Data Engineering and Applications

Research Paper

Freight-Signal-Corridor-Driven Cooperative Speed Guidance Using a DeepONet-GRU Model

Kaido Hirt[1, *] and Rainer Rebane[2]

[1] School of Information Technologies, Tallinn University of Technology (TalTech), Tallinn, 19086, Estonia
[2] Institute of Data Science, University of Tartu Narva College, Narva, 20308, Estonia
*Corresponding author: kaido.hi@taltech.ee

Abstract. Freight signal corridors combine slow heavy-vehicle responses, heterogeneous payloads, and interacting queue spillbacks, which can make isolated eco-driving rules unreliable. A cooperative speed guidance model based on operator learning and gated recurrent units is proposed in this paper. The branch network encodes the rolling truck trajectory, signal plan, queue estimate, grade and payload descriptor, and the trunk network queries the future space-time position. A Feasibility projection maps the predicted corridor speed field to bounded vehicle commands. A model was trained using 1.94 million truck-state records from 420 calibrated simulation hours and tested on unseen demand, cycle, grade, penetration and communication conditions. In a six-intersection corridor, it reduced mean delay by 18.7%, stops by 31.4%, and energy-equivalent consumption by 12.6% compared with unguided actuated control. Speed mean absolute error was 2.06 m/s, 10.8% lower than the compact transformer baseline, and the 90th-percentile transferred-condition error was still 2.18 m/s. Median roadside inference for batches of 8-128 trucks increased from 3.6ms to 21.4ms, and the 99th percentile was still below 47ms. Heavy 34-40t trucks achieved an energy saving of 14.6%, and no projected commands exceeded the modelled red-light envelope. The simulation evidence indicates that a queryable corridor response operator can connect signal-state prediction with computationally feasible truck-level guidance; field validity remains to be established.

Keywords: *Freight Signal Corridor, Cooperative Speed Guidance, Deep Operator Network, Gated Recurrent Unit, Queue-Aware Control*

Received on 11 June 2023, Accepted on 16 October 2023, Published on 28 October 2023

Introduction

Urban and peri-urban freight corridors have a high proportion of heavy-vehicle traffic at close-set signalized intersections. They operate differently from the regular passenger car; they have a slower acceleration, larger braking loss, payload-constrained feasible motion, and a single extra stop can disrupt several following cycles. Signal coordination has been used for a long time to create progression bands, but fixed offsets do not fully utilize vehicle connectivity or the short-term trajectory information now available from roadside and fleet systems [1]. Connected-vehicle messages enable individual arrival control, but the guidance should still be reasonable when the nominal green window is congested [2]. Freight eco-driving research has shown that anticipation speed adjustment can reduce propulsion demand [3]. However, the advantages of these often rely on isolated-intersection assumptions or a deterministic signal phase and timing feed [4]. Corridor deployment adds interacting queues, uncertain discharge, grade, payload and communication latency, and thus turns a simple speed recommendation into a constrained spatio-temporal coordination problem [5].

The current guidance method computes a target arrival time and then solves a local trajectory problem. Although the division is clear, any error in queue discharge will directly affect the command and require another modification. Model-predictive controllers explicitly handle constraints [6], but for corridor-scale optimization involving many trucks updated every second, it can be too slow. Reinforcement-learning approaches can learn policies without solving an online program [7], but the actions are difficult to interpret outside the training distribution. Recurrent neural networks address temporal dependence in traffic conditions [8], and graph models consider connections among intersections [9]. Recently, Lu et al. [10] introduced a different abstraction by learning the mapping from historical functions and boundary conditions to queryable response fields, rather than approximating a single finite-dimensional input-output mapping. This feature is beneficial for freight corridors because the output naturally provides continuous speed recommendations in terms of future distance and time. However, relying solely on operator learning cannot ensure temporal robustness, vehicle feasibility, or safe queuing methods.

Therefore, the research gap is in the intersection of corridor-field prediction and feasible truck guidance. A practical model should incorporate both long-term and short-term temporal dependencies, be conditioned on vehicle-specific payload and power limits, query the response at any downstream location, and filter the result by safety constraints. It should be light enough for roadside inference and exhibit failure modes under demand, penetration, latency and signal-plan shifts. This paper introduces a freight-signal-corridor-driven cooperative speed guidance model that combines a Deep Operator Network (DeepONet) and a gated recurrent unit (GRU). The contributions are three testable advances: first, a branch-trunk representation that provides continuous space-time queries rather than a fixed forecast vector; second, a GRU fusion layer evaluated through transfer and ablation tests for rolling queue and phase context; and third, an independently auditable feasibility projection evaluated for command smoothness, latency, and modeled signal compliance.

The remaining parts of this paper are: Section 2 reviews connected freight speed guidance and learning-based traffic prediction, and locates the proposed approach between optimization and operator learning. Section 3 shows the corridor coordination problem and introduces the sensing, communication and safety workflow. Section 4 introduces the DeepONet-GRU architecture, training objective and online command generator. Section 5 shows the data, baselines, predictive accuracy, operating effects, robustness tests and ablations. Section 6 presents the principal conclusions, study limitations, and directions for field validation.

Related Work

Freight corridor speed guidance

Most connected eco-driving research has used signal phase and timing information to plan a vehicle path before an intersection. Analytical advisory systems build a feasible arrival interval and select a low-energy profile [11]. Model-predictive formulations add explicit acceleration, spacing or powertrain constraints to the idea [12]. The load and gradient of heavy vehicles will significantly alter the available trajectory; therefore, the parameterization of passenger cars may underestimate braking distance and energy consumption [13]. Corridor studies add offset coordination and queue prediction, but many still solve intersections sequentially; if the first queue estimate is wrong, the downstream advice will inherit this error [14]. These studies establish strong local trajectory-control principles but do not fully represent interacting corridor queues as one queryable response system.

Learning-based traffic prediction and control

Traffic prediction models have moved from recurrent sequence models to graph and attention architectures. Jiang et al. [15] showed that GRU-based predictors are suitable for real-time traffic forecasting because of their gated-state mechanism and relatively low computational overhead. Spatiotemporal graph networks encode topology and correlated congestion, but fixed graphs do not directly represent a continuous query at an unseen position [16]. Transformer predictors have increased the receptive field but require more data and memory for roadside execution [17]. Deep operator learning maps an input function to an output function, and then evaluates the learned response at any location [18]. It is still in the early days of cooperative driving, and few published formulations integrate operator queries, recurrent state estimation, payload conditioning and deterministic safety projection into a single corridor controller. DeepONet is used in this paper for continuous space-time querying, GRU for causal history compression, and a traditional constraint layer is retained for execution.

Corridor Coordination Framework

Problem formulation

Consider a one-way freight corridor with six signalized intersections and a roadside coordinator that receives detector data, signal phase and timing messages, and connected-truck states. At update time k, the coordinator observes the last H samples of speed, position, acceleration, queue length, phase state, residual green time, grade and communication quality. Vehicle descriptors are gross mass, power limit, rolling coefficient and length. The control objective is not to have all the trucks reach the green light as soon as possible; instead, we wish to create a stable speed field that balances progress, energy, comfort and queue safety over a finite horizon.

