EXPLORING ANTIFRAGILITY IN TRAFFIC NETWORKS: ANTICIPATING DISRUPTIONS WITH REINFORCEMENT LEARNING

Linghang Sun, Corresponding Author
Institute for Transport Planning and Systems
ETH Zurich, 8093 Zurich, Switzerland
Email: linghang.sun@ivt.baug.ethz.ch

Michail A. Makridis, Ph.D., Corresponding Author
Institute for Transport Planning and Systems
ETH Zurich, 8093 Zurich, Switzerland
Email: michail.makridis@ivt.baug.ethz.ch

Alexander Genser, Ph.D.
Institute for Transport Planning and Systems
ETH Zurich, 8093 Zurich, Switzerland
Email: alexander.genser@ivt.baug.ethz.ch

Cristian Axenie, Ph.D.
Computer Science Department
Technische Hochschule Nürnberg, 90489 Nürnberg, Germany
Email: cristian.axenie@th-nuernberg.de

Margherita Grossi, Ph.D.
Intelligent Cloud Technologies Lab
Huawei Munich Research Center, 80992 Munich, Germany
Email: margherita.grossi@huawei.com

Anastasios Kouvelas, Ph.D.
Institute for Transport Planning and Systems
ETH Zurich, 8093 Zurich, Switzerland
Email: kouvelas@ethz.ch

Word Count: 7923 words + 4 table(s) × 250 = 8923 words

Submission Date: October 11, 2023
ABSTRACT

The optimal operation of transportation networks is often susceptible to unexpected disruptions, such as traffic incidents and social events. Established control strategies rely on mathematical models that can not always deal with real-world uncertainty and their efficiency degrades remarkably under significant disruptions. While previous research works have dedicated efforts to enhancing the robustness or resilience of transportation systems against disruptions, in this paper, we use the concept of antifragility to better design a traffic control strategy for urban road networks. Antifragility sets itself apart from robustness and resilience as it represents a system’s ability to not only withstand stressors, shocks, and volatility but also thrive and enhance performance in the presence of such disruptions. Hence, modern transport systems call for solutions that are more antifragile. In this work, we propose a model-free deep Reinforcement Learning (RL) algorithm to regulate perimeter control in a two-region urban traffic network to exploit and strengthen the learning capability of RL under disruptions and achieve antifragility. By incorporating antifragility attributes based on the change rate and curvature of traffic states into the RL framework, the proposed algorithm further gains knowledge on the change rate of the traffic state, which helps with anticipating imminent disruptions. An additional dynamic redundancy term is also integrated into the RL algorithm to enhance the performance of the traffic control strategy under disruption scenarios. We present the proposed RL algorithm as being antifragile in comparison to a model-based control-theory method and a state-of-the-art RL algorithm, showcasing its ability to learn from past events and outperform the other methods, particularly under increasing disruptions.

Keywords: Antifragility, Reinforcement Learning (RL), Traffic disruptions, Perimeter control, Macroscopic Fundamental Diagram (MFD).
INTRODUCTION

Transportation networks serve as vital conduits for the movement of people and goods. The optimization of these transportation systems has become a focal point for transportation researchers, resulting in a multitude of research endeavors and practical implementations in the field of Intelligent Transportation Systems (ITS), as in (1–3). Given that various sorts of incidents, such as traffic accidents, social events, unfavorable weather conditions, etc., often occur unexpectedly in real-world networks, examining the robustness and resilience of the transportation system’s performance to handle disturbances and disruptions is crucial in the research of ITS (4).

With the ever-growing population in the cities and urbanization, traffic systems have gained both volume and complexity. For instance, according to the Federal Statistical Office of Switzerland, private motorized road traffic experienced a steady increase of 54% in total between 1980 and 2019 (5). Also, as pointed out in (6), the rise in traffic volume can result in an escalation of intensified congestion and more traffic incidents. Therefore, new requirements have been constantly posed to the urban road networks, to secure a decent level of service even when confronted by various disruptions with unforeseen magnitude. This also raises the question of whether the current levels of robustness and resilience in transportation systems are sufficient to handle such challenges.

To address such issues, “antifragility” has shed light on a feasible solution and provides a possible new evaluation criterium. The concept of antifragility, first introduced in the bestseller Antifragile: Things that Gain from Disorder by Nassim Nicholas Taleb in 2012 and mathematically explained in (7) and (8), provides insights on designing systems that can benefit from disruptions and perform better under growing disorder. Since then, antifragility has gained great interest from both the public and also academia, particularly in the field of risk engineering (9). The potential of antifragility can also be leveraged within transportation systems to tackle the increasingly severe traffic problems modern cities face from day to day.

However, how to identify and even design antifragile transportation systems remains understudied. One promising approach to induce antifragility in urban road networks is by using learning-based algorithms. With the rapid advancement of big data and sensor techniques, using Machine Learning (ML), particularly Reinforcement Learning (RL)-based methods has also become a trending practice in designing traffic management and control strategies in ITS (10–12). Through interacting with a given environment over time, an RL agent enhances its decision-making ability, maximizing the cumulative reward it can receive (13). One advantage of RL over established controllers is that it allows for more flexibility and competence to deal with multivariate nonlinearities in complex environments (14, 15). When deployed to an environment subject to variations, RL agents can gradually adjust their decision-making by interacting with the new environment, whereas established controllers, such as PID controllers, may need intensive manual tuning of parameters (16). Also, additional information can be fed to the RL agent with ease as a representation of the observable state of the environment, regardless of the knowledge of the model dynamics. In contrast, structured model dynamics of the system are often required when designing a controller (17). This feature of RL can also be exploited to explore hidden information from the abundance of data in the field of transportation, and modern traffic control systems can potentially learn from traffic disruptions, react proactively with merely early signs, and exhibit increasingly better performance as disruptions escalate.

The main goal of this paper is to first distinguish the concept of antifragility as opposed to other more commonly used terms, i.e., robustness and resilience, under the context of trans-
portation. Then we demonstrate how an antifragile RL algorithm can be designed to empower traffic control strategies, such as perimeter control in this study, to be antifragile against disruptions with various magnitudes. We also propose a quantitative approach to evaluate and compare the antifragility property among different methods.

LITERATURE REVIEW

This section provides a review of the relevant literature with three topics covered in this work. First, a macroscopic traffic model and the control strategy applied in this paper are introduced, which serve as the basis of model dynamics for our simulation environment. Following that, since antifragility itself is a relatively new topic and the link between RL and antifragility is yet absent, we present state-of-the-art research on leveraging RL algorithms to enhance the robustness and resilience of control strategies. Finally, based on the literature, we introduce antifragility conceptually and how an antifragile system can be designed.

Alleviating urban network congestion can be realized through various traffic control strategies. One common practice of achieving less congested networks is through traffic signal control. This microscopic control approach on an intersection level can be extended to a corridor or a larger urban network with proper coordination. However, fine-grid coordinated microscopic signal control can be computationally demanding to be implemented in real-time on a large-scale urban network (18). Since (19) proved the existence of the Macroscopic Fundamental Diagram (MFD) theoretically and (20) demonstrated the presence of the MFD with empirical data, the relationship between average traffic flow and density has been established through the aggregation of individual microscopic data points. This relationship has paved the way for the development of control strategies on a macroscopic level, enabling more computationally feasible real-time control strategies for large-scale networks (21). With the MFD and the identification of the critical parameters, several control methods have been proposed to enhance the performance of road networks, such as perimeter control (22–24), pricing (25–27) and route guidance (28–30).

