Detection and Analysis of Crash- and Weather-Induced Freeway Congestion Using Floating Car Data

submitted by

Lennart Querfurth

Full reprint of the dissertation accepted by the Faculty of Civil Engineering and Environmental Sciences of the University of the Bundeswehr Munich in fulfillment of the requirements for the degree of

Doctor of Engineering (Dr.-Ing.)

Reviewers:

1. Prof. Dr.-Ing. Silja Hoffmann
2. Prof. Dr.-Ing. Robert Hoyer

The dissertation was submitted to the University of the Bundeswehr Munich on June 08, 2026 and was accepted by the Faculty of Civil Engineering and Environmental Sciences on October 06, 2026. The oral examination took place on October 22, 2026.

Detektion und Analyse unfall- und witterungsbedingter Autobahnstaus mittels Fahrzeugflottendaten

vorgelegt von
Lennart Querfurth

Vollständiger Abdruck der von der Fakultät für Bauingenieurwesen und Umweltwissenschaften der Universität der Bundeswehr München zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

angenommenen Dissertation.

Gutachter/Gutachterin:

1. Prof. Dr.-Ing. Silja Hoffmann
2. Prof. Dr.-Ing. Robert Hoyer

Die Dissertation wurde am 08.06.2026 bei der Universität der Bundeswehr München eingereicht und durch die Fakultät für Bauingenieurwesen und Umweltwissenschaften am 06.10.2026 angenommen. Die mündliche Prüfung fand am 22.10.2026 statt.

Abstract

Road congestion in the European Union is estimated to cost around one percent of GDP annually. A large share of this is non-recurrent congestion, caused by crashes, vehicle breakdowns, and adverse weather rather than routine demand. Effective response depends on rapid detection, but the conventional path from crash to traffic-management response still takes tens of minutes, and the roadside sensors that would be needed for automated detection cover only part of the network. The mechanisms by which localized weather disturbances trigger and sustain traffic breakdowns are not well quantified either, which limits both real-time operations and the underlying theory. Floating Car Data (FCD) provide continuous position and speed traces from ordinary vehicles already on the network, and can close these gaps without dedicated sensor infrastructure.

This work detects crashes indirectly by analyzing the spatiotemporal structure of congestion patterns from fleet vehicle trajectories. The crash itself is not detected directly. The approach uses a known property of crash-induced congestion: its downstream front, where traffic returns to free flow, stays anchored at the bottleneck location. The position of this front therefore provides the bottleneck location more precisely than the road-segment ranges reported by traffic message channels. A binary classifier can separate crash-induced from routine congestion patterns. Two methods are developed in parallel: a rule-based approach building on three-phase traffic theory, and a set of convolutional and graph neural network (CNN, GNN) models that operate on graph and image representations of the same trajectory data. All methods are evaluated on three dimensions, namely crash detection accuracy, spatial localization, and detection timeliness, and benchmarked against each other as well as against conventional traffic information services.

Validated across four countries on 9,355 non-recurrent congestion patterns containing 445 verified crashes, the rule-based approach offers a tunable timeliness-accuracy trade-off: a high-precision setting reaches an F1-score of 75.0% and identifies crash-induced congestion 1.7 minutes earlier than conventional services; a fast setting widens that lead to 5.7 minutes at the cost of additional false alerts. Both settings detect congestion clearance nearly 9.5 minutes earlier than conventional services, without roadside infrastructure or specialized in-vehicle equipment. When compared with three deep-learning detectors trained on the

same data, the rule-based approach achieves the most predictable detection latency, the GNN the fastest accurate detections, and the partial-convolution CNN the highest overall accuracy. The choice between approaches is therefore operational rather than technical.

An iterative spatiotemporal model predicts how post-crash congestion propagates through the network. On the events for which a comparable forecast is available from the deployed incident feed, and during the active phase of each jam, the model outperforms this feed at every horizon up to 30 minutes. The improvement is largest for jams between 5 minutes and 30 minutes of realised delay, where the mean absolute error of the travel-time estimate drops by 16% to 50%, giving actionable travel-time estimates during the most uncertain phase of an incident.

An empirical study examines the role of localized rain cells as traffic breakdown triggers on German freeways. From 30 rain events observed with high-resolution weather radar, a directional asymmetry emerges: rain cells moving with the flow of traffic are 2.45x as likely to cause extended congestion as those moving against it, even though initial breakdown probabilities are similar. This adds empirical support for three-phase traffic theory and shows how the relative velocity between a weather disturbance and the traffic stream affects flow stability.

Taken together, these contributions show that three-phase traffic theory with modern data-driven methods enables reliable detection and spatiotemporal localization of non-recurrent congestion, particularly crash-induced mega jams, from fleet vehicle trajectories alone.

Zusammenfassung

Verkehrsstaus in der Europäischen Union kosten jährlich schätzungsweise rund ein Prozent des Bruttoinlandsprodukts. Ein großer Teil davon entsteht nicht durch die täglichen Nachfragespitzen im Berufsverkehr, sondern durch Unfälle, Pannen oder plötzliche Wetterumschwünge. Gerade bei solchen nicht-wiederkehrenden Störungen entscheidet die Geschwindigkeit der Erkennung über das Ausmaß der Folgen. Dennoch vergehen zwischen einem Unfall und der Reaktion der Verkehrsleitzentrale auf konventionellem Meldeweg häufig noch Verzögerungen von mehr als zehn Minuten und die stationäre Sensorik, die eine automatisierte Detektion leisten könnte, deckt lediglich ausgewählte Netzabschnitte ab. Auch die Mechanismen, durch die lokal begrenzte Wetterstörungen Verkehrszusammenbrüche auslösen und aufrechterhalten, sind quantitativ wenig untersucht, was den operativen Betrieb ebenso wie das zugrunde liegende Verständnis der Verkehrsdynamik betrifft. Zur Schließung dieser Forschungslücken werden Trajektoriendaten herangezogen, die im Regelbetrieb kontinuierlich von Flottenfahrzeugen generiert werden.

Statt einen Unfall direkt zu detektieren, untersucht ein skalierbares Verfahren die räumlich-zeitliche Struktur des resultierenden Staumusters: Die stromabwärtige Staufront verrät implizit den Ort der Blockade, und zwar genauer als die streckenabschnittsbezogenen Angaben herkömmlicher Verkehrsmeldungen. Ein binärer Klassifikator unterscheidet dabei unfallbedingte Staumuster von solchen mit anderer Ursache. Zwei Methoden werden parallel entwickelt: ein regelbasierter Ansatz auf Grundlage der Drei-Phasen-Verkehrstheorie sowie Faltungs- und Graphen-Neuronalnetzmodelle (CNN, GNN), die auf Graph- und Bildrepräsentationen derselben Trajektoriendaten arbeiten. Alle Verfahren werden anhand dreier Kriterien bewertet, nämlich Erkennungsgenauigkeit, räumlicher Lokalisierung und frühzeitiger Erkennung, und untereinander wie auch gegenüber konventionellen Verkehrsinformationsdiensten verglichen.

Der regelbasierte Ansatz wurde an 9,355 nicht-wiederkehrenden Staumustern mit 445 verifizierten Unfällen aus vier Ländern validiert. Er ermöglicht eine konfigurierbare Abwägung zwischen Erkennungslatenz und Genauigkeit: In einer präzisionsorientierten Konfiguration erreicht er einen F1-Score von 75.0% und erkennt unfallbedingte Staus 1.7 Minuten früher als etablierte Referenzdienste. Eine latenzoptimierte Konfiguration weitet diesen

zeitlichen Vorsprung auf 5.7 Minuten aus, allerdings auf Kosten zusätzlicher Fehlalarme. Die Auflösung des Staus wird in beiden Konfigurationen 9,5 Minuten früher erkannt, ohne stationäre Sensorik und ohne spezielle fahrzeugseitige Ausstattung. Im Vergleich mit drei auf denselben Daten trainierten lernenden Detektoren liefert der regelbasierte Ansatz die berechenbarste Erkennungslatenz, das GNN die schnellsten korrekten Detektionen und das CNN mit partieller Faltung die höchste Gesamtgenauigkeit. Die Wahl zwischen den Ansätzen ist damit eine betriebliche und keine technische Frage.

Ein iteratives räumlich-zeitliches Modell prognostiziert die Ausbreitung unfallbedingter Staus im Netz. Auf der Teilmenge der Ereignisse, für die der ausgelieferte Verkehrsmeldungsdienst eine vergleichbare Prognose liefert, übertrifft das Modell diesen Dienst während der aktiven Phase jedes Staus auf allen Vorhersagehorizonten bis 30 Minuten. Der Vorteil ist bei Staus mit 5-30 Minuten realisierter Verzögerung am größten. Dort sinkt der mittlere absolute Fehler der Reisezeitschätzung um 16 % bis 50 %, was belastbare Reisezeitschätzungen in der dynamischsten Phase eines Vorfalls ermöglicht.

Eine empirische Studie untersucht die Rolle lokal begrenzter Regenzellen als Auslöser von Verkehrszusammenbrüchen auf deutschen Autobahnen. Eine Analyse von 30 Regenereignissen anhand hochaufgelöster Wetterradardaten zeigt eine Richtungsasymmetrie: Regenzellen, die sich mit dem Verkehrsfluss bewegen, führen 2,45-mal häufiger zu persistenten Staus als solche, die sich entgegen dem Verkehrsfluss bewegen, obwohl die anfänglichen Zusammenbruchswahrscheinlichkeiten vergleichbar sind. Dies stützt die Drei-Phasen-Verkehrstheorie empirisch und zeigt, wie die Relativgeschwindigkeit zwischen Wetterstörung und Verkehrsstrom die Flussstabilität beeinflusst.

Gemeinsam zeigen diese Beiträge, dass die Verbindung der Drei-Phasen-Verkehrstheorie mit datengetriebenen Verfahren eine zuverlässige Erkennung und räumlich-zeitliche Lokalisierung nicht-wiederkehrender Staus, insbesondere unfallbedingter ”mega jams”, allein aus Trajektoriendaten von Flottenfahrzeugen ermöglicht.

Acknowledgements

This dissertation would not have been possible without the support and encouragement of many people, to whom I owe sincere gratitude.

I am deeply grateful to my supervisor, Prof. Dr. Silja Hoffmann, whose academic guidance and critical feedback shaped the scientific quality of this work from its earliest stages to its completion. I thank Prof. Dr. Robert Hoyer for his constructive comments and for kindly agreeing to serve as the second examiner of this thesis.

This work was carried out at Mercedes-Benz Group AG, which provided the rare opportunity to pursue doctoral research embedded in an industrial setting. I am particularly thankful to Dr. Hubert Rehborn for his continuous mentorship and the trust that allowed me to conduct this research with a high degree of independence. I thank Marc Zürn for the many discussions and collaborative problem-solving that sharpened both the technical and conceptual aspects of this thesis, and the wider team for the stimulating and collegial environment throughout these years. I would also like to thank Song Cheng, whom I had the pleasure of supervising during his master's thesis; his work fed into the learning-based components of this dissertation.

I acknowledge the University of the Bundeswehr Munich for the academic affiliation under which this dissertation was conducted, and thank my colleagues there for the valuable conversations along the way.

Finally, I thank my partner Eva and my parents for their unwavering support and patience. Their understanding during the long stretches of focus this work required made its completion possible.

Contents

Abstract v

Zusammenfassung vii

Acknowledgements ix

1 Introduction 1
- 1.1 Motivation . . . 2
- 1.2 Problem Statement . . . 3
- 1.3 Research Questions . . . 3
- 1.4 Thesis Structure . . . 4

2 Foundations of Traffic Breakdown and Crash Detection 5
- 2.1 Traffic Variables and Observation Methods . . . 5
  - 2.1.1 From Stationary Sensors to Floating Car Data . . . 7
- 2.2 Spatiotemporal Representation of Microscopic Data . . . 7
- 2.3 Traffic Flow Theories and Breakdown Mechanisms . . . 8
  - 2.3.1 Schools of Traffic Flow Theory . . . 8
  - 2.3.2 Three Phase Traffic Theory . . . 9
  - 2.3.3 Traffic Phase Detection from FCD . . . 13
- 2.4 Bottlenecks and Congestion Patterns . . . 14
  - 2.4.1 Congestion Patterns at Bottlenecks . . . 14
- 2.5 Precipitation-Induced Congestion . . . 15
  - 2.5.1 Mechanisms of Influence . . . 15
  - 2.5.2 Macroscopic Effects of Precipitation . . . 16
  - 2.5.3 Localized Rain Cells . . . 16
- 2.6 Crash Detection on Freeways . . . 17
  - 2.6.1 Mechanisms and Stages of Disturbance . . . 17
  - 2.6.2 Approaches to Crash Detection . . . 18
- 2.7 Convolutional Neural Networks . . . 19
  - 2.7.1 Principles of Convolutional Neural Networks . . . 20

2.7.2 CNN-Based Crash Detection on TSDs . . . 21
2.8 Graph Neural Networks . . . 22
2.8.1 Principles of Graph Neural Networks . . . 22
2.8.2 Sequence Modelling Components . . . 23
2.8.3 GNN Applications in Traffic Forecasting and Crash Detection . . . 23

3 Dataset: Acquisition, Preprocessing, and Labeling 25
3.1 Data Characteristics . . . 25
3.1.1 Sampling and Transmission . . . 25
3.1.2 Fleet Penetration Rate . . . 26
3.2 Preprocessing and Feature Computation . . . 26
3.2.1 Temporal Alignment and Kinematic Computation . . . 26
3.2.2 Map Matching and Spatial Filtering . . . 27
3.2.3 Traffic Phase Detection . . . 27
3.3 Region Clustering . . . 28
3.4 Precipitation Data . . . 32
3.4.1 Precipitation Coverage . . . 32
3.5 TPEG Event Data . . . 34
3.5.1 Preprocessing and Spatial Referencing . . . 35
3.5.2 Ground-Truth Labeling . . . 35
3.6 Ground-Truth Sets . . . 37
3.6.1 Dataset Statistics . . . 39
3.6.2 Train-Validation-Test Split . . . 40
3.7 Input Preparation for Deep Learning Models . . . 40
3.7.1 Observation Window . . . 40
3.7.2 Class Imbalance . . . 41
3.7.3 Window Sampling Strategy . . . 41
3.7.4 Dataset Statistics . . . 44
3.7.5 Training Set Augmentation . . . 45
3.7.6 Validation and Test Configuration . . . 45

4 Influence of Rainfall on Freeway Traffic Breakdown 47
4.1 Detection and Characterization of localized rainfall cells (LRCs) . . . 47
4.2 Rain-Induced Capacity Reduction and Habituation . . . 47
4.3 Direction-Dependency of LRCs . . . 48
4.4 Empirical Evidence from Bidirectional Freeway Sections . . . 50

4.5 Statistical Analysis . . . 55
4.5.1 Sample Construction and Breakdown Probability . . . 55
4.5.2 Direction-Dependent Coupling and Velocity Effects . . . 55
4.5.3 Breakdown Location and the Habituation Effect . . . 57
4.5.4 Crash Occurrence Under Rainfall . . . 57
4.6 Discussion . . . 58

5 Crash Detection: Congestion Patterns and Task Formulation 61
5.1 Distinct Features of Recurrent and Crash-Induced Congestion . . . 61
5.1.1 Theoretical Basis for Crash Severity . . . 62
5.1.2 Classification of Post-Crash Congestion Patterns . . . 63
5.1.3 Task Definition . . . 64
5.2 Overview of Approaches . . . 65

6 Rule-Based Crash Detection Based on Distinct Phase Regions 67
6.1 Methodology . . . 67
6.1.1 Positive Indicators . . . 67
6.1.2 Plausibility Filters . . . 72
6.1.3 Aggregation into Crash Probability . . . 75
6.1.4 End of Crash Detection . . . 76
6.1.5 Crash Localization . . . 77
6.2 Parameter Calibration . . . 77
6.2.1 Parameter Space Reduction . . . 77
6.2.2 Pareto-Front Exploration . . . 77
6.3 Operating Points . . . 78
6.4 Ablation and Feature Analysis . . . 78
6.5 Sensitivity to Fleet Density . . . 80
6.6 Summary of Findings . . . 81

7 Data-Driven Crash Detection Using Learning-Based Models 83
7.1 Task Formulation and Shared Setup . . . 83
7.1.1 Learning-Based Task Formulation . . . 83
7.1.2 Loss Functions . . . 84
7.1.3 Experimental Setup . . . 84
7.2 CNN-Based Crash Detection . . . 85
7.2.1 Grid Encoding . . . 85
7.2.2 Sparsity Handling . . . 86

7.2.3 U-Net with Transformer Head . . . 88
7.2.4 CNN Model Configuration and Ablation Study . . . 89
7.3 GNN-Based Crash Detection . . . 91
7.3.1 Graph Construction . . . 91
7.3.2 Model Architecture . . . 95
7.3.3 Training Configuration . . . 97
7.3.4 Ablation Analysis . . . 98

8 Comparison of Crash Detection Approaches 101
8.1 Summary of Model Selection . . . 101
8.2 Prediction Workflow and Comparability of Approaches . . . 102
8.3 Quantitative Results . . . 103
8.3.1 Classification Results . . . 103
8.3.2 Regression Results . . . 108
8.3.3 Inference Efficiency . . . 110
8.4 Case Studies . . . 112
8.4.1 Multi-Sample Overview . . . 112
8.4.2 Detailed Probability-Curve Analysis . . . 113
8.5 Analysis and Discussion . . . 115
8.5.1 Model Performance Analysis . . . 115

9 Post-Crash Congestion Prediction 121
9.1 Introduction . . . 121
9.2 Methodology . . . 122
9.2.1 Congestion Evolution and Prediction Stages . . . 122
9.2.2 Congestion Area Estimation . . . 123
9.2.3 Detection of Transition Points . . . 125
9.2.4 Default Prediction Parameters . . . 126
9.3 Travel-Time Delay Estimation . . . 127
9.4 Results . . . 128
9.4.1 Dataset and Validation Setup . . . 128
9.4.2 Intersection over Union Performance . . . 128
9.4.3 Travel-Time Delay Prediction Accuracy . . . 129
9.4.4 Case Studies . . . 131
9.5 Discussion . . . 134
9.5.1 Limitations . . . 134
9.5.2 Extensions and Future Work . . . 134

10 Discussion and Conclusion **135**
10.1 Revisiting the Research Questions . . . 135
10.2 Methodological Observations . . . 137
10.3 Practical Implications . . . 137
10.4 Limitations . . . 138
10.5 Future Work . . . 139
10.6 Closing Remarks . . . 140

A Data and Preprocessing **141**
A.1 TPEG Event Codes . . . 141
A.2 Detection Window Sampling . . . 141

B Weather-Induced Traffic Breakdown **143**

C Rule-Based Detection **145**
C.1 Complete Parameter Configurations . . . 145
C.2 Detailed Ablation Results . . . 145
C.3 Feature-Isolation Experiment . . . 145
C.4 Confusion Matrices and Extended Timeliness Statistics . . . 149

D Learning-Based Crash Detection **151**
D.1 Implementation Details . . . 151
D.1.1 PSM Inference Pipeline . . . 151
D.1.2 Regularization Details . . . 151
D.2 GNN Architecture Details . . . 153
D.3 CNN Ablation Details . . . 153
D.4 GNN Ablation Details . . . 155
D.4.1 Hyperparameter Selection . . . 155
D.4.2 Design Choice Ablations . . . 155
D.4.3 Feature Ablation at Additional Thresholds . . . 160
D.4.4 Per-Run Validation Results . . . 160
D.5 Exponential Decay Smoothing Function . . . 160

E Crash Detection Comparison **161**
E.1 Prediction Workflow . . . 161
E.2 Additional Material . . . 162

F Post-Crash Congestion Prediction **166**
F.1 Empirical Example of Upstream Flow Starvation 166
F.2 Cluster Analysis of Congestion Parameters 166
F.2.1 K-Means Clustering and Group Structure 168
F.2.2 ANOVA Effect-Size Analysis 168
F.2.3 Early-Classification Experiment 169
F.2.4 Implications for Parameterization 169
F.3 Additional Case Studies 170
F.3.1 A81, Germany (March 18, 2022): Behavior without Within-Stage Refinement 170
F.3.2 NJ 3, United States (March 3, 2024): Extended Starvation without Error Growth 171
F.4 TPEG-based delay error analysis 171
F.5 Robustness to Simulator Velocities 172
F.6 Lifecycle-Stage Decomposition of the ETA Comparison 172

List of Own Publications **175**

Abbreviations **177**

Symbols **181**

References **185**

List of Figures **211**

List of Tables **219**

1 Introduction

Traffic congestion costs the European Union an estimated one percent of GDP each year [17], and the problem is not subsiding. On Germany's freeways alone, the Automobile Club of Germany (ADAC) recorded 516,000 traffic jams in 2024, totaling 448,000 hours and spanning 859,000 kilometers [18]. While recurring bottlenecks during rush hours receive much attention, a substantial share of congestion arises from non-recurring causes. In the United States, traffic incidents account for 25 % of all delay, adverse weather for 15 % and road work for 10 % [19, 20]. Of these, traffic crashes pose the most acute challenge: they occur unpredictably, carry the highest safety risk, and generate disproportionately large delays relative to their duration as their downstream effects compound rapidly if detected late [21].

Among non-recurring causes, crashes and adverse weather dominate. Road crashes claim more than 1.19 million lives worldwide each year and trigger significant secondary congestion on major corridors [22, 23]. Each minute a crash remains uncleared raises the likelihood of a secondary crash by an estimated 2.8% [24]. Rain alone increases travel time by 12–20 % in metropolitan networks and raises crash risk by more than 70% [19, 25]. In Europe, severe weather events impose substantial transport disruption and delay costs [26, 27].

Meanwhile, the data landscape for traffic monitoring is shifting rapidly. Connected vehicles continuously report global navigation satellite system (GNSS)-based position traces to central servers, generating what is commonly termed floating car data (FCD). In 2021, connected vehicles accounted for roughly 24 % of the European fleet, a figure projected to reach 83 % by 2035 [28]. Mobile phones complement vehicle systems with hybrid positioning methods combining GNSS, network, and inertial sensors. The resulting growth in floating car data (FCD) opens two opportunities: situation-aware information for individual vehicles, and large-scale analysis of traffic conditions across entire networks.

1.1 Motivation

Conventional crash detection still relies largely on manual reporting by road users — a process that is slow and error-prone [21, 29]. While eCall, mandatory in new European vehicles since 2018 [30], can automatically notify emergency services after severe collisions, it is designed for rescue coordination, not traffic management, and does not cover low-severity incidents.

Official crash records suffer from severe underreporting: fatality underreporting reaches 84 % in low-income countries and 51 % in middle-income countries [31], while approximately 50 % of non-fatal crashes and up to 90 % of minor-injury crashes go unreported even in developed regions [32, 33]. Spatial precision is equally problematic: 54.5 % of GNSS-recorded crash locations exhibit discrepancies, with nearly 20 % located off the actual roadway [34].

Temporal delays compound these issues. Traditional traffic message channel (TMC) operations aggregate traffic events from multiple sources and distribute them to road users, but the underlying chain of witness notification, emergency dispatch, and responder delay to the TMC can require up to 30 minutes [35]. Crowdsourced systems such as Waze achieve detection 9.8 minutes faster than commercial traffic information services based on aggregated FCD [36], yet direct analysis of individual vehicle trajectories offers the potential for even earlier detection with higher spatial precision.

FCD provides a scalable and infrastructure-independent alternative. Because the approach relies on GNSS position traces from surrounding traffic rather than from the crashed vehicle itself, it does not require the involved vehicle to carry any sensors. Unlike camera-based or fixed-sensor systems, it leverages existing commercial fleets without additional roadside infrastructure, enabling network-wide deployment at a fraction of the capital investment required by traditional systems [37].

Beyond crashes, the same data can reveal weather-induced traffic disturbances. Understanding the interaction between rain intensity, spatiotemporal rainfall patterns, and traffic is essential for resilient traffic management. Most existing studies rely on aggregated weather data that cannot capture the high spatial and temporal variability of localized rainfall cells (LRCs), limiting the ability to link specific rainfall characteristics to traffic phase transitions [26].

This dissertation investigates the use of FCD for two related applications:

i. Detection of crashes through analysis of crash-induced congestion patterns.

ii. Identification and characterization of localized rain cells and their influence on traffic breakdown.

1.2 Problem Statement

Despite considerable progress in crash detection using advanced sensors, surveillance cameras, and machine learning [38], most operational systems remain constrained by infrastructure costs and data availability [e.g., 39, 40]. Consequently, operational incident detection continues to suffer from multi-minute delays inherent in manual reporting.

While the aggregate effect of rainfall on traffic flow is well established [25, 41], the specific role of LRCs in triggering traffic breakdown remains insufficiently quantified. Aggregated weather data cannot capture the fine spatial and temporal structure of individual rain cells, limiting understanding of how rainfall intensity, cell movement, and cell position relative to the traffic stream influence phase transitions.

The central research gap lies in the absence of scalable, infrastructure-independent methods for crash detection using widely available FCD, without requiring the crashed vehicle to be connected or to carry sensors. A related but secondary gap concerns the integration of FCD with high-resolution environmental information to isolate and assess the effect of localized rain cells on traffic dynamics.

1.3 Research Questions

This research assesses the viability of using FCD as an infrastructure-independent framework for the detection, prediction, and explanation of non-recurrent freeway congestion. Three research questions support this investigation; the first represents the core focus of the work.