For truck i, the corridor state history is written as the multichannel function $u_i(\tau)$, and the query $\mathbf{y} = (x, t)$ specifies a downstream location x and future time t. The desired operator maps this history and the signal boundary conditions to a feasible reference speed field:

$$\mathcal{G}: (u_i(\tau), s(\tau), m_i) \mapsto v_i^*(x, t) \quad \text{Eq.(1)}$$

This formulation separates what has been observed from where a prediction is required. Therefore, it does not have a fixed output grid and can query a location close to a stop line densely while using coarser points between intersections. The operating cost attributed to a candidate field includes travel time, tractive energy, stopping and smoothness:

$$J_i = \sum_{h=1}^{P} \left(w_t \Delta t + w_e E_{i,h} + w_s \mathbf{1}_{\{v_{i,h} < 0.5\}} + w_j j_{i,h}^2 \right) \quad \text{Eq.(2)}$$

The weights are normalized by the training-set scale so that no term is disproportionately large due to units. A stop is defined as motion at or below $0.5\ \mathrm{m/s}$ for a continuous duration of 2 s, and thus excludes short-lived low-speed movement caused by the discharge wave.

Signal-Corridor Information Pipeline

Figure 1 presents the coordination workflow. Detector and connected-vehicle streams are first clock-aligned, plausibility-checked and then placed in a queue estimator. A corridor tensor then combines phase trajectories, predicted discharge, truck states, payload descriptors, grade and link occupancy. DeepONet-GRU model queries this tensor at candidate space-time points. A constraint projector verifies the learned speed field against vehicle kinematics, car-following headway, red-light compliance and emergency-braking distance prior to command emission. Vehicle acknowledgements and realized trajectories return to the state estimator to form a closed loop, rather than an open-loop advisory. Separation of learned prediction and deterministic projection can also keep the projector active when the model's confidence drops below the deployment threshold.

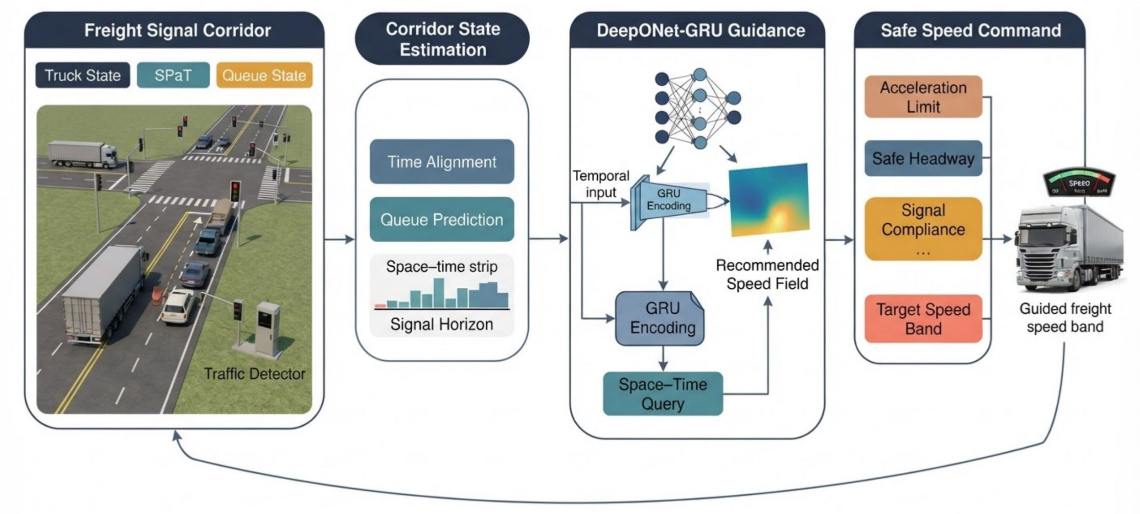

Figure 1. Closed-loop freight signal corridor coordination workflow from sensing and phase prediction to DeepONet-GRU field inference, feasibility projection, vehicle guidance, and feedback.

Feasibility and Safety Layer

The deterministic projector clips the raw recommendation using vehicle-dependent acceleration and deceleration limits. A jerk limiter avoids a sudden change in the previous update, and a queue envelope limits the reachable position when the predicted back of the queue is upstream of the stop line. Let d_q denote the distance to the effective queue boundary and T_r the response delay. The admissible speed must satisfy

$$v_{i,k}^2 \leq 2b_i^{safe} max\left(0, d_q - v_{i,k}T_r - d_0\right) \tag{Eq.(3)}$$

If a bound is exceeded, the reference is projected onto the nearest admissible speed and kept below the red-phase approach envelope. The coordinator also has a freshness constraint; signal messages older than 1.5s and vehicle messages older than 2.0s are no longer considered for green-window targeting. A conservative car-following fallback then substitutes the learned recommendation. Thus, the communication breakdown will be a planned decline in the goal of guidance, not an unplanned spread.

Coordination of multiple guided trucks is done by ordered reservations rather than independent green window selections. The queue estimator predicts the departure time of each vehicle rank, and then the coordinator sets a soft arrival window with a width that considers the uncertainty in saturation flow. A downstream truck will be eligible for the same green window if its expected headway remains above the payload-dependent threshold during the approach. Otherwise, it is shifted to the next admissible band. This rule prevents the learned speed field from compressing several trucks into an infeasible platoon merely because their recent histories are similar. Reservations are recalculated after every accepted message, but an already issued band will not be displaced by more than 3s in one update unless the signal plan changes or the safety envelope shrinks.

The safety layer is intentionally independent of the neural network weights. The inputs and decisions are logged as human-readable reason codes: acceleration bound, stopping distance, stale message, queue occupancy, red envelope, leading-vehicle headway, or low confidence. Separation can be used to verify and strengthen some rules for a corridor operator without retraining the prediction model. It will have a reasonable deployment area. The neural component proposes a desirable field, while the projector verifies only that the immediate command lies within the modeled admissible set; it does not constitute a general safety certification. Long-term safety will still need several state updates, conservative fallbacks, and the vehicle's on-board collision-avoidance system.