Perimeter control is among the methods that have attracted immense attention and research. Real-world implementation as shown in (31) also demonstrated the applicability of perimeter control as an effective approach to regulating urban traffic. By refraining the incoming vehicles from adjacent regions into a protected zone, the traffic density in the protected area remains below the critical density, so a satisfactory level of service can be upheld (32). (22) proposed an optimal perimeter control method using Model Predictive Control (MPC) and proved its effectiveness compared to a greedy controller in a cordon-shaped network. However, within the context of this paper, the problem with MPC is that it necessitates an abundance of a priori knowledge (33), which may not be readily available in real-world applications and is hard to forecast with high fidelity. Also, since disruptions, such as traffic accidents, can neither be known nor accurately predicted in advance, the performance will be negatively affected when large disruptions would happen in real-world networks. Another problem with the previous works is the MFD heterogeneity and uncertainty. To address the heterogeneity of the MFDs in different sub-regions, a substantial amount of effort has been made in investigating the partitioning algorithms so that a well-defined MFD can be referred to for a certain sub-network (34, 35). Following the partitioning, the cordon network has been expanded into multi-reservoir systems and these works implemented perimeter control based on methods either without coordination (36) or with coordination (37). (23) combined a model-based perimeter control design with an online adaptive fine-tuning algorithm. Although these studies are based on the assumption of static well-defined MFDs, in reality,
MFDs can hardly be always well-defined, as shown in (38) with loop detector data over a year. Research works based on real-world data have demonstrated that certain types of disruptions, such as adverse weather conditions (39) and traffic incidents (40), can alter the shape of the MFDs and even a recovery from the peak-hour congestion may lead to a hysteresis (41). These phenomena could potentially violate the mathematical model that serves as the foundation for the established model-based perimeter controllers. Also, increasing model errors and disturbances will cause the effectiveness of these control strategies to decline progressively (42).

To tackle the parameter uncertainties in the models caused by real-world disruptions, recent years have also witnessed a growing trend towards utilizing non-parametric learning-based approaches in traffic control (43), owing to the abundance of data collected from multiple sources such as loop detectors, video cameras, and mobile networks. Among different ML algorithms, RL has been researched extensively in transportation operations, e.g., traffic light control (44, 45), dynamic pricing (46, 47), delay management (48, 49). Particularly for perimeter control, recent works from (50) and (51) have illustrated the capability of RL algorithms to achieve comparable or even better performance in perimeter control compared to the established control methods.

Before introducing the concept of antifragility, two related terms robustness and resilience commonly used to evaluate traffic control strategies are compared here. As stated in (52), the concepts of robustness and resilience vary significantly according to their usage in different disciplines. Also, (53, 54) mentioned that these terms are sometimes used interchangeably in transportation research works. While researchers have endeavored to produce review articles in either the applications of RL in transportation (12, 55) or robustness/resilience in transportation systems (56, 57), a summarization of using RL algorithms in transportation to achieve robust or resilient urban networks remains absent. Here we made a brief review of the established works using RL to regulate urban traffic, either microscopically or macroscopically, that touched upon proving the robustness or resilience of their proposed methods. By transferring the knowledge of robust or resilient design, we can achieve antifragility in urban networks through similar approaches. Table 1 shows a selection of papers around perimeter control and traffic signal control designed with RL algorithms. The objectives, the application scenarios, and the property of being either robust or resilient, as claimed by the authors, are also listed in the table.

As can be seen, 2 out of the 10 reviewed papers are focused on the resilience of the RL-based traffic control strategies and the studied scenarios are traffic demand uncertainties and MFD errors (60) as well as lane closures (66). When examining the system’s robustness, the types of scenarios under consideration become more diverse, including incidents (61, 63), sensor failures (65), other than varying demand patterns. However, as mentioned above, the common interchangeable usage of robustness and resilience makes it challenging to establish a definitive differentiation between robustness and resilience based on their application scenarios. Therefore, we follow the principles proposed by (56), that robustness is concerned with assessing a system’s capacity to preserve its initial state and resist performance deterioration in the presence of uncertainty and disturbances, while resilience stresses the capability and speed of a system to recover from disruptions to the original state, as in the lane closure scenario in (66). Since maintaining the original state is difficult for systems when confronted with significant disruptions, such events are usually associated with the study of resilience (67).

Table 2 summarizes the RL setups to demonstrate the property of being either robust or resilient of each traffic control strategy. In these papers, some authors proved the superior robustness or resilience of their proposed methods directly through testing against the established methods as...
### TABLE 1: Traffic Control Strategies Using RL with Demonstrated Robustness or Resilience

<table>
<thead>
<tr>
<th>Literature</th>
<th>Category</th>
<th>Objective</th>
<th>Scenarios</th>
<th>Robust</th>
<th>Resilient</th>
</tr>
</thead>
<tbody>
<tr>
<td>(51)</td>
<td>Perimeter control</td>
<td>Maximize trip completion under uncertainties</td>
<td>Demand and MFD uncertainties</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>(58)</td>
<td>Perimeter control</td>
<td>Minimize total time spent under uncertainties/disturbances</td>
<td>Demand uncertainties, demand surge</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>(59)</td>
<td>Perimeter control</td>
<td>Maximize trip completion under uncertainties</td>
<td>Demand uncertainties</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>(60)</td>
<td>Perimeter control</td>
<td>Maximize trip completion under uncertainties and errors</td>
<td>Demand and MFD uncertainties</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>(61)</td>
<td>Traffic signal control</td>
<td>Minimize average travel time under disturbances</td>
<td>Incidents, sensor noise</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>(62)</td>
<td>Traffic signal control</td>
<td>Minimize queue length and delay, maximize vehicle speed and trip completion under demand uncertainties</td>
<td>Demand uncertainties, demand surge</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>(63)</td>
<td>Traffic signal control</td>
<td>Minimize total travel time, total delay, and total stop time</td>
<td>Demand surges, incidents, sensor failures</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>(64)</td>
<td>Traffic signal control</td>
<td>Minimize queue length at the intersection under uncertainties</td>
<td>Demand uncertainties</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>(65)</td>
<td>Traffic signal control</td>
<td>Minimize average vehicle delay under different scenarios</td>
<td>Demand uncertainties, sensor noise</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>(66)</td>
<td>Traffic signal control</td>
<td>Minimize average travel time under disruptions</td>
<td>Lane closure</td>
<td>×</td>
<td></td>
</tr>
</tbody>
</table>

1. benchmarks, whereas others induced such properties by adding specific terms (italicized in Table 2) in the state space of the RL-algorithm. As a result, the algorithms are given additional information, which potentially empowers the algorithms to anticipate ongoing disruptions. For instance, (63) induced robustness by adding the elapsed time since the last green signal for each phase, (64) experimented with speed or pressure (residual queue) as an additional state representation in the state space of the RL algorithm, (62) supplemented the control policies of neighboring intersec-
### TABLE 2: RL Setups for Studying Robustness or Resilience in Traffic Control

