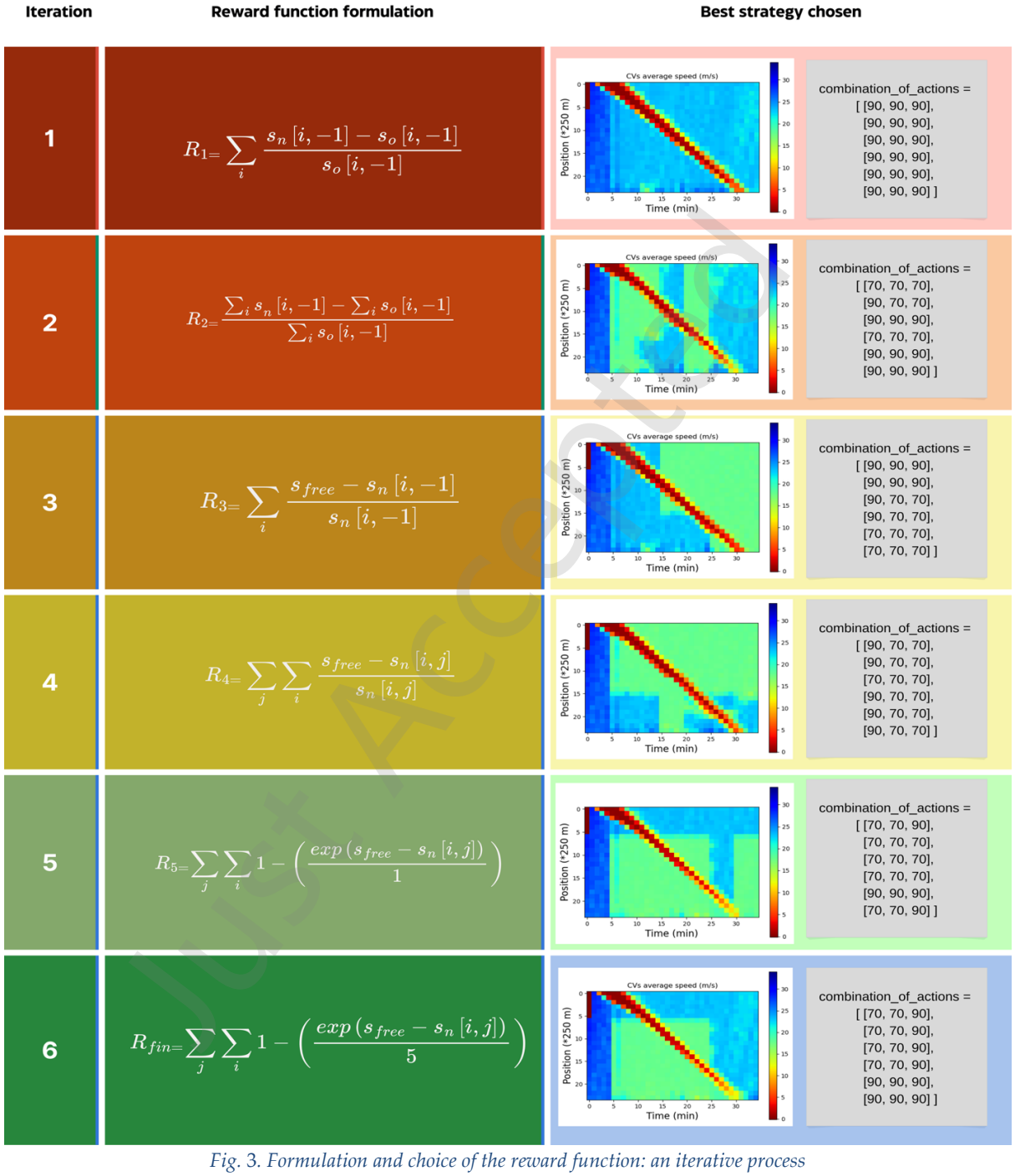


Fig. 3. Formulation and choice of the reward function: an iterative process

highest cumulative reward, and assessing whether it corresponds to the operationally optimal strategy. This comparison constitutes an ablation study of the reward function: the six candidate formulations cover the two conventional families used in the RL-VSL literature, namely improvement with respect to the previous state (R1, R2) and deviation from a free-flow reference (R3, R4). The selection criterion is not the numerical value of the reward but the operational coherence of the argmax action sequence, i.e., whether it restricts upstream of the congestion front and refrains from restricting downstream of it. The failures of R1 to R5 identified below are therefore structural properties of the formulations rather than artefacts of one stochastic realisation.

The six formulations are analysed as follows:

R1 compares, cell by cell, the oldest column of the previous

Just Accepted

state matrix to the most recent column of the new state matrix. The optimal strategy under R1 is [90,90,90] — i.e., doing nothing — because the natural dissipation of the shockwave between time steps produces apparent speed improvements regardless of the action taken. R1 is therefore confounded by temporal autocorrelation in the traffic state.

R2 replaces the spatial comparison with a global average, partially correcting R1's bias. The selected strategy applies some upstream restrictions and partially reduces the shockwave. However, R2 does not penalise speed limits applied unnecessarily downstream of the congestion front.

R3 and **R4** compare current average speeds to free-flow speed across all cells of the new state matrix, providing a reference-anchored formulation. R3 selects a strategy that applies restrictions exclusively downstream of the shockwave (the inverse of the desired behaviour). R4 improves upon R3 by computing the ratio cell-by-cell rather than aggregating across cells, but it still fails to discriminate between upstream and downstream restrictions.

R5 introduces an exponential penalty for deviations from free-flow speed. This formulation effectively identifies upstream restrictions as beneficial, and successfully reduces the shockwave. However, it still applies unnecessary restrictions downstream.

R6 adjusts the exponential scaling factor. The larger denominator softens the penalisation for mild speed reductions, making the reward less sensitive to free-flow speed fluctuations downstream. Under R6, the selected optimal strategy maximises upstream restriction while avoiding downstream penalisation — precisely aligned with the operational objective. R6 is therefore retained as the reward function for all subsequent training stages. The retained formulation is given explicitly in Eq. (4). It is computed at the cell level, as a double sum over all spatial cells i and all temporal sub-steps j of the current state matrix. $R_6 = \sum_j \sum_i 1 - exp(\frac{s_{free} - s_t[i,j]}{5})$ (Eq. (4)), where s_t[i,j] is the average speed of the connected vehicles observed in cell (i, j), expressed in m/s, and s_free = 30 m/s the free-flow speed. Cells for which no observation is available carry the sentinel value −1 and therefore contribute a strongly negative term, which discourages the agent from ignoring unobserved regions. The deviation is therefore measured against a fixed free-flow reference rather than against the previous state, thereby removing the temporal autocorrelation bias identified for R1. No spatial weighting is applied, with all cells contributing equally.

This iterative, dataset-driven reward selection process constitutes a reproducible methodology for reward engineering in RL-VSL applications, addressing a gap identified in the literature (Eschmann, 2021; Kušić et al., 2020b).