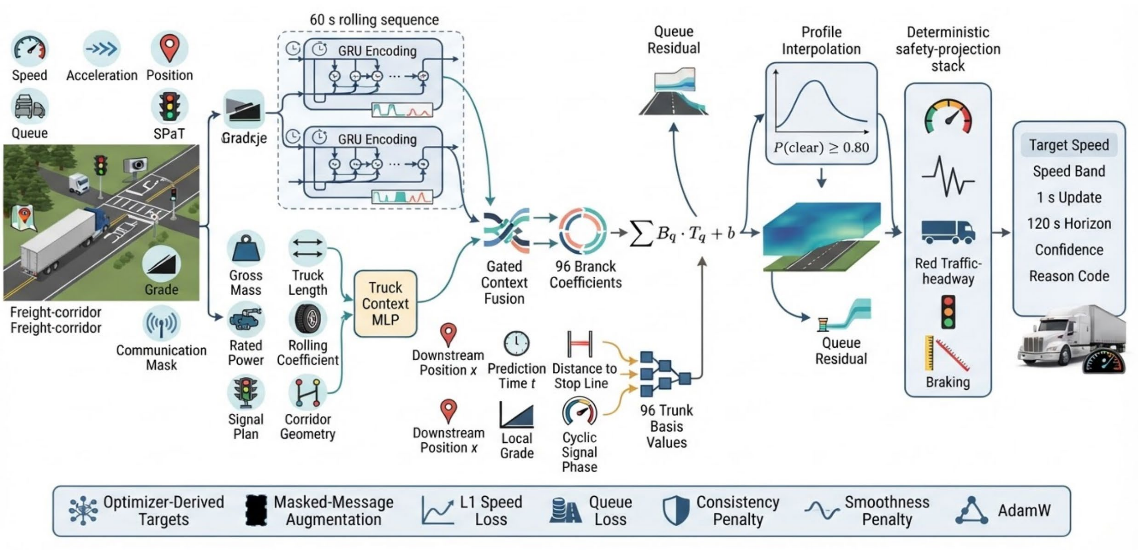

Figure 2. DeepONet-GRU architecture with temporal branch encoding, static truck-context fusion, continuous space-time trunk queries, queue residual head, and constrained command output.

DeepONet-GRU Guidance Model

Branch-Trunk Representation

Figure 2 shows the proposed architecture. The branch pathway contains two encoders. A GRU is used to process the$H$-step corridor sequence, and a multilayer perceptron embeds the static truck attributes and signal-plan descriptors. The two resulting vectors are fused by a gated context-fusion module. The trunk network encodes the normalized position, prediction time, distance to the next stop line, local grade and phase-cycle coordinates. Branch and trunk outputs share a latent dimension p, and their inner product is used to define the speed-field value at any query point.

For hidden state $\mathbf{h}_k$ and input $\mathbf{z}_k$, the GRU update is

$$\boldsymbol{h}_k = (1 - \boldsymbol{r}_k) \odot \boldsymbol{h}_{k-1} + \boldsymbol{r}_k \odot tanh[\boldsymbol{W}_h \boldsymbol{z}_k + \boldsymbol{U}_h(\boldsymbol{g}_k \odot \boldsymbol{h}_{k-1}) + \boldsymbol{b}_h] \quad \text{Eq.(4)}$$

Here, $\mathbf{r}_k$ is the update gate and $\mathbf{g}_k$ is the reset gate. The DeepONet output for query $\mathbf{y}$ is then expressed as

$$\hat{v}_i(\boldsymbol{y}) = v_{lim}\sigma\left[\sum_{q=1}^{p} B_q(\boldsymbol{h}_k, m_i, \boldsymbol{s}_k)T_q(\boldsymbol{y}) + b\right] \quad \text{Eq.(5)}$$

Before the feasibility projection, the sigmoid function constrains the original field through link speed limits. A small residual head predicts the queue boundary error and is only used to move the safety boundary; it does not directly increase the commanded speed.

Temporal Fusion and Training

Training samples are rolling windows taken from corridor episodes. Connected-vehicle messages are randomly masked so that the GRU learns to use detector and signal context when connected-vehicle penetration is low. Payload, demand and cycle settings are stratified by mini-batch. The supervised target is obtained from a constrained trajectory optimizer using realized queues, and an auxiliary head predicts queue positions over horizons of 30 to 120 s. The training loss is:

$$\mathcal{L} = \lambda_v\|\hat{\boldsymbol{v}} - \boldsymbol{v}^{opt}\|_1 + \lambda_q\|\hat{\boldsymbol{q}} - \boldsymbol{q}\|_1 + \lambda_c\Phi_{cons} + \lambda_s\|\Delta\hat{\boldsymbol{v}}\|_2^2 \quad \text{Eq.(6)}$$

The consistency penalty Φ_{cons} discourages speed fields that imply position reversal or red-light crossing. Training uses AdamW, early stopping based on corridor-level validation episodes, and cosine learning-rate decay. Hyperparameters are selected based on the validation scenario, not the reported test conditions. Therefore, the scenario-specific tuning should not be regarded as transfer ability.

Sensor channels are standardized based on robust statistics calculated only from the training set, and phase time is shown in sine and cosine coordinates to prevent a discontinuity at cycle rollover. Missing values contain both an imputed value and a binary observation mask. The sequence encoder contains two GRU layers with 96 hidden units. The branch and trunk projections each have three 128-unit layers and produce $p = 96$ latent coefficients. Static mass and power descriptors pass through a 32-unit embedding before gated fusion. The above dimensions were chosen from a bounded validation search with latent widths of 48 to 160 and one-to-three recurrent layers. Larger models improved the training error but had little transfer gain and increased roadside latency.

To reduce the mismatch between optimizer targets and executable commands, the last third of the training adds projected targets generated from perturbed queue and latency estimates. The network thus observes cases where the nominal optimum cannot be realized exactly. Gradient clipping at 1.0 prevents destabilization of the recurrent encoder caused by occasionally saturated episodes. Each mini-batch contains complete, partially masked and delayed-message versions of the same corridor episode to promote a common latent state rather than separate behaviors for individual communication conditions. Validation stopping uses a combined score of speed error, queue error and projected red-envelope violations, and thus a numerically precise but operationally fragile checkpoint is rejected.

Online Command Generation

At every 1 s control update, the trunk network evaluates 24 candidate query points over the next 120 seconds and the next two stop lines. Cubic Hermite interpolation is a smooth local curve, and the earliest feasible arrival window is selected only if its predicted queue-clearing probability exceeds 0.80. The command sent to the truck is the first part of the projected profile:

$$v_{i,k}^{cmd} = \Pi_{\Omega_i(k)}\left[\alpha\hat{v}_i\left(x_{i,k} + v_{i,k}\Delta t, k+1\right) + (1-\alpha)v_{i,k}^{cmd,-}\right] \quad \text{Eq.(7)}$$

Here, $\Pi_{\Omega_i(k)}$ denotes projection onto the vehicle-specific feasible set, and $\alpha = 0.35$ suppresses command oscillation. The roadside unit returns a target speed, validity horizon, confidence value, and reason code. A driver-oriented implementation will show a rounded speed band instead of an exact number. The experiment isolates the controller's behavior from the driver interface effects by using the centerline command.

Experimental Evaluation

Data and experimental design

A microscopic corridor testbed was established at a 7.8 km industrial arterial with six coordinated intersections and link grades ranging from -2.1% to 3.4%, and a 70 km/h limit. The dataset has 420 simulated hours and 1.94 million truck state records, and these have been divided into episodes of 70% training, 10% validation and 20% testing. The range of truck gross mass was 12–40 t, the cycle length was 80–140 s, the freight demand was 180-620 veh/h, and the connected penetration ranged from 20% to 100%. Calibration followed observed link travel-time distributions and queue maxima rather than a single mean statistic [19]. Emission-equivalent energy was calculated based on tractive power and idle consumption; it is more sensitive to truck mass and grade than a distance-only factor [20].

