Proceedings of the 4th International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/2026.37153

A Case Study on the Robustness of Traffic Signal Control from a Data Science Perspective: The RobustLight Diffusion Reinforcement Learning Framework

Peiyu Qian

Jiangsu Tianyi High School, Wuxi, China
ireneqian0428@gmail.com

Abstract. Damaged sensor data, often caused by noise interference and data loss, impairs the decision-making accuracy of Deep Reinforcement Learning (DRL) traffic signal control models. This paper takes RobustLight (ICML 2025) as the core case and analyzes it from a data science perspective. We introduce DRL-based traffic signal control and diffusion model principles, then examine RobustLight's dual-process framework, DSI algorithm, Denoise and Repaint modules, and its non-Markov loss design. Experiments on Jinan, Hangzhou and New York datasets under four adversarial attacks and sensor damage show that RobustLight achieves up to 50.43% improvement in state recovery, with ATT reduced by up to 42%. Finally, we discuss real-world challenges including anomaly-detection dependency, inference latency, and cross-domain generalization.

Keywords: Traffic signal control, Deep reinforcement learning, Diffusion model, Case study, Data science

1. Introduction

With increasing urbanization, traffic congestion has become a prominent problem in major cities around the world. Traffic Signal Control (TSC), which allocates passage time for traffic flows in different directions at intersections, is the key way to alleviate congestion and improve the efficiency of the road network [1]. In recent years, deep reinforcement learning (DRL) has made significant progress in the TSC field. Agents can continuously interact with the traffic environment and learn the optimal timing strategies based on reward feedback [1, 2]. However, most of the existing DRL-TSC studies assume that the sensor data are clean and reliable, while ignoring quality problems in data.

In the real traffic system, sensor data face two types of threats [2], and one is the adversarial noise attacks, including Gaussian noise, uniform random noise (U-rand), maximum action difference (MAD) attack and minimum Q-value (MinQ) attack. The second is the damage of physical sensors, which is caused by weather, human factors or hardware failures, making it less likely to observe some state dimensions. Sensor attacks targeting traffic signal systems have been demonstrated at security conferences such as DEF CON 22, where attackers can artificially create congestion by changing geomagnetic sensor data [3, 4]. When the input states of DRL models are contaminated or

DOI: 10.54254/2755-2721/2026.37153

partially missing, decision-making performance drops sharply. For instance, under MAD attacks, the average travel time of some baseline models can rise from approximately 300 seconds to over 700 seconds [2].

At the industry level, adaptive traffic signal systems such as Surtrac in Pittsburgh and Google Project Green Light have been deployed in cities and have achieved reductions in delay [5, 6]. However, these systems mainly focus on optimizing timing itself and lack specific defense mechanisms against sensor data corruption. For instance, Surtrac optimizes signal timing in real time based on roadside cameras and radars [5], and Google Project Green Light extracts transportation mode from Google Map's floating-car data, providing optimized time-allocation suggestions to urban engineers [6]. Neither system is equipped with a dedicated data-recovery module. This means that when the roadside sensors malfunction or encounter adversarial perturbation, Surtrac receives a damaged input, which may result in a suboptimal timing scheme. Although Green Light has a relatively low reliance on physical sensors, it still lacks data robustness guarantee in areas where the coverage of floating vehicles is sparse or biased. Therefore, maintaining the robustness of TSC systems in unreliable data environments has become a key challenge in data science.

This article takes RobustLight [2] (ICML 2025) as the core case. Analyzing from the perspective of data science, specific goals are: (1) Explain the principles of diffusion models and its applicability in restoring time series traffic data, (2) Conduct an analysis of the framework design, core algorithms, data processing flow and model architecture of RobustLight, (3) Analyze its experimental datasets, perturbation settings, evaluation indicators and key results, (4) Discuss the data science challenges and limitations this framework may face in actual deployment.

2. Technical background

2.1. Traffic signal control and MDP modeling

TSC can be modeled as Markov decision process (MDP) [1, 7]. At each decision moment t, the agent gets the environmental state (such as the queue length of each lane and the number of vehicles traveling), selects an action a_t (signal phase), receives a reward r_t, and transitions to the next state s_{t+1}. The reward is commonly defined as the negative queue length because minimizing queues directly reduces vehicle delay and congestion, so maximizing this negative value encourages the controller to serve accumulated demand and improve throughput over time.