RQ1: **Crash Detection and Localization.** Can FCD detect and localize freeway crashes without requiring roadside infrastructure or direct involvement of the crashed vehicle? Specifically:

a) What detection performance can a theory-guided, rule-based approach grounded in three-phase traffic theory achieve?

b) Can deep-learning models operating on theory-inspired representations of the same data improve upon the rule-based baseline?

c) To what extent can these trajectory-based methods improve timeliness and spatial precision compared to conventional traffic information services?

RQ2: **Post-Crash Congestion Prediction.** Can the spatiotemporal propagation of post-crash congestion be predicted with sufficient accuracy to provide actionable travel-time estimates during the early phase of an incident?

RQ3: **Weather-Induced Traffic Breakdown.** How do localized rain cells influence traffic breakdown on freeways, and does the movement direction of a rain cell relative to the traffic stream govern the severity of the resulting congestion?

1.4 Thesis Structure

The remainder of this thesis is organized as follows. Chapter 2 establishes the theoretical foundations and reviews related work. Chapter 3 describes the datasets and the preprocessing framework supporting the subsequent analyses. The core contributions are presented in three parts:

- Chapter 4 addresses **RQ3** with empirical evidence on rain-cell-induced traffic breakdown,
- Chapters 5 to 8 address **RQ1** developing an infrastructure-independent crash detection framework,
- Chapter 9 addresses **RQ2** with a post-crash congestion prediction model.

Chapter 10 synthesizes these findings to discuss methodological advances and concludes the thesis and outlines directions for future research.

2 Foundations of Traffic Breakdown and Crash Detection

Freeway traffic is inherently dynamic and develops in both space and time. Understanding its mechanisms requires a consistent theoretical framework and empirical validation. The study of congestion phenomena serves not only to improve mobility and efficiency but also to enhance safety and resilience of transportation systems [42].

This chapter establishes the theoretical and empirical foundation for analyzing weather- and crash-induced congestion. It introduces relevant traffic variables and data sources, and reviews the major traffic flow theories that explain how traffic states emerge and evolve.

Notes on symbols and units. Unless stated otherwise, distances are given in km, speeds in $\mathrm{km\,h^{-1}}$, and accelerations in $\mathrm{m\,s^{-2}}$. Mathematical objects follow a consistent typographic convention: lowercase italic for scalars and elements (d, t, x), uppercase italic for totals and thresholds (T_b, D), Δ-prefixed for differences (Δt), calligraphic uppercase for sets and domains ($\mathcal{T}$), lowercase bold for vectors ($\mathbf{h}$), uppercase bold for matrices ($\mathbf{W}$), and bold calligraphic for higher-order tensors ($\boldsymbol{\mathcal{X}}$). Subscripts identify the parent object (e.g., $d_{\psi,k}$: k-th distance sample of trip ψ); statistical parameters such as μ and σ are subscripted by the variable they characterize and defined locally. All variables belong to $\mathbb{R}$ unless otherwise noted.

2.1 Traffic Variables and Observation Methods

The macroscopic state of traffic on a roadway segment is described by three fundamental variables: flow, density, and speed [42–45]. Although classically defined from the perspective of stationary observers, these quantities provide the common vocabulary for all traffic flow models and are therefore introduced here before discussing how they are obtained from FCD.

Flow (q) denotes the rate at which vehicles pass a fixed cross-section d during a time interval Δt. If $N(d, \Delta t)$ vehicles cross the section, the flow is

$$q(d,t) \approx \frac{N(d,\Delta t)}{\Delta t}. \tag{2.1}$$

Typical units are vehicles per hour per lane.

Density (ρ) represents the number of vehicles N occupying a road segment of length D around position d at time t:

$$\rho(d,t) \approx \frac{N([d,d+D],t)}{D}. \tag{2.2}$$

It is expressed in vehicles per kilometer. Density cannot be measured directly by most fixed detectors and is therefore often derived from other quantities [45, 46].

Speed (v) The space-mean speed $v(d,t)$ denotes the average speed of all vehicles present in a road segment and is computed as the harmonic mean of individual speeds v_i:

$$v(d,t) = \left(\frac{1}{N} \sum_{i=1}^{N} \frac{1}{v_i} \right)^{-1}. \tag{2.3}$$

It is distinguished from the *time-mean speed* obtained from spot measurements at a fixed location, which is systematically higher. Because the space-mean speed satisfies the fundamental identity

$$q(d,t) = \rho(d,t)\, v(d,t), \tag{2.4}$$

it is the appropriate measure for macroscopic analysis [45, 47]. The fundamental identity equation holds for sufficiently homogeneous traffic and is central to the construction of fundamental diagrams and the theoretical analysis of flow regimes.

Related quantities follow directly from these definitions. The time headway $\tau_i = t_i - t_{i-1}$ between consecutive vehicles relates to flow by $q \approx 1/\bar{\tau}$, while the mean space headway s determines density as $\rho \approx 1/\bar{s}$. Under quasi-stationary conditions, $\bar{s} \approx v\,\bar{\tau}$ links the temporal and spatial domains. Capacity $q_{\max}$ denotes the maximum sustainable flow under prevailing conditions and serves as an operational benchmark [44].

2.1.1 From Stationary Sensors to Floating Car Data

The definitions above assume either a fixed cross-section or a snapshot of a road segment. Traditional infrastructure-based sensors operate in exactly this manner: inductive loops provide continuous counts and occupancies at discrete cross-sections [48], while radar, LiDAR, and video-based systems extend the concept to multi-lane speed and vehicle classification. Although these technologies have matured considerably, their installation and maintenance costs limit spatial coverage, and adverse weather can compromise data quality [49].

FCD invert this perspective. Connected vehicles periodically transmit their position, generating discrete records along their space–time trajectory. Speed is directly observable from these records. Aggregating trajectories over space and time yields estimates of macroscopic traffic variables comparable to those from stationary sensors, yet covering a much wider portion of the network [50, 51]. Flow and density, however, cannot be reconstructed in absolute terms without knowledge of the penetration rate — the fraction of connected vehicles in the total traffic stream [52]. Empirical and simulation-based studies indicate that penetration rates as low as 1–3 % already suffice to reconstruct freeway traffic states with reasonable accuracy [53–57], while higher penetration improves temporal resolution and robustness against sampling noise [58].

A distinctive strength of FCD is that, unlike fixed detectors, they capture the evolution of traffic states along a corridor rather than at isolated points. This provides direct insight into the propagation of congestion fronts and moving disturbances [59, 60] and makes FCD particularly suitable for studying localized disruptions — such as weather-induced breakdowns or crash events — that may not coincide with the location of fixed detectors.

2.2 Spatiotemporal Representation of Microscopic Data

The dynamics of individual vehicles can be represented in a time space diagram (TSD), where each trajectory depicts the temporal evolution of a vehicle's position. Figure 2.1 illustrates this for microscopic FCD: discrete position reports are connected to form trajectories whose slope corresponds to instantaneous speed. Changes in slope reveal acceleration or deceleration, and when many trajectories are plotted together, collective phenomena such as congestion formation and the upstream propagation of traffic fronts become visible [59]. This type of diagram is the primary tool for visualizing FCD and congestion patterns throughout this dissertation and is referred to as a TSD.

Aggregating probe data across vehicles and over short time intervals enables the reconstruction of macroscopic quantities such as local speed, flow, and density [61]. Regions

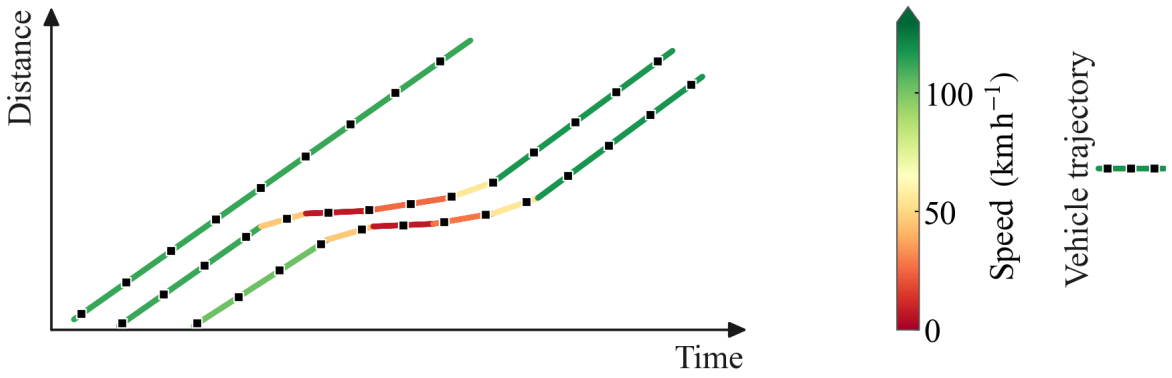


Figure 2.1: Qualitative TSD of microscopic FCD. Dots denote position reports; consecutive points of the same vehicle are connected to establish trajectories. Colors indicate speed, with warmer colors representing higher speeds. The diagram illustrates how trajectories converge in congested regions and how the slope of trajectories corresponds to vehicle speed.

where trajectories converge indicate temporary congestion, and the velocity at which these boundaries propagate is a key observable in empirical studies of traffic breakdown and recovery [59, 60]. The TSD thus provides a direct link between measured vehicle trajectories and the macroscopic traffic flow frameworks discussed in the following section.

2.3 Traffic Flow Theories and Breakdown Mechanisms

Traffic flow theory provides the conceptual framework to explain how collective motion of vehicles emerges from individual behavior and how traffic states form on freeways. Multiple theoretical traditions exist, each offering different levels of abstraction and mathematical formulation. The following overview briefly situates the main approaches before introducing three phase traffic theory (3PTT) [42], which serves as the theoretical basis for this dissertation.

2.3.1 Schools of Traffic Flow Theory

Macroscopic continuum models. The Lighthill-Whitham-Richards (LWR) model [62, 63] treats traffic as a compressible fluid and derives flow dynamics from a fundamental diagram – a fixed relationship between flow and density. This makes it suited to network-level planning and shockwave analysis, but requires flow and density to match the diagram at every instant, ruling out capacity drops, multivalued congested states, and spontaneous

stop-and-go waves. Second-order models relax this by adding a momentum equation for driver reaction and anticipation. The Payne-Whitham formulation [64, 65] reproduces stop-and-go instabilities but lets disturbances outrun the vehicles carrying them, effectively influencing drivers from behind [66] – which the Aw-Rascle-Zhang model [67, 68] resolves by imposing anisotropic wave propagation. Relative to LWR, second-order models trade analytical solvability and calibration simplicity for dynamic realism. Gas-kinetic models [69, 70] instead derive macroscopic equations from statistical assumptions about microscopic interactions, bridging individual and collective behavior at considerable analytical cost.

Microscopic and particle-based models. Car-following models describe individual vehicle motion as a function of spacing and relative speed to the preceding vehicle [e.g., 71–73]. Their direct connection to trajectory-level observables – spacing, speed, acceleration – makes them well suited wherever individual driving behavior matters. The cost is computational: every vehicle must be tracked individually. Cellular automata [e.g., 74] sit at the opposite trade-off: discretizing space and time with stochastic rules sacrifices quantitative accuracy for efficient large-scale simulation.

Stochastic capacity and breakdown models. Stochastic capacity models treat breakdown as a probabilistic event rather than a deterministic instability, triggered when demand or behavioral fluctuations exceed a local stability threshold [45, 75, 76]. These models quantify capacity reliability and breakdown flow distributions at a single cross-section, which is exactly what infrastructure design at known bottlenecks requires. The trade-off is spatial: they describe when breakdown occurs, not where the resulting congestion goes.

Each framework has its domain of strength: LWR for planning and queue analysis, second-order continuum and car-following models for dynamic phenomena, cellular automata for large-scale simulation, stochastic models for capacity reliability. The specific problem of detecting and classifying congestion from sparse vehicle trajectories sits at the intersection of three requirements: operationally distinct states identifiable from local speed, a probabilistic account of breakdown, and propagation signatures observable at low penetration rates. 3PTT is the framework that meets all three, and is adopted as the foundation for this work.

2.3.2 Three Phase Traffic Theory

Empirical studies using single-vehicle and aerial observations revealed that congested traffic does not follow a single, well-defined curve in the flow–density plane [42]. Instead,

measurements form a wide two-dimensional region, indicating that several traffic states may coexist at similar flow levels. To explain these findings, Kerner introduced the 3PTT, which distinguishes three empirically observed traffic phases [42, 77]:

- **Free flow traffic phase (F):** Vehicles move freely at low density, maintaining steady speeds close to the speed limit, with variable overtaking and lane changing.
- **Synchronized flow traffic phase of congested traffic (S):** Congested traffic characterized by continuous flow without significant stoppage and a tendency towards synchronization of vehicle speeds across lanes, typically observed at moderate to high densities. The downstream front of the congested region remains localized at a specific bottleneck, which serves as the focal point for the formation and dissipation of congestion.
- **Jam traffic phase of congested traffic (J):** Within synchronized flow traffic phase of congested traffic (S), disturbances can lead to the spontaneous emergence of wide moving jams (WMJs). A WMJ is characterized by low-speed congestion that is not localized at a specific bottleneck and propagates upstream with a characteristic velocity v_g of its downstream front. On heavily loaded freeways, multiple WMJs can merge into a *mega-jam*: a jam traffic phase of congested traffic (J) of large spatial extent. Such mega-jams typically arise from crashes with high traffic impact.

The three phases are ordered by decreasing mean speed, $\bar{v}_F > \bar{v}_S > \bar{v}_J$, a property that enables their distinction from FCD on the basis of speed thresholds alone (cf. Section 2.3.3).

Figure 2.2 illustrates the three phases and their transitions in the space–time domain. The upper panel shows the velocity profile of a single vehicle trajectory as it traverses the congested region: speed decreases upon entering S and drops sharply within the J before recovering downstream. The lower panel depicts the same trajectory in the distance–time plane, where the three phases appear as distinct spatial regions. Filled markers denote traffic phase transitions (F→S, S→J, and their reversal), while open markers indicate traffic state transitions. The small schematic on the right shows the road profile, with an on-ramp indicating a structural bottleneck at which S persists.

Within this framework, traffic breakdown is interpreted as a *nucleation process*. Small disturbances in free flow traffic phase (F), such as short-term speed reductions or lane changes, dissipate when flow rates are low but may trigger a transition from F to S when traffic demand exceeds a metastable threshold. This metastability implies that F and congested traffic can coexist at comparable flow rates. Once S is established, local self-compression, known as the ”pinch effect”, can occur within the congested region, leading

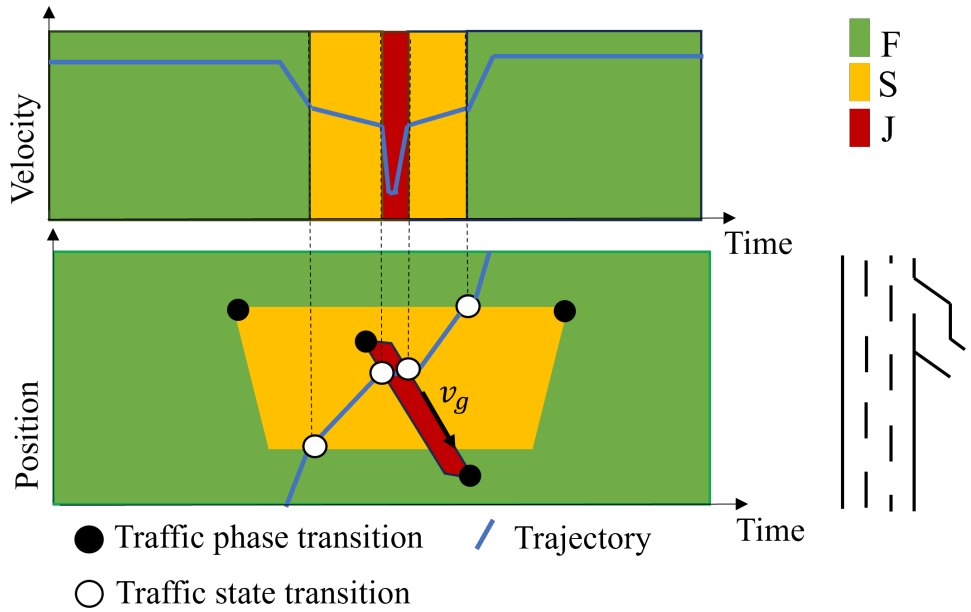


Figure 2.2: Qualitative explanation of traffic phases, their transitions and characteristics (adapted from [78]).

to the spontaneous formation of narrow moving jams. As these narrow jams merge and grow, they transform into J, giving rise to the characteristic sequence $F \rightarrow S \rightarrow J$.

J are spatially bounded by two characteristic fronts. At the upstream front, vehicles decelerate to the low speed inside J, while at the downstream front they accelerate to higher speeds. This behavior is represented in the flow-density plane by the so-called *J-line*, whose slope corresponds to v_g (Figure 2.3).

Breakdown Probability and Capacity Parameters

A key implication of 3PTT is that the outflow rate q_{out} from a J is significantly lower than the empirical maximum free-flow rate $q_{\max}$. In this theory, drivers maintain a minimum safe space gap G_{safe}. A vehicle stopped in a jam accelerates only after the preceding vehicle has reached a sufficient headway, which corresponds to a finite acceleration delay $\tau_{\text{del}}^{(a)}$ [42]. This delay determines both the characteristic jam outflow

$$q_{\text{out}} = \left(1 - \frac{\rho_{\min}}{\rho_{\max}}\right) \frac{1}{\tau_{\text{del}}^{(a)}} \approx \frac{1}{\tau_{\text{del}}^{(a)}}, \tag{2.5}$$

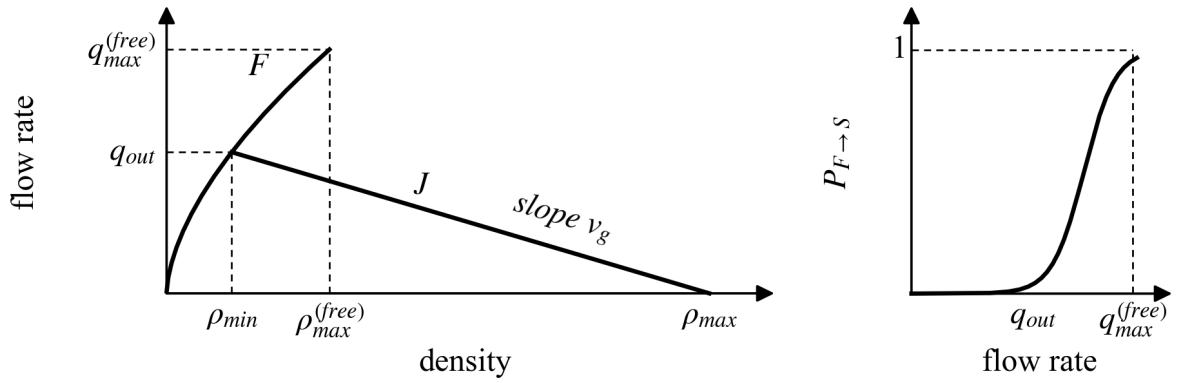


Figure 2.3: Empirical characteristics of metastable F and breakdown probability [42]. Left: J-line demonstrating the metastability of F at high flow rates. Right: Breakdown probability as a function of traffic flow.

and the downstream jam front velocity

$$v_g \;=\; -\frac{1}{\rho_{\max}\,\tau_{\mathrm{del}}^{(a)}}. \tag{2.6}$$

The likelihood of breakdown can be expressed as a function of the prevailing flow:

$$P_{\mathrm{F}\to\mathrm{S}}(q), \quad q \in [q_{\mathrm{out}}, q_{\max}^{\mathrm{free}}], \tag{2.7}$$

where the probability of an F→S transition increases with traffic demand (Figure 2.3). This relationship implies that, at sufficiently high flow rates, both F and congested traffic can coexist. Kerner describes this as the nucleation nature of traffic breakdown [42].

Terminology and Conventions

All subsequent analyses in this dissertation adopt the terminology and conceptual distinctions of 3PTT as introduced above. In particular, WMJs denote congestion patterns that propagate upstream with characteristic velocity v_g at their downstream front, independent of fixed bottleneck locations. When multiple WMJs merge, they form *mega jams* - extended congested regions that can be spatially localized to a persistent root cause such as a crash site (mega jam). All other congested states are referred to as S. For consistency, this dissertation adopts the terminology of 3PTT throughout; equivalent concepts from other frameworks, such as ”shock waves” or ”stop-and-go waves”, are not used.

Table 2.1: Phase detection parameters adapted from [79]. Speed thresholds in kmh^{-1}; dwell times in seconds.

	$v_{\mathrm{F,low}}$	$v_{\mathrm{S,low}}$	$\Delta t^{\mathrm{F}\to\mathrm{S}}$	$\Delta t^{\mathrm{F}\to\mathrm{J}}$	$\Delta t^{\mathrm{S}\to\mathrm{F}}$	$\Delta t^{\mathrm{S}\to\mathrm{J}}$	$\Delta t^{\mathrm{J}\to\mathrm{F}}$	$\Delta t^{\mathrm{J}\to\mathrm{S}}$
Value	75	25	15	25	10	10	10	20

2.3.3 Traffic Phase Detection from FCD

To apply 3PTT to empirical FCD, each spatiotemporal observation along a vehicle's trajectory must be assigned to one of the three phases: F, S, or J. Building on the practical adaptation of this framework by [57, 79], the detection is governed by two complementary criteria:

1. **Velocity condition:** Each phase transition $\Phi \to \Phi'$ is triggered when the instantaneous speed $v(t)$ crosses a defined threshold. Typical predicates include $v(t) > v_{\mathrm{F,low}}$ for the free-flow phase, $v_{\mathrm{S,low}} < v(t) < v_{\mathrm{F,low}}$ for S, and $v(t) < v_{\mathrm{S,low}}$ for J, where $v_{\mathrm{F,low}}$ and $v_{\mathrm{S,low}}$ are calibrated speed thresholds.

2. **Temporal stability:** To prevent misclassification due to short-lived fluctuations or measurement noise, each transition must persist for a minimum dwell time $\Delta t_{\mathrm{dwell}}^{\Phi\to\Phi'}$.

Table 2.1 lists the parameter set proposed by [79], which serves as the baseline configuration in this study.

The detection proceeds in two steps. First, the speed time series is evaluated against the phase-transition predicates to identify contiguous intervals where each condition holds. These intervals are compared to the corresponding dwell-time requirements, yielding valid transition indicators for all pairs $\Phi \to \Phi'$.

Second, a consistent phase sequence is constructed by evolving forward from an initial phase inferred from $v(0)$, allowing transitions only when the corresponding indicator is active and the minimum dwell time since the last change is satisfied. The result is a temporally coherent sequence of phase labels $\Phi \in \{\mathrm{F}, \mathrm{S}, \mathrm{J}\}$ aligned with the original trajectory timestamps.

From this sequence, transitions between phases are used to localize congestion fronts. A transition into a more congested state defines an upstream front Φ_{up}, while a transition into a less congested state defines a downstream front Φ_{down}: $\mathrm{F} \to \mathrm{S}$ and $\mathrm{S} \to \mathrm{J}$ yield S_{up} and J_{up}, while $\mathrm{S} \to \mathrm{F}$ and $\mathrm{J} \to \mathrm{F}$ yield S_{down} and J_{down}. Additional contextual rules resolve ambiguous transitions involving three consecutive phases.

2.4 Bottlenecks and Congestion Patterns

Traffic breakdowns on freeways occur at bottlenecks where local capacity is temporarily or permanently reduced. Two categories are distinguished: *structural* and *event-based* bottlenecks [80, 81]. Structural bottlenecks are fixed points of recurring congestion caused by roadway geometry, merging behaviour, or control restrictions — typically on- and off-ramps, lane drops, or gradients. Their position is stationary and their effect largely predictable, making them well represented by classical macroscopic models [45, 82]. Event-based bottlenecks originate from temporary disturbances such as crashes, vehicle breakdowns, or adverse weather. They are characterized by temporal unpredictability and heterogeneous severity [83, 84]. A further distinction applies within this category: fixed event-based bottlenecks remain at a specific location, while moving bottlenecks (MBs) are caused by slow vehicles traveling within the traffic stream, creating a speed disturbance that propagates at the velocity of the slow vehicle [42, 85]. As vehicles approach a MB, lane-changing manoeuvres generate local speed reductions that move with the slow vehicle rather than remaining stationary in space [86].

2.4.1 Congestion Patterns at Bottlenecks

Both structural and moving bottlenecks can trigger traffic breakdown, which leads to the emergence of diverse synchronized flow congestion patterns (SPs) in congested traffic [42, 78]. Depending on prevailing traffic conditions, the velocity of the bottleneck, and the magnitude of the disturbance, different congestion pattern types can occur in the space–time plane (Figure 2.4):

- Localized congestion pattern (LP): A congestion pattern that remains spatially limited and typically stays attached to the bottleneck.
- Moving congestion pattern (MP): A congestion pattern in which the congested region detaches from the bottleneck and propagates upstream with a characteristic velocity of the upstream front. The bottleneck location and the congested region become spatially separated.
- General congestion pattern (GP): An extended congestion pattern that develops under high traffic flow rates. The pattern covers a large spatial region upstream of the bottleneck and contains multiple WMJs that emerge within synchronized traffic through the pinch effect.