The two-part distribution view in Figure 3 is the data context for the model. As shown in Figure 3(a), the 12, 20, 28, 34 and 40 t payload-class samples across the five demand bands show a median-speed decline from 15.8 to 11.4 m/s and a 90th-percentile queue increase from 74 to 286 m as demand rises. Figure 3(b) pairs hourly demand density with green-ratio and grade distributions: 62% of the samples are between 300 and 500 veh/h, the green ratio is 0.34-0.61, and uphill observations reach 3.4%. The split retains rare high-mass, high-demand combinations in the test set to avoid an easily achieved evaluation [21].

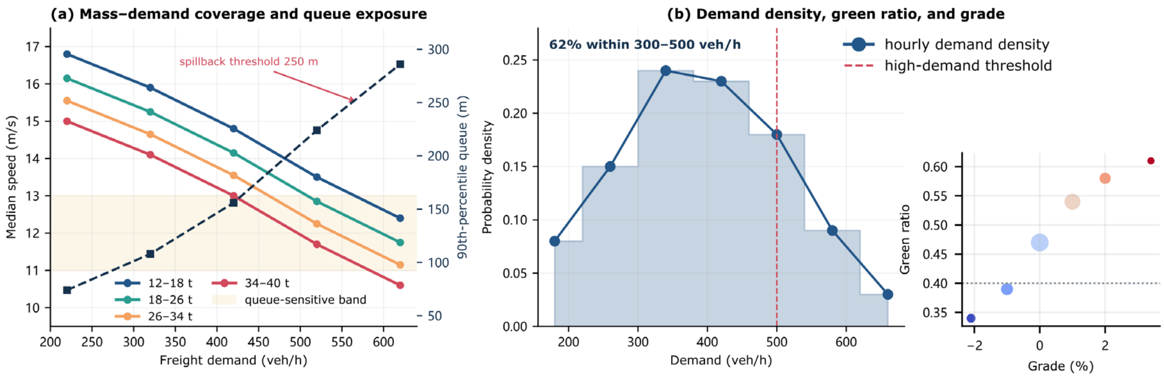

Figure 3. Experimental data distributions: (a) speed and queue statistics across truck-mass and demand classes; (b) joint demand density, green-ratio distribution, and corridor-grade coverage.

The baselines comprise actuated signals without guidance, green-light optimal speed advisory (GLOSA), corridor model-predictive control (MPC), GRU, spatiotemporal graph convolution (STGCN), and a compact transformer. All learning models use the same history and descriptor input; only their function approximators differ. The primary performance indicators are speed MAE, queue MAE, corridor travel time, stops, energy-equivalent consumption, jerk RMS, hard braking and roadside inference latency. Ten random seeds and their bootstrap 95% confidence intervals are shown. Network performance is calculated for all vehicles, and speed-error indicators are only available for connected trucks [22].

The corridor is simulated at a 0.1 s resolution, detector measurements are aggregated every second, and advice is refreshed at the same 1 s interval. Training windows have a history of 60 s and a prediction context of 120 s. The demand has a non-stationary arrival process with shift changes, lunch peaks and stochastic platoons released from an upstream junction. Lane 1 is for passenger cars, rigid trucks, tractor-trailers and occasional buses. Only freight vehicles are given advice; all classes contribute to the queue and occupancy estimates. Lane changes use calibrated gap acceptance, and a 7% fraction of unequipped vehicles show slow-response driving to stress the discharge prediction model.

Evaluation scenarios are blocked by day and a random seed to prevent neighboring windows from leaking between training and testing. Five scenario families test nominal operation, unseen cycle plans, grade transfer, reduced penetration and delayed messages. Each family has at least 120 episodes, and all baselines receive the

https://doi.org/10.64972/dea.2023.v2i4.3784d:41-53

same vehicle arrivals, signal randomization and incident perturbations. Paired comparison therefore removes some of the variance due to random demand. Confidence intervals are based on episode-level differences, not individual one-second samples, as adjacent samples share the same queue evolution. This episode-level construction produces wider but more defensible uncertainty intervals.

Predictive and operational performance

The three-view accuracy assessment in Figure 4 distinguishes between average error and calibration/spatial failure. Figure 4(a) shows the speed MAE, with DeepONet-GRU at 2.06 m/s; the others are 2.31 m/s (transformer), 2.48 m/s (STGCN), 2.73 m/s (GRU), and 3.18 m/s (GLOSA), and the error bars across seeds are 0.07–0.16 m/s. Figure 4(b) is the horizon-wise queue MAE: The proposed model is 7.8 m at 30 s and 22.6 m at 120 s; the Transformer is 27.9 m and the GRU is 34.7 m. Figure 4(c) shows the predicted and realized speeds, a slope of 0.96, R^2=0.91, and a 90% prediction band that widens near queue discharge. The remaining underpredictions above 18 m/s represent relatively conservative high-speed cases and not unsafe queue approaches [23]. This combination of low average error and stable calibration supports reliable speed-field estimation across both free-flow and queue-transition regimes.