In DRL-TSC, frequently-used state representations include efficient pressure (EP) and the number of running vehicles (RV). The action is defined as traffic signal phases, and the reward is the negative value of queue length [2]. The state trajectory $\tau_t^s = \{s_1, s_2, \ldots, s_t\}$ records historical time sequence, providing context for restoring time series data.

2.2. The decision-making mechanism of DRL in traffic signal control

Deep reinforcement learning (DRL) combines deep neural networks with reinforcement learning to handle high-dimensional state spaces [8]. In TSC, DRL agents use traffic states (e.g., queue length, traffic flow) as input and output signal phase decisions, optimizing time-allocation strategies through reward feedback [1]. When sensor input is contaminated by noise, the phase decisions made by the network based on wrong states will directly lead to congestion aggravation, which makes RobustLight need to restore data before making decisions.

DOI: 10.54254/2755-2721/2026.37153

2.3. Foundations of diffusion models

Diffusion models are a class of probabilistic generative models that generate samples through a forward noising process and a reverse denoising process [9]. In the forward process, Gaussian noise is gradually added to the raw data x_0 within T steps. The noise data at any step t can be obtained through reparameterization:

$$q(x_t|x_0) = N(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I) \tag{1}$$

where $\alpha_t = 1 - \beta_t$ and β_t follow a predefined variance schedule [9].

Guided diffusion introduces external conditions during the reverse process to control the generated content. Classifier-free guidance integrates conditional information into the mean network:

$$\mu_{\text{guided}} = \mu_\theta(x_t, t) + \omega(\mu_\theta(x_t, t, c) - \mu_\theta(x_t, t)) \tag{2}$$

in which ω controls the trade-off between conditional generation and unconditional generation [10]. This mechanism enables the diffusion model to infer the missing state dimensions based on known sensor data.Diffusion models are suitable for restoring time-series traffic data because they can learn the high-dimensional distribution of traffic states and restore the original distribution under noise corruption. The conditional guidance mechanism supports the completion of missing values with known data as constraints, and the step-by-step denoising process is naturally suitable for processing time series data.

2.4. Problem definition of data corruption

RobustLight addresses two types of data corruption [2]:

(1) Adversarial noise attacks: perturbations are added to the state s_t. This includes four types:

1. The Gaussian noise attack: it simulates natural and continuous physical disturbances. For instance, thermal noise generated by sensors due to temperature changes and electromagnetic wave jitter.

$$\tilde{s}_t = s_t + k \cdot N(\mu, \sigma^2) \tag{3}$$

2. The uniform random (U-rand) noise attack: this represents bounded but unpredictable measurement errors, such as quantization errors or irregular interference within a known range.

$$\tilde{s}_t = s_t + k \cdot U(-I, I) \tag{4}$$

3. The maximum action difference (MAD) attack: this attack selects noise within an l_∞ ball that maximizes the difference between two policies, which models policy-divergence attacks in multi-policy redundant systems [2].

DOI: 10.54254/2755-2721/2026.37153

4. The minimum Q-value (MinQ) attack: it selects the noise that minimizes the Q-value as the perturbation, which models Q-value poisoning that tricks the agent into overly conservative timing [2].

(2) Physical sensor damage: some sensor dimensions are unobservable, denoted as:

$$\tilde{s}_t = Mask \cdot s_t \tag{5}$$

In the experiments, two damage ratios were set: 25% (westbound sensor failure) and 50% (westbound + eastbound sensor failure) [2].

3. Case study: RobustLight

3.1. Overview of the framework

RobustLight adopts a "dual-process, four-algorithm" framework that can be integrated into any existing TSC algorithm without modifying the original algorithm.

The dual process alternates between no-attack training (simultaneously training TSC and DSI agents) and attack testing (detecting corruption, restoring via Denoise or Repaint, then feeding the restored state to the TSC agent). The four algorithms consist of a TSC controller, a DSI agent, a Denoise module for noise attacks, and a Repaint module for sensor damage [2].