3.4 Step 2.2 — Surrogate model construction and hyperparameter calibration

To avoid running SUMO for each hyperparameter configuration, which would be computationally prohibitive, we train a surrogate model M that approximates the environment transition function: $S_{t+1} \cong M(S_t, a_t)$. The surrogate is implemented as a deep neural network trained on the Step 1 dataset using Mean Absolute Error (MAE) as the loss function. Its architecture and training settings are reported in Table 2: the network maps a 124-dimensional input, concatenating the 120 cells of the current state matrix and the four speed limit values of the action, onto the 120 cells of the next state matrix, through one hidden layer of 124 ReLU units and one of 128 tanh units. Inputs and outputs are standardised to zero mean and unit variance using coefficients computed on the training split only, and the dataset is divided into 70 % for training and 30 % for validation. Training converges in approximately 12 minutes to MAE = 0.039 on standardised speeds (zero mean, unit variance; the per-cell standardisation coefficients are stored alongside the model). This MAE can be expressed back in physical units as a mean absolute error of approximately 1.0 km/h per cell, confirming that the surrogate captures the dominant dynamics of SUMO's traffic response with high fidelity.

Characteristic	DQL policy network	Transition surrogate M	State reconstruction
Purpose	Q-value estimation	S_t+1 from S_t and a_t	Full 6 km state from partial observation
Section	2.4	3.4	5.2
Input dimension	120	124 (120 state + 4 action)	120 (masked 6 km grid)
Hidden layers	5 x 400, ReLU	124 ReLU, then 128 tanh	120 ReLU, then 128 tanh
Output layer	68, linear	120, linear	120, linear
Loss function	MSE on TD error	MAE	MAE
Optimiser	Adam	Adam, lr 2e-3 halved twice	Adam, lr 2e-3 halved twice
Target network	none (single online network)	n/a	n/a
Input / output scaling	none (raw m/s, -1 if unobserved)	z-score	z-score
Training epochs	100 replay steps / iteration	up to 5000, early stopping	Up to 5000, early stopping
Train / validation split	-	70 / 30	70/30
MAE (standardised)	-	0.039	0.1389
MAE (physical units)	-	approx. 1.0 km/h per cell	approx. 3.7 km/h per cell
Batch size	512	256	256
Stored weights	trained_model_<ep>.h5	full_coverage_6 km/best_model _2D.h5	reduced_cover age_2km/best_ model_2D.h5

Table 2. Architecture and training settings of the neural networks used in the study. The same reconstruction architecture is instantiated for the 6 km and the 2 km coverage cases. Standardised MAE values are converted to physical units using the per-cell standard deviations stored with each model.

Just Accepted

Hyperparameter calibration is then performed by training the DQL agent on the surrogate rather than SUMO, reducing the cost of each training run and making the exploration reported in Fig. 4 feasible. Conducting the same grid search directly in the microscopic simulator was beyond our computational budget. It is worth stressing that this is an argument of feasibility for the calibration phase only. The contribution claimed for the surrogate at the training stage is of a different nature and is discussed in Section 3.5. The parameters explored are: batch size and replay buffer limits (linked to action set cardinality), training epochs (balancing computation time and learning quality), learning rate (starting high and decreasing), and discount factor γ (controlling the weight of long-term rewards). As shown in Fig. 4, only a small subset of configurations achieves convergence toward the optimal cumulative reward of approximately −17,552 (compared to a baseline of around −20,000 for most strategies). The retained configuration is: learning rate = 0.01, training epochs = 100, γ = 0.7. The full set of environment, interface and learning parameters is reported in Table 3, for the calibration phase and for the two training stages respectively, so that the experiments can be reproduced from the companion repository. The preference for γ = 0.7 over γ = 0.8 is confirmed by direct comparison in SUMO: γ = 0.7 converges to the optimal policy, whereas γ = 0.8 does not, suggesting that placing excessive weight on distant future rewards is counterproductive in this scenario, even though the shockwave evolves over multiple decision steps.

Parameter (symbol, unit)	Calibration	Pre-training	Refinement
RSU coverage, dist_coverage (m)	6000	6000	6000
RSU position on the 10 km link (m)	7000	7000	7000
Spatial cell size, dist_cell (m)	250	250	250
CAM polling interval (s)	20	20	20
Sub-step, time_GetInfo (s)	60	60	60
Decision step, time_TakeAction (s)	300	300	300
State matrix shape (rows x cols)	5 x 24 = 120	5 x 24 = 120	5 x 24 = 120
Unobserved-cell sentinel value	-1	-1	-1
Subsection length (m)	2000	1500	1500
Action vector shape	3	4	4
Speed limit set (km/h)	70, 90	50, 70, 90, 110	50, 70, 90, 110
Admissible action set size	4	68	68
Hidden layers x width	5 x 400, ReLU	5 x 400, ReLU	5 x 400, ReLU
Batch size	512	512	512
Replay buffer, min / max	12000 / 24000	12000 / 24000	60 / 6000
Replay steps per iteration	100	100	100
Learning rate	0.1, 0.01, 0.001	0.01	0.0001
Optimiser	Adam	Adam	Adam
Loss function	MSE	MSE	MSE
Discount factor, gamma	0.7, 0.8, 0.9, 0.99	0.7	0.7
Total episodes	6000	4000	400
Parallel simulations per iteration	10	10	10
Epsilon schedule	0.01 + 0.99 exp(-4e-4 x ep.)	0.01 + 0.99 exp(-4e-4 x ep.)	0.3 initial
Weights saved every	100 episodes	100 episodes	100 episodes

Table 3. Environment, interface and learning parameters of the RL-VSL agent. Comma-separated values denote the set explored during calibration; the retained configuration is learning rate = 0.01, 100 replay steps per iteration, gamma = 0.7.

3.5 Step 3 — Agent training with gradual knowledge transfer

Step 3.1 — Pre-training on the surrogate model.

With the reward function and hyperparameters fixed, the agent is trained in the full action space: speed limits ∈ {50, 70, 90, 110} km/h, length_subsection = 1.5 km, action_shape = 4, yielding 68 admissible actions after safety filtering. Training on the surrogate over 4,000 episodes converges to a cumulative reward of −12,500. Episodes are run in batches of 10 parallel simulations. After each batch, 100 replay steps of 512 transitions are drawn from a buffer holding between 12 000 and 24 000 transitions, and the exploration rate follows an exponential schedule decaying from 1 to 0.01. The optimal strategy found at episode 4,000 successfully resolves the shockwave in the surrogate environment.

Step 3.2 — Few-shot refinement in SUMO.

The pre-trained model is transferred to SUMO and refined through few-shot training. Several constraints are relaxed at this stage: the simulation seed is no longer fixed, introducing stochastic vehicle distributions and thus a wider range of observed states. The learning rate is lowered to 10^{-4}, two orders of magnitude below the value used for pre-training, and the replay buffer is reduced to 6,000 transitions so that refinement is driven by recently observed states. Furthermore, exploration is limited to avoid destructive re-exploration, which is a known risk in few-shot fine-tuning where the agent may unlearn behaviours acquired during pre-training (Kirkpatrick et al., 2017).