<table>
<thead>
<tr>
<th>Literature</th>
<th>Robustness</th>
<th>Resilience</th>
<th>Benchmark</th>
<th>RL method</th>
<th>State</th>
<th>Action</th>
<th>Reward</th>
</tr>
</thead>
<tbody>
<tr>
<td>(51)</td>
<td>Tested</td>
<td></td>
<td>No-control, MPC</td>
<td>DDPG</td>
<td>Number vehicles, traffic demand</td>
<td>Perimeter control variables</td>
<td>Trip completion</td>
</tr>
<tr>
<td>(58)</td>
<td>Tested</td>
<td></td>
<td>MPC</td>
<td>AC-IRL</td>
<td>Number vehicles</td>
<td>Perimeter control variables</td>
<td>Divergence from critical accumulation</td>
</tr>
<tr>
<td>(59)</td>
<td>Tested</td>
<td></td>
<td>FT/Webster/ MP + PI</td>
<td>DQN</td>
<td>Flow, density, speed</td>
<td>Perimeter control variables</td>
<td>Ratio of flow over max flow</td>
</tr>
<tr>
<td>(60)</td>
<td>Induced</td>
<td></td>
<td>No-control, MPC</td>
<td>MADDPG</td>
<td>Number vehicles, traffic demand, congestion indicator</td>
<td>Perimeter control variables</td>
<td>Trip completion</td>
</tr>
<tr>
<td>(61)</td>
<td>Tested</td>
<td></td>
<td>Cross comparison</td>
<td>DQN, AC, SARSA</td>
<td>Phase, pressure</td>
<td>Green time duration</td>
<td>Queue length</td>
</tr>
<tr>
<td>(62)</td>
<td>Induced</td>
<td></td>
<td>IA2C, IQL-LR, IQL-DNN</td>
<td>MAA2C</td>
<td>Total delay, number vehicles, adjacent info</td>
<td>Signal configuration</td>
<td>Queue length, total delay</td>
</tr>
<tr>
<td>(63)</td>
<td>Induced</td>
<td></td>
<td>Self-comparison</td>
<td>DDQN</td>
<td>Queue length, elapsed time in each phase</td>
<td>Time extension, phase selection</td>
<td>Numerical derivative of queue length</td>
</tr>
<tr>
<td>(64)</td>
<td>Induced</td>
<td></td>
<td>Self-comparison</td>
<td>DDQN</td>
<td>phase info, queue length, speed, pressure</td>
<td>Phase configuration</td>
<td>Queue length, duration over detection</td>
</tr>
<tr>
<td>(65)</td>
<td>Tested</td>
<td></td>
<td>Fixed-time, actuated</td>
<td>DDQN</td>
<td>Phase, number vehicles, elapsed time, neighbour phases</td>
<td>No-change/ extend/ terminate</td>
<td>Queue length</td>
</tr>
<tr>
<td>(66)</td>
<td>Tested</td>
<td></td>
<td>Random, cyclical, demand, analytical+</td>
<td>DQN, DQN + heuristic</td>
<td>Vehicle occupancies</td>
<td>Green time duration</td>
<td>Negative pressure</td>
</tr>
</tbody>
</table>

1 Tested/Induced means whether the property of robustness or resilience is demonstrated through testing against other benchmark methods or the author claimed to have implemented specific methods to induce the robustness or resilience of the proposed method.

2 Italicized terms represent the additional information the authors applied and claimed as effective to design a robust/resilient system.

---

1 Since the concept of antifragility has been proposed, it has become an increasingly
2 important aspect to consider in system design.
3
4 Ever since the concept of antifragility has been proposed, it has become an increasingly
5 important aspect to consider in system design.
popular concept in many disciplines, such as economy \((68)\), business \((69)\), biology \((70)\), medicine \((71)\), and robotics \((72)\). However, current studies on antifragility in the field of engineering mostly pertain to post-disaster reconstruction efforts \((73–75)\). Leveraging the potential of antifragility for the daily operations and optimization of transportation systems is a new and unexplored notion.

Although not explicitly stated in the book or in any scientific paper, researchers’ understanding of antifragility is two-fold. The first understanding resembles adaptiveness. A robust system tends to maintain its original state amidst model and input uncertainties, resilience enables a system to recover from disruptions within a period of time. In contrast, antifragile systems aim to enhance performance in a progression of unexpected events, such as in \((73)\). These events are often referred to as black swan events \((76)\), suggesting that the magnitude of disruptions is unexperienced and likely to exceed the capacity of system robustness to manage. Another understanding, which may be less intuitive but can be grasped by considering its antonym, i.e., fragility, refers to an object or a system that suffers from exponentially growing loss when faced with linearly increasing disruptions \((8)\). This concept is characterized by concavity and can be mathematically formulated with Jensen’s inequality \(E[g(X)] \leq g(E[X])\). On the contrary, the nonlinear relationship between external stressors and responses for antifragile systems is convex with \(E[g(X)] \geq g(E[X])\). Since the difference is not clearly distinguished, in our work, we will refer to the different understandings as antifragility type I (adaptiveness) and antifragility type II (concavity).

Researchers have also proposed methods to incentive antifragility property of a system by emphasizing the derivatives to capture the temporal evolution patterns of the system dynamics, i.e., how fast the system state deviates towards a possible black swan event, and the curvature of this deviation \((8, 71, 77)\). With this additional information, the system can anticipate ongoing disruptions and be more responsive to drastic changes. Similar to the function of redundancy in resilience \((78, 79)\), another feasible approach to induce system antifragility is to add redundancy in the system \((76, 80, 81)\).

In this work, we simulate a cordon-shaped urban road network, subject to demand disruptions. Rather than relying on a mathematical model and static traffic indicators to determine the best policies for perimeter control, we use a model-free RL-based approach that adopts the concept of antifragility. To achieve this, we integrate supplementary state and reward components derived from derivatives and redundancy into the RL algorithm. This integration allows the algorithm to exhibit the characteristic to acquire knowledge from previous disruptions and become antifragile. The key contributions of this work can be summarized as follows: This work is the first to embrace the concept of antifragility in the context of the daily operation of transportation systems. The study demonstrates the effectiveness of incorporating derivatives and redundancy to handle system disruptions. Also, the proposed antifragile model-free RL-based algorithm achieves better performance compared to both a model-based control strategy that requires a priori knowledge and the other state-of-the-art model-free RL-based methods.

To enhance the clarity of the paper’s structure, the study will proceed as follows: Section 6 formulates mathematically the cordon network and the model dynamics of perimeter control, and summarizes all the applied notations in this paper. Section 5 explains the applied control methods MPC and RL-based algorithms in detail and describes specifically the mathematical definition of antifragility, and how we can induce antifragility in a transportation system. Section 6 introduces the specifications of the tested environment and the evaluation metrics, while Section 7 presents the results to showcase the effectiveness and antifragility of our proposed method.
PROBLEM FORMULATION

This paper studies the problem of perimeter control between two homogeneous regions. A cordon-shaped urban network is investigated as in (22, 51), with the inner region representing a city center, as shown in Figure 1a, and traffic demand for an Origin-Destination (OD) pair from region i to region j at time t is denoted as \( q_{ij}(t) \). The inner and outer regions have different MFDs due to the difference in capacity to accommodate vehicles in the road networks within the city center and the surrounding region, defined as \( G_i(n_i(t)) \) as illustrated in Figure 1b. Given the total number of vehicles, denoted as \( n_i(t) \) with presence in region i at time t, the total trip completion rate, denoted as \( M_i(t) \) for this region i can be determined using the corresponding MFD, which comprises both the intraregional trip completion, i.e., \( M_{ii}(t) \) and the interregional transfer flow, i.e., \( M_{ij}(t) (i \neq j) \) with \( i, j \in \{1, 2\} \). In order to protect both regions from being overflown by possible high traffic demand, the percentage of the transfer flow allowed to go across the region perimeter at time t is regulated by two perimeter controllers denoted as \( u_{ij}(t) (i \neq j) \). A list of all notations used in this paper, including the notations used in defining the RL algorithm and the antifragile terms, is summarized in Table 3.