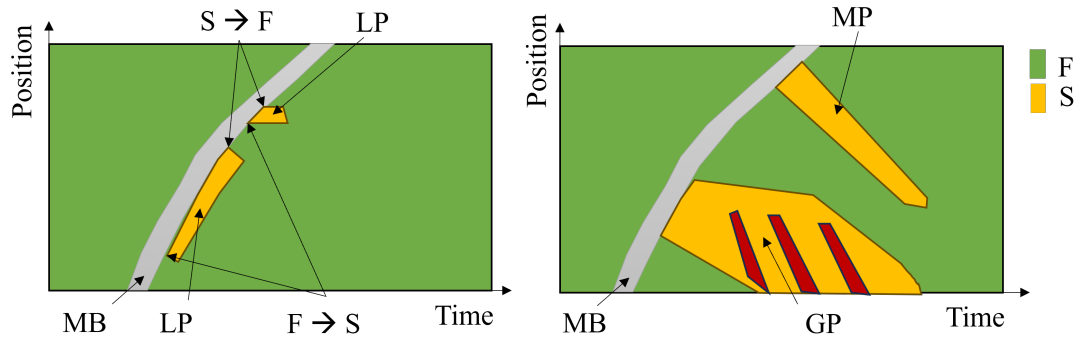


Figure 2.4: Qualitative illustration of diverse congestion patterns and spontaneous breakdowns (F→S) at a MB in the space–time plane [78].

2.5 Precipitation-Induced Congestion

Adverse weather is among the most relevant external factors affecting traffic flow and safety. Rain, snow, and reduced visibility alter driver behavior, modify vehicle-road interactions, and decrease the effective capacity of the road. Climate projections indicate that short, high-intensity rainfall events are becoming more frequent, which highlights the importance of understanding their effects on freeway performance [87–89].

2.5.1 Mechanisms of Influence

Rainfall affects traffic operations through both physical and behavioral mechanisms. Reduced tire-road friction decreases available braking and cornering forces [90, 91], while spray and glare impair visibility and increase perception-reaction times [92]. Sudden environmental noise generated by heavy rainfall can impair drivers' cognitive performance [93], and surface water and spray interfere with environmental sensors that modern driver assistance systems rely on [94, 95]. Simultaneously, drivers adapt by lowering speeds, increasing headway, and adopting more cautious maneuvers [96, 97]. These responses are rational at the individual level but collectively reduce the flow rate. The resulting capacity loss and the higher variability of driver reactions lead to a higher probability of breakdown, particularly when demand approaches the nominal capacity under dry conditions. A meta-analysis of 34 studies confirms these combined effects, showing that precipitation raises crash rates by 71% compared to dry conditions [98].

Table 2.2: Rain intensity classifications [105].

Rain intensity I (mm/h)				Classification
	I	$<$	2.5	Slight
2.5	$\leq I <$	10		Moderate
10	$\leq I <$	50		Heavy
	I	$\geq$	50	Violent

2.5.2 Macroscopic Effects of Precipitation

Numerous empirical studies quantify how precipitation modifies aggregate traffic variables. Reported capacity reductions range between 10–17%, while free-flow speeds typically decrease by 4–9.5% [99–101]. Discharge flow rates at bottlenecks can drop by up to 11% during heavy rain [101]. The probability of breakdown increases systematically with rainfall intensity [102]. These effects are often expressed as multiplicative adjustment factors applied to dry-weather capacity estimates. The magnitude of impact depends not only on rainfall intensity but also on visibility, driver composition, and the duration of exposure. [103] and [104] confirmed consistent speed reductions with increased rainfall intensity using FCD, while [97] and [96] advanced this understanding by analyzing driver headway behaviors during heavy rain, highlighting how adverse conditions lead to increased spacing between vehicles.

Rainfall intensity is conventionally categorized by [105] into four levels, summarized in Table 2.2. These categories are frequently used in traffic studies to relate meteorological observations to measurable traffic effects [103, 104, 106].

Empirical analyzes further show that precipitation can alter the sensitivity of drivers to surrounding vehicles, leading to smoother but slower traffic. In this sense, rain not only changes mean speeds but also modifies the parameters that govern stability. From the perspective of traffic flow theory, these changes correspond to shifts in the equilibrium speed-density relation and to a lower threshold for instability, thereby facilitating earlier transitions from F to congestion. Severe weather has also been linked to the development of general congestion patterns [107]. German crash statistics indicate that approximately 10% of all injury crashes are weather-related [108], reinforcing the broader safety implications of rain-induced disruptions.

2.5.3 Localized Rain Cells

While the general effects of precipitation are well established, localized and transient weather phenomena have received less systematic attention. Short-lived rain cells or convective

showers can create sharp spatial gradients of visibility and friction along a corridor. Vehicles entering such zones experience abrupt changes in conditions and consequently adapt their behavior within seconds. This produces temporary speed discontinuities that can act as MBs, similar in effect to short-lived lane closures. Radar-based studies confirm that these local effects can induce moving congestion patterns that propagate beyond the area directly affected by rainfall [109, 110]. [111] found that rainfall-induced speed reductions led to flow rate declines, suggesting thresholds at which capacity loss becomes significant, while weather-responsive traffic prediction models have been proposed to adjust demand management during inclement conditions [112, 113]. Existing studies predominantly address precipitation as a large-scale and spatially homogeneous phenomenon, relating rainfall to macroscopic reductions in speed and capacity [103, 114, 115]. By focusing on aggregated precipitation measures, they implicitly assume uniform conditions and therefore do not capture the spatiotemporal structure of many real-world rainfall events.

2.6 Crash Detection on Freeways

Traffic crashes represent one of the most disruptive forms of exogenous disturbance in freeway traffic flow. They combine sudden capacity reductions with driver reactions such as braking, lane changes, and visual distraction, resulting in sharp and often long-lasting congestion that is irregular in both space and time [116, 117].

Terminology. The term *crash* is used throughout, following the recommendation of the traffic safety community: whereas *accident* suggests an unavoidable event and may discourage investigation into preventable causes, *crash* emphasizes that collisions have identifiable factors that can be analyzed and addressed [118–121]. The related term *incident*, common in operational literature [e.g., 122–124], covers a broader set of events including vehicle breakdowns and debris; since the present work focuses specifically on vehicle collisions and their effect on traffic flow, the more precise term *crash* is preferred.

2.6.1 Mechanisms and Stages of Disturbance

A crash typically produces two interacting effects. First, the physical obstruction caused by damaged or stopped vehicles reduces the number of available lanes, lowering the effective capacity. Second, drivers approaching the scene react by decelerating and increasing headways, even in unaffected lanes, extending the influence area far beyond the obstruction itself [125].

Empirical studies identify three characteristic stages. Immediately after the crash, a WMJ forms at the crash location and propagates upstream. During this phase, speeds decline sharply and density rises until a quasi-stationary congested region forms downstream of the crash site. Once the scene is cleared and lanes reopen, the recovery phase begins, characterized by gradual acceleration. Recovery is generally slower than breakdown, and the total delay depends on both the duration of obstruction and the intensity of the initial demand [126, 127].

2.6.2 Approaches to Crash Detection

Crash detection methods span a wide range of data sources and reasoning strategies [128]. The following overview groups them by methodology, from single-vehicle and infrastructure-based approaches through theory-guided methods to data-driven and deep learning techniques.

Single-vehicle and broadcast-based systems. The most direct approach detects crashes at the vehicle level. Onboard sensor systems combine accelerometers, GNSS, and gyroscope data to identify collisions from abrupt changes in motion signals [129, 130]. The European eCall system automates emergency calls with vehicle location upon severe impact [131]. These methods enable rapid alerting but are limited to equipped vehicles and may not capture the resulting congestion dynamics. Once detected, crash information is disseminated via broadcast protocols such as transport protocol experts group (TPEG), which encodes traffic events from heterogeneous sources, including eCall, police reports, and traffic control centers, for real-time transmission to navigation systems [132].

Infrastructure-based methods. A large body of work relies on macroscopic traffic flow parameters collected by fixed detectors. Early rule-based methods applied thresholds to loop detector measurements: the Standard Normal Deviate algorithm [133] flagged deviations in lane occupancy, the California Algorithm [134, 135] introduced decision trees with temporal smoothing, and [136] incorporated spatial disparity between upstream and downstream detectors. Classical machine learning subsequently replaced fixed thresholds with supervised pattern recognition, including neural networks [137, 138], Bayesian classifiers [139], and support vector machines [140]. Video-based approaches using object detection and classification networks [141–146] provide rich spatial detail but are constrained by camera coverage and sensitivity to weather and lighting conditions. All infrastructure-based methods share a fundamental limitation: their spatial coverage is determined by detector

or camera placement, leaving large portions of the network unmonitored. [147] reports a precision of just 0.49 — meaning that more than half of its detections are false positives.

Theory-guided methods. A complementary direction grounds detection criteria in traffic flow theory rather than purely statistical patterns.[148] modeled the abrupt transition from F to congestion as a discontinuous process and used the resulting pattern signatures for incident detection. Within 3PTT, the ASDA/FOTO system [42, 149] reconstructs and tracks spatiotemporal congested patterns by identifying S and WMJs, and has been deployed operationally on freeway networks in multiple countries. [79] extended this principle to probe vehicle data. These approaches offer interpretability and physical grounding, though they have primarily targeted congestion reconstruction rather than crash detection specifically.

Data-driven methods using probe vehicle data. With the growing availability of FCD, recent work has shifted towards detecting crashes from trajectory-derived features rather than fixed-point measurements [128, 150]. [150] proposed a grid-based parameter extraction combined with support vector classification and a spatiotemporal Markov model for crash detection in connected vehicle environments. While the reported detection accuracy is high, the method processes data in batch mode and evaluates performance using accuracy; a metric that tends to be biased towards the majority class when crashes are rare events. More broadly, deep learning models have been applied to traffic anomaly detection, including generative adversarial network (GAN)-based data augmentation [151], and graph neural networks (GNNs) [152]. Crash detection on time–distance diagrams derived from probe vehicle trajectories, however, remains largely unexplored.

Each of these directions offers distinct strengths: single-vehicle systems provide immediate alerting, infrastructure-based methods benefit from continuous monitoring at instrumented locations, theory-guided approaches ensure physical interpretability, and data-driven methods enable adaptive feature extraction from complex inputs.

2.7 Convolutional Neural Networks

Convolutional neural networks (CNNs) naturally align with spatiotemporal grids such as time–space diagrams, which provide structured representations of traffic dynamics.

2.7.1 Principles of Convolutional Neural Networks

CNNs are neural architectures designed to process spatially structured data by applying shared kernels across the input [153]. Given an input tensor $\mathbf{X} \in \mathbb{R}^{p_d \times p_t \times C_{\text{in}}}$, a convolutional layer applies C_{out} learnable kernels of size $k_d \times k_t \times C_{\text{in}}$, each computing a weighted sum over a local receptive field followed by a nonlinear activation $\sigma(\cdot)$:

$$\hat{\mathbf{X}}_{i,j,c} = \sigma\left(\sum_{u,v,c'} \mathbf{K}^{(c)}_{u,v,c'} \cdot \mathbf{X}_{i+u-1,\, j+v-1,\, c'} + b_c\right).$$

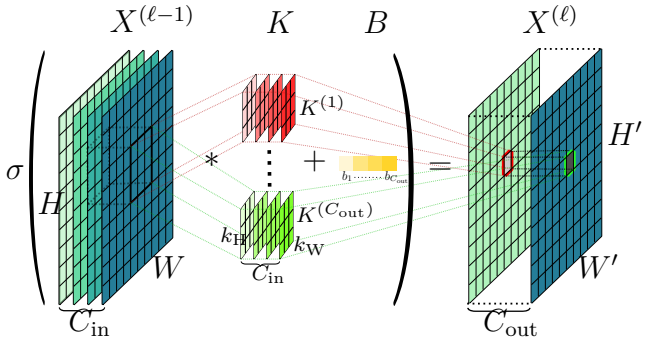


Figure 2.5: CNN principle

As illustrated in Figure 2.5, the convolution operation aggregates local regions from all input channels via shared kernels. In the following, convolution is consistently denoted by $*$.

Weight sharing and local connectivity yield two properties central to this work: *translational equivariance*, meaning that a shifted input produces a correspondingly shifted output, and *hierarchical feature extraction*, where stacking layers progressively captures patterns at increasing spatial scales.

Partial convolution. For incomplete input, such as sparse TSDs with missing observations, the partial convolution mechanism [154] restricts the weighted sum to valid entries and renormalizes accordingly:

$$x' = \begin{cases} \dfrac{\mathbf{W}^\top(\mathbf{X} \odot \mathbf{M})}{\text{sum}(\mathbf{M})} + b, & \text{if sum}(\mathbf{M}) > 0, \\ 0, & \text{otherwise}, \end{cases} \tag{2.8}$$

where $\mathbf{M}$ is a binary mask indicating valid positions and $\odot$ denotes element-wise multiplication; the mask is updated as $m' = \mathbf{1}[\text{sum}(\mathbf{M}) > 0]$. This mechanism is reused in two distinct contexts in the present work: within the 2-D CNN pipeline for sparse image

completion, and within the 1-D Partial Causal TCN for variable-length trajectory encoding (Section 7.3.2).

Encoder–decoder architectures and U-Net. Encoder–decoder CNNs compress the input into a low-resolution latent representation and then reconstruct spatial detail through successive upsampling stages. The U-Net architecture [155] augments this design with skip connections that concatenate encoder feature maps to the corresponding decoder stage, preserving fine-grained spatial information that would be lost during downsampling. Originally proposed for biomedical image segmentation, U-Net has since been applied broadly to dense prediction tasks.

Alternative dense prediction architectures include Feature Pyramid Networks [156], which construct multi-scale feature hierarchies through top-down pathways, and DeepLab-style models [157], which employ dilated convolutions to enlarge the receptive field without reducing resolution. Vision Transformers [158] have also been adapted for segmentation tasks, offering global context modeling through self-attention. For the present application, U-Net is preferred for three reasons: its skip connections preserve the spatial precision required for crash localization in TSDs, partial convolution integrates naturally into its encoder–decoder structure to handle sparse inputs, and its moderate parameter count is well suited to the limited training data available in this study.

2.7.2 CNN-Based Crash Detection on TSDs

The majority of CNN-based crash detection research operates on video streams, using object detection and classification architectures [141, 144, 146]. These methods are unsuitable for network-wide detection from probe vehicle data due to their dependence on camera infrastructure.

Crash detection on TSDs derived from FCD trajectories remains largely unexplored. The central challenge in this setting is data sparsity: limited probe-vehicle penetration yields sparse time–space representations, where large portions of the spatiotemporal field contain no observations. Standard convolutions assume dense inputs and degrade under such conditions. Therefore, Two reconstruction strategies have been proposed to address this: a phase-based smoothing method for traffic speed estimation (PSM) that exploits the propagation characteristics of traffic phases for spatiotemporal interpolation [159], and a CNN-based deep learning approach for impainting [160]. The present work integrates PSM into the CNN-based crash detection pipeline to provide completed inputs for the segmentation model.

2.8 Graph Neural Networks

While CNNs operate on gridded spatiotemporal representations, many aspects of traffic dynamics are more naturally described by relational structures: vehicles interact with their neighbors, and these interactions change as vehicles move. GNNs provide a framework for learning on such irregular and dynamic structures.

2.8.1 Principles of Graph Neural Networks

GNNs are deep learning models for data defined on irregular, non-Euclidean domains [161]. Given a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with node features $\mathbf{h}_i \in \mathbb{R}^d$, the core operation is *message passing* [162]: each node iteratively updates its representation by aggregating information from its neighborhood,

$$m_i = \text{AGGREGATE}(\{\mathbf{h}_j \mid j \in \mathcal{N}_i\}), \qquad \hat{\mathbf{h}}_i = f_\theta(\mathbf{h}_i, m_i),$$

where $\mathcal{N}_i = \{j \mid (j,i) \in \mathcal{E}\}$ denotes the in-neighbors of node i. The choice of aggregation function largely determines the GNN variant: sum or mean pooling yield graph convolution network (GCN)-style layers [163], while learned attention weights yield graph attention networks (GATs) [164].

GATv2. This work adopts GATv2 [165] as the message-passing operator. In the original GAT, attention scores depend effectively only on the sending node, producing a static importance ranking of neighbors. GATv2 modifies the order of operations so that attention scores depend on both the sending and receiving node, allowing the network to weight the same neighbor differently depending on the local context of the query node. Unnormalized attention scores are computed as

$$e(\mathbf{h}_i, \mathbf{h}_j) = \mathbf{a}^\top \text{LeakyReLU}(\mathbf{W}[\mathbf{h}_i \,\|\, \mathbf{h}_j]),$$

where $\mathbf{W} \in \mathbb{R}^{d' \times 2d}$ and $\mathbf{a} \in \mathbb{R}^{d'}$ are learnable parameters and $[\cdot\|\cdot]$ denotes concatenation. Scores are normalized via softmax over the neighborhood, and the updated representation is

$$\hat{\mathbf{h}}_i = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}\, \mathbf{W}_v \mathbf{h}_j\right), \qquad \alpha_{ij} = \frac{\exp(e(\mathbf{h}_i, \mathbf{h}_j))}{\sum_{j' \in \mathcal{N}_i} \exp(e(\mathbf{h}_i, \mathbf{h}_{j'}))}.$$

Given a node-feature matrix $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}| \times d}$, the element-wise update is lifted to a layer

mapping $\Psi : \mathbb{R}^{|\mathcal{V}|\times d} \to \mathbb{R}^{|\mathcal{V}|\times d'}$, $\hat{\mathbf{H}} = \Psi(\mathbf{H}; \mathcal{E})$. When edge type or layer depth must be distinguished, the operator is written $\Psi_a^{(\ell)}$; its specialization is detailed in Section 7.3.2.

2.8.2 Sequence Modelling Components

The GNN-based architecture in this work combines spatial message passing with temporal sequence encoders.

Temporal Convolutional Networks (TCN). Causal dilated convolutions can match or exceed recurrent models on a broad range of sequence tasks while offering full parallelism and stable training [166]. A temporal convolutional network (TCN) stacks causal convolutional layers with exponentially increasing dilation rates, yielding receptive fields that grow exponentially with depth. In Section 7.3.2, this design is extended with partial-convolution masking [154] to handle variable-length vehicle trajectories.

Long Short-Term Memory (LSTM). The LSTM [167] is a gated recurrent architecture that mitigates vanishing gradients through input, forget, and output gates controlling access to a persistent cell state. It is employed in Section 7.3.2 to encode fixed-length supernode history sequences, where its sequential nature is not a bottleneck due to the short, uniform sequence lengths.

Transformer encoder and set attention. The Transformer architecture [168] replaces recurrence with multi-head self-attention, enabling each position to attend to all others in parallel. For aggregating variable-size sets into fixed-size summaries, the Set Transformer [169] uses learnable query vectors to perform permutation-invariant pooling via cross-attention. Both mechanisms are used in the output head of the GNN model (Section 7.3.2).

2.8.3 GNN Applications in Traffic Forecasting and Crash Detection

Spatiotemporal GNNs have been widely adopted for traffic forecasting, as the physical layout of road segments and intersections maps naturally onto graph structures. Early work combined graph convolution for spatial modelling with gated temporal convolution for sequence learning [170]. Subsequent approaches replaced static adjacency matrices with attention-learned dynamic neighbor relations [171] or adaptively constructed graphs from node features and pairwise distances [172]. These forecasting methods share a common

design pattern of combining graph-based spatial operators with temporal sequence models, though they remain confined to prediction rather than detection.

Closer to crash detection, GNN frameworks have been developed for predicting crash-proneness of road segments by jointly modelling temporal, meteorological, and spatial factors [152]. However, all the above approaches model *static infrastructure*, road segments or intersections, as graph nodes. This design is ill-suited for crash detection, where distinguishing normal from crash-induced congestion requires finer, dynamic representations that capture the evolving interactions between individual vehicles.

Two lines of recent work inform the approach taken in this thesis. First, *heterogeneous graph neural networks* provide principled mechanisms for jointly processing multiple node and edge types within a single graph [173]. Second, *dynamic scene graphs* constructed from agent trajectories [174] demonstrate that building graphs on-the-fly from moving entities can capture interaction patterns invisible to static-topology models.

3 Dataset: Acquisition, Preprocessing, and Labeling

This chapter describes the construction of the research dataset, covering the pipeline from raw data acquisition to the generation of labeled traffic states and input construction for deep learning models. The study combines three heterogeneous but complementary data sources: FCD, TPEG traffic event data, and radar-based precipitation data.

3.1 Data Characteristics

The trajectory data were collected across 81 freeway sections in four countries (Germany, Italy, USA, and the United Kingdom) over a period between March 2022 and September 2025. The dataset comprises 1131 records, where each record represents all connected vehicle fleet trips observed on a specific day and road section, totaling 3.8 million individual vehicle trips and 529 million data points. Table 3.1 summarizes the geographic and temporal coverage.

Each trajectory is assigned an anonymous identifier to comply with privacy regulations, with a new identifier generated for each trip. To ensure origin-destination privacy, data are only analyzed on freeway sections with a maximum length of 200 km.

3.1.1 Sampling and Transmission

Vehicles record positional samples at a trip-specific sampling interval $\Delta t \in [2,\, 10]\,\mathrm{s}$, with the majority operating at 5 s. Samples are buffered over a fixed batch duration $T_\mathrm{b} = 120\,\mathrm{s}$ before transmission to the backend. For a trip ψ, this yields a discrete sequence $(d_{\psi,k},\, t_{\psi,k})_{k=0,1,\ldots}$, where k denotes the sample index along the trajectory, with approximately $\lfloor T_\mathrm{b}/\Delta t_\psi \rfloor + 1$ samples per batch. These discrete observations constitute the empirical realization of the continuous trajectory γ_ψ within the monitored domain.

Parts of this chapter are based on previous publications by the author [1–5, 8] as well as on a master's thesis supervised by the author [175].

Table 3.1: Geographic and temporal coverage of the trajectory dataset.

Country	Files	Days	Roads	Sections	Trips	Records (M)
Germany	800	44	6	22	2,904,732	450.1
Italy	16	1	7	16	36,404	6.3
USA*	3	3	3	3	52,582	0.64
United Kingdom	312	10	12	40	794,480	71.7
Total	**1131**	**54**	**28**	**81**	**3,788,198**	**528.7**

*US data provided by HERE Technologies, with higher fleet penetration.

3.1.2 Fleet Penetration Rate

The connected vehicle fleet represents only a fraction of total traffic, with the penetration rate α varying by region, time, and location. On Germany's A81 freeway on March 18, 2022, 331 connected vehicles per hour were detected, while a governmental counting station at the same location recorded 4,135 total vehicles per hour [176], yielding $\alpha \approx 0.084$. Prior research [57, 177] demonstrates that penetration rates as low as $\alpha = 0.02$ are sufficient for effective traffic flow analysis, confirming the adequacy of the observed rates for the methods developed in this work.

3.2 Preprocessing and Feature Computation

The preprocessing pipeline transforms raw trip records into a clean, spatially referenced dataset suitable for traffic state analysis. The following subsections describe each stage in sequential order: temporal alignment and kinematic computation, map matching and spatial filtering, and traffic phase detection.

3.2.1 Temporal Alignment and Kinematic Computation

Timestamps are converted to local time using the geographic coordinates of each position, and missing or corrupted entries are removed. For consecutive measurements at geographic positions $(\varphi_{\psi,k},\ \lambda_{\psi,k})$ and $(\varphi_{\psi,k+1},\ \lambda_{\psi,k+1})$, the traveled distance is obtained via the haversine formula:

$$d_{\psi,k} = R{\cdot}2\arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_{\psi,k+1}-\varphi_{\psi,k}}{2}\right) + \cos(\varphi_{\psi,k})\ \cos(\varphi_{\psi,k+1})\ \sin^2\left(\frac{\lambda_{\psi,k+1}-\lambda_{\psi,k}}{2}\right)}\right), \tag{3.1}$$

where $R = 6371\,\text{km}$ is the Earth's radius. The cumulative distance $D_{\psi,k} = \sum_{j=0}^{k-1} d_{\psi,j}$ serves as the spatial index along the road segment. Velocities and accelerations are obtained by temporal differentiation:

$$v_{\psi,k} = \frac{d_{\psi,k}}{\Delta t_{\psi,k}}, \qquad a_{\psi,k} = \frac{v_{\psi,k+1} - v_{\psi,k}}{\Delta t_{\psi,k}}, \tag{3.2}$$

where $\Delta t_{\psi,k} = t_{\psi,k+1} - t_{\psi,k}$. A smoothing filter is applied to reduce measurement noise.

To identify physically consistent trajectory segments, the time gaps $\Delta t_{\psi,k}$ are analyzed. If a gap exceeds 1.5 times the median gap of the corresponding session, the trajectory is split into separate trips to avoid unrealistic interpolation.

3.2.2 Map Matching and Spatial Filtering

Raw geographic coordinates are projected onto the digital road network using the open-source routing engine Valhalla [178], which provides map-matching functionality based on the OpenStreetMap network [179]. Each GNSS observation is assigned to the most probable road segment, transforming the geographic tuple into (d, t, ψ)—the fundamental representation used throughout subsequent analysis. The matched data additionally provide edge attributes including road class, lane count n_{lane}, posted speed limit v_{lim}, and geometric properties such as curvature, grade, and elevation. A corridor-specific set of way identifiers constrains the dataset to a freeway section defined by $d_{\min}$ and $d_{\max}$; only the portions of each trajectory that fall within this section are retained. As a consequence, a vehicle that exits the freeway and later re-enters via an on-ramp will produce two spatially close subsets of observations that may be separated by a large time gap. Conversely, a vehicle passing through the section twice within a short period may produce observations that are close in time but far apart in space. In both cases, naively connecting consecutive observations would create unphysical jumps. Such discontinuities are detected and the trajectory is split into distinct trips to preserve spatial and temporal consistency. Valhalla is operated on local infrastructure, ensuring full control over data privacy and reproducibility.