The refinement stage runs 400 episodes under the same batching scheme, *i.e.*, 40 training iterations, which makes it a few-shot stage relative to the 4,000 episodes of pre-training, and the cumulative reward improves from −12,500 to −5,000 over that budget. This stage combines two transfers that are worth separating. The first is a transfer of fidelity, from the learned surrogate to the microscopic simulator. The second, and in our view the more consequential, is a transfer from a deterministic to a stochastic environment. Every earlier stage, including the exhaustive dataset generation in Step 1 and the

Just Accepted

calibration in Step 2, operates with a fixed seed, whereas the seed is released here. The agent is therefore exposed to vehicle distributions, and hence to interactions, that it has never encountered during pre-training. This second axis is the closest analogue, within our pipeline, to what a controller would face when moved from a simulated environment to a field deployment, and it is the reason we present the pipeline as a stepping stone toward sim-to-real transfer rather than as a computational shortcut. The benefit of pre-training is accordingly measured in episodes rather than in computing time. At a fixed seed, training the same agent from a random policy directly in the microscopic simulator required 4,000 episodes to converge. Starting from the pre-trained policy, 400 episodes suffice to improve the cumulative reward while additionally absorbing seed variability, i.e., one tenth of the budget for a strictly harder problem. We deliberately refrain from claiming that the pipeline outperforms end-to-end training at an equal computational budget, because the two procedures do not attempt the same task. Indeed, end-to-end training under a released seed would have to discover the control structure and absorb the stochasticity of the environment simultaneously, and whether a value-based agent converges at all under these conditions remains an open question.

A key limitation of the current implementation is the absence of explicit mechanisms to prevent catastrophic forgetting during the SUMO refinement phase (Kirkpatrick et al., 2017). Future work will explore elastic weight

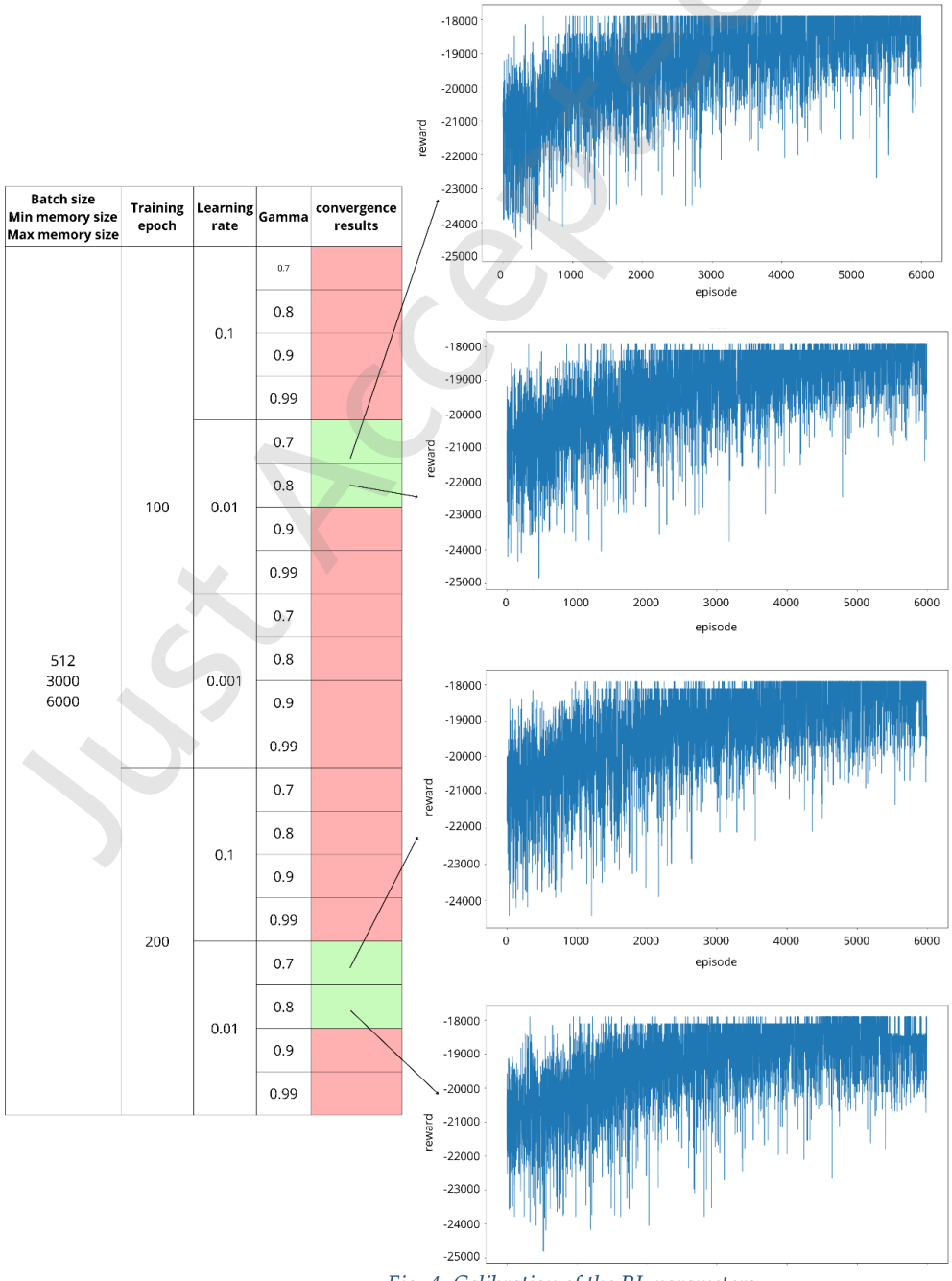


Fig. 4. Calibration of the RL parameters

Just Accepted

consolidation (EWC) or progressive network architectures to address this.

Just Accepted

4 Performance analysis — single RSU, large coverage area

4.1 Experimental setup

The trained agent is evaluated over 30 independent random seeds at MPR ∈ {0%, 10%, 20%, 30%, 50%, 75%, 100%}, keeping all other scenario parameters unchanged. While the scenario itself is held constant, the evaluation is not deterministic: the 30 seeds generate independent stochastic vehicle distributions, and seven penetration rates are tested. Performance is assessed using two Key Performance Indicators (KPIs):

Total Time Spent (TTS) (vehicle-hours): the aggregate travel time of all vehicles in the network, capturing the overall efficiency impact of the VSL strategy.
Coefficient of Variation of Travel Time (Cv-TT): the ratio of standard deviation to mean individual travel time, measuring the equity of the system's performance across vehicles — a dimension often neglected in VSL evaluations.

Statistical comparisons between the RL-based and Full CV-based VSL systems are performed using the Wilcoxon signed-rank test across the 30 seeds, with a significance threshold of $\alpha = 0.05$. An implementation of the Full CV-based VSL system (Fauchet et al., 2022), an adaptation of the SPECIALIST algorithm (Hegyi et al., 2008) for connected-vehicle data, serves as the primary benchmark. It represents the state of the art in physics-based, CV-enabled VSL control and provides a meaningful upper bound for comparison. Please note that this version of the Full CV-based SPECIALIST is fed with a state reconstruction per cell, similar to the original Eulerian approach, but restricted by the MPR of CVs.