\[
\frac{dn_{ii}(t)}{dt} = q_{ii}(t) + u_{ji}(t) \cdot M_{ji}(t) - M_{ii}(t) \tag{1}
\]

\[
\frac{dn_{ij}(t)}{dt} = q_{ij}(t) - u_{ij}(t) \cdot M_{ij}(t), (i \neq j) \tag{2}
\]

The total trip completion, i.e., \( M_i(t) \) for region i at time t is calculated based on the trip
### TABLE 3: List of Notations

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. General notations in problem formulation</strong></td>
<td></td>
</tr>
<tr>
<td>$t$</td>
<td>Time</td>
</tr>
<tr>
<td>$\Delta t$</td>
<td>Time step</td>
</tr>
<tr>
<td>$t_{end}$</td>
<td>Total simulation time</td>
</tr>
<tr>
<td>$n_{ij}(t)$</td>
<td>Vehicle accumulation with OD from region $i$ to $j$ at time $t$</td>
</tr>
<tr>
<td>$n_i(t)$</td>
<td>Vehicle accumulation in region $i$ at time $t$</td>
</tr>
<tr>
<td>$u_{ij}(t)$</td>
<td>Perimeter control variables regulating flow from region $i$ to $j$ at time $t$</td>
</tr>
<tr>
<td>$q_{ij}(t)$</td>
<td>Traffic demand with OD pair $i$ and $j$ at time $t$</td>
</tr>
<tr>
<td>$G_i(n_i(t))$</td>
<td>Sum of trip completion and transfer flow in region $i$ at time $t$, equals to $M_i(t)$</td>
</tr>
<tr>
<td>$M_{ij}(t)$</td>
<td>Trip completion rate with OD from region $i$ to $j$ at time $t$</td>
</tr>
<tr>
<td>$n_{i,cap}(t)$</td>
<td>Maximal possible number of vehicles (jam accumulation) in region $i$ at time $t$</td>
</tr>
<tr>
<td>$n_{i,cru}(t)$</td>
<td>Critical number of vehicles with highest completion rate in region $i$ at time $t$</td>
</tr>
<tr>
<td>$J$</td>
<td>Objective function</td>
</tr>
<tr>
<td><strong>2. Notations in reinforcement learning</strong></td>
<td></td>
</tr>
<tr>
<td>$\mathcal{S}$</td>
<td>State space, the whole set of states the RL agent can transition to</td>
</tr>
<tr>
<td>$s_t$</td>
<td>$s_t \in \mathcal{S}$, the observable state in simulation at time $t$</td>
</tr>
<tr>
<td>$\mathcal{A}$</td>
<td>Action space, the whole set of actions the RL agent can act out</td>
</tr>
<tr>
<td>$a_t$</td>
<td>$a_t \in \mathcal{A}$, the action taken in simulation at time $t$</td>
</tr>
<tr>
<td>$\mathcal{R}$</td>
<td>The reward function for the RL agent</td>
</tr>
<tr>
<td>$r_t$</td>
<td>$r_t = \mathcal{R}(s_t,a_t)$, the received reward in simulation at time $t$</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Discount factor to favor rewards in the near future</td>
</tr>
<tr>
<td>$Q(s(t),a(t))$</td>
<td>Expected long-term return for taking action $a(t)$ in state $s(t)$ at time $t$</td>
</tr>
<tr>
<td>$\theta^\mu$</td>
<td>Weight of the deep neural network for the actor network</td>
</tr>
<tr>
<td>$\theta^Q$</td>
<td>Weight of the deep neural network for the critic network</td>
</tr>
<tr>
<td>$y_i$</td>
<td>Expected long-term return calculated with the target critic network</td>
</tr>
<tr>
<td>$L$</td>
<td>The loss of the critic network</td>
</tr>
<tr>
<td>$\rho^\beta$</td>
<td>All possible trajectories of $s_t$</td>
</tr>
<tr>
<td>$I$</td>
<td>The objective function for the actor network</td>
</tr>
<tr>
<td><strong>3. Notations for the additional antifragile terms applied in reinforcement learning</strong></td>
<td></td>
</tr>
<tr>
<td>$\epsilon(t)$</td>
<td>Additional reward term in RL based on derivatives and redundancy</td>
</tr>
<tr>
<td>$\omega_h$</td>
<td>The weight of first derivative in the additional reward term $\epsilon(t)$</td>
</tr>
<tr>
<td>$\omega_{\Delta h}$</td>
<td>The weight of second derivative in the additional reward term $\epsilon(t)$</td>
</tr>
<tr>
<td>$\alpha_i(t)$</td>
<td>Binary variable determining the term to be reward/penalty</td>
</tr>
<tr>
<td>$h_i(t)$</td>
<td>The first derivative of traffic state at time $t$</td>
</tr>
<tr>
<td>$\Delta h_i(t)$</td>
<td>The second derivative of traffic state at time $t$</td>
</tr>
</tbody>
</table>
accumulation, denoted by \( n_i(t) \) through MFD, defined as \( G_i(n_i(t)) \) and consists of two parts, intraregional trip completion, i.e., \( M_{ii}(t) \), in Eq. 3 and interregional transfer flow, i.e., \( M_{ij}(t) \) (\( i \neq j \)), in Eq. 4:

\[
M_{ii}(t) = \frac{n_{ii}(t)}{n_i(t)} \cdot G_i(n_i(t)) \quad (3)
\]
\[
M_{ij}(t) = \frac{n_{ij}(t)}{n_i(t)} \cdot G_i(n_i(t)), \quad (i \neq j) \quad (4)
\]
\[
n_i(t) = \sum_{j=1,2} n_{ij}(t) \quad (5)
\]

The objective function is to maximize the throughput of this cordon-shaped network, which is the sum of the intraregional trip completion, which is \( M_{ii}(t) \) in both regions.

\[
J = \max_{u_{ij}(t)} \int_0^{t_{\text{end}}} \sum_{i=1,2} M_{ii}(t) dt \quad (6)
\]
subject to the following boundary conditions:

\[
n_{ij}(t) \geq 0 \quad (7)
\]
\[
n_i(t) \leq n_i,\text{cap} \quad (8)
\]
\[
u_{\text{min}} \leq u_{ij}(t) \leq u_{\text{max}} \quad (9)
\]
\[
n_{ii}(t) \text{ and } n_{ij}(t) \text{ are intraregional and interregional vehicle accumulation and are non-negative values. } n_i,\text{cap} \text{ is the maximal possible number of vehicles accumulated in region } i, \text{ which can lead to a gridlock as shown in Figure 1b. } u_{\text{min}} \text{ and } u_{\text{max}} \text{ represent the lower and upper limit for the perimeter control variable } u_{ij}(t) \text{ for both directions and such applications are in line with (22, 51).}
\]
\[
\text{These bounds are due to the fact that perimeter control is normally implemented through signalization. While } u_{\text{max}} \text{ accounts for the lost time caused by the interchange between the red and green phases, } u_{\text{min}} \text{ describes that complete perimeter control through an indefinite long red light is rare in real-world cases.}
\]
In contrast to the control-based strategies, for the RL-based algorithms, following the idea of redundancy, an additional term \( \varepsilon(t) \) is added into the objective function \( J_{RL} \), referred to as reward
\[
r_i \in \mathcal{R} \text{ under the context of RL, leading to:}
\]

\[
J_{RL} = \max_{u_{ij}(t)} \int_0^{t_{\text{end}}} \left[ \sum_{i=1,2} M_{ii}(t) + \varepsilon(t) \right] dt \quad (10)
\]

The term \( \varepsilon(t) \) aims to build up a proper redundancy so that the proposed RL algorithm does not reward the agent for targeting the exact critical accumulation point. A comprehensive explanation of the term \( \varepsilon(t) \) for the reward in RL will be provided in the following Section 5.3.