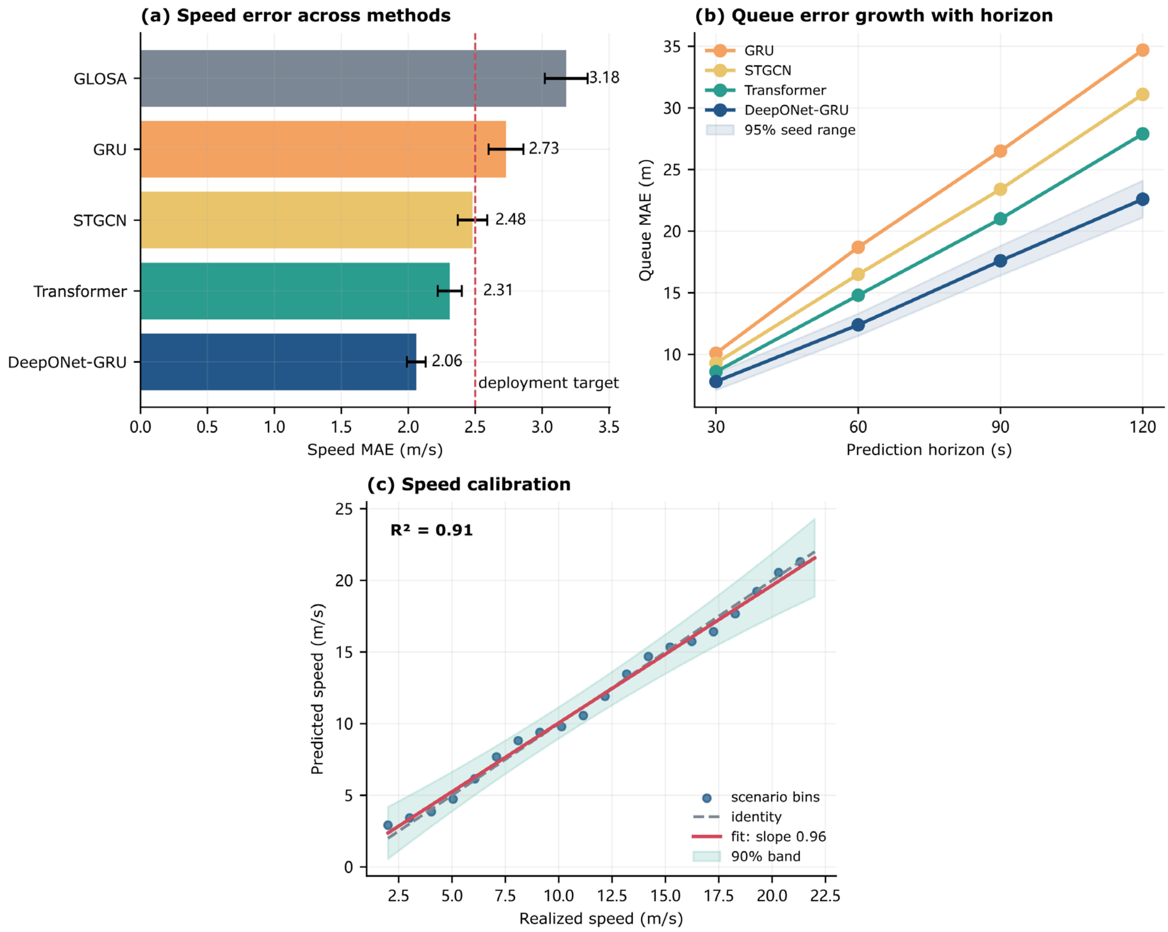

Figure 4. Predictive performance: (a) speed MAE with seed uncertainty; (b) queue MAE versus prediction horizon; (c) predicted-versus-realized speed calibration and 90% band.

Operational outcomes appear in the four-part Figure 5. Figure 5(a) shows mean corridor travel times of 694 s without guidance, 641 s for GLOSA, 612 s for MPC, 585 s for the transformer, and 564 s for DeepONet-GRU; the proposed reduction is 18.7%. Figure 5(b) gives stops per trip of 4.33, 3.72, 3.18, 3.11, and 2.97, respectively, so the proposed model removes 31.4% of baseline stops. Figure 5(c) displays energy-equivalent consumption by mass class: savings relative to no guidance are 9.4%, 11.8%, 13.1%, and 14.6% for 12–18, 18–26, 26–34, and 34–40 t trucks. Figure 5(d) presents travel-time cumulative distributions; the median shifts from 676 to 552 s and the 90th percentile from 842 to 687 s. The larger tail improvement is consistent with queue-spillback avoidance rather than uniform cruising alone [24].

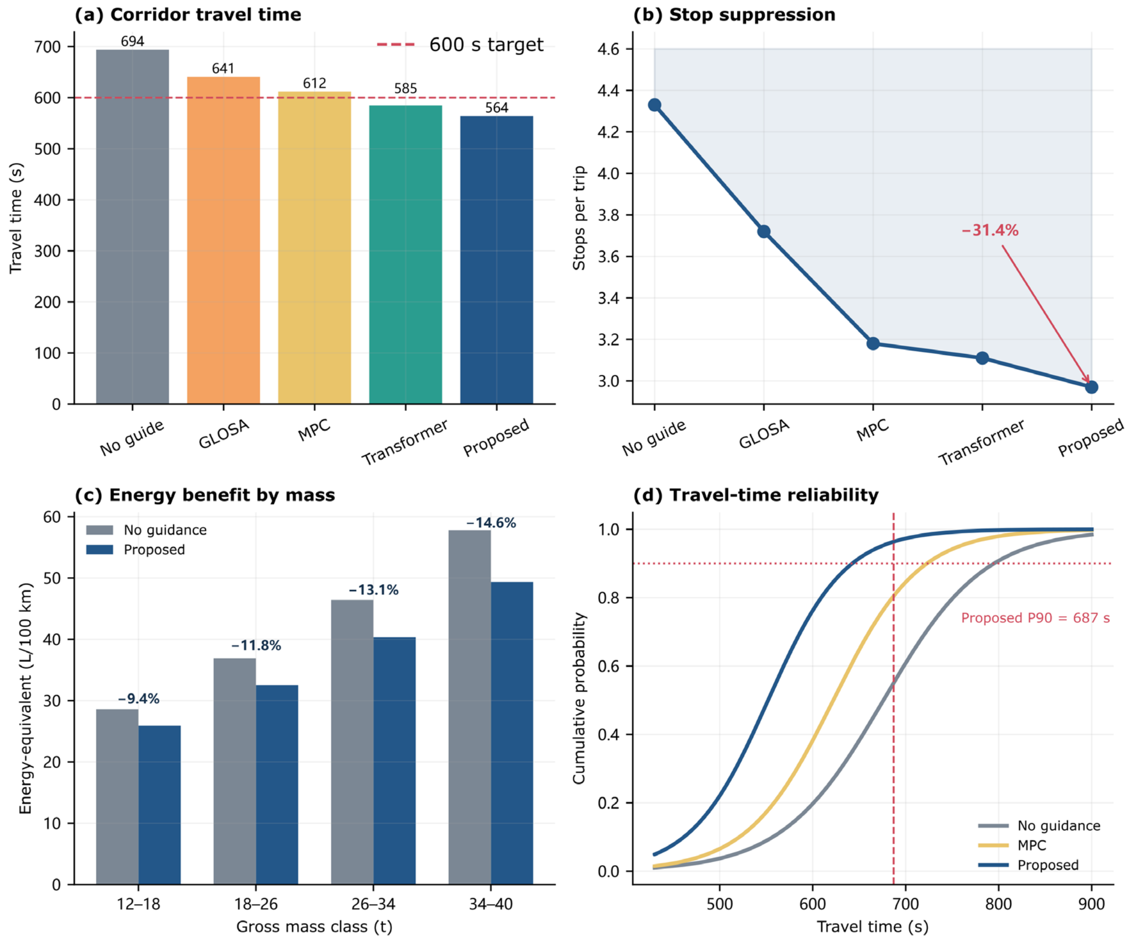

Figure 5. Corridor outcomes: (a) travel time by method; (b) stops per trip; (c) energy-equivalent consumption and savings by mass class; (d) travel-time cumulative distributions.

Comfort and execution cost are shown in the two-part Figure 6. Figure 6(a) combines jerk RMS, hard-braking frequency, and speed-command revision magnitude; DeepONet-GRU records 0.41 m/s³, 0.18 events/trip, and 0.74 m/s, compared with MPC values of 0.52, 0.24, and 1.02, respectively. Figure 6(b) presents the distribution of roadside inference latency for fleet batch sizes of 8–128 trucks; the median latency increases from 3.6 ms to 21.4 ms, while the 99th percentile remains below 47 ms at 128 trucks. The compact Transformer exceeds 100 ms at that batch size. Since the deployment update interval is 1 s, the proposed implementation retains a substantial computational margin after message decoding and feasibility checks [25]. These results demonstrate that the model improves command smoothness without compromising the real-time responsiveness required for corridor-scale deployment.