4.2 Results and discussion

Qualitative analysis.

Space-Time Diagrams for a representative seed (seed 25) at MPR = 0%, 30%, and 100% are presented in Fig. 5. At MPR = 0%, no CV data are available, and the agent correctly applies free-flow speed limits throughout (effectively doing nothing, as expected). At MPR = 30% and MPR = 100%, the agent selects the same strategy: a progressive upstream speed reduction that intercepts the shockwave before it fully develops. At MPR = 30%, the shockwave is attenuated but not fully eliminated: non-equipped vehicles tend to overtake CVs, particularly in the left lane, partially undermining the speed limit enforcement. This lane-differentiated compliance effect is consistent with findings on VSL control in mixed traffic flows (Bhattacharyya et al., 2025) and highlights the importance of evaluating VSL strategies across all lanes rather than on aggregated metrics alone.

Quantitative analysis.

Fig. 6 presents the distribution of TTS and Cv-TT across 30 seeds for both VSL systems at all tested MPRs. Key findings are as follows:

At MPR = 100%, the RL-based and Full CV-based systems show no statistically significant difference: TTS p-value = 0.061, Cv-TT p-value = 0.062, both above the $\alpha = 0.05$ threshold. The RL-based system exhibits lower inter-seed variance for both KPIs, suggesting greater robustness to stochastic vehicle distributions — an advantage for real-world deployment where traffic conditions are inherently variable.

At MPR ≥ 50%, the Full CV-based system achieves better mean KPIs. This is attributable to its explicit use of shockwave theory (SPECIALIST): with sufficient CV density, the physics-based system can precisely localise the shockwave front and apply targeted restrictions, whereas the RL agent relies on a learned approximation of traffic dynamics that may not generalise optimally to all shockwave configurations within this MPR range.

At MPR < 30%, both systems show degraded performance relative to the no-control baseline for some seeds, reflecting the inherent difficulty of VSL control with sparse observations. Notably, the RL-based system's variance is consistently lower than that of the Full CV-based system across all MPR levels, providing a meaningful advantage for practitioners seeking predictable performance guarantees.

Robustness of the state representation.

A notable result is that the same strategy is selected by the agent at both MPR = 30% and MPR = 100%, despite the substantially different data density. This insensitivity to MPR in strategy selection (rather than strategy effectiveness) suggests that the space-time matrix representation successfully abstracts the congestion pattern from the density of CV observations, at least above a threshold MPR. This is consistent with findings on FCD-based traffic state estimation, which report acceptable reconstruction accuracy at moderate penetration rates under favourable conditions (Seo et al., 2017).

5 Performance analysis — multiple RSUs, smaller coverage areas

5.1 Motivation and experimental design

The single-RSU configuration evaluated in Section 4 assumes a continuous 6 km coverage area — a setting that may not reflect real deployment constraints, where RSU placement is governed by infrastructure costs, right-of-way agreements, and existing roadside equipment. This section investigates whether the trained agent can be adapted to operate with RSUs of reduced coverage (2 km), and whether multi-RSU coordination can compensate for the resulting gaps in perception.

The experimental design considers a discontinuous communication layer with MPR = 100%, and progressively introduces four mechanisms: (i) reduced coverage with

Just Accepted

naive state extension, (ii) reduced coverage with DL-based state reconstruction, (iii) multi-RSU operation without coordination, and (iv) multi-RSU operation with state and action sharing. This incremental design allows the contribution of each mechanism to be isolated, providing a cleaner attribution of performance gains than a single end-to-end comparison. This campaign is run over 10 random seeds rather than the 30 used in Section 4, because it involves nine infrastructure configurations, each combining the reconstruction model inference loop with the coordination mechanisms. The seeds used should be a subset of those of Section 4, so that the two campaigns remain comparable. This sample does not support formal statistical inference, and no significance test is therefore reported in this section. The results are discussed comparatively, in terms of the ordering of the configurations and of the presence or absence of shockwave resolution in the space-time diagrams.