METHODOLOGY

This section gives a brief explanation of how MPC and RL are applied in this paper, and how we realize designing an antifragility perimeter control strategy based on the derivatives and redundancy in detail.
MPC algorithm

MPC is an established control method with wide applications in engineering to regulate dynamic systems (82), and is used as one of the benchmark methods in this work. First applied in (22), MPC has been proven to be an effective and robust method to regulate perimeter control in comparison to a greedy controller under both normal situations and traffic demand uncertainties as well as MFD errors. For details of the implementation of MPC in general practice or under the context of perimeter control, we refer the readers to (83) and (22). The applied MPC toolkit in this paper is introduced in (84), which uses the CasADi framework (85) and the NLP solver IPOPT (86).

The disadvantage of MPC is that it assumes an abundance of a priori knowledge, which includes traffic demand at each time step, but this information is hardly available in real-world applications. Although traffic forecasting (87, 88) may serve as a substitute for real traffic demand, the prediction accuracy is not always guaranteed. Potentially the information on disruptions, such as the distribution of the surging demand, can also be fed into MPC so that the algorithm can yield more accurate results. However, it is self-evident that disruptions are mostly fully unexpected events without any a priori knowledge. Therefore, in our research, when disruption takes place in the simulation, we apply the same control policies generated by MPC under an aggregated scenario. This aggregated scenario is the weighted average of the past traffic demand profiles, which includes all the experienced disruptions except the current one.

RL algorithm

A perimeter control strategy using RL has been proven feasible in (50, 51, 58). In RL algorithms, an agent or multiple agents interact with a preset environment and improve the performance of decision-making, defined as action \( a_t \) in an action space \( \mathcal{A} \), based on the observable state \( s_t \) in the state space \( \mathcal{S} \) and the reward, defined as \( r_t = R(s_t, a_t) \), where \( R \) is the reward function. The improvement of decision-making is commonly realized through a deep neural network as a function approximator. The RL algorithm applied in this work is Deep Deterministic Policy Gradient (DDPG) as proposed in (89). By applying an actor-critic scheme, DDPG can manage a continuous action space, i.e., allowing the perimeter control variables to be continuous instead of only choosing from a limited set of discrete values. In contrast, the traditional Deep Q-Network (DQN) algorithm (90) commonly applied as in Table 2 shows, can only be applied in an environment with a discrete action space. Even though a discrete action space is also viable to determine the change rate of the perimeter variables, (51) has proven that an RL algorithm with continuous action space can achieve better performance compared to discrete action space. The DDPG algorithm can be divided into two main components, namely the actor and the critic, which are updated at each step through policy gradient and Q-value, respectively. The scheme of the DDPG algorithm applied in this paper is schematically illustrated in Figure 2.

The state \( s_t \in \mathcal{S} \) is defined distinctively according to different methods applied in this work. Our proposed method consists of three terms, the vehicle accumulation regarding the OD pair \( n_{ij}(t) \), the change rate of vehicle accumulation at each time step \( dn_{ij}(t) \) (first derivative) as well as the second derivative \( d^2n_{ij} \). In (51), a state \( s_t \) defined as \( [n_{ij}(t), q_{ij}(t)] \) is adopted. However, since traffic demand in the real world is hardly measurable, meaning \( q_{ij}(t) \) should be an unobservable state for the agent. In our work, only variables that are observable or deducible for the agents can be incorporated, which are \( n_{ij}(t), dn_{ij}(t), \) and \( d^2n_{ij}(t) \) in our case. The action \( a_t \in \mathcal{A} \) is defined the same as the control variables \( u_{ij}(t) \). The reward for our proposed method is defined as in Eq. 10 describes while the baseline method uses only the completion rate as the reward term.
FIGURE 2: DDPG scheme

The actor network is represented by $\mu(\cdot)$ and it determines the best action $a_t$, which is the percentage of vehicles that are allowed to travel across the periphery, based on the current state $s_t$ given the weight $\theta_\mu$ at a certain time step $t$:

$$a_t = \mu(s_t | \theta_\mu)$$  \hspace{1cm} (11)

The critic network, denoted by $Q(\cdot)$, takes the responsibility to evaluate whether a specific state-action pair at a certain time step yields the maximal possible discounted future reward $Q(s_t, a_t)$. A common technique used in DDPG is to create a target actor network $\mu'(\cdot)$ and a target critic network $Q'(\cdot)$, which are a copy of the original actor and critic network but updated posteriorly to stabilize the training process and prevent overfitting (91). The target maximal discounted future reward for the target critic network can be calculated as in Eq. 12.

$$y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1} | \theta_\mu')) | \theta_\mu'$$  \hspace{1cm} (12)

Similar to DQN, the critic network can be updated through calculating the temporal difference between the predicted reward and the target reward and minimizing the loss for a mini-batch $N$ sampled from the replay buffer:

$$L = \frac{1}{N} \sum_i (y_i - Q(s_i, a_i | \theta_Q))^2$$  \hspace{1cm} (13)

Afterward, the actor network can be updated with sampled deterministic policy gradient (92):

$$a_t = \mu(s_t | \theta_\mu)$$  \hspace{1cm} (11)
\[ I = \mathbb{E}_{s_i \sim p} [r(s, \mu(s|\theta^\mu))|_{s=s_i}] \]  
(14)
\[ \nabla_{\theta^\mu} I = \mathbb{E}_{s_i \sim p} \left[ \nabla_a Q(s, a|\theta^Q) \right|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s|\theta^\mu) \right|_{s=s_i} \]  
(15)
\[ \nabla_{\theta^\mu} I \approx \frac{1}{N} \sum_t \nabla_a Q(s, a|\theta^Q) \right|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s|\theta^\mu) \right|_{s_i} \]  
(16)

When training the agent, we add intermittent disruptions into the simulation environment at certain training episodes, in the form of a temporary surging traffic demand. The following Algorithm 1 summarizes all the essential steps to implement the DDPG algorithm under the scenario with the assumption of disruption occurrences.

5 Antifragility and the antifragile terms in RL

In this paper, we demonstrate the two types of antifragility as introduced in Section 3. We refer to the antifragility gained from experiencing disruptions sequentially as antifragility type I, and the nonlinear response of linearly increasing disruptions regardless of prior experience as antifragility type II.

As illustrated in Section 3, researchers have been working around the state space \( \mathcal{S} \) of the RL algorithms to induce robustness of various traffic control strategies (60, 62–64). In our work, following the same logic, we add additional terms based on derivatives (8) and redundancy (76), in the state space \( \mathcal{S} \) and the reward function \( \mathcal{R} \) of the RL algorithm.