Pairwise significance tests extend the scale comparison. Compared with the strongest learning baseline, the proposed model has reduced travel time per trip by 21 s (95% confidence interval: 16-27s), energy-equivalent consumption by 3.7% (2.9-4.5%), and command revision magnitude by 0.19 m/s (0.14-0.24m/s). The two strengths are at the second and fourth intersections, and the upstream release patterns are short-storage links. All learned models behave similarly on the longest link because the available cruise distance accommodates the arrival error. Therefore, queue prediction is more suitable for areas with a small recovery space, and this spatial interpretation is more informative about infrastructure placement than a corridor-wide average.

The no-guidance and GLOSA cases show a secondary travel-time mode near 810s, which occurs because trucks miss two consecutive progression bands after a long first stop. That mode is weak under MPC and missing in DeepONet-GRU. Inspection of the individual traces shows that the proposed model sometimes accepts a 9-14s

https://doi.org/10.64972/dea.2023.v2i4.3784d:41-53

slower arrival at the first intersection to keep a reasonable band across the next two. A local advisory generally moves towards the first green light and then brakes at the downstream queue. The operator formulation learns the downstream effect without listing individual band combinations online, and the projector avoids using unrealistically sharp acceleration in the field.

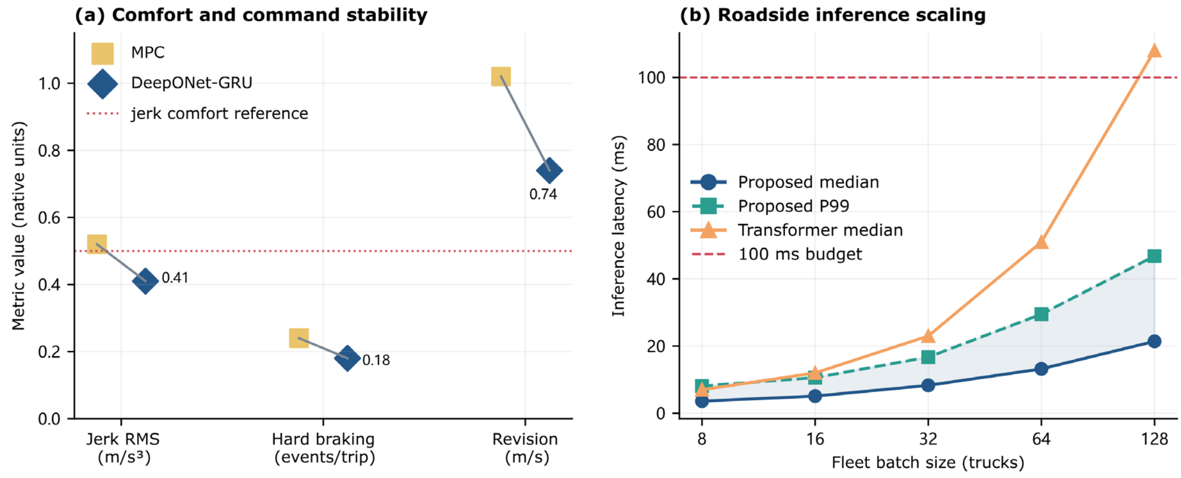

Figure 6. Execution characteristics: (a) jerk, hard braking, and command revisions; (b) median and tail inference latency across fleet batch sizes.

Robustness, ablation, and operational interpretation

Robustness and component effects are summarized by the four-part Figure 7. Figure 7(a) shows delay reduction across connected penetrations of 20%, 40%, 60%, 80%, and 100%: 6.1%, 10.4%, 14.2%, 17.0%, and 18.7%. Figure 7(b) reports speed MAE under 0, 0.5, 1.0, 1.5, and 2.0 s communication delays; it remains below 2.50 m/s through 1.5 s and reaches 2.83 m/s at 2.0 s, where freshness fallback activates more frequently. Figure 7(c) is the single heat map in the figure set and gives energy savings over demand–mass cells, ranging from 5.8% in low-demand light-truck traffic to 15.2% for heavy trucks at 500 veh/h; savings fall to 11.9% at 620 veh/h because saturated queues reduce arrival flexibility. Figure 7(d) gives the ablation radar and numeric annotations: removing the GRU raises speed MAE by 21%, removing trunk queries raises queue MAE by 17%, removing payload conditioning reduces heavy-truck energy savings by 4.1 percentage points, and removing the projection multiplies red-envelope violations from 0.03 to 0.41 per 100 trips [26].

Demand transfer was assessed by withholding one cycle-length family and the highest-grade band during training. DeepONet-GRU achieved an MAE of 2.18 m/s and a delay reduction of 15.3%, and was better than the transformer's 2.64 m/s and 11.2%. Operator queries appear to interpolate the response field more effectively than a fixed 24-step decoder, especially at irregular intervals of stop lines. However, the advantage decreases with full saturation, and no model can form a feasible green window after the spillback covers the upstream link. The boundary is operationally necessary; otherwise, the confidence gate would drive the controller towards safe queue following instead of providing unattainable progression advice [27].

Paired-episode analysis shows that 71% of the energy reduction is due to avoiding acceleration after stops and 29% to smoother speed regulation at signals. The effect increases with mass, and time savings are relatively uniform across payload classes. The model does not materially disadvantage unequipped vehicles in the simulated scenarios; at a 60% penetration rate, connected trucks reduce the average delay for discharge oscillations by 7.6%, but lane change may not achieve the same result in a highly heterogeneous environment. Across all 840 test episodes, no red-light crossing occurs after projection, and the minimum simulated time-to-collision is still above 2.1 s. The above observations support deployment as a roadside advisory with an explicit safety layer, and field trials still need to address driver compliance, positioning bias, and mixed-fleet communication behavior [28].

Sensitivity to the confidence threshold shows a typical operating trade-off. Lowering the queue-clearance threshold from 0.80 to 0.65 increases the nominal delay reduction from 18.7% to 20.1%, but hard braking rises to 0.29 events per trip and red-envelope projection is activated 46% more frequently. Set it to 0.90 to reduce

https://doi.org/10.64972/dea.2023.v2i4.3784d:41-53

hard braking to 0.15 events per trip, but lose 2.8 percentage points in delay benefit. Therefore, the selected value is an engineering operating point rather than an accuracy hyperparameter. The road authority can increase the conservatism without altering the learned weights, and the logged projection rate indicates how far the model is from matching the actual traffic conditions at present.

Failure inspection focuses on cases exceeding the 95th percentile of speed error. Forty-three percent involve an unobserved lane blockage, 31% arise from abrupt platoons entering between detectors, 17% combine maximum grade with a 40-t truck, and 9% reflect signal-message delay beyond the freshness limit. The first two groups have errors mainly due to queue location, not free-flow speed. Providing perfect blockage information recovers about half of the lost delay benefit; thus, sensing coverage is the primary solution. More detailed powertrain states are required for the heavy-uphill group, as gross mass and rated power do not indicate gear selection or thermal derating. The latency group is already limited by fallback, and its direction is less effective.