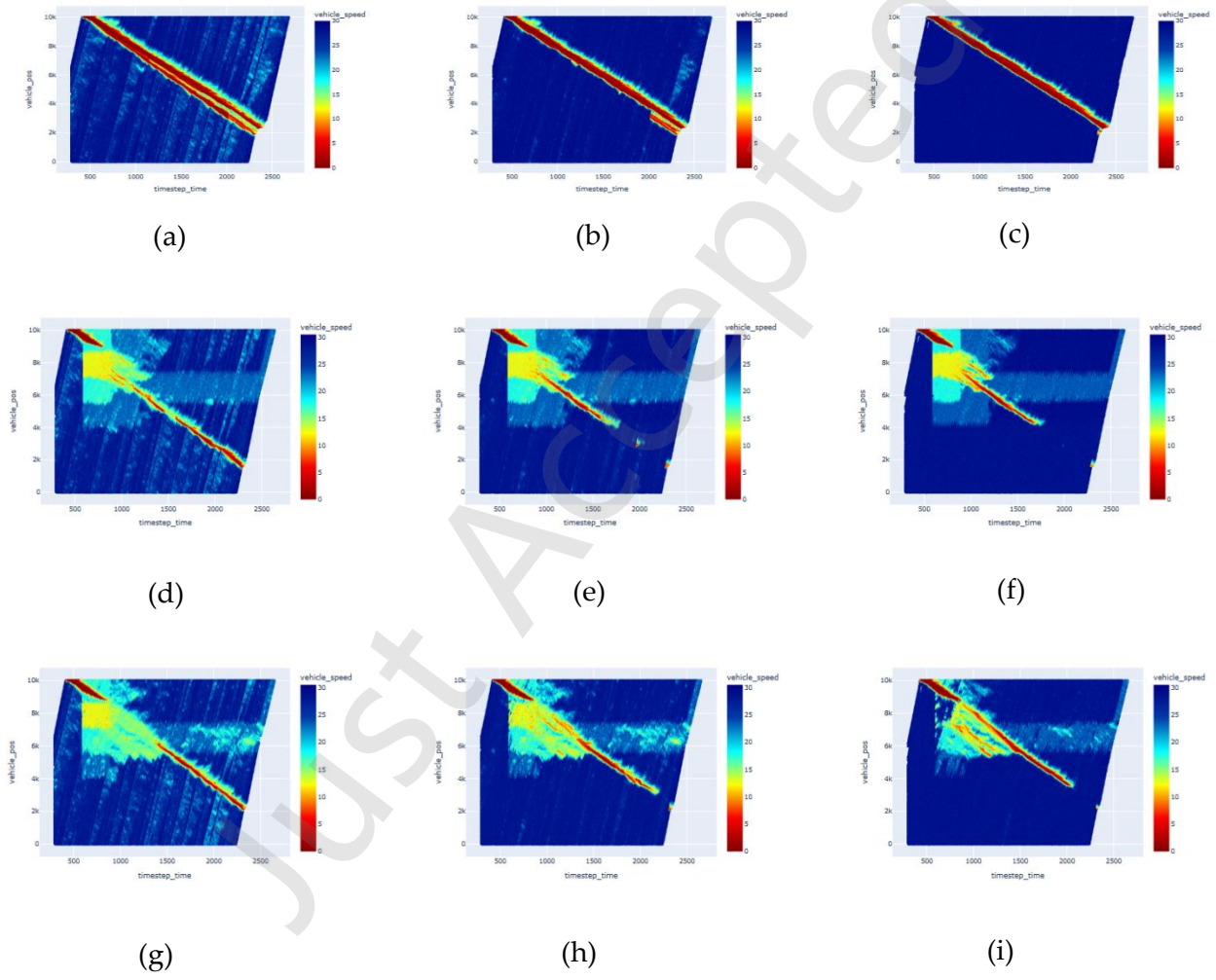


Fig. 5. Space-Time Diagram (seed 25) without control (TTS=497.4h and Cv-TT=0.46) in the right lane (a), middle lane (b), left lane (c); for RL-based VSL - MPR=100% (TTS=458.1h and Cv-TT=0.27) in the right lane (d), middle lane (e), left lane (f); for RL-based

Just Accepted

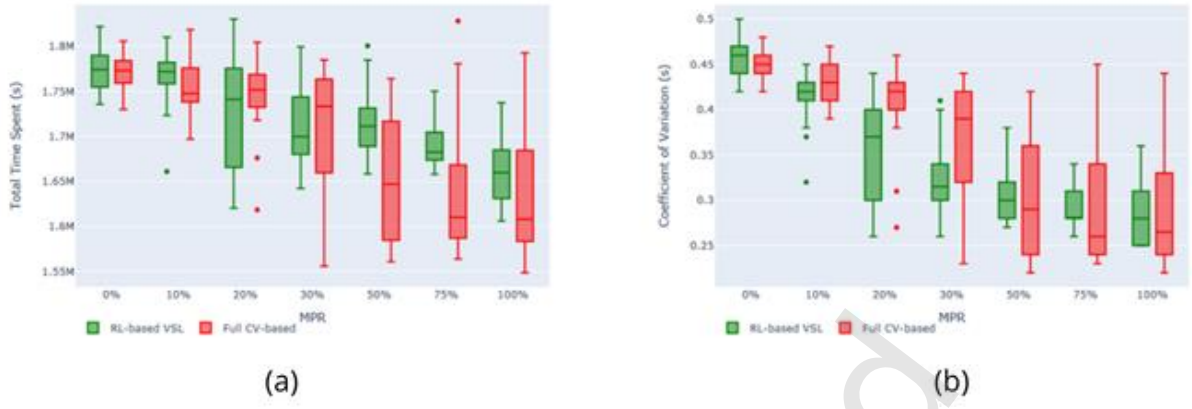
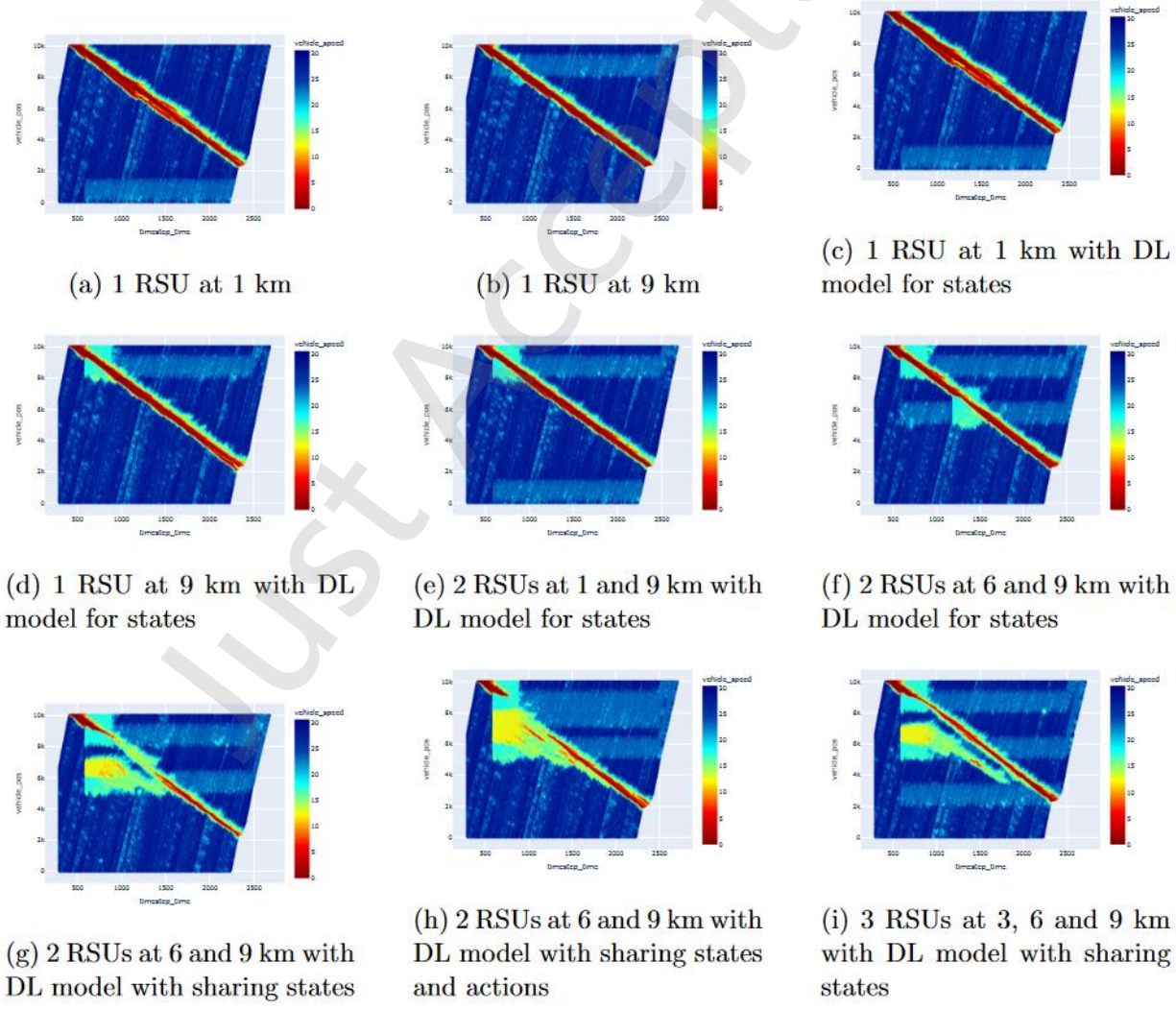


Fig. 6. Full-CV_based and RL-based VSL systems performances comparison (a) TTS, (b) Cv-TT