For the state space \( \mathcal{S} \), the baseline RL algorithm (51) uses traffic demand \( q_{ij} \) as additional information fed to the agent other than accumulation \( n_{ij} \). In our work, we replace \( q_{ij} \) using the first and second derivatives of the vehicle accumulation \( d n_{ij}(t) \) and \( d^2 n_{ij}(t) \). With this additional information, the RL agent is aware of a possible demand surge and \( d^2 n_{ij}(t) \) can reflect the curvature of the demand profile, i.e., how fast the demand is soaring so that the agent would anticipate the possible magnitude of demand in the future time steps.

For the reward function \( \mathcal{R} \), the trip completion at each time step acts as the main component as in MPC. The term \( \varepsilon(t) \) in the objective function \( J_{RL} \) in Equation (10) acts as an additional term to build up redundancy in the system. Similar to the creation of the additional term in the state space, we create redundancy also through the calculation of the derivates, but instead of the derivatives of the vehicle accumulation, we calculate the derivatives of the traffic state. First, we summarize \( \varepsilon(t) \) as the sum of two terms, with \( H(t) \) being an overall term representing the first derivative of the traffic state and \( \Delta H(t) \) representing the second derivative:

\[ \varepsilon(t) = H(t) + \Delta H(t) \]  
(21)

\( H(t) \) and \( \Delta H(t) \) can be expanded as:

\[ H(t) = \sum_{i=1,2} H_i(t) = \omega_h \sum_{i=1,2} f(n_i(t), n_{i,\text{crit}}, n_{i,\text{cap}}) \cdot \alpha_i(t) \cdot h_i(t) \]  
(22)

\[ \Delta H(t) = \sum_{i=1,2} \Delta H_i(t) = \omega_{h\Delta} \sum_{i=1,2} f(n_i(t), n_{i,\text{crit}}, n_{i,\text{cap}}) \cdot \Delta h_i(t) \]  
(23)

\( h_i(t) \) and \( \Delta h_i(t) \) are the first and second numerical derivatives of the traffic states on the MFD, \( h_i(t) \) is defined as the difference of trip completion over vehicle accumulation at the end of a time step versus at the beginning of the same time step, as in Eq. 24 shows, and the second
Algorithm 1 RL-based algorithm for perimeter control with disruption occurrence

Initialize actor $\mu(s|\theta^\mu)$ and critic network $Q(s,a|\theta^Q)$ with random weight $\theta^Q$, $\theta^\mu$
Initialize target actor network $\mu'$ and target critic network $Q'$ with the same weight
Initialize the replay memory, converge_flag
for episode = 1 to max_epi do
    Initialize a normal distribution $N$ as random noise for action exploration
    Receive initial observation and reward $s_1, r_1$ through the agent interacting with environment
    for $t = 1$ to max_step do
        Choose action $a_t = \mu(s_t|\theta^\mu)$ with noise $n_t$
        if disruption = True then
            Update the demand profile regarding the predefined disruption
            $q(t) \leftarrow q(t) + q\text{dist}(t)$
        end if
        Agent interacting with the environment and gets reward $r_t$ and new state $s_{t+1}$
        Record $(s_t, a_t, r_t)$
        if Training = True then
            Store into the replay memory and get a mini-batch $(s_i, a_i, r_i, s_{i+1})$
            Determining maximal future reward with target actor and critic network
            $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^\mu')|\theta^Q')$
            Update critic network weight $\theta^Q$ by minimizing the loss
            $L = \frac{1}{N} \sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$
            Update the actor network weight $\theta^\mu$ through policy gradient
            $\nabla_{\theta^\mu} I \approx \frac{1}{N} \sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s|\theta^\mu)|_{s_i}$
            Update the target actor network and target critic network
        end if
    end for
    Test the performance on the same environment and check convergence
    if episode = converge_episode and converge_flag = False then
        Abort
    end if
end for

1 derivative $\Delta h_i(t)$ is calculated as the difference between the first derivatives of two consecutive
2 time steps, as in Eq. 25 shows:

$$h_i(t) = \frac{M_i(t) - M_i(t-1)}{n_i(t) - n_i(t-1)}$$

$$\Delta h_i(t) = h_i(t) - h_i(t-1)$$

3 Since in the RL algorithms implemented in this paper, all variables involved in the deep
4 neural network should be normalized to facilitate the training process, meaning the exact values of
5 the derivatives are not of importance, $\omega_h$ and $\omega_{\Delta h}$ are introduced as the weight constants for the
first and second derivatives to regulate their impact on the reward side $R$.

$\alpha_i(t)$ is a binary variable designed in the first derivative to reward the agent when moving towards the desired direction on the MFD. For instance, the derivative of any data point in the congested zone of the MFD is negative. In this case, when the vehicle accumulation is still getting larger, a penalty will be applied. However, if the vehicle accumulation is decreasing through perimeter control, this binary $\alpha_i(t)$ variable will turn it into a reward. For the second derivative, an additional binary variable is not necessary since the two consecutive first derivatives are able to determine whether $\Delta h_i(t)$ is either positive or negative.

$$\alpha_i(t) = \begin{cases} 1, & \text{if } n_i(t) \geq n_i(t-1), \\ -1, & \text{otherwise.} \end{cases} \quad (26)$$

The term $f(n_i(t), n_{i,crit}, n_{i,cap})$ is a reduction factor to constrain the impact of the $\varepsilon(t)$ term when the accumulation is either on a very lower level or on a very higher level (gridlock). The area around critical accumulation is where the $\varepsilon(t)$ term should have the greatest impact. Here we use a modified trigonometric function to realize this purpose. It should be noted that other functions, such as normal distribution, are also valid for achieving the same purpose.

$$f(n_i(t), n_{i,crit}, n_{i,cap}) = \begin{cases} 1 + \cos\left(-\pi \cdot \frac{n_{i,crit} - n_i(t)}{n_{i,crit}}\right), & \text{if } n_i(t) \geq n_{i,crit}, \\ \frac{2}{1 + \cos\left(-\pi \cdot \frac{n_i(t) - n_{i,crit}}{n_{i,cap} - n_{i,crit}}\right)}, & \text{otherwise.} \end{cases} \quad (27)$$

After considering all the modifiers above, we show $H(t)$ and $\Delta H(t)$ using a single MFD as an example in Figure 3.

![Figure 3](image)

(a) Overall term $H$ for the first derivative $h_i(t)$  
(b) Overall term $\Delta H$ for the second derivative $\Delta h_i(t)$

**FIGURE 3**: Illustration of overall term $H(t)$ and $\Delta H(t)$ on the reward $R$.

As can be noticed, the first derivative $H(t)$, in Figure 3a, rewards the agent more when it’s moving towards the critical accumulation to maximize its trip completion rate. However, when the number of vehicles approaches the critical accumulation, the reward drops significantly and
becomes a penalty when it exceeds the critical point. For the second derivative $\Delta H(t)$ in Figure 3b, it always applies a penalty to the agent, and a darker blue indicates a greater penalty. The x-axis is the accumulation, same as in Figure 3a, while the y-axis represents how fast the traffic state is changing on the MFD. The faster it changes around the critical accumulation, the greater the penalty will be on the RL agent. With $H(t)$ and $\Delta H(t)$, the agent learns to be conservative when regulating the perimeter control variables when the accumulation is about to reach critical, in case of disruptions take place. Therefore, as can be concluded, although $H(t)$ and $\Delta H(t)$ apply the same concept of the derivative as the $dn_{ij}(t)$ and $d^2n_{ij}$ in the state space $\mathcal{S}$, the purpose of $H(t)$ and $\Delta H(t)$ is preserving redundancy in the system instead of feeding additional information to the agent.