Figure 3.1 shows an example velocity-colored TSD for the A81 segment between Stuttgart and Bad-Dürrheim on 2022-03-18, illustrating the data after preprocessing and map matching.

3.2.3 Traffic Phase Detection

The traffic phase detection method introduced in Section 2.3.3 is applied to classify each observation into F, S, or J, using the baseline parameter set of [79]. For the German data,

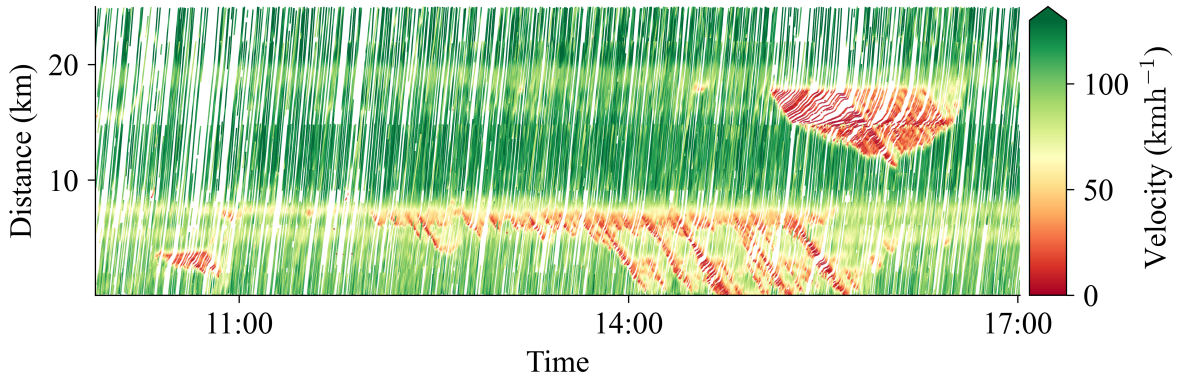


Figure 3.1: Velocity TSD for the A81 segment between Stuttgart and Bad-Dürrheim on 2022-03-18. Colors encode instantaneous speed (kmh^{-1}); horizontal axis is time, vertical axis is distance along the route.

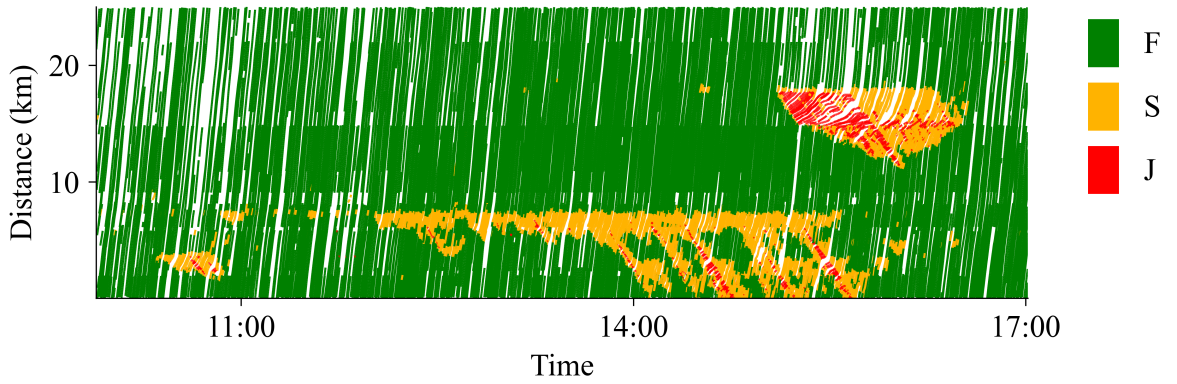


Figure 3.2: Phase-colored TSD for the A81 segment between Stuttgart and Bad-Dürrheim on 2022-03-18. Colors encode traffic phases following [42].

the speed thresholds are adjusted to $v_{\mathrm{F,low}} = 65\,\mathrm{km\,h^{-1}}$ and $v_{\mathrm{S,low}} = 15\,\mathrm{km\,h^{-1}}$ to account for higher prevailing speeds on unrestricted freeway sections and differences in driving behavior. All dwell-time parameters remain unchanged.

Figure 3.2 shows an example phase-colored TSD for the same A81 segment, illustrating the output of the detection pipeline.

3.3 Region Clustering

The region clustering procedure follows the method introduced by the author in [8].

To ensure the reliability of subsequent tasks such as crash detection and congestion cause inference, coherent spatiotemporal regions of congested traffic must be identified with high precision. Therefore, a density-based clustering procedure is applied to J and S in the time–distance domain, using a two-dimensional feature space spanned by absolute time and longitudinal road coordinate. A modified density-based spatial clustering of applications with noise (DBSCAN) algorithm with a custom distance metric—referred to here as Anisotropic Phase-Aware DBSCAN (AP-DBSCAN)—is used to reflect spatial proximity and the dynamic propagation of congestion fronts. The distance metric is aligned with the directional movement of WMJ fronts, which propagate upstream at approximately $v_g = -15\,\mathrm{km\,h^{-1}}$. DBSCAN is suitable because it requires no predefined number of clusters, is robust to noise, and detects clusters of arbitrary shape [180, 181].

For each phase $\Phi_{\text{congested}} \in \{J, S\}$, observations are represented as feature vectors $\mathbf{z}_{\psi,k} = (t_{\psi,k},\, d_{\psi,k})^T$, where $t_{\psi,k}$ denotes the absolute time and $d_{\psi,k}$ the longitudinal road coordinate of sample k in trip ψ.

Anisotropic distance metric. The distance between two observations m and n is computed by first rotating the coordinate system to align with the expected propagation speed v_{ideal}, then applying anisotropic scaling that penalizes deviations perpendicular to this direction. Let $\Delta t = t_n - t_m$ and $\Delta d = d_n - d_m$ denote the temporal and spatial separation. The rotated and scaled coordinates are

$$\Delta_{\parallel} = \Delta t \cos\theta + \Delta d \sin\theta, \qquad \Delta_{\perp} = -\Delta t \sin\theta + \Delta d \cos\theta, \tag{3.3}$$

where $\theta = \arctan(v_{\text{ideal}})$. The anisotropic core distance is then

$$d_{\text{core}} = \sqrt{\left(\frac{\Delta_{\parallel}}{\sigma_{\parallel}}\right)^2 + \left(\frac{\Delta_{\perp}}{\sigma_{\perp}}\right)^2}, \qquad \sigma_{\perp} > \sigma_{\parallel}, \tag{3.4}$$

where $\sigma_{\parallel}$ and $\sigma_{\perp}$ are scaling semi-axes along and across the propagation direction, respectively. Two observations are considered neighbors if either of the following conditions is met: $d_{\text{core}} < r_{\text{core}}$ for a local radius threshold r_{core}, or the observed propagation speed $\Delta d/\Delta t$ lies within $v_{\text{ideal}} \pm \Delta v_{\text{tol}}$—provided both separations satisfy $|\Delta d| < D_{\text{gap}}$ and $|\Delta t| < T_{\text{gap}}$. Otherwise, the distance is set to a prohibitively large value to prevent spurious connections. With $\sigma_{\perp} > \sigma_{\parallel}$, the neighborhood favors continuity along the propagation direction and suppresses cross-structure links, keeping it consistent with WMJ propagation.

Table 3.2: Tuned AP-DBSCAN hyperparameters for German freeway sections.

$\Phi_{\text{congested}}$	r_{core} [m]	v_{ideal} [m/s]	Δv_{tol} [m/s]	D_{gap} [m]	T_{gap} [s]	ε	*minPts*	κ
J	202	−4.77	1.12	2085	270	3831	14	5
S	389	−2.29	4.78	840	931	3520	43	1

Phase-aware gap adaptation. Because FCD coverage is sparse, the clustering must bridge gaps in observations without merging unrelated traffic structures. A phase-aware gap field $P_\Phi(t, d)$ is constructed for each phase $\Phi \in \{\mathrm{F}, \mathrm{S}, \mathrm{J}\}$ by computing the relative proportion of that phase in local time–space bins. During clustering, the field is evaluated at the midpoint between candidate neighbors:

$$g_{mn} = P_\Phi(t, d)\big(\tfrac{1}{2}(\mathbf{z}_m + \mathbf{z}_n)\big). \tag{3.5}$$

The effective neighborhood radius is then adapted as

$$\varepsilon_{\text{eff}} = \varepsilon\,(1 + \kappa\, g_{mn}^{\eta}), \tag{3.6}$$

where $\kappa \in [0, 1]$ controls adaptation strength and $\eta \geq 1$ emphasizes strong phase consistency. When $g_{mn} \approx 1$, indicating homogeneous phase coverage, the radius expands to bridge likely data gaps. When $g_{mn} \approx 0$, the standard radius ε is maintained to avoid merging across phase boundaries.

Parameter selection. The hyperparameters of the distance metric and gap adaptation, together with the DBSCAN parameters (ε, `minPts`), are tuned per country and phase to account for differences in probe density and regional traffic behavior. An observation becomes a core point if at least `minPts` neighbors lie within ε_{eff}, and clusters with fewer than 50 observations are discarded as noise. Parameter values are selected through iterative tuning and visual validation so that identified regions align with characteristic jam propagation patterns; the tuned values for German freeway sections are listed in Table 3.2.

Figure 3.3 shows an example clustering result for the A81 segment between Stuttgart and Bad-Duerrheim on 2022-03-18.

Comparison to state-of-the-art algorithms. To evaluate the performance of the proposed AP-DBSCAN method, it is compared against several established DBSCAN variants, including Spatio-Temporal DBSCAN (ST-DBSCAN) [182], HDBSCAN [183], and standard DBSCAN [180]. Each algorithm is applied to the same set of traffic phase data for

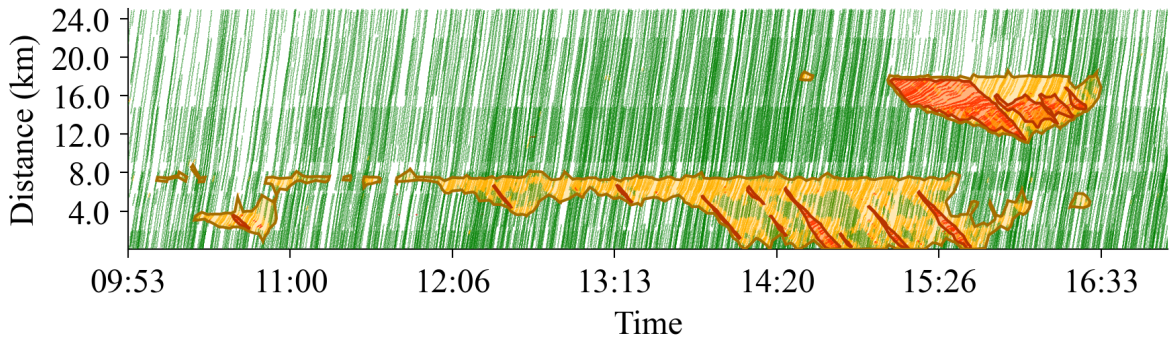


Figure 3.3: Clustering of congested regions based on traffic patterns on A81 near Stuttgart. Colors indicate distinct clusters identified by the DBSCAN algorithm; horizontal axis is time, vertical axis is distance along the route.

multiple freeway segments in Germany and optimized using a Tree-structured Parzen Estimator (TPE) [184] implemented in Optuna [185] to select hyperparameters that maximize clustering quality. Ground-truth labels for congested regions are obtained through manual annotation by the author, who delineates the spatial and temporal extents of congestion phases based on speed time–space diagrams and known traffic events. Predicted clusters are converted into phase-specific polygons in the time–distance domain and compared against the labeled regions. As the primary evaluation measure (score function $f_S(\cdot)$), a combined intersection over union (IoU)-count metric is used, defined as

$$f_S = 0.7\, f_S^{\mathrm{J}} + 0.3\, f_S^{\mathrm{S}}, \tag{3.7}$$

$$f_S^{\Phi} = \frac{1}{2}\,\mathrm{IoU} + \min\left(\frac{M_{\mathrm{gt}}}{M_{\mathrm{pred}}},\ \frac{M_{\mathrm{pred}}}{M_{\mathrm{gt}}}\right), \tag{3.8}$$

where M_{gt} and M_{pred} are the numbers of annotated and reconstructed congestion regions. The score is bounded in $[0, 1]$ and attains its maximum only when both the spatial extent and the structural composition of congestion phases are recovered faithfully. It is adopted as the primary measure because it directly couples the two practical objectives of congestion-phase reconstruction—recovering the correct number of coherent jam regions and approximating their aggregated spatiotemporal footprint—without relying on precise polygonal boundaries or instance-level matching heuristics. For completeness, Variation of Information (VI), Panoptic Quality (PQ), and F1 are additionally reported as established cross-checks on segmentation structure and geometric fidelity.

Figure 3.4 compares the clustering results of the proposed AP-DBSCAN method with the other DBSCAN variants for three freeway segments in Germany, and Table 3.3 summarizes

Table 3.3: Comparison of clustering results. Best score in **bold**, second best underlined. Arrows indicate optimization direction.

Method	$f_S \uparrow$	$\mathrm{VI}_{\mathrm{total}} \downarrow$	PQ ↑	F1 ↑	$\mathrm{IoU}_J \uparrow$	$\mathrm{IoU}_S \uparrow$
Standard DBSCAN	0.5867	0.988	0.059	0.087	0.592	0.381
ST-DBSCAN	0.5484	1.092	0.052	0.078	0.511	0.418
HDBSCAN	0.4653	1.441	0.042	0.007	0.392	0.333
AP-DBSCAN	**0.5940**	**0.706**	**0.165**	**0.248**	**0.607**	**0.456**

the corresponding quantitative results. AP-DBSCAN achieves the best score across all reported metrics, confirming both improved geometric coverage and more faithful recovery of the congestion-phase structure.

3.4 Precipitation Data

Precipitation data from the German Meteorological Service (*Deutscher Wetterdienst*) (DWD) [186] are used. The Radar-Online-Adjustment (German: *RAdar-OnLine-ANeichung*) (RADOLAN) dataset combines observations from 17 C-band weather radars with measurements from more than 1400 automatic rain gauges to produce gauge-adjusted areal precipitation estimates on a 1 km grid at 5-minute intervals [187]. The high spatial and temporal resolution allows for detailed observation of moving rain fronts and is widely used in hydrological modeling and meteorological applications (e.g., [187, 188]).

To align precipitation and traffic data, trajectory timestamps are converted to UTC and rounded to the nearest 5-minute RADOLAN interval. Each observation is assigned to the precipitation intensity of the nearest grid cell, with invalid pixels masked. Figure 3.5 illustrates the resulting integrated dataset for an exemplary freeway section, combining vehicle speeds, precipitation intensity, and infrastructural features in a single time-space diagram.

3.4.1 Precipitation Coverage

The German subset including precipitation data spans 13 corridors (26 directed sections, $\approx 1\,060$ km) across 43 days between March 2022 and January 2025. The analysis is performed at the level of *section-day* observations, where each observation corresponds to one freeway segment, one travel direction, and one calendar day. A section-day is classified as *wet* when at least one spatially coherent rain cluster - five or more connected

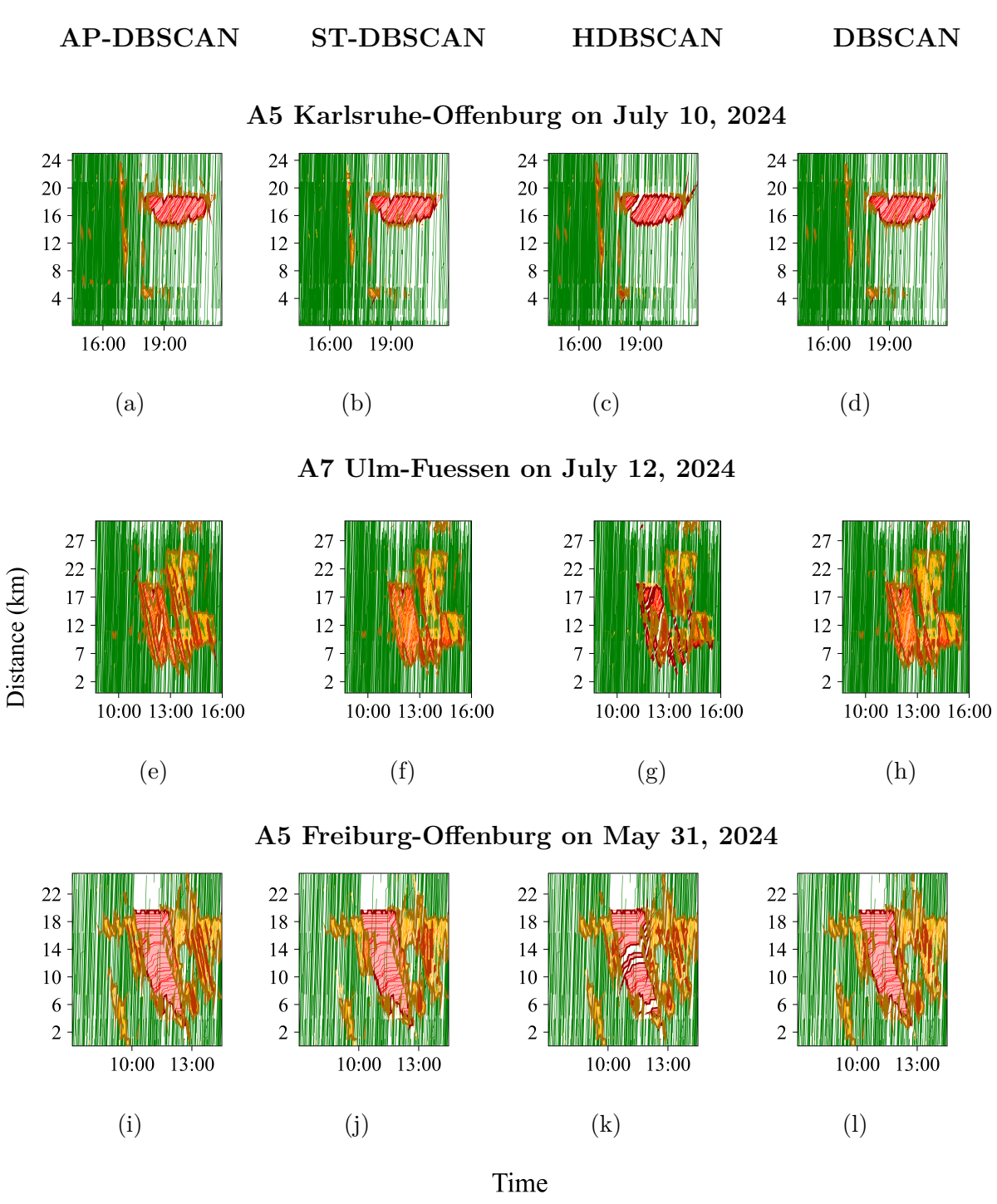


Figure 3.4: Comparison of region clustering results for three freeway segments in Germany using different DBSCAN variants: (a) AP-DBSCAN (this work), (b) ST-DBSCAN [182], (c) HDBSCAN [183], and (d) standard DBSCAN [180]. Each row corresponds to a different freeway segment and date, with colors indicating distinct clusters identified by each algorithm. Horizontal axis is time, vertical axis is distance along the route.

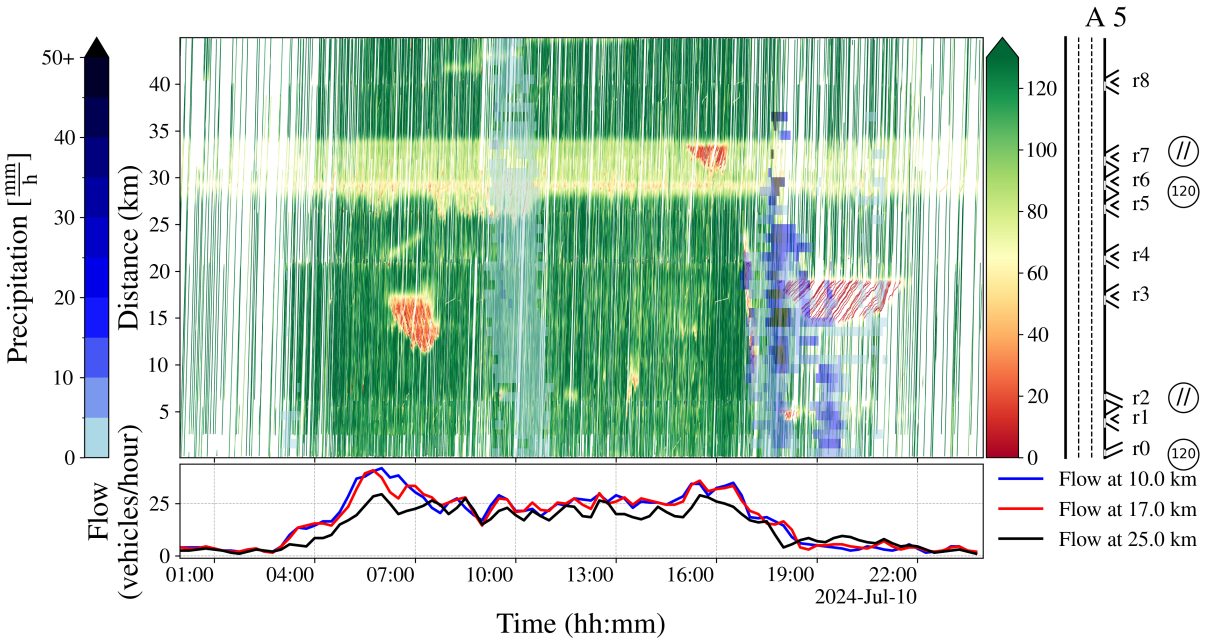


Figure 3.5: Dataset including rain data, infrastructural bottlenecks, traffic flow and FCD. A5 between Karlsruhe and Offenburg, on July 10, 2024.

grid cells exceeding 10 mm/h - is detected, restricting the analysis to heavy and violent events. Under this definition 509 of 857 section-days (59.4 %) are wet. Even on wet days, precipitation remains localized, covering on average 22.6 % of the time–space observation window (Table 3.4).

3.5 TPEG Event Data

TPEG defines a standardized framework for broadcasting traffic and travel information via digital radio, cellular networks, and internet-based services [132, 189, 190]. It is widely

Table 3.4: Rain-cluster characteristics of wet section-days (peak intensity > 10 mm/h, $n = 509$).

Variable	Mean	SD	Max
Peak intensity (mm/h)	41.0	40.1	347.4
Mean intensity (mm/h)	2.1	1.1	7.7
Wet area fraction (%)	22.6	16.0	94.6
Rain clusters per section	15.9	10.6	64.0
Cluster duration (min)	76.2	89.3	740.0
Cluster extent (km)	22.4	12.8	123.9

adopted in commercial real time traffic information (RTTI) services across Europe and forms the basis for most current event broadcasting systems [191]. Each message describes a traffic event by a start and end coordinate on the road network, with optional offsets refining the spatial extent.

Archived TPEG messages covering all recorded days were obtained from a commercial provider (TomTom AG) and aligned with FCD. Each TPEG event is valid for two minutes and may be reissued if the event persists, with updated temporal or spatial attributes. The dataset aggregates incident reports from heterogeneous sources, including public traffic control centers, emergency call systems, FCD, road work notifications from highway authorities, and weather-related hazard alerts from meteorological services.

3.5.1 Preprocessing and Spatial Referencing

The preprocessing converts raw TPEG broadcast messages into a road-aligned spatiotemporal representation that serves as a common reference layer for manual annotation and subsequent analysis. The reference coordinates of each message are projected onto the freeway axis using a two-stage map-matching procedure: a coarse nearest-neighbor search first assigns the message to the correct freeway section, followed by a refined search in a locally projected coordinate system to obtain precise linear positions d_{start} and d_{end}. If offset information is present, the effective spatial extent is computed as

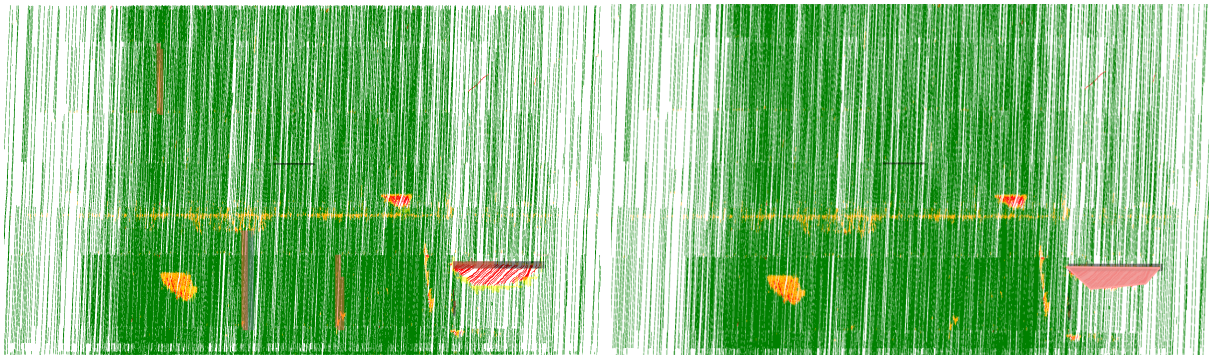
$$d^*_{\text{start}} = d_{\text{start}} + d_+, \qquad d^*_{\text{end}} = d_{\text{end}} - d_-, \tag{3.9}$$

where d_+ and d_- denote the positive and negative spatial offsets provided in the TPEG message. Messages with inconsistent spatial ordering are discarded to ensure a unique travel direction along the freeway. The resulting road-aligned event representation can be directly overlaid with time-space diagrams and preserves all original messages for transparent comparison with automated detection methods.