(a) 1 RSU at 1 km

(b) 1 RSU at 9 km

(c) 1 RSU at 1 km with DL model for states

(d) 1 RSU at 9 km with DL model for states

(e) 2 RSUs at 1 and 9 km with DL model for states

(f) 2 RSUs at 6 and 9 km with DL model for states

(g) 2 RSUs at 6 and 9 km with DL model with sharing states

(h) 2 RSUs at 6 and 9 km with DL model with sharing states and actions

(i) 3 RSUs at 3, 6 and 9 km with DL model with sharing states

Fig. 7. Right Lane Space-Time Diagram for Scenario: MPR = 100% (seed 9)

Just Accepted

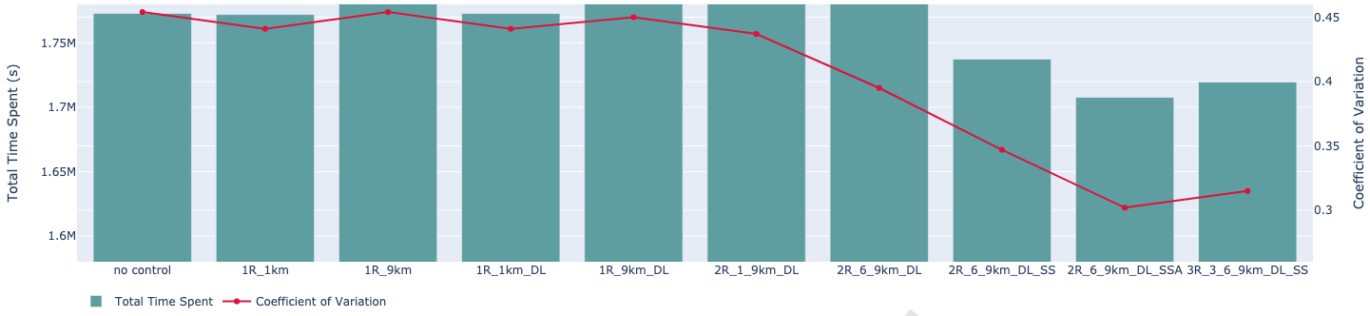


Fig. 8. Results for 10 seeds and the various configurations

5.2 Single RSU with reduced coverage

Naive state extension.

The trained agent, designed for a 6 km state matrix, is tested with a 2 km RSU by padding the unobserved cells with free-flow speed values. The agent systematically selects free-flow speed limits and fails to resolve the shockwave (Figs. 7a–b, Fig. 8). This result confirms that the agent's policy is sensitive to the absolute values in the state matrix: free-flow padding creates a misleading state representation that the agent interprets as an absence of congestion, regardless of what is observed within the 2 km window.

DL-based state reconstruction.

To address this limitation, a deep learning model (architecturally similar to the surrogate model of Section 3) is trained to predict the full 6 km state matrix from partial observations (2 or 4 km coverage), using the SUMO dataset from Step 1. Its architecture and training settings are reported in Table 2, alongside those of the surrogate model. A separate instance of the same architecture is trained for the 2 km coverage case. The achieved MAE = 0.1389 is expressed on the same standardised scale as the surrogate and corresponds to approximately 3.7 km/h per cell, so that extrapolating unobserved cells is roughly three to four times less accurate than predicting the next state from a fully observed one. Despite this, the reconstructed states enable the agent to apply stronger speed restrictions in congested zones (Figs. 7c–d). However, shockwaves remain unresolved, confirming that a single RSU with 2 km coverage (even with state reconstruction) is insufficient to control a 10 km freeway section. These two configurations constitute an ablation of the state representation. The same trained agent, on identical scenarios, is fed either a state matrix padded with free-flow values or one completed by the reconstruction model, and the resulting policy changes qualitatively. This establishes that it is the informative content of the space-time matrix, and not merely its dimensions, that drives the decision.

5.3 Multiple RSUs without coordination

Two RSUs are deployed at kilometre positions {1, 9} and {6, 9}, each using the DL-based state reconstruction independently. Both configurations produce suboptimal results (Figs. 7e–f, Fig. 8), with the {6, 9} placement performing marginally better. This placement advantage is attributable to the proximity of the km 6 RSU to the typical shockwave formation zone: better coverage of the congestion front improves reconstruction accuracy and yields more targeted restrictions. This finding has a practical implication for infrastructure planning: RSU placement optimisation (even without coordination) has a measurable impact on VSL effectiveness (Bhattacharyya et al., 2024).

5.4 Multiple RSUs with state sharing

RSUs within each other's extended perception range exchange, for the spatial cells located in the overlap of their coverage areas, the per-cell average speeds aggregated from the CAMs they have received; these enriched observations are then fed to each RSU's own reconstruction model. Reconstructed state matrices are not exchanged, so that each RSU remains responsible for its own state estimate and no reconstruction error is propagated between agents. This cooperative perception mechanism is consistent with the collective perception service defined in the ETSI C-ITS standard (ETSI, 2019c). With state sharing enabled, the shockwave is nearly resolved for the {6, 9} configuration (Fig. 7g, Fig. 8). The residual inefficiency is attributed to uncoordinated speed limit application: the two RSUs may independently select conflicting restrictions for the shared spatial zone, creating spatial discontinuities in the VSL profile that reduce driver compliance and shockwave attenuation.

5.5 Multiple RSUs with state and action sharing

To resolve the coordination gap identified in Section 5.4, RSUs are equipped with an action-sharing mechanism: each RSU broadcasts its currently selected speed limits, and for any subsection covered by multiple RSUs, the most restrictive value is applied. This conservative coordination rule ensures spatial consistency of the VSL profile while

Just Accepted

remaining implementable within the existing ITS messaging infrastructure. With this mechanism enabled, the {6, 9} configuration achieves near-complete shockwave resolution (Figs. 7h–i, Fig. 8), with TTS and Cv-TT approaching the single-RSU large-coverage benchmark. The addition of a third RSU at km 3 provides only a marginal improvement, suggesting that two well-placed, coordinating RSUs constitute a near-optimal configuration for the tested scenario.

These results demonstrate that, despite relying on (i) a small number of communicating RSUs, (ii) a single communication protocol, and (iii) no fixed sensing infrastructure, a distributed, infrastructure-light VSL system can achieve performance comparable to a fully connected single-RSU system, provided that both state reconstruction and action coordination are implemented.

6 Conclusions

This paper introduced an RL-based VSL system embedded in RSUs within a C-ITS architecture, designed to mitigate freeway shockwaves using only CV-derived observations and V2I communication. The system requires no fixed sensing infrastructure, operates under a discontinuous communication layer, and is designed to be deployable with a minimal number of RSUs.

Three main results were demonstrated.

First, a structured three-stage training methodology yields an agent whose design is deployment-oriented and which generalises across MPRs and scenario seeds within the tested scenario. The transfer it organises is twofold: from a lightweight surrogate to a microscopic simulator, and from a deterministic setting to a stochastic one. Starting from the pre-trained policy, 400 refinement episodes suffice with a released seed, compared with the 4,000 episodes required to converge from a random policy at a fixed seed. The three-stage training process consists of (i) an exhaustive combination of reward function evaluation based on a SUMO dataset, (ii) hyperparameter calibration on a surrogate model, and (iii) few-shot refinement in SUMO. This gradual knowledge transfer approach addresses a recognised gap in the RL-VSL literature regarding the reproducibility and transferability of training procedures (Pineau et al., 2021).