3.2. Data perspective: dataset and perturbation settings

RobustLight uses the CityFlow simulator [11] and real road-network datasets from three cities [2]: Jinan (12 intersections, 3 datasets), Hangzhou (16 intersections, 4 datasets), and New York (192 intersections, 1 dataset). Jinan and Hangzhou represent small-to-medium scale road networks, and New York represents a large-scale urban network. This scale-based sampling tests whether the framework remains effective as network size increases. All tests lasted 60 minutes, and results were averaged over 10 independent runs. Experiments were conducted on an NVIDIA P100 GPU [2].The perturbation settings cover all scenarios defined in Section 2.4: four adversarial attacks are tested under different noise intensities (k = 3.5, 4.0), the sensor damage is set at two ratios, 25% and 50%. This perturbation design enables the experimental results to reflect the robustness boundaries of the model under different data quality conditions.

3.3. Model architecture and core algorithms

3.3.1. DSI agent

The DSI agent adopts the actor-critic architecture [2]. Its state is $S_t = \left(a_{t-1}, \tau_{t-1}^{s}\right)$, which is the trajectory of TSC's actions and states in the previous moment. The action is $\hat{s}_t$, which is the clean state after recovery. The reward is negative queue length. Critic network adopts the same U-Net architecture as actor [12], and its last layer outputs the Q-value, which is updated through the Bellman error. The actor network employs an action gradient ascent update strategy and performs a reverse denoising process through an improved conditional diffusion model to predict the original state.

DOI: 10.54254/2755-2721/2026.37153

3.3.2. Non-Markov loss function

Traditional diffusion models use Markov losses that only consider the current moment. RobustLight proposes a non-Markov loss function [2], which, in addition to the denoising error at the current moment, also takes into account the cumulative error at future moments, enabling the model to consider the impact on following states' prediction when restoring the current state. This design is important for time series traffic data because the recovery error at the current moment can propagate along the time axis.

3.3.3. Customized beta schedule

For medium and small-scale noise, RobustLight has designed a customized linear beta scheduling [2]:

$$\beta_i = 1 - \alpha_i = \frac{\bar{\alpha}_{i-1}}{1-\bar{\alpha}_i} \cdot e^{-\frac{b}{(i+a+c)}} \tag{6}$$

Unlike standard cosine or linear scheduling that generates data from pure noise, this scheduling, in conjunction with the U-Net conditional diffusion model for noise prediction, is more suitable for traffic state denoising tasks. Specifically, a controls the initial noise level of the diffusion process to determine the extent to which the initial state is disturbed, b controls the rate at which noise changes with the number of diffusion steps, and c controls the residual noise level at the end of diffusion.

3.3.4. Denoise algorithm

During the testing phase, when detecting a noise attack, the Denoise algorithm inputs the corrupted state and the diffusion timestep i into the actor diffusion network, and performs reverse denoising:

$$\widetilde{A}_t^{i-1} = \frac{1}{\sqrt{\alpha_i}} (\widetilde{A}_t^i - \frac{1-\alpha_i}{\sqrt{1-\bar{\alpha}_i}} \epsilon_\theta(\widetilde{A}_t^i, S_{t-1}, i)) + \sqrt{\widetilde{\beta}_i} \cdot \epsilon \tag{7}$$

After K denoising steps, the recovered state is obtained [2].

3.3.5. Repaint algorithm

The Repaint algorithm is adapted from the image restoration method [13], and its core idea is to infer unknown dimensions from known sensor data. For the mask matrix m, the known parts are directly sampled from the known states, and the unknown parts are inferred from the conditional diffusion model. The combination of the two yields the complete recovery state. The algorithm repeats the sampling process U times at each diffusion step to enhance the constraint effect of the known data, and outputs the complete recovery state.

3.4. Evaluation dimensions

RobustLight is evaluated from two perspectives: transportation business and data recovery [2].