3.5.2 Ground-Truth Labeling

Raw TPEG messages cannot serve directly as ground truth for crash detection. Crashes are not consistently labeled using the dedicated event code *crash*; instead, related codes are frequently transmitted without an explicit crash label. Limitations regarding timeliness, completeness, and spatial accuracy of RTTI systems are moreover well documented in the literature [192–194].

To establish a robust ground truth, the dataset is manually labeled in Label Studio [195]:



(a) TSD diagram colored by velocity and crash-related TPEG event codes

(b) Annotated TSD diagram with GT polygon label

Figure 3.6: TSD files used for labeling are plotted without axes or borders to allow easy matching of annotated pixel positions to the time-space domain of the dataset. The example shows two raw TPEG event codes and a TPEG – Mega jam aligned with the second event code.

time-space diagrams are overlaid with road-aligned TPEG messages, and annotated polygons are drawn to span the spatiotemporal extent of each event. This approach enables identification of post-crash congestion grounded in traffic flow dynamics, while avoiding unintended learning of delayed or incomplete TPEG signals. Figure 3.6 illustrates the labeling procedure.

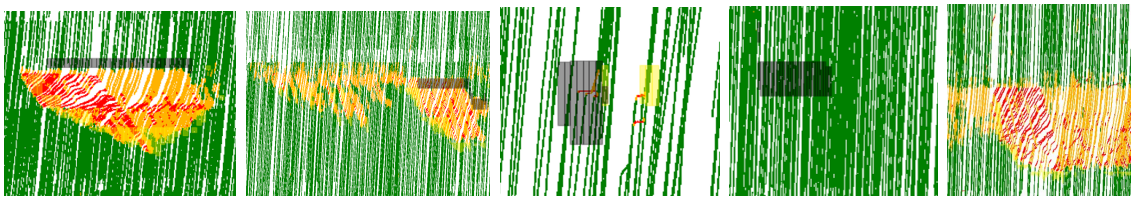
TPEG defines 31 event codes covering traffic conditions, incidents, and regulatory information (see Section A.1 for the complete list) [132]. Event codes *crash*, *fire*, and *rescue and recovery work in progress* are used to identify congestion patterns likely caused by crashes. Since not all crash-related messages induce observable congestion, three ground-truth categories are defined:

TPEG – Mega jam A large-scale J-phase congestion pattern temporally and spatially aligned with one or more crash-related TPEG event codes.

TPEG – General Pattern A congestion pattern of $\Phi_{\text{congested}}$, typically containing extended S-phase with multiple WMJs, aligned with a crash-related event code.

TPEG – Full blockage (FB), <3 trajectories A crash-related event code coinciding with near-complete traffic interruption, evidenced by fewer than three trajectories γ_ψ traversing the affected section.

Each congestion pattern that aligns with a TPEG event code is assigned exactly one of the above categories based on its observed characteristics. Original event codes are additionally preserved as *TPEG raw event code* for transparent comparison. Congestion patterns with



(a) TPEG Mega jam (b) TPEG General Pattern (c) TPEG < 3 Trajectories (d) TPEG raw event code (e) Mega jam (no event)

Figure 3.7: Examples for each labeling category present in the dataset. TPEG event code is annotated in $(a) - (d)$

mega-jam characteristics but no corresponding TPEG message are labeled as *Mega jam (no event)*, since their traffic flow signature is indistinguishable and they should be detected equally by data-driven approaches. Examples of each category are shown in Figure 3.7.

Labeling statistics and justification. All 1,140 files were manually annotated. Of these, roughly half contained at least one congestion event; the remainder exhibited free flow or minor congestion only. The process identified 445 mega jams and FBs (Table 3.5) that serve as the target variable for subsequent detection algorithms.

To evaluate whether TPEG reports could serve directly as ground truth, all 828 incident notifications were cross-referenced with the FCD. Two systematic deficiencies emerged: approximately 43 % of reports lacked any observable traffic impact, and 20–25 % of severe events visible in the trajectories had no corresponding report at all, with the highest shares in Germany and the United Kingdom. GPs were excluded as detection targets because they are indistinguishable from recurrent congestion, and FBs with fewer than three trajectories lack sufficient observability for model training. These findings, together with the inherent reporting delay Δt_{TPEG} and spatial inaccuracies of TPEG, confirm that manual trajectory-based annotation is necessary to establish reliable labels. Throughout the manuscript, the specific ground-truth label set used in each experimental phase is explicitly stated.

3.6 Ground-Truth Sets

The previous section established that raw TPEG messages are neither complete nor fully reliable as crash labels. This raises the question of what constitutes an appropriate ground-truth definition for model training and evaluation. Depending on the analytical objective and the degree of trust placed in TPEG reporting, three complementary ground-truth sets can be constructed from the labeled data, each with a distinct rationale.

Table 3.5: Distribution of labeled congestion events by source and country.

Event Category	Total	Germany	UK	Italy	USA
Target labels: Mega jams and FBs					
Mega jams with TPEG report	326	266	53	4	3
Mega jams without TPEG report	91	73	16	2	—
FBs (no report)	28	22	6	—	—
Total target labels	**445**	361	75	6	3
TPEG incident reports (candidate labels)					
All TPEG reports	828	721	101	6	3
with impact	474	388	80	6	3
without impact	354	333	21	0	0

Set A — TPEG Raw events. Set A comprises all TPEG messages whose event code relates to a crash (*crash*, *fire*, *rescue and recovery work in progress*), regardless of whether the event produced measurable traffic impact. This set reflects the perspective that every reported crash is relevant, even if traffic flow recovered before congestion became observable in the probe data. It is the most permissive definition and directly inherits the limitations of the TPEG system: false positives from rapidly cleared incidents and missing events that were never reported.

Set B — TPEG event codes with measurable traffic impact. Set B retains only those TPEG crash-related messages that coincide with an observable congestion pattern in the FCD, i.e. events labeled as *Mega jam*, *General Pattern*, or *FB*. By requiring measurable traffic impact, this set removes false positives but remains dependent on TPEG coverage and therefore still misses unreported events.

Set C — All mega jams. Set C includes all congestion patterns that exhibit mega-jam characteristics, irrespective of whether a corresponding TPEG message exists. Events without a TPEG report are included because their traffic flow signature is indistinguishable from reported crash events (cf. Section 3.5.2). FBs are deliberately excluded from this set: with fewer than three trajectories γ_ψ traversing the affected section, they are too sparsely covered to be reliably detected from FCD alone. Set C is therefore the most suitable definition for training detection models that operate solely on FCD, as it captures the full population of FCD-observable severe congestion.

Throughout this work, Set C is used exclusively for model training and evaluation, as it does not depend on the availability or reliability of external TPEG reports; Sets A and B

Table 3.6: Overview of crash-related congestion categories, cumulative durations, and membership in ground-truth sets. Rows are not mutually exclusive and must not be summed (see text).

Category	Count	Duration [h]	A	B	C
TPEG — Raw event	822	727.3	✓		
TPEG — General Pattern	102	133.8		✓	
TPEG — FB < 3 Trajectories	42	15.0		✓	
TPEG — Mega jam	324	588.5		✓	✓
Mega jam (no event)	89	151.6			✓
FB (no event)	28	13.1			✓

are presented as alternative definitions but are not used in the experiments.

3.6.1 Dataset Statistics

The total observation time T_{obs} is defined as the aggregated monitoring duration over all road sections:

$$T_{\text{obs}} = \sum_{i=1}^{K} T_i, \tag{3.10}$$

where T_i denotes the observation time of road section i and K is the total number of road sections. The dataset comprises $T_{\text{obs}} = 27\,103\,\text{h}$ of freeway traffic. Table 3.6 summarizes the number of events and cumulative duration per labeling category, together with the ground-truth sets to which each category contributes. The categories are not mutually exclusive: every TPEG-reported event is preserved both under its raw event code and under the category assigned from the observed traffic pattern, so the rows of Table 3.6 must not be summed. The duration columns are likewise not directly comparable — for TPEG Raw events the duration reflects the broadcast duration of the TPEG message, whereas for the categorized rows it reflects the observed temporal extent of the congestion structure. In total, distinct crash-related congestion events are identified; the remaining observation time corresponds to free-flow or non-crash congestion.

For each ground-truth set the crash-related congestion time T_{cr} and corresponding crash ratio η_{cr} are computed as

$$\eta_{\text{cr}} = \frac{T_{\text{cr}}}{T_{\text{obs}}}. \tag{3.11}$$

Table 3.7: Crash-related congestion time and crash ratio per ground-truth set.

Set	T_{cr} [h]	η_{cr} [%]
A	727.3	2.68
B	737.3	2.72
C	740.1	2.73

Table 3.7 reports the results for sets A, B, and C. Across all three definitions, crash-related congestion accounts for approximately 2.7 % of total freeway observation time, confirming severe class imbalance that must be addressed during model training.

3.6.2 Train-Validation-Test Split

The dataset is split into training (70 %), validation (15 %), and test (15 %) partitions, stratified by country and date. Data-driven models are trained on the training set, with the validation set used for hyperparameter tuning. The rule-based expert-knowledge method is calibrated on the validation set; no training partition is required for this approach. All models are evaluated on the same held-out test set, ensuring a fair comparison.

3.7 Input Preparation for Deep Learning Models

Data-driven approaches enable the extraction of complex spatiotemporal patterns directly from trajectory observations without requiring a fully analytical formulation of traffic flow dynamics. In this work, a CNN-based and a GNN-based model are considered as complementary learning-based approaches for crash detection. Both models address the same binary classification task: given a detection time t_{det} and a spatial anchor on the freeway, determine whether a crash-induced congestion pattern is present.

3.7.1 Observation Window

Both models operate on the spatiotemporal observation window $\mathcal{W}_{D,T}$. Given a detection time t_{det} and a downstream boundary d_{down}, the window is defined as

$$\mathcal{W}_{D,T} = [d_{\text{down}} - D,\ d_{\text{down}}] \times [t_{\text{det}} - T,\ t_{\text{det}}]. \tag{3.12}$$

3.7.2 Class Imbalance

Traffic data is strongly imbalanced: free-flow conditions dominate, non-crash congestion is less frequent, and crash-related congestion is rare (Section 3.6). Naive uniform sampling at the window level would produce a crash-to-non-crash ratio of approximately 1:36 (2.7% crash rate), hindering the learning of characteristic post-crash congestion patterns. To address this, three complementary sampling modes are applied sequentially to generate candidate observation windows for training and validation. Crash windows are moderately oversampled to obtain an effective ratio of approximately 1:4, while the test set remains unmodified to reflect realistic deployment conditions.

3.7.3 Window Sampling Strategy

Velocity-weighted sampling. This mode overrepresents congested traffic relative to free flow, which is essential for learning traffic breakdown and congestion formation. The spatial domain is divided into N_d segments and the temporal domain into N_t segments, yielding an $N_d \times N_t$ grid of spatiotemporal cells. Each cell (m, n) is assigned a sampling weight inversely proportional to its mean speed:

$$p_{mn} = \frac{w_{mn}}{\sum_{m,n} w_{mn}}, \qquad w_{mn} = \frac{1}{v_{mn} + \epsilon}, \tag{3.13}$$

where v_{mn} is the mean speed in cell (m, n) and ϵ is a small constant preventing division by zero. Cell-level probabilities are smoothed and normalised to avoid degenerate cases.

For each trajectory record, $M_{\text{seg}} = 10$ cell pairs are drawn according to p_{mn}. From each selected pair, $M_{\text{win}} = 4$ candidate windows are uniformly sampled within the cell bounds. Regions containing labelled crash events are excluded from this mode to avoid overrepresentation of crash-specific patterns in the velocity-weighted subset. Figure 3.8 illustrates the procedure.

Crash-centred sampling. This mode generates samples exclusively around manually annotated crash events (Section 3.5.2). For each crash with temporal extent $[t_{\text{start}}, t_{\text{end}}]$ and spatial location d_{crash}, window centres are drawn from

$$t_{\text{centre}} \in \left[t_{\text{start}} - \Delta t_{\text{before}},\ t_{\text{end}} + \Delta t_{\text{after}}\right], \tag{3.14}$$

where Δt_{before} and Δt_{after} are temporal buffers that extend the sampling range into pre-crash and post-crash recovery conditions. Per crash event, $M_{\text{crash}} = 40$ windows are drawn. The

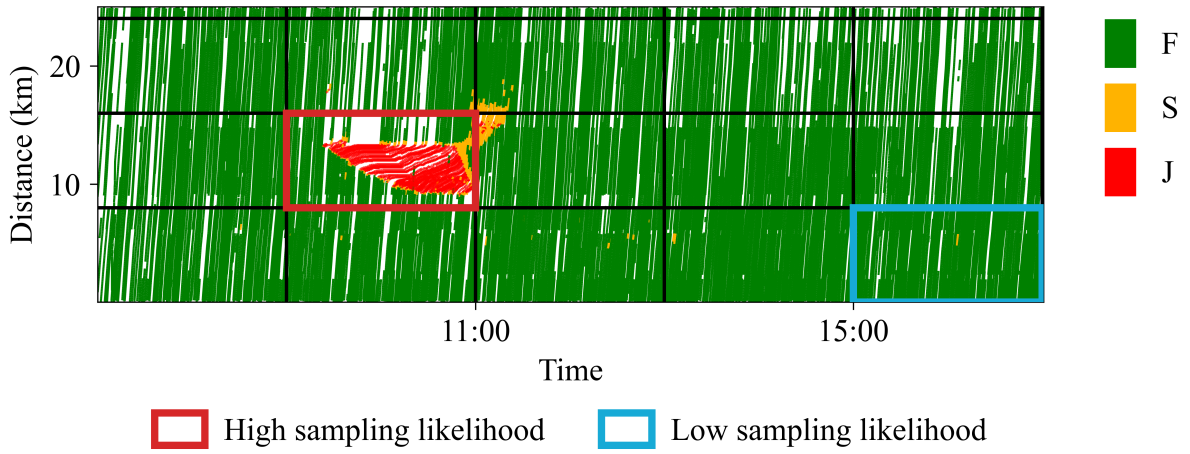


Figure 3.8: Velocity-weighted sampling on a speed-distance diagram, with $N_d = 3$ and $N_t = 6$. Two candidate cells are highlighted, with sampling probabilities inversely proportional to their mean speeds. The red cell shows a congested region with higher sampling probability, while the blue cell shows a free-flow region with lower sampling probability.

downstream boundary d_{down} is randomized such that the crash location d_{crash} is uniformly distributed across the full spatial extent of the window. This prevents the model from exploiting a fixed spatial prior rather than learning to identify crash-induced patterns across the full segment. Figure 3.9 illustrates the procedure.

Sequential congestion sampling. This mode operates on all manually labelled congestion regions, including both recurrent and crash-induced congestion. Each region is defined by a spatial anchor d_{anchor} and temporal boundaries $[t_{\text{start}}, t_{\text{end}}]$. Sampling proceeds sequentially along the time axis:

$$t_k = t_{\text{start}} + k \cdot \Delta t_{\text{step}}, \quad k = 0, 1, \ldots, \left\lfloor (t_{\text{end}} - t_{\text{start}})/\Delta t_{\text{step}} \right\rfloor. \tag{3.15}$$

During training, random spatial and temporal perturbations increase data variability:

$$d_{\text{sample}} = d_{\text{anchor}} + \xi_d, \qquad \xi_d \sim \mathcal{U}(-\Delta d_{\text{pert}}, \Delta d_{\text{pert}}), \tag{3.16}$$

$$t_{\text{sample}} = t_k + \xi_t, \qquad \xi_t \sim \mathcal{U}(-\Delta t_{\text{pert}}, \Delta t_{\text{pert}}). \tag{3.17}$$

For the test set, perturbations are disabled and complete congestion regions are evaluated as contiguous sequences to ensure reproducibility and direct comparability with the rule-based approach. Figure 3.10 illustrates the procedure.

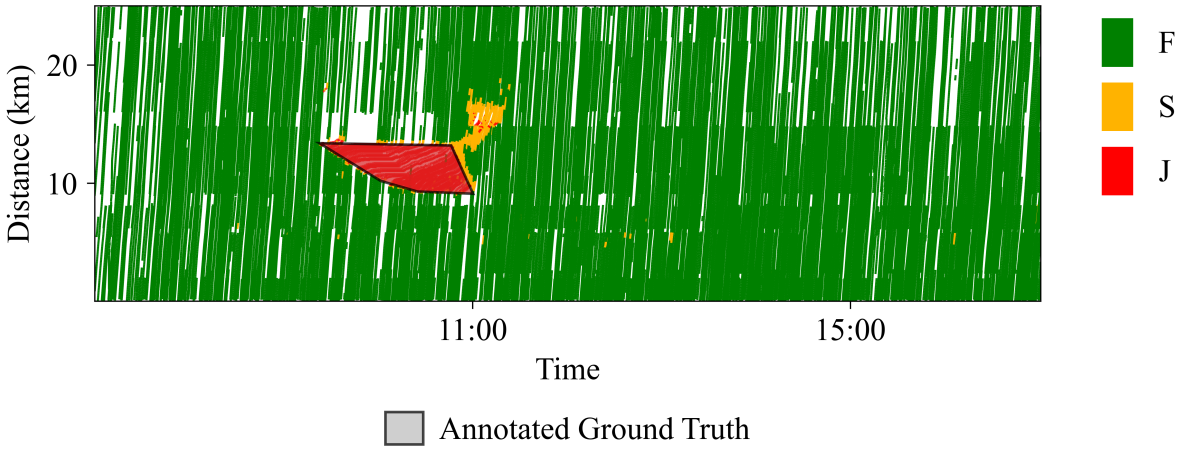


Figure 3.9: Crash-centred sampling around a manually annotated crash event. Windows are generated within a temporal buffer around the event, covering pre-crash, crash, and recovery conditions.

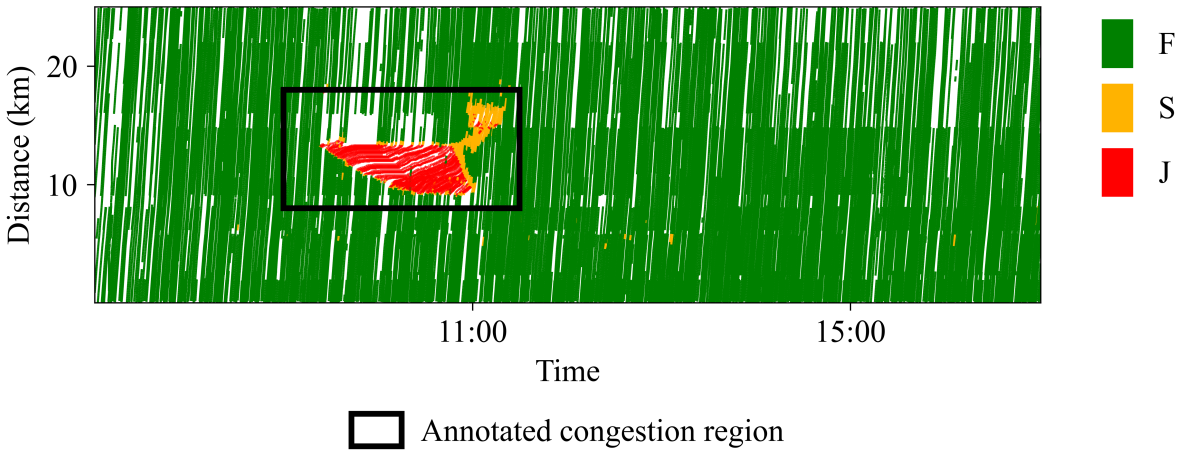


Figure 3.10: Sequential congestion sampling. Windows are placed at regular intervals along the time axis within a labelled congestion region; spatial and temporal perturbations are applied during training.

Table 3.8: Contribution of each sampling mode to the training set.

Sampling mode	Windows	Share [%]
Velocity-weighted	29,832	26.2
Crash-centred	11,266	9.9
Sequential congestion	72,736	63.9
Total	113,834	100.0

Table 3.9: Classification dataset statistics. The test set is shared across both pipelines and evaluated at the region level.

(a) Window-level statistics (train / validation).

Split	Total	Crash	Non-crash
Train	113,834	18,775 (16.5 %)	95,059 (83.5 %)
Val	28,392	4,637 (16.3 %)	23,755 (83.7 %)

(b) Region-level statistics (test).

Split	Total	Crash	Non-crash
Test	205	66 (31.2 %)	141 (68.8 %)

A visual representation of the sampled windows for an exemplary road section time-space diagram, demonstrating the oversampling of crash events, is provided in Figure A.1.

3.7.4 Dataset Statistics

Table 3.8 summarizes the contribution of each sampling mode to the training set.

Table 3.9 summarizes the full dataset partitioning used for the crash detection task. Training and validation sets are evaluated at the window level, while the test set is evaluated at the level of complete congestion regions. This setup ensures consistency with realistic deployment conditions and allows a direct comparison with the rule-based baseline.

The regression task operates on the subset of crash-positive windows identified in Table 3.9. This restriction is necessary because the regression model predicts the spatial crash position within a congestion region and therefore requires samples in which a crash is present.

3.7.5 Training Set Augmentation

Three augmentation strategies are applied independently within each training window $\mathcal{W}_{D,T}$, modifying only input features and never altering window boundaries or ground-truth labels. Let $n_\psi = |\{\psi \in \mathcal{W}_{D,T}\}|$ denote the number of unique trips in a window, with empirical mean $\mu_{n_\psi} \approx 121$ and standard deviation $\sigma_{n_\psi} \approx 55$. *(a) Cross-trajectory subsampling* adaptively reduces the retained trajectories γ_ψ in dense windows, where large n_ψ produces oversized graphs and tensors that raise computational cost and may bias the model toward extreme congestion. A randomised retention ratio is drawn from a uniform distribution whose bounds tighten as n_ψ grows: windows with $n_\psi \leq \mu_{n_\psi}$ are left unchanged, those with $\mu_{n_\psi} < n_\psi \leq \mu_{n_\psi} + \sigma_{n_\psi}$ retain roughly 70–100 %, and the densest windows ($n_\psi > \mu_{n_\psi} + 2\,\sigma_{n_\psi}$) retain only 40–50 %. In practice, about half of all windows remain unmodified. *(b) Within-trajectory subsampling* removes individual observations along each trajectory γ_ψ to simulate reduced penetration rates α and sensor dropouts. A density-dependent retention probability

$$r_{\text{save}}(n_\psi) \in [0.6,\ 1.0] \tag{3.18}$$

decreases from 1.0 for sparse windows to 0.6 for the densest. In the GNN pipeline, structurally relevant nodes and their immediate neighbors are protected, while remaining nodes are sampled independently at rate r_{save}; the CNN pipeline retains all nodes uniformly at the same probability. *(c) Feature perturbation* adds small stochastic noise to continuous features to prevent deterministic overfitting. Speed v is perturbed by additive Gaussian noise scaled to 20 % of the original value and clamped to a plausible range, and phase labels Φ are replaced with probability 0.1 by a uniformly sampled alternative. In the GNN, where time t and cumulative distance $D_{\psi,k}$ are explicit node features, both receive zero-mean Gaussian noise before normalization ($\sigma_t = 0.5\,\text{s}$, $\sigma_d = 0.01\,\text{km}$); in the CNN these coordinates are implicitly encoded by the grid and left unchanged.

3.7.6 Validation and Test Configuration

The validation set applies only cross-trajectory subsampling to maintain comparability across traffic densities; within-trajectory subsampling and feature perturbation are disabled. The test set follows the same principle but is evaluated over complete congestion regions rather than individual windows: for each region, a streaming evaluation constructs the latest window at every time step and generates a prediction until the traffic state returns to free flow.

4 Influence of Rainfall on Freeway Traffic Breakdown

Non-recurring congestion arises from temporary disturbances in freeway traffic [196, 197]. Precipitation is a common trigger of such breakdowns [99, 198]. In operational conditions, precipitation rarely occurs as spatially uniform rainfall, but instead appears in LRCs with limited spatial extent and duration that propagate across the freeway network. These LRCs can act as MBs that increase the probability of traffic breakdown. Two mechanisms are considered. First, rainfall reduces the effective freeway capacity by increasing safe following distances and reaction times. Second, the propagation direction of LRCs relative to traffic flow determines how rainfall-induced disturbances interact with the upstream propagation of congestion.

4.1 Detection and Characterization of LRCs

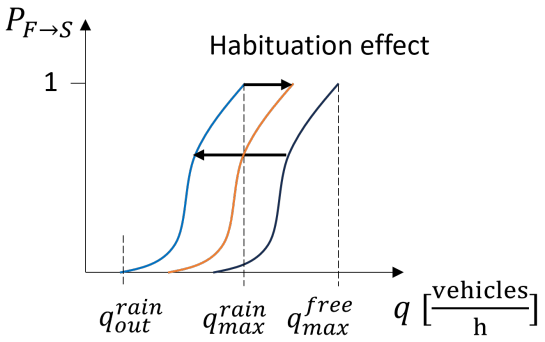
LRCs are identified from radar precipitation data using DBSCAN [180]. The clustering procedure groups spatiotemporal precipitation observations into contiguous rainfall structures. For each detected cluster, the centroid trajectory provides an estimate of the LRC propagation velocity. Congestion patterns are identified using the same clustering procedure applied to the spatiotemporal traffic state. Clusters are classified into LPs, MPs, and GPs.