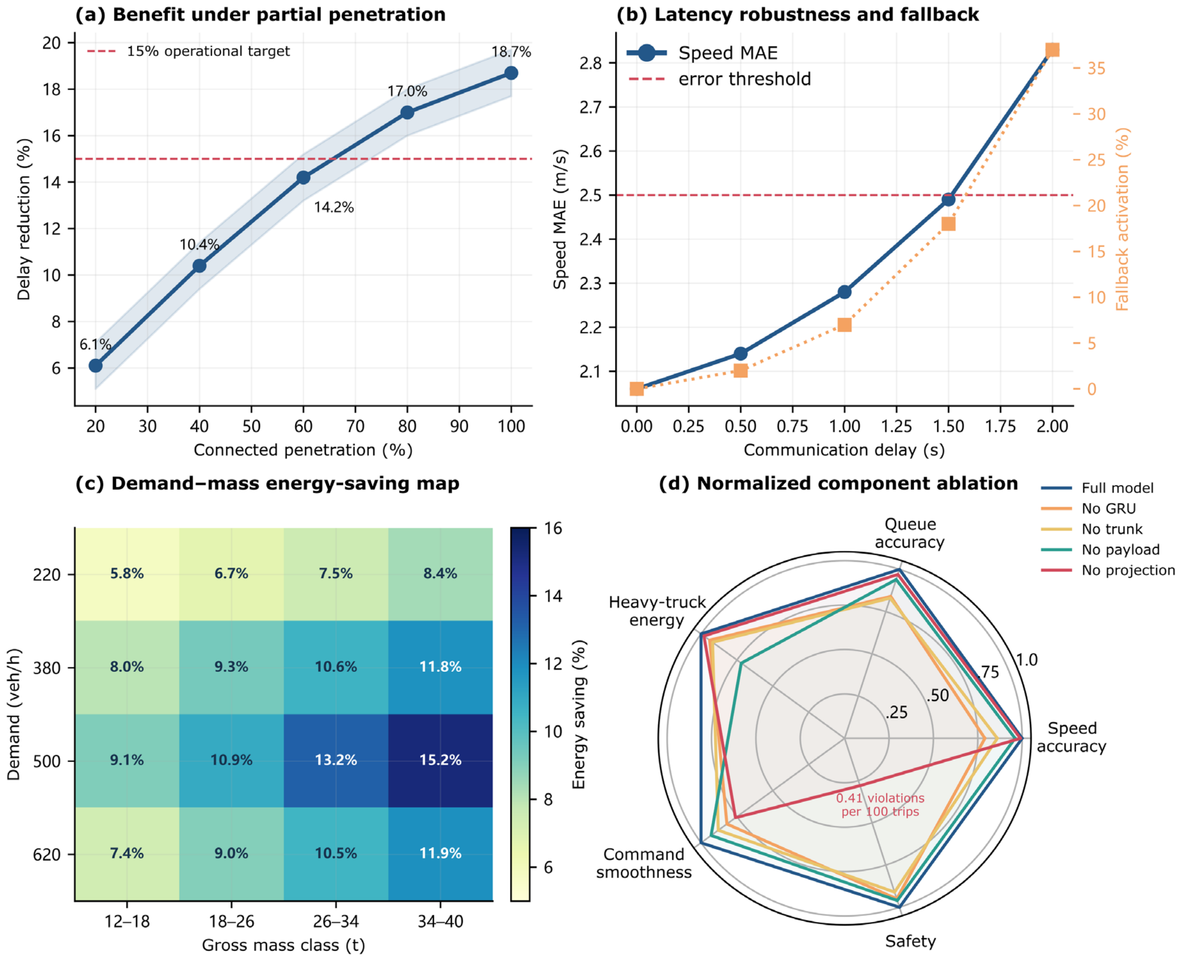

Figure 7. Robustness and ablation: (a) benefit versus connected penetration; (b) speed error under communication delay; (c) demand–mass energy-saving heat map; (d) normalized component-ablation radar with safety annotations.

Conclusion

This paper proposes a freight-signal-corridor-driven cooperative speed guidance method that combines DeepONet with GRU temporal encoding and a deterministic feasibility projection. The formulation learns a queryable corridor response operator from rolling signal, queue, vehicle, grade and payload information, and converts the predicted field into stable truck commands. The experiments examine predictive accuracy, corridor efficiency, comfort, latency, connected penetration, communication delay, demand transfer, and component-level ablation. The above evidence supports the application of corridor-state prediction and vehicle-level guidance in operator learning. The Structure also separates the data-driven prediction and rule-based execution;

therefore, transport authorities can modify safety constraints, confidence thresholds or communication policies without retraining the entire model.

The study relies on a calibrated microscopic environment, centralized roadside coordination, and modeled driver behavior rather than field observations. Rare cases, bad weather, detector failures and complex multi-lane interactions were only shown as bounded perturbations. The energy model has included mass, grade, acceleration and idle but does not cover all powertrain control modes. The feasibility projector also guarantees that it meets the modelled envelopes and does not certify safety under all physical and behavioral uncertainties. The experiments also assume correct vehicle identification and payload information. In field deployment, inaccurate mass estimation, map errors, delayed signal-controller updates and inconsistent driver responses can all reduce prediction accuracy and stability of cooperative arrival assignments.

Future work will carry out hardware-in-the-loop and field tests of heterogeneous diesel, battery-electric and human-driven fleets. Distributionally robust operator training can assess the confidence of unseen incidents and weather, and federated adaptation may update corridor models without centralizing proprietary fleet data. A decentralized vehicle-roadside implementation and formal reachability analysis are also required before safety-critical deployment. Future extensions should coordinate lane selection and charging constraints and connect tactical guidance with network-level freight scheduling. Other research will explore interpretable uncertainty decomposition, corridor-to-corridor transfer with limited calibration data, and incentive mechanisms to promote participation by competing freight operators.

Author Contributions

Kaido Hirt contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision, project administration, and funding acquisition. Rainer Rebane contributes to analysis, investigation, data collection, draft preparation. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

References

[1] Xu, N., Li, X., Liu, Q., & Zhao, D. (2021). An Overview of Eco-Driving Theory, Capability Evaluation, and Training Applications. Sensors, 21(19), 6547. https://doi.org/10.3390/s21196547

[2] Yu, G., Li, H., Wang, Y., Peng, C., & Zhou, B. (2022). A review on cooperative perception and control supported infrastructure-vehicle system. Green Energy and Intelligent Transportation, 1(3), 100023. https://doi.org/10.1016/j.geits.2022.100023

[3] Smith, S. W., Kim, Y., Guanetti, J., Li, R., Firoozi, R., Wootton, B., Kurzhanskiy, A. A., Borrelli, F., Horowitz, R., & Arcak, M. (2020). Improving Urban Traffic Throughput With Vehicle Platooning: Theory and Experiments. IEEE Access, 8, 141208–141223. https://doi.org/10.1109/access.2020.3012618

[4] Maadi, S., Stein, S., Hong, J., & Murray-Smith, R. (2022). Real-Time Adaptive Traffic Signal Control in a Connected and Automated Vehicle Environment: Optimisation of Signal Planning with Reinforcement Learning under Vehicle Speed Guidance. Sensors, 22(19), 7501. https://doi.org/10.3390/s22197501