DOI: 10.54254/2755-2721/2026.37153

(1) Transportation business indicator is ATT, average travel time, defined as the average time taken by all vehicles from origin to destination (in seconds), where lower is better,

(2) Data recovery indicators: $\mathrm{E}_{\text{denoise}} = \frac{1}{\mathrm{N}} \sum |\hat{\mathrm{s}} - \mathrm{s}|$, which is the absolute error between the restored state and the original state, PSNR (Peak Signal-to-Noise Ratio), MAE (Mean Absolute Error)

(3) Distribution visualization: Using t-SNE, the high-dimensional traffic state is reduced to two dimensions [14]. By comparing the distributions among the original state, the attacked state, and the recovered state, the violin plots show the degree of morphological matching of the data distribution.

3.5. Experiment results and ablation analysis

3.5.1. Performance under noise attacks

On the Jinan and Hangzhou datasets, RobustLight significantly outperformed the baselines under all four types of attacks [2]. Take the JiNan1 dataset and the CoLight algorithm as examples: when there is no attack, the ATT is approximately 285 seconds, Under the MAD attack (k=3.5), the ATT of the original CoLight soared to 538 seconds (±132), while that of the RobustLight after recovery was only 312 seconds (±4). Under the MinQ attack, the original CoLight ATT lasted for 549 seconds, and the RobustLight recovered for 320 seconds. Overall, RobustLight significantly reduced the average travel time (ATT) of existing TSC algorithms, with improvement in state recovery performance reaching up to 50.43% [2].

3.5.2. Performance under sensor damage

With 25% sensor damage, for example on HangZhou1 with Advanced-CoLight, the baseline ATT was 386 seconds, and it dropped to 328 seconds after RobustLight recovery. At 50% damage, the original algorithm's ATT took 842 seconds, and after RobustLight recovery, it was 542 seconds [2]. The PSNR and MAE indicators also verified the recovery quality: under Gaussian noise, after using RobustLight, the PSNR increased from 6.71 to 7.20, and the MAE decreased from 6.26 to 5.42 [2]. Across different road networks and baseline controllers, RobustLight achieves up to 50.43% state-recovery accuracy for corrupted sensor inputs. Table 1 summarizes these results.

Table 1. Summary of key experimental results of RobustLight

Scenario	Dataset / Algorithm	Baseline ATT (s)	ATT with RobustLight (s)	ATT Improvement
MAD attack k=3.5	JiNan1 / CoLight	538.66±132.69	312.27±4.07	−42.0%
MinQ attack k=3.5	JiNan1 / CoLight	549.41±120.08	320.48±3.29	−41.7%
25% sensor damage	HangZhou1 / Adv-CoLight	401.87±48.31	328.26±12.67	−18.3%
50% sensor damage	HangZhou1 / Adv-CoLight	842.29±235.36	541.56±122.99	−35.7%

Note. Improvement = (ATT with RobustLight − baseline ATT) / baseline ATT. A negative value denotes reduction in average travel time and thus the improved traffic efficiency.

DOI: 10.54254/2755-2721/2026.37153

3.5.3. Ablation experiments and cross-city migration

The ablation experiments verified the contributions of each component: removing the linear beta scheduling degraded performance, confirming its effectiveness for medium and small-scale noise; removing the non-Markov loss caused a significant drop, demonstrating its critical role in time-series recovery. In cross-city migration, the model trained on Jinan transferred to New York and still alleviated noise attacks effectively [2].

3.6. Limitations in experiments

(1) Inference delay: The larger the noise range, the more diffusion steps are required and the longer the recovery time. In large-scale road networks, the dual increase in the number of diffusion steps and intersections leads to slow reasoning [2]. The author plans to use DDIM to reduce the number of diffusion steps and divide the intersection into sub-regions to accelerate reasoning [15].

(2) Anomaly detection dependency: RobustLight assumes that the attack has been identified by the anomaly detection module, which means that noise is detected through outlier detection and analysis of variance, and missing data are identified by checking null/NaN values [2]. When the detection rate drops, recovery performance also degrades, so the overall robustness of the system is bounded by the performance of the anomaly detector.

4. Discussion

4.1. Generalization and simulation-to-reality gap

(1) Cross-domain generalization: although migration experiments show some cross-city transferability, traffic-flow distributions (peak patterns, road topologies) differ significantly across cities, and performance may degrade in unseen cities. Meta-learning or domain adaptation could mitigate this issue. (2) Simulation-to-reality gap: the four attack types are mathematically defined and may not fully match real sensor faults such as temperature drift or rain interference; validation on physical sensor data is needed [2]. These gaps highlight the need for evaluation on real-world sensor data rather than solely on simulated attack scenarios.