4.2 Rain-Induced Capacity Reduction and Habituation

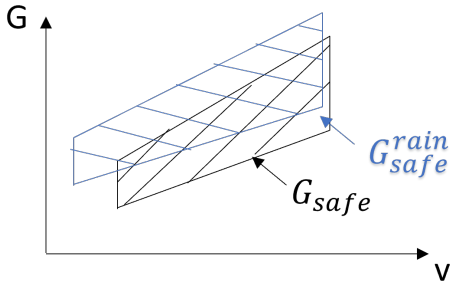
Rainfall affects driving behavior by increasing the accepted safe following distance

$$G_{\text{safe}}{}^{\text{rain}} > G_{\text{safe}}, \tag{4.1}$$

Parts of this chapter are based on previous work by the author [5, 11].



(a) Breakdown probability shift due to rainfall and partial recovery through the habituation effect during prolonged precipitation.



(b) Following distance G as a function of speed v, showing the increased safe distance under rainy conditions.

Figure 4.1: (a) Breakdown probability shift and habituation effect, (b) Increased safe following distance in rainy conditions [5].

which leads to longer reaction delays and reduced flow:

$$q_{\text{out}}{}^{\text{rain}} < q_{\text{out}}{}^{\text{free}}. \tag{4.2}$$

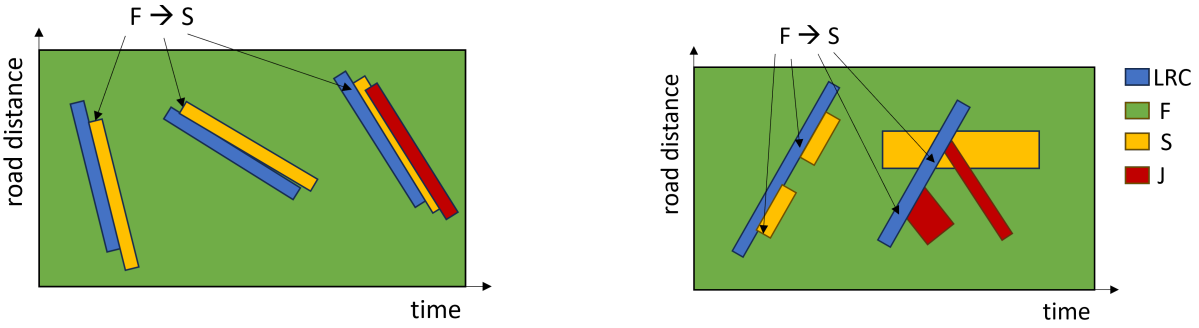
Consequently, the capacity interval $[q_{\text{out}}, q_{\text{max}}]$ shifts toward lower flow rates during rainfall. This shift increases breakdown probability (see Figure 4.1).

During prolonged rainfall, drivers gradually adapt their behavior by consistently reducing speed and increasing headway. This behavioral adaptation reduces the disturbance introduced by the precipitation front and partially restores traffic stability. The effect is referred to as the *habituation effect*. Rainfall introduces the strongest disturbance during the transition from dry to wet conditions, whereas sustained precipitation leads to more stable driving behavior (see Figure 4.1a).

4.3 Direction-Dependency of LRCs

Congestion patterns propagate upstream against traffic flow. The interaction between rainfall disturbances and congestion dynamics depends on the propagation direction of the LRC relative to traffic.

If a LRC moves downstream with traffic flow, the precipitation front travels perpendicularly to the propagation direction of WMJs. Disturbances generated at the rain front can detach from the MB and propagate upstream. This mechanism allows congestion to expand spatially and develop into extended patterns such as MPs or GPs. If the LRC propagates upstream against the traffic flow, the precipitation front moves in the same



(a) Rain front moving upstream

(b) Rain front moving downstream

Figure 4.2: Conceptual illustration of congestion dynamics for upstream- and downstream-moving LRCs.

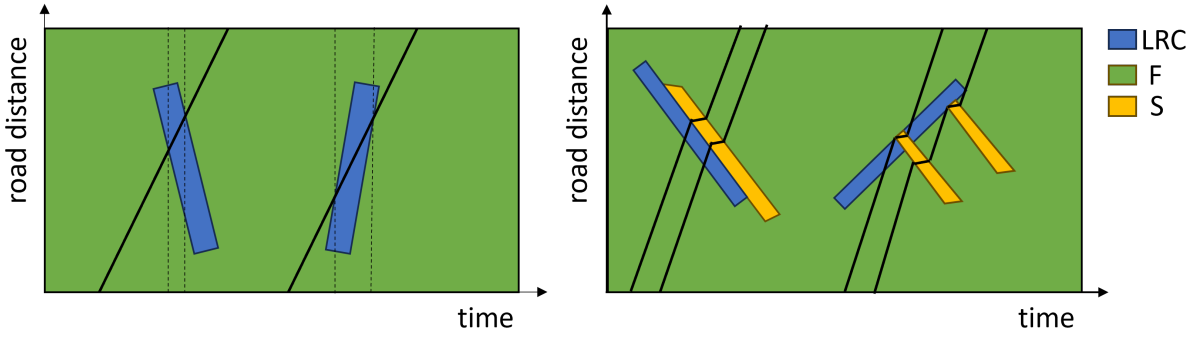


Figure 4.3: Conceptual explanation of direction-dependent congestion patterns induced by LRCs. Left: downstream-moving LRC with prolonged exposure. Right: upstream-moving LRC with limited congestion growth.

direction as congestion propagation. In this case, disturbances remain coupled to the MB and congestion remains spatially constrained to the precipitation region.

Two mechanisms contribute to this direction-dependent behavior (see Figure 4.3).

- Vehicles remain longer inside downstream-moving LRCs because the relative velocity between vehicles and the precipitation front is reduced. Longer exposure increases the probability of traffic breakdown.
- The spatiotemporal alignment between the upstream-moving rain front and the upstream propagation of congestion prevents congestion from detaching from the MB. Consequently, congestion remains confined to the LRC.

4.4 Empirical Evidence from Bidirectional Freeway Sections

The direction-dependent mechanism introduced in Section 4.3 can be directly observed on bidirectional freeway sections, where identical LRCs interact simultaneously with traffic moving in opposite directions. Because both directions experience the same precipitation field, bidirectional sections allow the effect of LRC propagation direction on the congestion response to be observed directly. For ease of comparison, the subsequent examples show downstream-moving LRCs on the left panel and upstream-moving LRCs on the right panel, unless noted otherwise.

Figure 4.4 shows both travel directions of the German freeway A7 on June 2, 2024. The same LRCs propagate upstream relative to traffic flow in one direction and downstream in the other. When precipitation propagates upstream, congestion remains confined to the spatial extent of the LRC and dissolves once the precipitation disappears. When the same LRC propagates downstream, congestion detaches from the precipitation front and develops into extended structures that persist after the rainfall has passed. The bottom picture shows a cutout at 60 km-90 km where the direction-dependent effect is clearly visible. An upstream moving LRC (left panel) constraints the congestion, whereas downstream-moving LRC triggers congestion that propagates away from the LRC (right panel). This example confirms that identical LRCs produce fundamentally different traffic responses depending solely on their propagation direction relative to traffic.

Figure 4.5 presents additional bidirectional examples that support the general pattern while illustrating the variability of the resulting congestion structures. Downstream LRC propagation more frequently triggers congestion that detaches from the LRC front and extends spatially, whereas upstream propagation tends to produce spatially constrained congestion whose propagation speed remains synchronized with the LRC front. Both examples also contain crashes seeded within the precipitation zone: one under upstream LRC propagation (Figure 4.5a) and two under downstream propagation (Figure 4.5b).

Interplay of intensity and direction. Figure 4.6 presents a case where low-intensity and high-intensity LRCs interact within the same bidirectional section (A7 between Füssen and Ulm, August 18, 2024). In the Ulm-bound direction, low-intensity precipitation propagates upstream relative to traffic while localized high-intensity cells ($I > 10\,\mathrm{mm/h}$) propagate downstream. The high-intensity downstream cells seed congestion that detaches from the LRC front, while the upstream low-intensity precipitation does not trigger breakdown. In the Füssen-bound direction the LRC propagation directions are reversed relative to traffic.

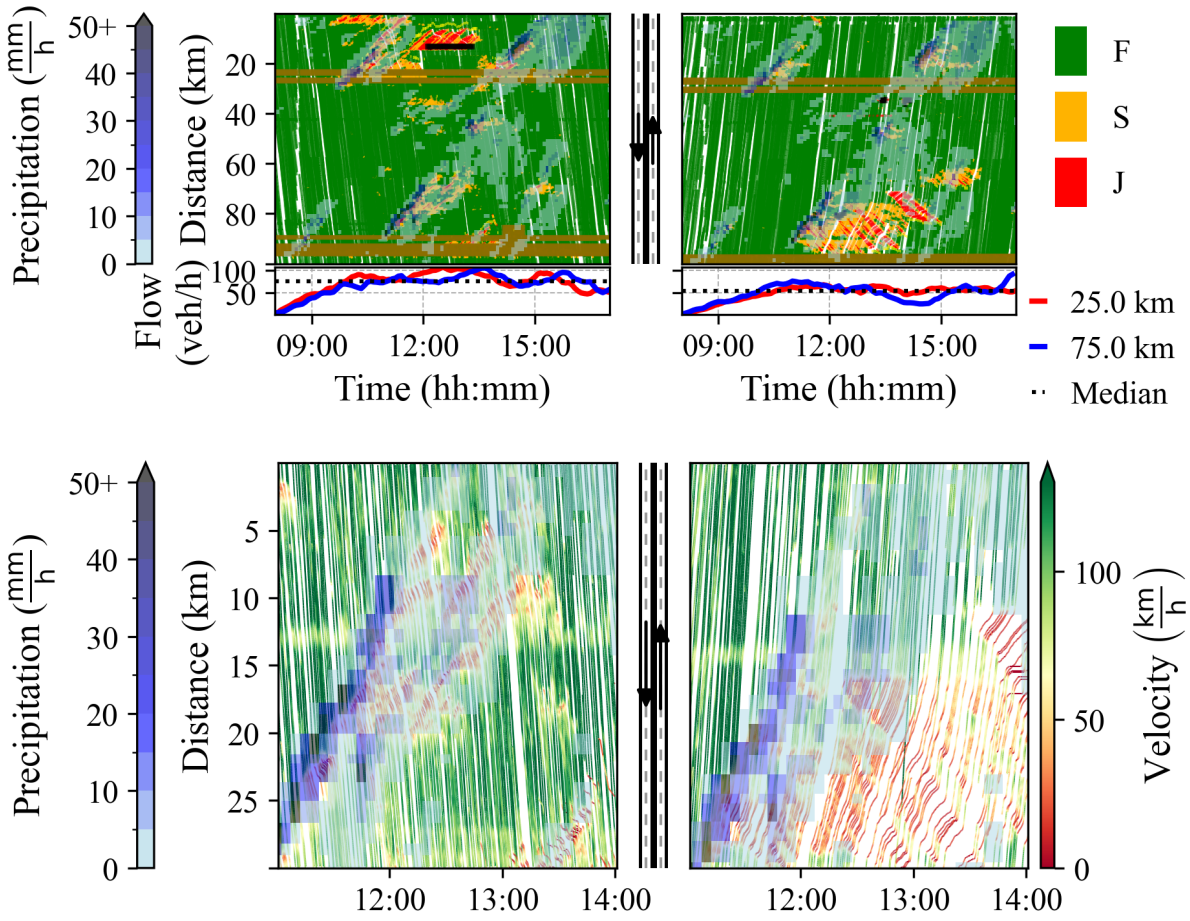
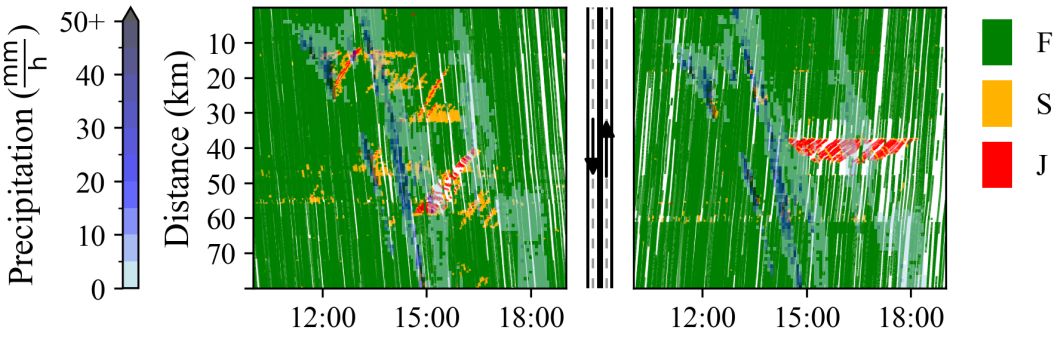
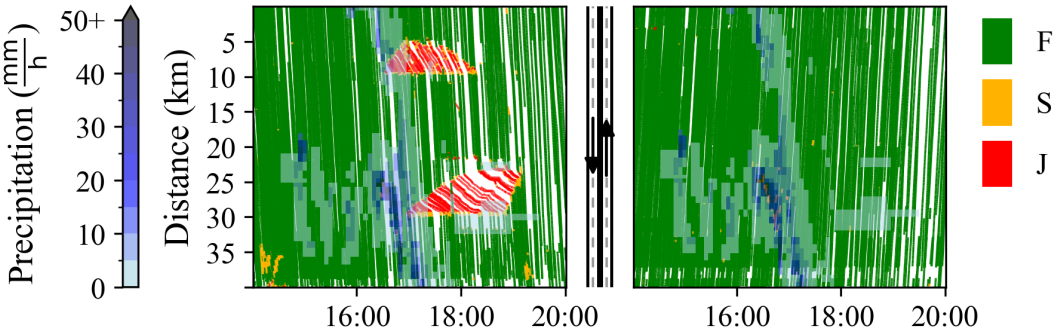


Figure 4.4: Both travel directions of German freeway A7 on June 2, 2024. Identical LRCs act as upstream-MBs in one direction and downstream-MBs in the other. (Top) shows the full distance. (Bottom) shows zoom-in at 80 km clearly showing direction-dependent impact.



(a) A6 between Heilbronn and Nuremberg on July 12, 2024.



(b) A6 between Mannheim and Heilbronn on June 21, 2024.

Figure 4.5: The same LRC propagates downstream relative to traffic on the right and upstream on the left, showing the direction-dependent congestion response.

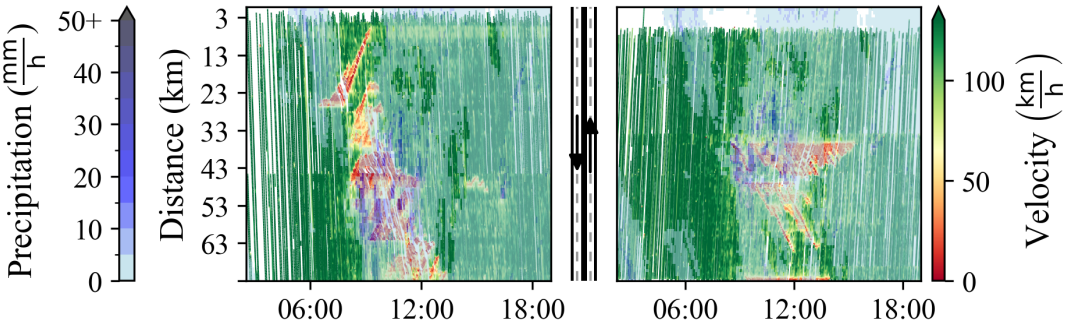


Figure 4.6: A7 between Füssen and Ulm on August 18, 2024. Opposite driving directions. Low-intensity and high-intensity LRC cells interact differently depending on their propagation direction relative to traffic.

Here, the now downstream-propagating low-intensity LRC triggers congestion that detaches from the precipitation front—despite its low intensity—while the now upstream-propagating high-intensity LRC produces large-scale congestion that remains spatially bound to the LRC front. These observations indicate that the congestion response is governed by the interplay of intensity and propagation direction: direction predominantly determines whether congestion remains coupled to the LRC front or detaches, while intensity modulates the likelihood and severity of breakdown.

Sub-breakdown effects and intensity thresholds. Not all LRCs trigger congestion. Figure 4.7 shows a low-intensity LRC on the A81 between Bad Dürrheim and Stuttgart on September 8, 2024. No traffic breakdown occurs, but a clear velocity reduction is visible as drivers adapt their speed in response to the precipitation, indicating that even below the breakdown threshold, LRCs measurably influence driving behavior. This is consistent with the literature on adverse-weather speed reductions [e.g., 101, 199]. Note that the permanent speed reduction between km 32 and km 40 reflects an enforced speed limit due to roadworks ($80\,\mathrm{km\,h^{-1}}$) and is unrelated to precipitation.

A related pattern is shown in Figure 4.8, where a high-intensity LRC ($I > 10\,\mathrm{mm/h}$) is embedded within a broader low-intensity LRC ($2 < I < 10\,\mathrm{mm/h}$) on the A5 between Karlsruhe and Heidelberg. The surrounding low-intensity precipitation causes only moderate speed reductions, while the nested high-intensity cell triggers a localized traffic breakdown that exhibits the direction-dependent pattern observed in the preceding examples. Together with the sub-breakdown case in Figure 4.7, this supports the existence of an intensity threshold below which precipitation modulates traffic speed but does not induce congestion formation.

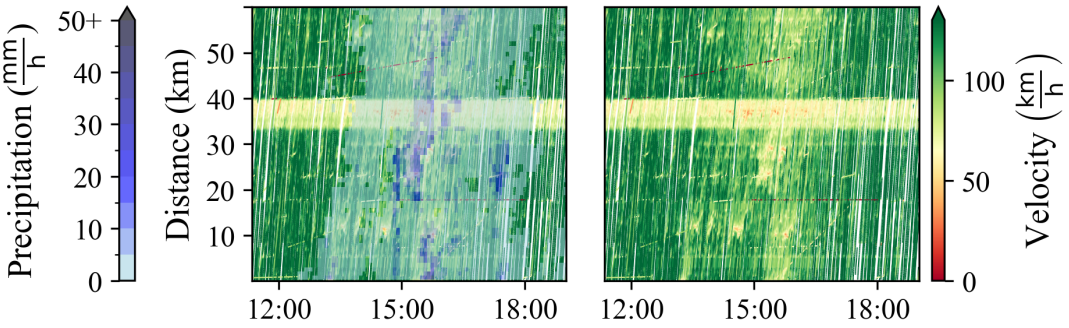


Figure 4.7: A81 between Bad Dürrheim and Stuttgart on September 8, 2024. Same driving direction shown with precipitation overlay (left) and without (right). A low-intensity LRC reduces vehicle speed without triggering traffic breakdown.

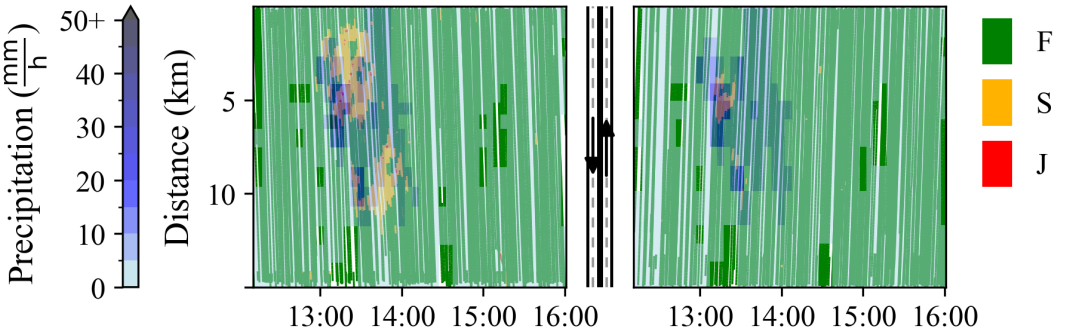


Figure 4.8: A5 between Karlsruhe and Heidelberg on August 18, 2024. Opposite driving directions. A high-intensity LRC cell embedded in a broader low-intensity field triggers localized breakdown, while the surrounding low-intensity precipitation only reduces speed.

4.5 Statistical Analysis

The preceding sections introduced two mechanisms through which LRCs affect freeway traffic: a capacity reduction that increases breakdown probability (Section 4.2), and a direction-dependent coupling between the rain front and congestion propagation (Section 4.3). This section tests both mechanisms against the full dataset. All key effect sizes are collected in Table 4.3 at the end of this section.

4.5.1 Sample Construction and Breakdown Probability

The analysis draws on the section-day observations introduced in Section 3.4.1. Wet section-days are roughly twice as likely to contain congestion as dry ones (80.9 % vs. 66.7 %; $\chi^2 = 21.8$, $p < 0.001$, odds ratio (OR) = 2.12). Among wet observations, a logistic regression [200] identifies rainfall intensity as the dominant predictor of rain-related congestion: each unit increase in log(Intensity) more than doubles the odds ($OR = 2.43$, 95 % confidence interval (CI) [1.61, 3.67], $p < 0.001$; full model in Table B.1), while precipitation area does not reach significance after controlling for intensity. A dose–response analysis confirms a monotonic relationship: the congestion rate rises from 40.6 % in the lowest intensity quartile to 71.7 % in the third ($OR = 1.54$ per quartile step, $p < 0.001$). Five-fold cross-validation yields an area under the curve (AUC) of 0.64 ± 0.02; the moderate discriminative power is expected, given that breakdown occurrence also depends on demand level and geometric bottlenecks not included as predictors.

The directional analysis is restricted to congestion clusters whose breakdown onset lies inside a LRC and for which the LRC propagation velocity can be estimated. Two velocity sources are available: a direct estimate from the uniquely matched precipitation cluster ($n = 125$), and the velocity of the nearest LRC ($n = 240$; correlation $r = 0.91$). The larger nearest-LRC sample is used throughout unless noted otherwise. Clusters are classified as upstream- or downstream-moving relative to traffic flow using a velocity threshold of $\pm 5\,\mathrm{km\,h^{-1}}$, yielding 190 clusters (90 upstream, 100 downstream). Table 4.1 summarizes the resulting constraint categories.

4.5.2 Direction-Dependent Coupling and Velocity Effects

Once a breakdown has occurred inside a LRC, the subsequent evolution of congestion depends on whether—and how fast—the LRC moves relative to traffic. As shown in Table 4.1, 76.7 % of congestion clusters under upstream-moving precipitation remain spatially constrained to the rain front, compared with 66.0 % under downstream-moving

Table 4.1: Rain–congestion constraint categories by LRC direction (onset in rain, $|v_{\mathrm{LRC}}| > 5\,\mathrm{km\,h^{-1}}$).

	Upstream ($n = 90$)		Downstream ($n = 100$)	
Constraint category	n	%	n	%
Constrained to rain front	69	76.7	66	66.0
Partially rain-linked	14	15.6	16	16.0
Onset in rain, then free	4	4.4	10	10.0
Triggered by rain, persists	3	3.3	8	8.0

Table 4.2: Constraint rate by upstream LRC speed ($v_{\mathrm{LRC}} < -1\,\mathrm{km\,h^{-1}}$, $n = 110$).

Speed category	n	Constrained (%)	Pct. in precip. (%)	Persists
Slow (1–10 $\mathrm{km\,h^{-1}}$)	44	79.5	88.3	0
Moderate (10–20 $\mathrm{km\,h^{-1}}$)	37	89.2	91.2	0
Fast (> 20 $\mathrm{km\,h^{-1}}$)	29	62.1	74.7	3

precipitation. The unadjusted difference is suggestive but not significant (Fisher exact test: $OR = 1.69$, $p = 0.112$). However, downstream-moving LRCs tend to produce more severe congestion events that are inherently harder to constrain. A logistic regression that adjusts for congestion extent, duration, intensity, and LRC extent reveals the underlying relationship: upstream motion more than doubles the odds of rain-front constraint ($OR = 2.45$, 95 % CI [1.15, 5.21], $p = 0.020$; full model in Table B.2). The effect direction is consistent across all tested velocity thresholds from $\pm 1\,\mathrm{km\,h^{-1}}$ to $\pm 10\,\mathrm{km\,h^{-1}}$ ($OR = 1.69$–2.06; Table B.3).

Upstream velocity magnitude modulates the coupling strength. The constraint rate peaks at 89.2 % for moderate upstream velocities ($|v| \in [10, 20]\,\mathrm{kmh^{-1}}$), a range that coincides with the typical propagation speed of congestion in the dataset ($\overline{v_g} \approx -4.9\,\mathrm{km\,h^{-1}}$; ${v_g}^{\mathrm{WMJ}} \approx -15\,\mathrm{km\,h^{-1}}$). When the rain front moves at approximately the same speed as the congestion, the LRC continuously reinforces the MB at the congestion front, producing a *speed-lock* that prevents decoupling. At higher upstream velocities ($|v| > 20\,\mathrm{km\,h^{-1}}$) the LRC outruns the congestion, the constraint rate drops to 62.1 % ($OR = 5.04$ vs. moderate, $p = 0.016$; Table 4.2), and congestion persistence beyond the LRC occurs (3 of 29 fast cases; zero among moderate or slow). Among the 174 clusters that do remain constrained, stronger upstream motion is associated with longer congestion duration ($r = -0.23$, $p = 0.002$) and larger congestion area ($r = -0.15$, $p = 0.044$), indicating that the speed-lock allows the constrained pattern to grow within the rain zone.