[5] Liu, M., Zhao, J., Hoogendoorn, S., & Wang, M. (2021). A single-layer approach for joint optimization of traffic signals and cooperative vehicle trajectories at isolated intersections. Transportation Research Part C Emerging Technologies, 134, 103459. https://doi.org/10.1016/j.trc.2021.103459

[6] Khayatian, M., Mehrabian, M., Andert, E., Dedinsky, R., Choudhary, S., Lou, Y., & Shirvastava, A. (2020). A Survey on Intersection Management of Connected Autonomous Vehicles. ACM Transactions on Cyber-Physical Systems, 4(4), 1–27. https://doi.org/10.1145/3407903

[7] Song, L., & Fan, W. (2021). Traffic Signal Control Under Mixed Traffic With Connected and Automated Vehicles: A Transfer-Based Deep Reinforcement Learning Approach. IEEE Access, 9, 145228–145237. https://doi.org/10.1109/access.2021.3123273

[8] Shaygan, M., Meese, C., Li, W., Zhao, X., & Nejad, M. (2022). Traffic prediction using artificial intelligence: Review of recent advances and emerging opportunities. Transportation Research Part C Emerging Technologies, 145, 103921. https://doi.org/10.1016/j.trc.2022.103921

[9] Yuan, H., & Li, G. (2021). A Survey of Traffic Prediction: from Spatio-Temporal Data to Intelligent Transportation. Data Science and Engineering, 6(1), 63–85. https://doi.org/10.1007/s41019-020-00151-z

[10] Lu, L., Jin, P., Pang, G., Zhang, Z., & Karniadakis, G. E. (2021). Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. Nature Machine Intelligence, 3(3), 218–229. https://doi.org/10.1038/s42256-021-00302-5

[11] Zhao, X., Wu, X., Xin, Q., Sun, K., & Yu, S. (2020). Dynamic Eco-Driving on Signalized Arterial Corridors during the Green Phase for the Connected Vehicles. Journal of Advanced Transportation, 2020, 1–11. https://doi.org/10.1155/2020/1609834

[12] Long, K., Ma, C., Jiang, Z., Wang, Y., & Yang, X. (2020). Integrated Optimization of Traffic Signals and Vehicle Trajectories at Intersection With the Consideration of Safety During Signal Change. IEEE Access, 8, 170732–170741. https://doi.org/10.1109/access.2020.3021082

[13] Hu, J., Zhang, Z., Xiong, L., Wang, H., & Wu, G. (2021). Cut through traffic to catch green light: Eco approach with overtaking capability. Transportation Research Part C Emerging Technologies, 123, 102927. https://doi.org/10.1016/j.trc.2020.102927

[14] Fang, S., Yang, L., Wang, T., & Jing, S. (2020). Trajectory Planning Method for Mixed Vehicles Considering Traffic Stability and Fuel Consumption at the Signalized Intersection. Journal of Advanced Transportation, 2020, 1–10. https://doi.org/10.1155/2020/1456207

[15] Jiang, M., Chen, W., & Li, X. (2021). S-GCN-GRU-NN: A novel hybrid model by combining a Spatiotemporal Graph Convolutional Network and a Gated Recurrent Units Neural Network for short-term traffic speed forecasting. Journal of Data Information and Management, 3(1), 1–20. https://doi.org/10.1007/s42488-020-00037-9

[16] Reza, S., Ferreira, M. C., Machado, J. J. M., & Tavares, J. M. R. S. (2022). A multi-head attention-based transformer model for traffic flow forecasting with a comparative analysis to recurrent neural networks. Expert Systems with Applications, 202, 117275. https://doi.org/10.1016/j.eswa.2022.117275

[17] Li, G., Knoop, V. L., & van Lint, H. (2021). Multistep traffic forecasting by dynamic graph convolution: Interpretations of real-time spatial correlations. Transportation Research Part C Emerging Technologies, 128, 103185. https://doi.org/10.1016/j.trc.2021.103185

[18] Wang, S., Wang, H., & Perdikaris, P. (2021). Learning the solution operator of parametric partial differential equations with physics-informed DeepONets. Science Advances, 7(40), eabi8605. https://doi.org/10.1126/sciadv.abi8605

[19] Leitner, D., Meleby, P., & Miao, L. (2022). Recent advances in traffic signal performance evaluation. Journal of Traffic and Transportation Engineering (English Edition), 9(4), 507–531. https://doi.org/10.1016/j.jtte.2022.06.002

[20] Xing, J., Wu, W., Cheng, Q., & Liu, R. (2022). Traffic state estimation of urban road networks by multi-source data fusion: Review and new insights. Physica A Statistical Mechanics and its Applications, 595, 127079. https://doi.org/10.1016/j.physa.2022.127079

[21] Wang, S., Zhao, J., Shao, C., Dong, C., & Yin, C. (2020). Truck Traffic Flow Prediction Based on LSTM and GRU Methods With Sampled GPS Data. IEEE Access, 8, 208158–208169. https://doi.org/10.1109/access.2020.3038788

[22] Zafar, N., Haq, I. U., Chughtai, J., & Shafiq, O. (2022). Applying Hybrid Lstm-Gru Model Based on Heterogeneous Data Sources for Traffic Speed Prediction in Urban Areas. Sensors, 22(9), 3348. https://doi.org/10.3390/s22093348

[23] Chen, K., Chen, F., Lai, B., Jin, Z., Liu, Y., Li, K., Wei, L., Wang, P., Tang, Y., Huang, J., & Hua, X. (2020). Dynamic Spatio-Temporal Graph-Based CNNs for Traffic Flow Prediction. IEEE Access, 8, 185136–185145. https://doi.org/10.1109/access.2020.3027375

[24] Ge, Z., Chai, W. K., Duanmu, J., & Katos, V. (2022). Hybrid deep learning models for traffic prediction in large-scale road networks. Information Fusion, 92, 93–114. https://doi.org/10.1016/j.inffus.2022.11.019

[25] Yu, H., Jiang, R., He, Z., Zheng, Z., Li, L., Liu, R., & Chen, X. (2021). Automated vehicle-involved traffic flow studies: A survey of assumptions, models, speculations, and perspectives. Transportation Research Part C Emerging Technologies, 127, 103101. https://doi.org/10.1016/j.trc.2021.103101

[26] Zhai, C., Chen, X., Yan, C., Liu, Y., & Li, H. (2020). Ecological Cooperative Adaptive Cruise Control for a Heterogeneous Platoon of Heavy-Duty Vehicles With Time Delays. IEEE Access, 8, 146208–146219. https://doi.org/10.1109/access.2020.3015052

[27] Sun, W., Wang, S., Shao, Y., Sun, Z., & Levin, M. W. (2022). Energy and mobility impacts of connected autonomous vehicles with co-optimization of speed and powertrain on mixed vehicle platoons. Transportation Research Part C Emerging Technologies, 142, 103764. https://doi.org/10.1016/j.trc.2022.103764

[28] Gong, J., Shang, J., Li, L., Changjian, Z., He, J., & Ma, J. (2021). A Comparative Study on Fuel Consumption Prediction Methods of Heavy-Duty Diesel Trucks Considering 21 Influencing Factors. Energies, 14(23), 8106. https://doi.org/10.3390/en14238106