4.2. Comparison with industrial systems

Commercial systems like Surtrac and Google Project Green Light have reduced delays in real deployments [5, 6], but they rely on floating-car data or traditional optimization and lack dedicated recovery modules for sensor data corruption. RobustLight addresses this gap with a data-driven robustness approach, offering a reference for next-generation intelligent transportation systems where data quality is as critical as algorithm design.

5. Conclusion

This paper takes RobustLight as the core case and deeply analyzes the data robustness issue in traffic signal control from the perspective of data science. Research shows that in intelligent transportation systems, model performance is often bounded more by input data quality than by the decision-making algorithm itself. As a powerful tool for time-series data restoration, the diffusion model, through conditional guidance and a non-Markov loss design, can effectively restore traffic states contaminated by noise or partially missing data, enabling existing TSC algorithms to maintain

DOI: 10.54254/2755-2721/2026.37153

performance in unreliable data environments (with a maximum ATT improvement of 50.43%). The framework points toward a future where data quality assurance is built directly into traffic control pipelines rather than treated as an afterthought.

References

[1] Rasheed, F., Yau, K.-L. A., Noor, R. M., Wu, C., & Low, Y.-C. (2020). Deep reinforcement learning for traffic signal control: A review. *IEEE Access, 8*, 208016–208044. https://doi.org/10.1109/ACCESS.2020.3034141

[2] Li, M., Wang, J., Yu, G., Wang, X., Chen, Q., Ni, W., Li, L., & Peng, H. (2025). RobustLight: Improving robustness via diffusion reinforcement learning for traffic signal control. In *Proceedings of the 42nd International Conference on Machine Learning (ICML 2025)*. PMLR.

[3] Laszka, A., Potteiger, B., Vorobeychik, Y., Amin, S., & Koutsoukos, X. (2016). Vulnerability of transportation networks to traffic-signal tampering. In *2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCPS)* (pp. 1–10). IEEE. https://doi.org/10.1109/ICCPS.2016.7479130

[4] Chen, Q. A., Yin, Y., Feng, Y., Mao, Z. M., & Liu, H. X. (2018). Exposing congestion attack on emerging connected vehicle based traffic signal control. In *Network and Distributed System Security Symposium (NDSS 2018)*.

[5] Carnegie Mellon University. (2019, October 25). Surtrac allows traffic to move at the speed of technology. *Carnegie Mellon University*. https://www.cmu.edu/news/stories/archives/2019/october/traffic-moves-at-speed-of-technology.html

[6] Boston. (2025, June 17). Mayor Wu announces expansion of Project Green Light signal optimization program. https://www.boston.gov/news/mayor-wu-announces-expansion-project-green-light-signal-optimization-program

[7] Abdulhai, B., Pringle, R., & Karakoulas, G. J. (2003). Reinforcement learning for true adaptive traffic signal control. *Journal of Transportation Engineering, 129*(3), 278–285. https://doi.org/10.1061/(ASCE)0733-947X(2003)129:3(278)

[8] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature, 518*(7540), 529–533. https://doi.org/10.1038/nature14236

[9] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems, 33*, 6840–6851.

[10] Ho, J., & Salimans, T. (2022). Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598.

[11] Zhang, H., Feng, S., Liu, C., Ding, Y., Zhu, Y., Zhou, Z., Zhang, W., Yu, Y., Jin, H., & Li, Z. (2019). CityFlow: A multi-agent reinforcement learning environment for large scale city traffic scenario. In *The World Wide Web Conference (WWW '19)* (pp. 3620–3624). https://doi.org/10.1145/3308558.3314139

[12] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015* (pp. 234–241). Springer. https://doi.org/10.1007/978-3-319-24574-4_28

[13] Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., & Van Gool, L. (2022). Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2022)* (pp. 11461–11471). https://doi.org/10.1109/CVPR52688.2022.01117

[14] Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research, 9*(11), 2579–2605.

[15] Song, J., Meng, C., & Ermon, S. (2020). Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.