**EXPERIMENT APPLICATION**

This section introduces the environment setup of the simulation and the evaluation procedure. We investigate a total number of three perimeter control strategies in addition to a no-control scenario, based on 2 different scenarios, i.e., under constant or incremental disruptions for testing antifragility type I and type II respectively. The model-based approach utilized is Model Predictive Control (MPC), whereas the other two perimeter control strategies are based on RL. While both of these RL-based methods adopt the same definition of action $a_t \in \mathcal{A}$ as the perimeter control variable, the definition of state $\mathcal{S}$ and reward $\mathcal{R}$ differ between the methods.

- No control
- MPC modified from Geroliminis et al. (22), averaging history demand profile
- State-of-the-art RL-based method proposed by Zhou and Gayah (51) as baseline:
  - State $s_t$: vehicle accumulation and traffic demand $[n_{ij}(t), q_{ij}(t)]$
  - Reward $r_t$: trip completion in both regions $[\sum_{i=1,2} M_{ii}(t)]$
- Our proposed RL-based antifragile method:
  - State $s_t$: vehicle accumulation and the first and second derivatives $[n_{ij}(t), dn_{ij}(t), d^2n_{ij}(t)]$
  - Reward $r_t$: trip completion and the $\epsilon(t)$ term $[\sum_{i=1,2} M_{ii}(t) + \epsilon(t)]$

**Simulation environment**

We simulate a cordon-shaped urban network with inner and outer regions with different MFDs, which resemble those in (51). Some minor modifications have been adapted in the congested zone of the MFD to ensure the derivative of the MFD is continuous. The MFD of the outer region is largely in line with the observations in Yokohama (20). Other critical indicators, e.g., critical accumulation and maximal trip completion, remain unchanged. The non-identical MFDs reflect the fact that the area of the periphery is much larger than the city center, and thus the maximal accumulation would be significantly higher. The traffic demand under no disruption is modified based on (20). Instead of a trapezoidal demand profile, the chosen profile is in the shape of a normal distribution in accordance with (93) to describe the peak hour traffic flow. The total simulation duration $t_{end}$ is 2 hours with each time step $\Delta t$ as 60 seconds, and the second hour has significantly fewer vehicles so that the network would be able to clear the vehicles accumulated in the first hour. Other notable constraints and initializations include the lower and upper bounds for the perimeter control variable $u_{ij} \in [0.1, 0.9]$ (22) and an initial accumulation in the network $n_{ij,0} = [600, 1300, 300, 2400]$. Two different scenarios are considered in this work. For the validation of antifragility type I, after the agent has been trained for the first 50 episodes with a base demand profile, as
shown in Figure 4a, we introduce a new demand profile with disruption, which characterizes a surging demand within a short period of time from the outer region into the inner region. The new environment lasts for 50 episodes with the disruption being a constant magnitude of 2500 vehicles following normal distribution, as shown in Figure 4b. To test antifragility type II, with the total number of episodes being the same 100 episodes, instead of a constant amplitude of disruption, an incremental disruption is applied for the last 50 episodes. The amplitude of disruption grows linearly from 0 to twice the number of vehicles in the scenario with constant disruptions. Both the RL-based methods are trained 15 times and the average is calculated as their learning curves.

![Base demand profile](image1)

![Demand profile under demand disruption](image2)

**FIGURE 4**: Demand profile with or without disruption

**Performance evaluation**

Although the reward defined in the RL algorithm is based on trip completion, to better showcase antifragility, the main performance indicator being evaluated here is the total time spent (TTS), which is calculated by adding the number of vehicles within the network at each second of the simulation. The main reason for choosing TTS over completion is that completion is lower bounded with increasing magnitude of disruption by the number of vehicles that finished their trips before the disruption, which means we may observe a concave trip completion with increasing disruptions at first but then it becomes approximately constant. This makes the manifestation of antifragility difficult since the performance curve could be partially concave and partially convex. However, by using TTS as the performance indicator, we circumvent this problem since TTS still grows linearly with increasing disruptions even under extreme cases instead of being lower-bounded. Therefore, the full convexity or concavity of the performance curve can be maintained and we can roughly approximate the simulation results with a second-degree polynomial.

As urban road networks are always subject to capacity constraints, fully antifragile traffic control strategies may be impossible to design. Therefore, we normalize all the other perimeter control strategies over the RL baseline method to study the relative antifragility. To quantify the antifragility type II of different methods, we calculate the skewness of the TTS distribution. A more negative value of skewness indicates the distribution has a longer or fatter left tail and thus a higher degree of concavity in the function.
RESULTS

This section presents the simulation results of the four methods, i.e., no-control, MPC, baseline RL-based method, and our proposed antifragile RL-based method under the two scenarios to show the property of antifragility type I and antifragility type II, respectively.

Antifragility type I

Systems with the property of antifragility type I can learn from past adverse events and anticipate possible ongoing disruptions to enhance future performance. To test this property, we apply disruptions with constant magnitude after training with the base demand profile.

Unlike the no-control and MPC approaches, RL-based methods are subject to performance variations over simulation episodes. Therefore, in Figure 5, other than the performance curves of the no-control and MPC approaches, we show the learning curves of the baseline RL method and our proposed antifragile RL method. The curves are averaged from 15 simulations, while the shadowed area indicates the standard deviation of the simulation results. After conducting 50 episodes of training together with testing, an additional 2 testing phases were implemented using the final weight parameters obtained from the training process. These tests were performed under or under no disruptions.

FIGURE 5: Learning curves under constant disruptions

It can be noticed that two significant phenomena occur during the initial 50 episodes with the base demand profile. First, the proposed antifragile method can achieve better performance with less TTS when there is no disruption in the network. While the baseline method has almost converged after 50 episodes, the proposed antifragile method seems to be capable of further improving the performance with more training episodes. The reason for not continuing the training
process until the convergence of both methods is to avoid overtraining the non-disruption-related weight parameters as well as the underfitting of the disruption-related parameters. Regardless of the better performance of the proposed method, the shadowed area represents a more significant performance variation compared to the baseline method, indicating that our proposed antifragile method exhibits comparably lower training stability.

After disruption with constant magnitude is introduced into the network. The performance of the two RL-based methods has first changed drastically. Then, while the proposed method has an obvious tendency of reducing TTS over 50 episodes, the baseline method shows no clear sign of improving performance. In contrast to the conclusion above, the proposed method has a higher training stability compared to the baseline method and also becomes increasingly stable over episodes as the width of the shadowed area is getting smaller.

While the results of testing under disruption follow the same pattern as the last episode of the training process, however, when the simulation environment reverts to the previous no-disruption condition, the performance of different methods behaves differently. Since our modified MPC averages demand history, after experiencing episodes with disruptions, the performance deteriorates when there is no disruption. A similar pattern can be observed for the baseline RL method. Not only is the TTS higher in this testing episode compared to the training episode 50, but the performance variation is also significantly larger. On the contrary, our proposed antifragile RL method can both maintain excellent performance and achieve smaller variation.