Downstream velocity affects persistence but not constraint. For downstream-moving LRCs, the constraint rate is generally lower (61–69 %) and does not vary meaningfully with speed (Spearman $r = -0.08$, $p = 0.39$). However, faster downstream LRCs significantly increase the probability that congestion persists after the rain has passed ($r = 0.22$, $p = 0.019$): five of 39 fast cases ($|v| > 20\,\mathrm{km\,h^{-1}}$) exhibit persistence, compared with one of 49 slow cases. This combination of low constraint and high persistence makes fast downstream-moving LRCs the most problematic configuration for traffic management.

4.5.3 Breakdown Location and the Habituation Effect

The habituation mechanism proposed in Section 4.2 predicts that breakdown probability is highest at the dry-to-wet transition front, where drivers first encounter rainfall, and relaxes during prolonged precipitation. Under the null hypothesis of uniform breakdown positions, the normalized distance to the LRC front d_{front} has expectation 0.5.

Among 93 high-intensity clusters ($I > 10\,\mathrm{mm/h}$) with known precipitation direction and onset inside the LRC, 61.3 % of breakdowns occur in the front half of the rain cell (binomial test: $p = 0.019$; Cohen's $d = 0.21$; Table B.4). The effect is driven by downstream-moving LRCs (Kolmogorov-Smirnov (KS): $p = 0.028$, $n = 43$), for which longer exposure inside the rain cell sharpens the contrast between dry approach and wet interior; the upstream-moving subset is non-significant ($p = 0.41$, $n = 50$), consistent with the shorter exposure that limits the habituation gradient.

Lowering the intensity threshold to $I > 2\,\mathrm{mm/h}$ enlarges the sample to $n = 1{,}347$ clusters (673 upstream, 674 downstream) and reveals a U-shaped distribution (Figure 4.9): breakdowns concentrate at both the dry-to-wet and wet-to-dry fronts, while the rain-cell interior shows significantly fewer onsets than expected ($\chi^2 = 9.6$, $p = 0.008$ for a front/center/back split). This generalizes the habituation hypothesis: any transition between dry and wet conditions raises breakdown probability, while sustained exposure allows drivers to adapt. As in the primary sample, the effect is significant for downstream-moving LRCs ($\chi^2_5 = 14.8$, $p = 0.005$) but not for upstream-moving cases ($p = 0.27$).

4.5.4 Crash Occurrence Under Rainfall

Beyond its effect on congestion, rainfall also elevates crash risk on the freeway sections studied. Of the 348 crashes that can be uniquely matched to a precipitation observation window, 74 occur during rainfall and 274 under dry conditions. Dry conditions account for 86.1 % of total space–time exposure but only 78.7 % of crashes; the resulting exposure-adjusted rate ratio of 1.52 is consistent with the 30–40 % increases reported in the literature [41,

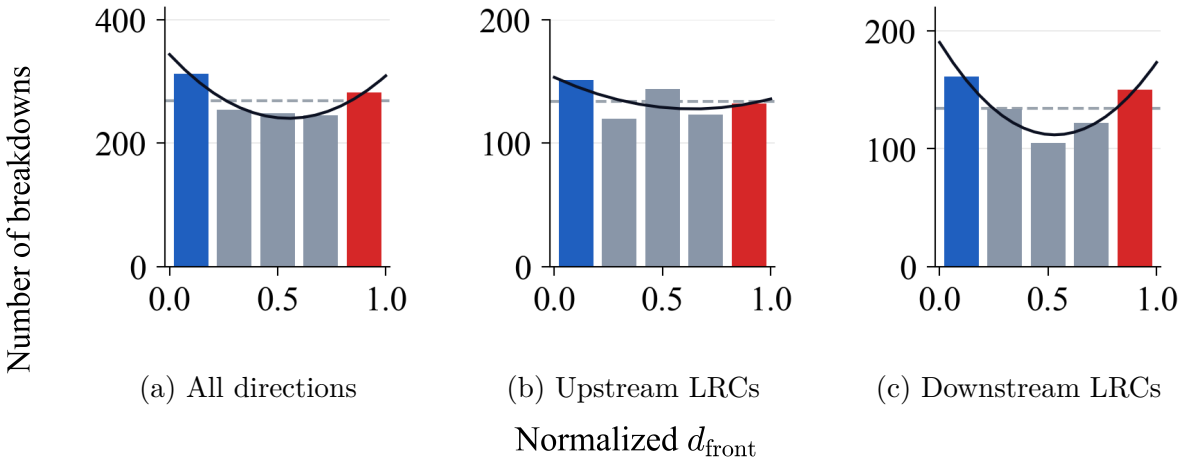


Figure 4.9: Breakdown onsets over normalized distance to the dry-to-wet front. The dashed line marks uniform expectation, the solid curve a quadratic fit. Breakdowns concentrate at both transition fronts; the effect is most pronounced for downstream-moving LRCs.

201, 202].

4.6 Discussion

The monotonic increase in breakdown probability across intensity quartiles is consistent with the capacity shift (Section 4.2). Intensity dominates over precipitation area, indicating that the local severity of the disturbance matters more than its spatial footprint. This is consistent with the observation that LRCs nested inside broader low-intensity rain are likelier to trigger breakdown (Figure 4.8).

The adjusted logistic regression confirms the direction-dependent coupling, which is masked in the unadjusted comparison because downstream-moving LRCs produce inherently more severe events that are harder to constrain. The speed-lock refines this finding: when the upstream LRC velocity matches the propagation speed of congestion, the rain front continuously reinforces the MB and yields the highest constraint rates; when it outruns congestion, the coupling breaks. Faster downstream LRCs are the most problematic configuration identified in this chapter, increasing post-rain persistence without affecting the already low constraint rate.

Breakdowns concentrate at the transition fronts of LRCs rather than in their interior. The U-shaped distribution in the larger sample generalizes this to both dry-to-wet and wet-to-dry boundaries. Whether this reflects gradual behavioral adaptation or the instantaneous

Table 4.3: Key statistical results. Effect sizes, test statistics, and p-values for all main findings.

Finding	**Method**	**Effect size**	***p***
Breakdown probability			
Rainfall increases congestion	Pearson χ^2	$OR = 2.12$	< 0.001
Intensity is dominant predictor	Logistic regr.	$OR_{\log(\text{Int})} = 2.43$; 95 % CI [1.61, 3.67]	< 0.001
Direction-dependent coupling			
Upstream rain increases constraint (adj.)	Logistic regr.	$OR = 2.45$; 95 % CI [1.15, 5.21]	0.020
Velocity–magnitude effects			
Speed-lock: moderate vs. fast (upstream)	Fisher exact	89.2 % vs. 62.1 %; $OR = 5.04$	0.016
Downstream persistence increases with speed	Spearman	$r = 0.22$	0.019
Constrained congestion grows under upstream rain	Spearman	$r_{\text{dur}} = -0.23$; $r_{\text{area}} = -0.15$	0.002; 0.044
Habituation effect			
Breakdowns at dry-to-wet front ($I > 10\,\text{mm/h}$)	Binomial	57/93 (61.3 %); Cohen's $d = 0.21$	0.019
U-shaped boundary clustering ($I > 2\,\text{mm/h}$)	χ^2	Three-zone: 496/403/448	0.008
Crash risk			
Crash rate elevated during rainfall	Rate ratio	1.52	–

impact of spray and reduced visibility cannot be resolved with FCD alone. For operational purposes the distinction is secondary: in either case, the LRC entry front constitutes the critical zone for breakdown risk, and its position can be tracked in real time from weather radar data.

The exposure-adjusted crash rate ratio of 1.52 aligns with the 30–40 % increases reported in the broader traffic safety literature [41, 201, 202]. This chapter treated precipitation as a trigger of non-recurring congestion, but crashes are another major source, and the elevated crash rate during rainfall suggests the two mechanisms are not independent.

Implications for traffic management. The congestion response to a LRC is largely characterized by two quantities observable in real time from weather radar: rainfall intensity and LRC propagation velocity relative to traffic. Under upstream-moving LRCs at moderate velocity, the speed-lock makes the MB trajectory predictable from the LRC motion alone, and variable speed limit systems or route guidance can target the dry-to-wet transition zone where breakdown risk is highest. Under downstream-moving LRCs, congestion more frequently persists after the rain has passed, so strategies should focus on dissolving residual congestion rather than tracking the rain front. The interplay of intensity and direction observed in Figure 4.6 further suggests that management responses should account for the internal structure of precipitation fields, not only their aggregate footprint.

Limitations. The dataset covers freeway sections in southwest Germany; transferability to other road networks and climatic regions remains to be verified. The reliance on connected vehicle data at approximately 5 % market penetration introduces potential sampling bias, particularly for short-lived breakdowns. Both LRCs and congestion patterns are identified using DBSCAN clustering (Section 4.1); alternative parameterizations could shift boundary cases between constraint categories. The analysis is restricted to rainfall; other adverse weather conditions such as snow or fog involve distinct physical mechanisms and are not addressed. Statistical power for some comparisons is limited by the number of events with estimable precipitation velocity.

5 Crash Detection: Congestion Patterns and Task Formulation

Crashes on freeways typically trigger congestion that manifests as mega jams. Mega jams exhibit characteristic patterns that differ systematically from recurrent congestion and can therefore, in principle, be identified from FCD alone. The following chapters analyze three conceptually distinct approaches to automated crash detection from FCD: a rule-based method derived from expert knowledge (Chapter 6), a CNN-based approach operating on structured spatiotemporal grid tensors (Section 7.2), and a GNN-based approach operating on spatiotemporal trajectory graphs (Section 7.3). All three approaches are evaluated on a shared test set with ground-truth labels established in Section 3.6.

5.1 Distinct Features of Recurrent and Crash-Induced Congestion

The concept of FCD-based crash detection relies on distinguishing congested phases and their characteristic features within the framework of 3PTT. A traffic breakdown induced congestion may persist at a bottleneck over an extended period; with growing congestion, narrow moving jams emerge and evolve into WMJs originating from the localized S phase.

Figure 5.1 illustrates two characteristic congestion scenarios. The left panel shows recurrent, non-crash-related congestion in which two WMJs emerge from S that is spatially fixed at a bottleneck. Such congestion typically occurs at road works, ramps, or other infrastructural bottlenecks. In contrast, a traffic breakdown induced by a crash often leads to the formation of a mega jam, as illustrated in the right panel.

Figure 5.2 illustrates the limitations of existing event-based reporting. The red area represents a mega jam whose upstream seed (top-left corner) marks the crash location. The crash notification is issued with a delay $\Delta t_1 > 0$ and resolved with an inaccuracy $\Delta t_2 \neq 0$; spatial inaccuracies Δd_1 and Δd_2 are observed between the reported and actual

Parts of this chapter and the subsequent model chapters are based on previous publications by the author [1–4, 9].

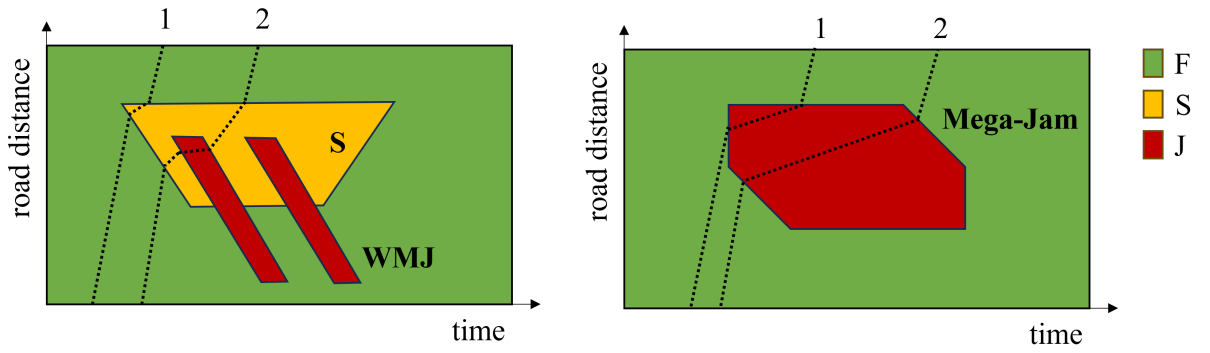


Figure 5.1: Congestion features in recurrent (left) and crash-induced (right) congestion. Dashed lines indicate individual vehicle trajectories [3].

crash location. The approaches developed in this work address these shortcomings by detecting crashes directly from traffic flow dynamics, enabling improved spatial localization and higher temporal accuracy.

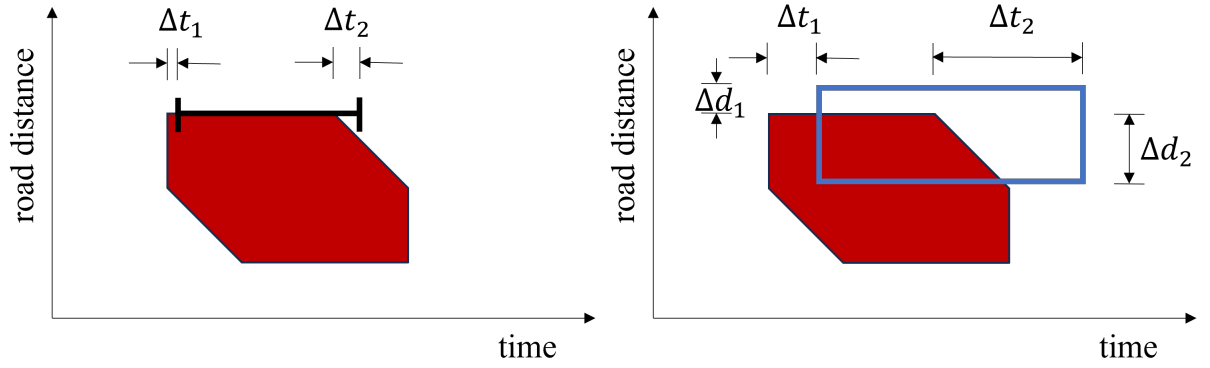


Figure 5.2: Spatial and temporal deviation of the proposed method (black line) compared to TPEG-based crash information (blue square). The red area indicates a mega jam evolving from a crash at the upstream seed of the congestion pattern [3].

5.1.1 Theoretical Basis for Crash Severity

The severity of crash-induced congestion can be related to the change in the safe time headway τ at the bottleneck created by the incident. [203] models this mechanism as follows. Within the bottleneck section, drivers increase their safety time gap to a value $\tau_{\text{safe}} > 1\,\text{s}$, compared with $\tau_{\text{safe}} \approx 1\,\text{s}$ outside the bottleneck. The magnitude of τ_{safe} determines the severity of the bottleneck and influences the resulting traffic flow.

A larger τ_{safe} reduces the maximum flow rate $q_{\text{max}}^{\text{free}}$, imposing significant limitations on

throughput. Given a traffic breakdown (F→S), a pinch region of characteristic width $D_{\text{pinch}} \approx 1\,\text{km}$ to $2\,\text{km}$ is forming. WMJs emerge in this region. As τ_{safe} increases, D_{pinch} becomes irregular, oscillations lose periodicity, and adjacent WMJs begin to merge at the bottleneck into a single mega jam. For very large τ_{safe} the pinch region vanishes entirely, leaving only a mega jam characterized by alternating flow interruptions and low-speed states. In extreme bottleneck scenarios where q approaches zero (e.g., full road closure), the mega jam transforms into a queue of stationary vehicles [203].

This theoretical progression motivates a classification of post-crash congestion based on the effective bottleneck severity: a minor incident may not increase τ_{safe} sufficiently to trigger a traffic breakdown, whereas a severe crash produces a bottleneck strong enough to form a mega jam directly at the crash site. A crash that results in lane blockage or full road closure drives the flow to near zero, creating a standing queue [4].

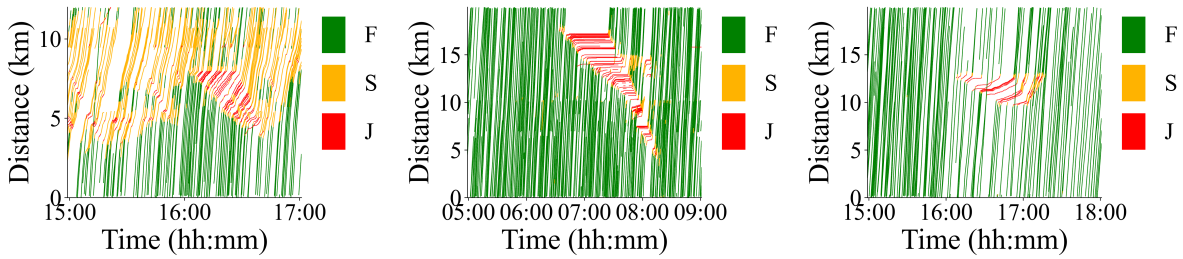
5.1.2 Classification of Post-Crash Congestion Patterns

Based on the theoretical framework above and an empirical analysis of 445 post-crash congestion events in the FCD dataset, three categories of crash-induced congestion are identified. The categories are ordered by the prevailing pre-crash traffic state and the severity of the induced bottleneck [4].

Category I: Crash within pre-existing congestion. Crashes in this category occur within an already congested traffic state, typically an extended S phase caused by road works or other persistent bottlenecks. The crash further reduces capacity and induces a pronounced J phase within the existing congestion. Figure 5.3a shows an example from the German A5 freeway, where a crash embedded in long-lasting road-work congestion leads to severe post-crash congestion despite low traffic demand.

Category II: Full blockage. Crashes causing a complete road closure result in a standing queue at the crash location. Vehicle delay increases approximately linearly with the duration of the blockage. An example from the German A3 freeway is shown in Figure 5.3b, where trajectories exhibit an abrupt deceleration to standstill and extended upstream queuing.

Category III: Crash in free flow. Crashes occurring under free-flow conditions introduce a sudden, temporary bottleneck. The congestion severity depends on the number of blocked lanes, crash duration, and traffic demand. Figure 5.3c shows an event on the German A81 freeway where a crash immediately generates a mega jam that persists until clearance and



(a) post-crash congestion within S phase (b) full blockage with standing queue (c) post-crash congestion in free flow

Figure 5.3: Categories of crash-induced congestion patterns.

subsequently dissolves with the characteristic downstream propagation velocity of WMJs ($\approx -15\,\mathrm{km\,h^{-1}}$; [78]).

A robust crash detection algorithm needs to be capable of detecting all three types of crash-induced congestion. Such congestion differs fundamentally from recurrent bottleneck congestion in that it can occur at any flow level, has an abrupt onset, and typically results in a mega jam or standing queue, whereas recurrent congestion typically develops gradually with increasing demand and is characterized by an extended S phase.

5.1.3 Task Definition

Crash detection addresses three objectives: classifying whether a crash has occurred, localising it in the time–distance plane, and detecting it as early as possible.

Consider a freeway segment of length D observed over a time period $\mathcal{T}$. The *classification* task determines whether a crash has occurred within the observed segment. The *localization* task estimates the crash position along the segment. The *timeliness* objective requires that the detection is issued with minimal delay after the crash onset.

Each instance is associated with two ground-truth labels:

$$y^{\mathrm{gt}}_{\mathrm{cls}} \in \{0,1\}, \qquad y^{\mathrm{gt}}_{\mathrm{loc}} \in (0,1), \tag{5.1}$$

where $y^{\mathrm{gt}}_{\mathrm{cls}} = 1$ indicates the presence of a crash. The localization label $y^{\mathrm{gt}}_{\mathrm{loc}}$ represents the normalized crash position along the segment and is defined only when $y^{\mathrm{gt}}_{\mathrm{cls}} = 1$.

Detection timeliness. Let t_{crash} denote the ground-truth crash onset time and t^{*}_{det} the earliest time at which the system issues a positive detection. The detection delay is

$$\Delta t_{\mathrm{det}} = t^{*}_{\mathrm{det}} - t_{\mathrm{crash}}\,. \tag{5.2}$$

This definition applies identically across all approaches, providing a consistent basis for comparison regardless of whether detection occurs through continuous streaming evaluation or classification over discrete observation windows.

5.2 Overview of Approaches

Three approaches to crash detection are developed in this work. They share the task definition of Section 5.1.3 and are evaluated on the same ground-truth set, which enables the direct comparison reported in Chapter 8. The approaches differ in how much of the detection logic is specified a priori from traffic flow theory and how much is inferred from labelled data. The rule-based detector encodes the distinguishing features of Section 5.1 as explicit decision rules and learns nothing beyond a small set of calibrated thresholds. The two learning-based models infer the detection mapping from data but differ in their inductive bias: the CNN operates on a regular grid representation with only weak structural priors, whereas the GNN embeds traffic flow theory directly in its graph construction and therefore occupies an intermediate position between the rule-based and the predominantly data-driven extreme.

Rule-based crash detection (Chapter 6). Detection rules are derived from traffic flow theory and domain expertise. The approach requires only parameter calibration on a small labelled subset and does not involve supervised learning.

CNN-based crash detection (Section 7.2). The CNN-based approach operates on structured spatiotemporal grid tensors derived from aggregated vehicle trajectories. Convolutional filters extract local spatiotemporal features that the classifier uses to distinguish crash-induced congestion from recurring congestion.

GNN-based crash detection (Section 7.3). The GNN-based approach constructs graphs directly from connected vehicle trajectories, representing individual positions as nodes and encoding interactions through edges based on spatial proximity, temporal overlap, and traffic-state similarity.

6 Rule-Based Crash Detection Based on Distinct Phase Regions

Crash-induced congestion produces a rapid $F \rightarrow S \rightarrow J$ phase transition, forming a stationary downstream front near the crash location and an upstream-propagating front that continuously expands the jam region. The proposed approach exploits these distinctive dynamics by continuously evaluating traffic phase regions and updating their spatiotemporal representation as new segments become available. Detection is formulated as an interpretable aggregation of features designed to capture three objectives: (i) early identification of abrupt breakdowns, (ii) confirmation of spatial fixation at the disturbance location, and (iii) robust handling of temporary road blockages.

6.1 Methodology

Building upon [3], the detection system evaluates each jam phase through two feature categories: *positive indicators* that quantify how closely the observed jam matches the expected crash signature, and *plausibility filters* that suppress artifacts, reduce false positives, and ensure physical consistency. The positive indicators contribute non-negative scores that raise the crash probability, while the plausibility filters impose penalty terms or hard gates that prevent spurious detections from reaching the alert threshold.

6.1.1 Positive Indicators

Positive indicators quantify features of the jam phase that are characteristic of crash-induced congestion. Each indicator produces a score in $[0, 1]$ reflecting how strongly the observed pattern matches the crash signature.

Pre-Jam Free-Flow Ratio. A crash disrupts otherwise free-flowing traffic, whereas recurring congestion builds gradually through an extended S phase. To distinguish these two scenarios, the traffic composition upstream of the jam seed is analyzed in a spatiotemporal

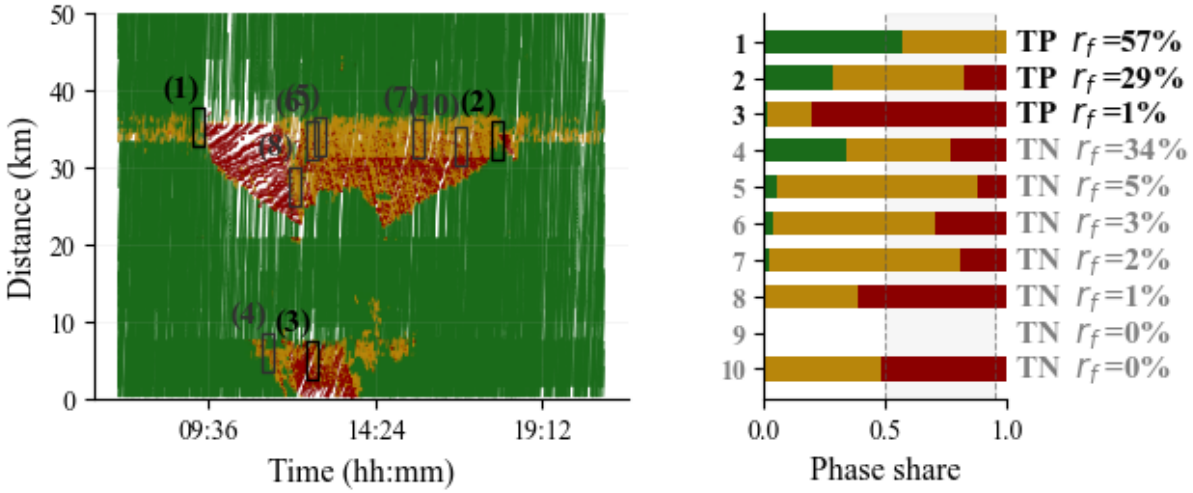


Figure 6.1: Pre-jam free-flow ratio for selected jam phases. Left: phase-colored trajectories with the upstream query windows shown as rectangles. Right: stacked bar chart of the F/S/J composition per window. Crash-induced jams (1-3) exhibit a higher free-flow share, while recurring congestion (4-10) shows a higher congested share.

window of width Δd and look-back duration Δt centered on the seed position and ending at the seed time. Within this window, all classified traffic segments are counted:

$$r_f = \frac{n_f}{n_f + n_s + n_j}, \tag{6.1}$$

where n_f, n_s, and n_j denote the number of free-flow, synchronized, and jam segments, respectively. A high r_f indicates that traffic was predominantly free-flowing before the jam emerged—consistent with a sudden onset. The score is obtained by linear normalization between configurable bounds $r_{f,\min}$ and $r_{f,\max}$. This indicator is computed once at seed time and remains constant throughout the phase lifetime, providing the earliest available signal. Figure 6.1 illustrates the pre-jam free-flow ratio for selected jam phases, showing that some crash-induced jams (1-3) exhibit a higher free-flow share, while recurring congestion (4-10) shows a higher congested share.