Second, the space-time matrix state representation, built exclusively from sparse CV observations, proves robust to MPR variations in terms of strategy selection rather than of strategy effectiveness: the same control strategy is selected at MPR = 30% and MPR = 100%, with full shockwave mitigation at full penetration and partial attenuation at MPR = 30%. At full penetration, performance is statistically comparable to the SPECIALIST-based physics benchmark and offers the additional advantage of lower inter-seed variance. This is a meaningful property for practitioners seeking predictable system behaviour.

Third, the multi-RSU analysis demonstrates that two communicating RSUs with 2 km coverage and DL-based state reconstruction can effectively manage a 10 km corridor, provided that both perceived states and selected actions are shared. The action coordination rule, which applies the most restrictive speed limit in shared zones, is simple, robust, and compatible with existing ITS messaging standards.

Several limitations of the current study deserve attention. The simulation scenario is restricted to a single incident type on a homogeneous, straight freeway, and the trained agent has not been evaluated on more complex geometries (on-ramps, lane drops, variable demand profiles). The absence of explicit catastrophic forgetting prevention during SUMO fine-tuning (Section 3.5) remains an open issue that future work will address through elastic weight consolidation (Kirkpatrick et al., 2017) or progressive network architectures. The DL-based state reconstruction model introduces an additional failure mode that has not been fully characterised: its accuracy degrades at very low MPRs. Three further limitations of the evaluation should be noted. First, the reward function was selected on a single fixed seed, and the robustness of the ranking of R1 to R6 across seeds has not been quantified. The paper's objective lies in pointing out the selection process of the reward formulation based on the surrogate, more than on the selected formulation itself. Second, we have not established whether a value-based agent trained end-to-end in the microscopic simulator under a released seed converges at all; the comparison with our pipeline at an equal computational budget is therefore left open, since the two procedures do not attempt the same task. Third, the multi-RSU campaign relies on 10 seeds and is therefore discussed comparatively rather than with statistical significance. Finally, the transition from simulation to real-world deployment will require online adaptation mechanisms to account for model mismatch, sensor and positioning noise in CAM data, communication losses, non-stationarity of traffic demand, and partial driver compliance with the broadcast speed limits, which are assumed to be total in the present simulations. The surrogate-based training pipeline is designed to facilitate this transition, but it has not yet been tested against these challenges.

Future work will focus on: (i) extension to larger and more heterogeneous networks with variable demand; (ii) online learning mechanisms for field adaptation; (iii) integration of vehicle type heterogeneity (trucks, autonomous vehicles) into the state representation; and (iv) a formal safety analysis of the VSL instructions generated by the RL agent under adversarial traffic conditions.

Replication and data sharing

The replication package, comprising the SUMO simulation scripts, the surrogate model implementation and the trained DQL agents used in this study, is available on the ETS Data platform (https://ets-data.sciopen.com) under

Just Accepted

https://doi.org/10.26599/ETSD.2026.9190099. The code is also available at https://github.com/EMob-Lab/SimTransfer-RLbasedVSLTailoredToRSU.git.

Author contributions

Eléonore Fauchet: Writing – original draft, Validation, Software, Methodology, Formal analysis, Data curation, Conceptualization. Pierre-Antoine Laharotte: Writing – original draft, Writing – review & editing, Supervision, Formal analysis, Methodology, Conceptualization, Funding acquisition. Nour-Eddin El Faouzi: Writing – review & editing, Formal analysis, Methodology, Funding acquisition.

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

Declaration of generative AI and AI-assisted technologies in the writing process

The authors did not use AI or AI-assisted technologies during the preparation of the code and experiments. During the preparation of this manuscript, the authors used Claude (Sonnet 5.0) to polish the language and to correct and reword English sentences, particularly in the introduction, the conclusions, and the transitions between paragraphs. After using this tool, the authors thoroughly reviewed and edited the content as necessary and take full responsibility for the final content of the published article.

References

Bengio, Y., Louradour, J., Collobert, R., Weston, J., 2009. Curriculum learning. In: Proceedings of the 26th Annual International Conference on Machine Learning, 41–48.

Bhattacharyya, K., Laharotte, P.A., Fauchet, E., Blache, H., El Faouzi, N.E., 2024. Enhancing traffic efficiency and sustainability through strategic placement of roadside units and variable speed limits in a connected vehicle environment. Sustainability, 16, 7495.

Bhattacharyya, K., Laharotte, P.A., El Faouzi, N.E., 2025. Detection of dangerously slow vehicles on highway corridor in a partially connected environment. Data Science for Transportation, 7, 13.

Eschmann, J., 2021. Reward function design in reinforcement learning. In: Reinforcement Learning Algorithms: Analysis and Applications. Belousov, B., Abdulsamad, H., Klink, P., Parisi, S., Peters, J., Eds. Cham: Springer, 25–33.

European Telecommunications Standards Institute (ETSI), 2019a. Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service. ETSI EN 302 637-2 V1.4.1.

European Telecommunications Standards Institute (ETSI), 2019b. Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specification of Decentralised Environmental Notification Basic Service. ETSI EN 302 637-3 V1.3.1.

European Telecommunications Standards Institute (ETSI), 2019c. Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Analysis of the Collective Perception Service. ETSI TR 103 562 V2.1.1.

European Telecommunications Standards Institute (ETSI), 2020. Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Facilities Layer Protocols and Communication Requirements for Infrastructure Services. ETSI TS 103 301 V1.3.1.

Fauchet, E., Bhattacharyya, K., Laharotte, P.A., El Faouzi, N.E., 2022. A variable speed limit strategy based on Lagrangian traffic indicators resulting from connected vehicles: Application to the SPECIALIST algorithm. In: Proceedings of the Transportation Research Board 101st Annual Meeting.

Glasson, P., Murard, F., Pelé, N., 2021. Régulation dynamique des vitesses. Projets de gestion dynamique du trafic, Fiche n°05. Cerema. https://doc.cerema.fr/Default/doc/SYRACUSE/21544/projets-de-gestion-dynamique-du-trafic-fiche-n-05-regulation-dynamique-des-vitesses

Han, Y., Hegyi, A., Zhang, L., He, Z., Chung, E., Liu, P., 2022. A new reinforcement learning-based variable speed limit control approach to improve traffic efficiency against freeway jam waves. Transportation Research Part C: Emerging Technologies, 144, 103900.

Han, Y., Wang, M., Leclercq, L., 2023. Leveraging reinforcement learning for dynamic traffic control: A survey and challenges for field implementation. Communications in Transportation Research, 3, 100104.

Hegyi, A., Hoogendoorn, S.P., Schreuder, M., Stoelhorst, H., Viti, F., 2008. SPECIALIST: A dynamic speed limit control algorithm based on shock wave theory. In: Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems, 827–832.

Hessel, M., Modayil, J., van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., et al., 2018. Rainbow: Combining improvements in deep reinforcement learning. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, 3215–3222.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., et al., 2017. Overcoming catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences, 114, 3521–3526.