We normalize all the other methods over the result of the baseline RL-based method averaged from the 15 simulations, as shown in Figure 6. The performance improvement of both MPC and the proposed antifragile methods can be observed. Since we average all the past experiences of the demand profile, including disruptions for MPC, it is capable of performing better under the same disrupted scenario, and even marginally better than the proposed antifragile method, demonstrating the property of antifragility type I. While maintaining roughly similar performance compared to the baseline method at the beginning of the scenario switch, and thus better performance than MPC, the proposed method gains capability in dealing with disruptions when experiencing more and more such disruptions, also showing the property of being antifragile.

**Antifragility type II**

Antifragility type II describes a system that exhibits a nonlinear response to a linearly increasing magnitude of disruptions. In terms of TTS, since an increase in the traffic density or vehicle accumulation in the network will inevitably lead to a decrease in the vehicle speed or trip completion rate based on the speed MFD, therefore, a concave response between TTS and vehicle accumulation indicates an antifragile type II system. The four studied methods are tested in an environment with linearly increasing traffic demand for 50 episodes after being trained with the same base demand as in the previous scenario for 50 episodes. Furthermore, another testing phase of 20 episodes is carried out following the same demand increment and the demand magnitude of the last training episode. Figure 7a shows the TTS of different methods under incremental disruptions in both training and testing phases. It should be noted that Figure 7a and Figure ?? do not share the same y-axis and the result of the no-control approach is not presented in ?? due to its exceedingly high value.

Under this scenario, the TTS of the no-control approach grows sharply, doubling the difference between the last and the first episode under the incremental disruption of the other methods. Similar to the previous scenario, the performance deviation of the proposed method is relatively
FIGURE 6: Performance difference normalized over the baseline method under constant disruptions

FIGURE 7: Learning curves under incremental disruptions

lower compared to the baseline method under the disruptions scenario. However, one more phenomenon to notice is that under incremental disruptions, the performance deviation of both meth-
ods also grows with the increase in the magnitude of disruption. And the deviation of the baseline method is significantly higher compared to the proposed antifragile method in the last simulation episodes. Although the performance of MPC under the constant disruptions scenario is even marginally better than the proposed antifragile method, the performance falls behind the other RL-based methods, as the amplitude of the disruption remains low and grows linearly after the first 50 episodes. Therefore, When averaging the past disruption experiences, MPC cannot gain sufficient knowledge of the upcoming magnitude of the disruptions. In the testing phase, it can be observed that the performance deviation of the baseline RL method is significantly larger compared to the proposed antifragile RL method. But since the weight parameters of the agents in the testing phase are not updated anymore, this could be a reason that the performance curves of the RL-based methods appear to be linear.

Since the performance differences between the baseline and the other methods are hard to observe in this figure, we need to normalize the performance of all the other methods over the baseline methods. Before the normalization process, to better showcase the nonlinear response and compare the concavity or convexity of different methods, the performance deviation in the learning curves of the RL-based methods needs to be minimized. Therefore, we approximate the simulation results of different methods each with a second-degree polynomial, and recreate the data points after fitting the polynomial coefficients, with which the normalization process can be implemented, and the results are shown in Figure 8.

![Figure 8](image_url)

**FIGURE 8:** Performance difference normalized over the baseline method under incremental disruptions

One of the benefits of a polar coordinate system over the cartesian coordinates is that it amplifies the performance difference, particularly when the disruption magnitude is high, and this
region is where we pay the most attention. While both the no-control approach and MPC have a positive increasing performance difference in comparison to the baseline method. The proposed antifragile method is the only one that remains a negative difference with its absolute value growing even larger after the disruption reaches around 4000 vehicles. However, it should be first noted that due to the characteristics of the polar coordinates, a concave or convex curve on this plot does not necessarily indicates that the data points of a specific method follow a concave or convex distribution. A linearly increasing curve can demonstrate itself as a spiral curve in the polar coordinates. Nevertheless, it can also be observed that the curve of the proposed method first draws near to the baseline curve and has, for instance, a performance difference of -0.5% under the disruption with roughly 3200 additional vehicles, then it recedes at some point, and reaches -0.5% again with around 5000 vehicles. This indicates that the proposed method has a relative concavity compared to the baseline method, showcasing the antifragility type II.

To quantify the antifragility type II of our proposed method, we demonstrate the distribution of performance difference in Figure 9. The proposed antifragile method exhibits the most concentrated distribution, indicating a lower TTS on average compared to the other methods under the same disruptions.

Moreover, in order to compare the (anti)fragility among different methods quantitively, we calculate the skewness of each distribution. In Table 4, we summarize the skewness of each studied method. Surprisingly, all 3 methods possess a negative skewness, indicating a relative antifragility type II compared to the baseline method. In other words, the baseline method is the most fragile of all the 4 methods investigated in this study. While the no-control approach has the least relative antifragility, our proposed antifragile method has demonstrated the strongest antifragility property, with its skewness of distribution being the most negative of all.
TABLE 4: Normalized Skewness over Baseline Method under Incremental Disruptions

<table>
<thead>
<tr>
<th>Methods</th>
<th>Skewness</th>
</tr>
</thead>
<tbody>
<tr>
<td>No control</td>
<td>-0.326</td>
</tr>
<tr>
<td>MPC</td>
<td>-0.753</td>
</tr>
<tr>
<td>Antifragile RL-based method</td>
<td>-1.106</td>
</tr>
</tbody>
</table>

CONCLUSION

First, this research work introduces the concept of antifragility by comparing it with two other terms robustness and resilience, which are commonly used in transportation and traffic control strategies. Through a literature review on using RL algorithms in traffic control to realize a robust or resilient design, whether and how such properties can be induced in RL is investigated. Following the same idea, we manage to induce antifragility in perimeter control by modifying the state space and reward function based on a state-of-the-art RL-based perimeter control strategy. We incorporate the change rate and curvature of the traffic states in the state space to leverage the potential of such derivatives to feed more information to the RL algorithm. Also, a redundancy term in the reward function has been carefully composed to empower the system to be more resistant to disruptions.

We conducted a comprehensive comparison between our proposed antifragile RL-based perimeter control approach and three other methods: no-control, MPC, and a baseline RL-based method. This comparison was carried out under two different scenarios, disruptions with constant and incremental magnitude, for validating the antifragility type I and type II individually. The results from the study clearly demonstrated the effectiveness and antifragility of our proposed method, for both antifragility type I and type II. Furthermore, we put forward a novel method for quantifying antifragility, utilizing the measure of skewness in the distribution. Our proposed antifragile method exhibits the greatest negative skewness among the methods examined, indicating its relative antifragility property against disruptions.

In conclusion, this study is the first of its kind to pioneer the application of the antifragility concept in enhancing the daily operation of engineering systems to enhance performance during unforeseen disruptions using a learning-based algorithm. It introduces a new possibility for evaluating system operation under disruptive conditions. Moreover, the concept can be extended not only to other traffic control strategies like traffic signal control and pricing but also to various engineering disciplines and industries, broadening its potential impact.
REFERENCES


58. Chen, C., Y. P. Huang, W. H. K. Lam, T. L. Pan, S. C. Hsu, A. Sumalee, and R. X. Zhong, Data efficient reinforcement learning and adaptive optimal perimeter control of network