Kušić, K., Ivanjko, E., Gregurić, M., 2018. A comparison of different state representations for reinforcement learning based variable speed limit control. In: Proceedings of the 26th Mediterranean Conference on Control and Automation, 1–6.

Kušić, K., Dusparic, I., Guériau, M., Gregurić, M., Ivanjko, E., 2020a. Extended variable speed limit control using multi-agent reinforcement learning. In: Proceedings of the 23rd IEEE International Conference on Intelligent Transportation Systems, 1–8.

Kušić, K., Ivanjko, E., Gregurić, M., Miletić, M., 2020b. An overview of reinforcement learning methods for variable

Just Accepted

speed limit control. Applied Sciences, 10, 4917.
Kušić, K., Ivanjko, E., Vrbanić, F., Gregurić, M., Dusparic, I., 2021a. Dynamic variable speed limit zones allocation using distributed multi-agent reinforcement learning. In: Proceedings of the IEEE International Intelligent Transportation Systems Conference, 3238–3245.
Kušić, K., Ivanjko, E., Vrbanić, F., Gregurić, M., Dusparic, I., 2021b. Spatial-temporal traffic flow control on motorways using distributed multi-agent reinforcement learning. Mathematics, 9, 3081.
Li, Z., Liu, P., Xu, C., Duan, H., Wang, W., 2017. Reinforcement learning-based variable speed limit control strategy to reduce traffic congestion at freeway recurrent bottlenecks. IEEE Transactions on Intelligent Transportation Systems, 18, 3204–3217.
Li, Z., Zhu, X., Liu, X., Qu, X., 2019. Model-based predictive variable speed limit control on multi-lane freeways with a line of connected automated vehicles. In: Proceedings of the IEEE Intelligent Transportation Systems Conference, 1989–1994.
Li, Z., Xu, C., Guo, Y., Liu, P., Pu, Z., 2020. Reinforcement learning-based variable speed limits control to reduce crash risks near traffic oscillations on freeways. IEEE Intelligent Transportation Systems Magazine, 13, 64–70.
Lopez, P.A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.P., Hilbrich, R., et al., 2018. Microscopic traffic simulation using SUMO. In: Proceedings of the 21st International Conference on Intelligent Transportation Systems, 2575–2582.
Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., et al., 2015. Human-level control through deep reinforcement learning. Nature, 518, 529–533.
Moerland, T.M., Broekens, J., Plaat, A., Jonker, C.M., 2023. Model-based reinforcement learning: A survey. Foundations and Trends in Machine Learning, 16, 1–118.
Papageorgiou, M., Kosmatopoulos, E., Papamichail, I., 2008. Effects of variable speed limits on motorway traffic flow. Transportation Research Record, 2047, 37–48.
Pineau, J., Vincent-Lamarre, P., Sinha, K., Larivière, V., Beygelzimer, A., d'Alché-Buc, F., et al., 2021. Improving reproducibility in machine learning research (a report from the NeurIPS 2019 reproducibility program). Journal of Machine Learning Research, 22, 1–20.
Schmidt-Dumont, T., Van Vuuren, J.H., 2015. Decentralised reinforcement learning for ramp metering and variable speed limits on highways. IEEE Transactions on Intelligent Transportation Systems, 14, 1–10.
Seo, T., Bayen, A.M., Kusakabe, T., Asakura, Y., 2017. Traffic state estimation on highway: A comprehensive survey. Annual Reviews in Control, 43, 128–151.
Shi, R., Mo, Z., Di, X., 2021. Physics-informed deep learning for traffic state estimation: A hybrid paradigm informed by second-order traffic models. In: Proceedings of the 35th AAAI Conference on Artificial Intelligence, 540–547.
Vidali, A., Crociani, L., Vizzari, G., Bandini, S., 2019. A deep reinforcement learning approach to adaptive traffic lights management. In: Proceedings of the 20th Workshop "From Objects to Agents", 42–50.
Vrbanić, F., Ivanjko, E., Mandžuka, S., Miletić, M., 2021. Reinforcement learning based variable speed limit control for mixed traffic flows. In: Proceedings of the 29th Mediterranean Conference on Control and Automation, 560–565.
Walraven, E., Spaan, M.T.J., Bakker, B., 2016. Traffic flow optimization: A reinforcement learning approach. Engineering Applications of Artificial Intelligence, 52, 203–212.
Wang, C., Zhang, J., Xu, L., Li, L., Ran, B., 2019. A new solution for freeway congestion: Cooperative speed limit control using distributed reinforcement learning. IEEE Access, 7, 41947–41957.
Zhao, W., Queralta, J.P., Westerlund, T., 2020. Sim-to-real transfer in deep reinforcement learning for robotics: A survey. In: Proceedings of the IEEE Symposium Series on Computational Intelligence, 737–744.
Zheng, Z., Ahn, S., Monsere, C.M., 2010. Impact of traffic oscillations on freeway crash occurrences. Accident Analysis and Prevention, 42, 626–636.
Zhu, F., Ukkusuri, S.V., 2014. Accounting for dynamic speed limit control in a stochastic traffic environment: A reinforcement learning approach. Transportation Research Part C: Emerging Technologies, 41, 30–47.

Just Accepted

Author biographies



Dr. Eléonore Fauchet received the Master's degree in 2021 from ENTPE, University of Lyon, Vaulx-en-Velin, France, and the Ph.D. degree in engineering from the same institution in 2025. She is currently a Traffic Engineer specialising in regulation and dynamic traffic simulation, within the mobility department of Cerema's Centre-Est division. Her research interests include mobility modelling, traffic flow regulation and control, intelligent transport systems, and connected vehicles.



Dr. Pierre-Antoine Laharotte is a researcher at the EMob-Lab (Energy and Mobility Lab) supported by the Université

Gustave Eiffel and ENTPE (Université de Lyon), Lyon, France. He received M.S. and Ph.D. Degrees respectively in 2013 and 2016 from the University of Lyon. He mainly works on Connected Vehicles and Intelligent Transport Systems (ITS), especially on applying machine learning and AI-based processes to address ITS challenges, from traffic state estimation to traffic control via Reinforcement Learning.



Pr. Nour-Eddin El Faouzi received his Ph. D. in Applied Mathematics from Montpellier University in 1992 and a Post-Doctoral Degree (HDR) in Traffic modelling from Lyon University in 2008. He has over 30 years of professional, academic, and research experience in traffic modelling and management, intelligent transportation systems, multimodal systems modelling and optimisation, and Data Analytics. He has served as principal investigator on various funded research projects sponsored by international, European, national, state, and metropolitan agencies, as well as private industry. Moreover, he is an Adjunct Professor at Queensland University of Technology (QUT), Australia.

He was the Deputy Director of the Components and Systems Department at Gustave Eiffel University, the former Head of the Transportation and Traffic Engineering Laboratory (LICIT), and the former Chair of the European Society of Traffic Management and Control – VCE NEARCTIS.

He has served in an advisory capacity to various institutes and programs and has performed several program assessments of leading international research institutes and corporate R&D departments. He was also a former member of various technical Committees of the Transportation Research Board.

Just Accepted