Downstream Front Velocity. The downstream front of a WMJ propagates upstream at the characteristic velocity $v_g \approx -15\,\mathrm{km\,h^{-1}}$, while a crash-induced congestion typically shows a spatially fixed downstream front ($v_g \approx 0\,\mathrm{km\,h^{-1}}$).

The score is computed through a confidence-ramped evaluation of consecutive near-zero readings. Let $\{(\hat{v}_{g,k})\}_{k=1}^{K}$ denote the sequence of downstream velocity estimates. A reading

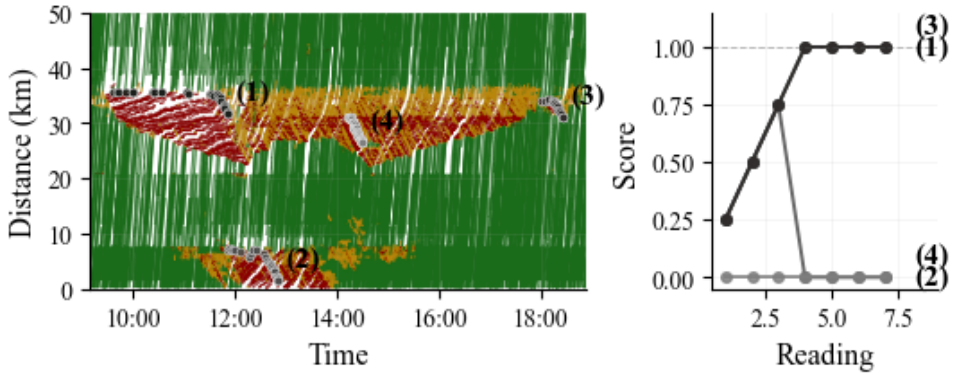


Figure 6.2: Downstream front velocity evolution for a crash-induced congestion (1-3) and a WMJ (4). The downstream front remains near zero until the crash is cleared, reaching full confidence after four consecutive in-band readings. The WMJ exhibits a negative propagation velocity.

is classified as *near-zero* if $|\hat{v}_{g,k}| \leq v_{\text{band}}$, where v_{band} is a configurable tolerance band. The score combines the ratio of near-zero readings with a confidence ramp:

$$f_S((v_g)) = \frac{n_{\text{near-zero}}}{K} \cdot \min\left(1,\ \frac{c_{\text{streak}}}{c_{\text{full}}}\right), \tag{6.2}$$

where c_{streak} counts the current consecutive near-zero streak and c_{full} is the number of consecutive readings required for full confidence. A single out-of-band reading resets the streak to zero. The score is frozen once confidence reaches 1.0 or the maximum number of evaluated readings is reached, so that a crash which clears before the score trigger retains the early spatially fixed downstream velocity rather than having it diluted by late-phase features.

Both v_{band} and c_{full} control the speed–selectivity trade-off: a tight band (e.g., $v_{\text{band}} = 0.2\,\text{km}\,\text{h}^{-1}$) with few required readings ($c_{\text{full}} = 2$) fires within 2–3 minutes of the seed but is susceptible to noise, while a wider band with more readings produces a more robust but slower signal. Both parameters are included in the calibration search space (Section 6.2). Figure 6.2 illustrates the downstream velocity evolution for selected jam phases, showing that the downstream front remains near zero until the crash is cleared, reaching full confidence after four consecutive in-band readings, while the WMJ exhibits a negative propagation velocity.

Full Blockage. In severe crashes, damaged vehicles may block parts of or the entire road, causing traffic speeds to drop to near zero for extended periods. Such a stationary queue can not form a downstream front, so the full blockage criterion serves as an alternative

indicator of a severe crash when the downstream velocity feature is unavailable. Since both address the same underlying phenomenon, only the highest-scoring member contributes to the crash probability, avoiding double-counting. The blockage criterion quantifies the fraction of observed sessions that exhibit sustained standstill. A session is considered blocked if it maintains a speed below v_{blk} for a consecutive duration exceeding T_{blk}, with at least n_{seg} observed segments. The blockage rate is defined as

$$\beta = \frac{N_{\text{blk}}}{N_{\text{total}}}\,. \tag{6.3}$$

The score is binary when a configurable session count is reached ($f_S = 1.0$) and otherwise uses the continuous rate β when it exceeds $\beta_{\min}$:

$$f_S(\beta) = \begin{cases} 1, & N_{\text{blk}} \geq N_{\text{FB}}, \\ \beta, & \beta \geq \beta_{\min}, \\ 0, & \text{otherwise.} \end{cases} \tag{6.4}$$

Figure 6.3 illustrates the blockage detection for a crash with full road closure, showing that trajectories entering the congested region decelerate to standstill and remain blocked beyond the duration threshold, causing the blockage rate β to rise as additional sessions confirm sustained standstill. Crashes without full blockage and recurring congestion show low or zero β and do not reach the session count threshold. These cases rely on the downstream velocity feature for detection, while full-blockage scenarios rely on the blockage feature, ensuring robust detection across a range of crash severities and congestion patterns.

Cluster Dominance. A crash-induced jam typically forms one dominant cluster that occupies a large share of the local spatiotemporal window, whereas recurring congestion fragments into several smaller clusters corresponding to individual WMJs. The cluster dominance ϕ_{cd} quantifies the fraction of the window occupied by the *current* cluster:

$$\phi_{\text{cd}} = \frac{|\{s \in \mathcal{W} : \text{cluster}(s) = c_k\}|}{|\mathcal{W}|}\,, \tag{6.5}$$

where c_k is the cluster identifier of the phase under evaluation and $\mathcal{W}$ is a spatiotemporal window of configurable spatial half-width Δd and temporal look-back Δt centered on the latest observation. For a crash with one coherent cluster, ϕ_{cd} is high (typically 0.6–0.8). For recurring congestion fragmenting into multiple WMJs (e.g., three clusters each covering

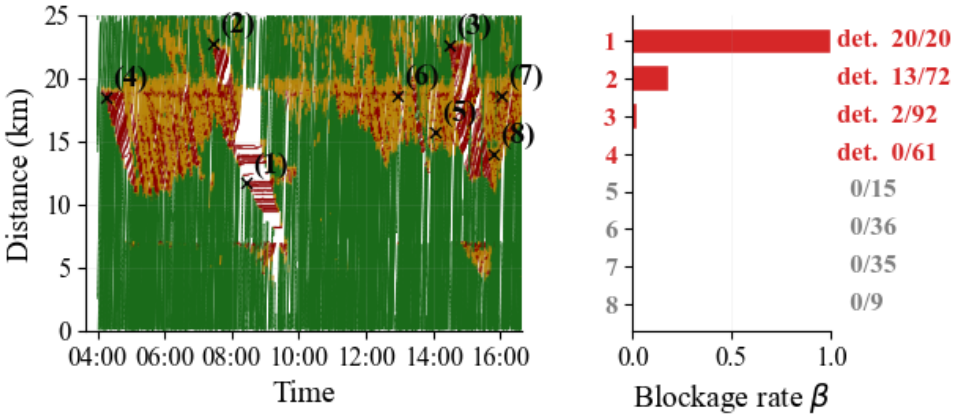


Figure 6.3: Blockage detection for a crash with full road closure (1). Trajectories entering the congested region decelerate to standstill (1) and remain blocked beyond the duration threshold. The blockage rate β rises as additional sessions confirm sustained standstill. Crashes without full blockage (2-4) and recurring congestion (5-8) show low or zero β and do not reach the session count threshold.

$\sim 15\,\%$ of the window), $\phi_{\mathrm{cd}} \approx 0.15$ per cluster, even when the total jam fraction is large. Cluster dominance is, therefore, a stronger discriminator than the raw jam-phase fraction. The score activates above a minimum threshold $\phi_{\mathrm{cd,min}}$ and is linearly normalized to $[0, 1]$. The look-back Δt controls the timeliness–robustness trade-off: a short window (e.g., 300 s) responds within minutes of the seed but is susceptible to transient dominance from short-lived clusters, while a long window (e.g., 900 s) yields a more stable but slower signal. Figure 6.4 shows the score for selected jam phases: crash-induced clusters (1–3) exhibit high dominance, while WMJs (4, 5) fragment into multiple clusters and score lower. Cluster 5 is an exception—spatially and temporally adjacent WMJs are merged by the clustering algorithm into a single high-dominance cluster, a common artefact that plausibility filters must address to prevent false positives (FPs).

Front Velocity Difference. The difference between the downstream and upstream front velocities captures the structural asymmetry of crash-induced jams:

$$\Delta v = v_g - v_u \,. \tag{6.6}$$

In a post-crash congestion, the downstream front is fixed ($v_g \approx 0$) while the upstream front propagates upstream ($v_u < 0$), yielding a negative Δv. For recurring congestion, both fronts move upstream ($\Delta v_{WMJ} \gg \Delta v_{crash}$). The score is linearly normalized with a configurable maximum. This indicator requires both a downstream and upstream velocity estimate,

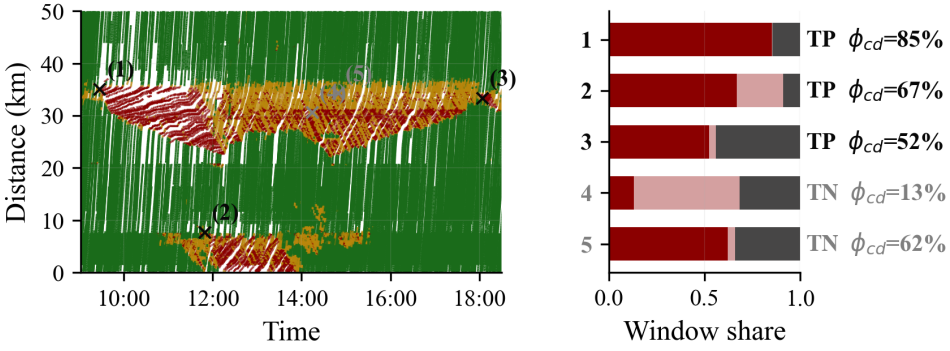


Figure 6.4: Cluster dominance for crash-induced congestion (1-3) and WMJs (4, 5). The crash shows a high cluster dominance due to the single coherent cluster, while the WMJ exhibits lower dominance as the jam fragments into multiple clusters.

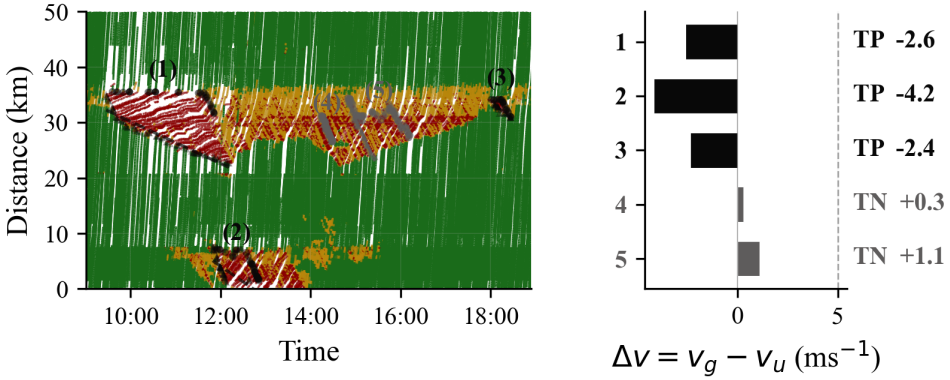


Figure 6.5: Velocity difference between upstream and downstream fronts for crash-induced congestion (1-3) and WMJs (4, 5). The crash shows a negative velocity difference due to the fixed downstream front and upstream-propagating front, while the WMJ exhibits a near-zero velocity difference as both fronts propagate upstream.

making it a supportive rather than primary indicator: it contributes when available but does not prevent an alert in full-blockage scenarios without a downstream front. Figure 6.5 illustrates the velocity difference for selected jam phases, showing that the crash-induced congestion (1-3) exhibits a negative velocity difference due to the fixed downstream front and upstream-propagating front, while the WMJ (4, 5) exhibits a near-zero velocity difference as both fronts propagate upstream.

6.1.2 Plausibility Filters

Real-world FCD contains noise, localization errors, and clustering imperfections that can produce spurious high-probability phases. Before an alert is accepted, a set of plausibility

conditions must be satisfied. These filters are designed to reject artifacts and reduce FPs rather than to perform fine-grained scoring; each enforces a necessary condition that any genuine crash must meet. Score evaluation is first gated by a hard prerequisite; the remaining filters contribute penalty terms that reduce the crash probability.

Phase Age (Hard Gate). A minimum observation period is required before the crash probability is evaluated at all, preventing premature detections from very short-lived phases. Let Δt_{age} denote the time elapsed since the jam seed was first observed. The score is only evaluated when

$$\Delta t_{\text{age}} \;\geq\; \Delta t_{\text{age,min}}\,. \tag{6.7}$$

Phases younger than $\Delta t_{\text{age,min}}$ are excluded from detection regardless of their indicator scores.

Prior Congestion History. A jam that emerges in a region with recent prior congestion is likelier to be demand-driven than crash-induced. Let n_{wmj} denote the number of distinct jam phases observed within a spatiotemporal window of width Δd_{wmj} and look-back Δt_{wmj} centered on the current seed. The penalty score is

$$f_{P,\text{wmj}} = \begin{cases} 1, & n_{\text{wmj}} \geq n_{\text{wmj,thr}}, \\ 0, & \text{otherwise}, \end{cases} \tag{6.8}$$

activating the full negative weight w_{wmj} whenever the local congestion count exceeds the threshold. Like the pre-jam free-flow ratio, this filter is evaluated at seed time and provides an immediate constraint: a jam emerging in a region already saturated with prior WMJs receives a penalty before any time-dependent indicators are available.

Mean Jam Speed. The mean speed of vehicles within the jam phase provides a context signal for congestion severity. A crash-induced standstill produces near-zero speeds, whereas recurring congestion typically maintains residual flow at moderate speeds. Jams with moderate mean speed are unlikely to reflect crash conditions and receive a penalty scaling with the observed speed. Let $\bar{v}_j$ denote the mean speed within the jam phase. The penalty

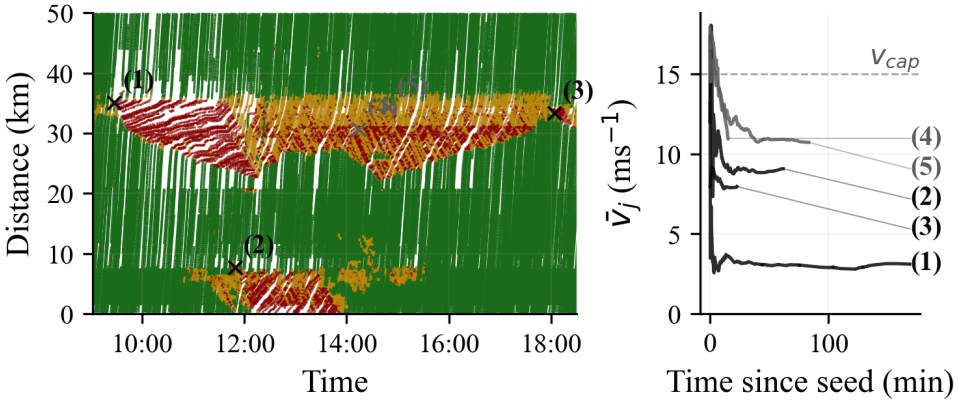


Figure 6.6: Mean speed within the jam phase for crash-induced congestion (1–3) and WMJs (4,5). Crashes exhibit near-zero mean speed and, therefore, no penalty, while WMJs retain residual flow that activates the penalty ramp.

activates on a ramp between a lower bound v_{lo} and an upper bound v_{hi}:

$$f_{P,\text{speed}}(\bar{v}_j) = \begin{cases} 0, & \bar{v}_j \leq v_{\text{lo}}, \\ \dfrac{\bar{v}_j - v_{\text{lo}}}{v_{\text{hi}} - v_{\text{lo}}}, & v_{\text{lo}} < \bar{v}_j < v_{\text{hi}}, \\ 1, & \bar{v}_j \geq v_{\text{hi}}, \end{cases} \tag{6.9}$$

so that standstill jams are unaffected ($f_{P,\text{speed}} = 0$) while jams with residual flow above v_{hi} incur the full penalty weight $w_{\text{avg-speed}} < 0$. Figure 6.6 illustrates the mean speed within the jam phase for selected jam phases: the crash-induced congestion cases (1–3) exhibit near-zero mean speed, while the WMJs (4,5) maintain higher residual flow and receive the full penalty.

Minimum Queue. A genuine crash-induced jam must produce a measurable queue of vehicles. Let N_q denote the number of vehicles within the jam region which speed is below a standstill threshold v_q. The penalty score is

$$f_{P,\text{queue}} = \begin{cases} 0, & N_q \geq N_{q,\min}, \\ 1, & \text{otherwise}, \end{cases} \tag{6.10}$$

penalizing phases where the queued vehicle count falls below the required minimum.

Upstream Velocity R^2. The upstream front velocity must be consistent with a coherent jam front propagating upstream. A regression line is fitted to the sequence of upstream velocity estimates $\{\hat{v}_{u,k}\}$, yielding a coefficient of determination R^2. The penalty score is

$$f_{P,R^2} = \begin{cases} 0, & R^2 \geq R^2_{\min}, \\ 1, & \text{otherwise}, \end{cases} \tag{6.11}$$

suppressing phases where scattered upstream velocity observations indicate noise or clustering artifacts rather than a genuine jam front.

Jam Growth Rate. The spatiotemporal area growth rate $\dot{A}$ of the jam cluster must not exceed a physical upper bound. Extremely high growth rates indicate measurement artifacts or clustering errors rather than real congestion growth. The penalty score is

$$f_{P,\text{growth}} = \begin{cases} 1, & \dot{A} > \dot{A}_{\max}, \\ 0, & \text{otherwise}, \end{cases} \tag{6.12}$$

penalizing phases whose area expansion exceeds the configurable limit $\dot{A}_{\max}$.

6.1.3 Aggregation into Crash Probability

The scored positive indicators and plausibility penalties are combined by an additive weighting scheme. Let $\mathcal{E}_{\text{excl}} = \{f_{S,\text{ds}}, f_{S,\text{FB}}\}$ denote the mutually exclusive downstream-front group (downstream velocity, full blockage), $\mathcal{I}$ the remaining independent positive indicators (pre-jam free-flow ratio, cluster dominance, velocity difference), and $\mathcal{P} = \{f_{P,\text{wmj}}, f_{P,\text{queue}}, f_{P,R^2}, f_{P,\text{growth}}, f_{P,\text{speed}}\}$ the set of plausibility penalty filters defined in equations (6.8) to (6.12). The crash probability is computed only for phases that pass the hard gate (equation (6.7)):

$$y^{\text{pred}}_{\text{cls}} = \min\left(1, \left(\underbrace{\sum_{i\in\mathcal{I}} w_i\, S_i \;+\; \max_{j\in\mathcal{E}_{\text{excl}}} w_j\, f_{S_j}}_{\text{positive indicators}} \;+\; \underbrace{\sum_{p\in\mathcal{P}} w_p\, f_{P_p}}_{\text{plausibility penalties}}\right)\cdot\gamma\right) \tag{6.13}$$

where $w_i > 0$ are non-negative weights reflecting relative importance, $w_p < 0$ are negative penalty weights that reduce the crash probability when plausibility conditions are violated (equations (6.8) and (6.10) to (6.12)), γ is a score multiplier, and only indicators with

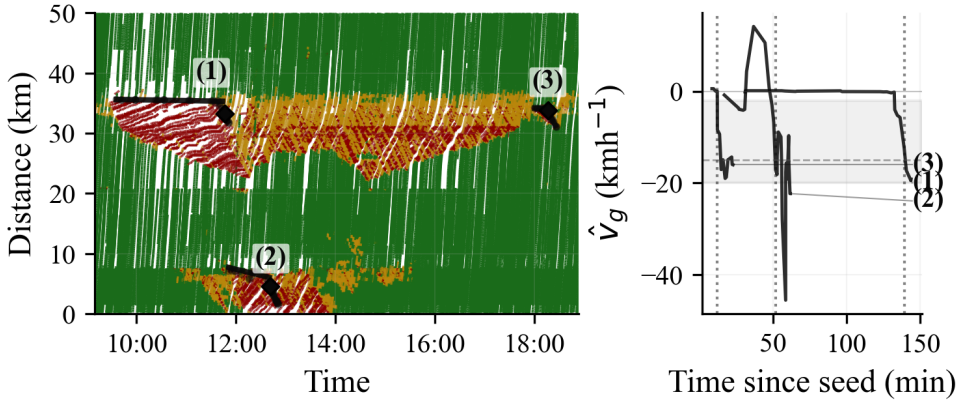


Figure 6.7: End-of-crash detection. Left: downstream front transitions with regression lines for the stationary crash phase (solid, $\hat{v}_g \approx 0$) and the dissipation phase (dashed, $\hat{v}_g \approx -15\,\mathrm{km\,h^{-1}}$); the diamond marks the detected regime transition. Right: downstream front velocity over time, showing the transition from spatial fixation to upstream propagation.

non-null scores participate. When all plausibility conditions are met, every $f_{P_p} = 0$ and the penalty sum vanishes, leaving the score determined entirely by the positive indicators. When one or more conditions are violated, the corresponding negative terms suppress the score, preventing artifactual phases from reaching the alert threshold. A crash alert is issued once $y_{\mathrm{cls}}^{\mathrm{pred}}$ exceeds a fixed threshold τ.

6.1.4 End of Crash Detection

Once the crash site is cleared, the stationary downstream front detaches and begins propagating upstream at the characteristic WMJ velocity $v_g \approx -15\,\mathrm{km\,h^{-1}}$. The system monitors the estimated downstream velocity $\hat{v}_g$ and issues an end-of-crash alert when

$$\hat{v}_{g,k} \leq v_{\mathrm{end}} \quad \text{for } c_{\mathrm{end}} \text{ consecutive evaluations,} \tag{6.14}$$

where $v_{\mathrm{end}} < 0$ is a velocity threshold consistent with upstream propagation and c_{end} is the required number of consecutive readings to suppress transient fluctuations. Figure 6.7 illustrates the transition from near-zero downstream velocity during the active crash phase to negative velocity once the site is cleared for crash-induced congestion (1–3). Post-crash congestion (2) shows a noisy signal, yet the sustained velocity shift is still detected.

6.1.5 Crash Localization

The system estimates the crash position along the monitored road segment using two complementary strategies derived directly from the phase representation computed by the detection pipeline.

Seed Position. The position of the jam seed—the first observed upstream-front transition of the detected phase—provides a single-shot localization estimate available at the moment of detection. Because the jam nucleates at or near the crash site, the seed position approximates the crash location, with accuracy bounded by the spatial resolution of the underlying FCD.

Downstream Front Position. Since the downstream front of a crash-induced jam remains spatially fixed at the crash location (Section 6.1.1), each observed downstream-front transition ($J \to S/F$ boundary) constitutes an independent position measurement. As transitions accumulate, a running mean over all observed positions averages out spatial noise. To prevent dilution by post-clearance front detachment, only transitions within a fixed time horizon after the seed (e.g., $\Delta t_{\max} = 7\,\text{min}$) are aggregated.

6.2 Parameter Calibration

The detection system involves over 30 parameters spanning feature weights, scoring ranges, plausibility filter thresholds, blockage configuration, penalty strengths, and alert logic. Calibration proceeded in two phases: parameter-space reduction through sensitivity analysis, followed by exploration of the F_1-vs-timeliness Pareto front via Bayesian optimisation (Optuna [185]) and systematic grid search.

6.2.1 Parameter Space Reduction

An initial Optuna study (339 trials, 22 tunable parameters) optimising F_1 on the validation set identified approximate operating ranges. These were frozen at the values listed in Table C.4, reducing the effective dimensionality to feature weights, penalty weights, scoring parameters, and alert logic.

6.2.2 Pareto-Front Exploration

The remaining parameter space was explored through two complementary strategies. Multiple Optuna runs with different objective weightings (pure F_1, F_1 and timeliness) produced

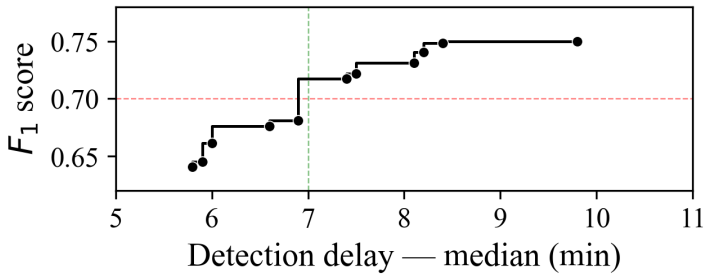


Figure 6.8: Pareto front of F_1 vs. median seed-to-alert delay for the 224 configurations evaluated in the grid search.

Table 6.1: Pareto-optimal operating points on the independent test set (65 ground-truth crashes, 50 datasets). Delays are seed-relative; $\tilde{\Delta}t$ denotes the median.

	Profile	$\boldsymbol{F_1}$	$\boldsymbol{F_2}$	**P**	**R**	**TP**	**FP**	$\tilde{\boldsymbol{\Delta}}\boldsymbol{t}$	**< 7 min**
Accuracy	Best F_1	**0.750**	0.743	0.762	0.738	48	15	9.8 min	12
Balanced	F_1 at speed	0.718	0.721	0.712	0.723	47	19	8.1 min	17
Speed	Fastest	0.653	0.710	0.577	0.754	49	36	**5.8** min	32

a set of candidate configurations. A systematic grid search then interpolated between these anchors at 11 blend levels, followed by a threshold sweep ($\tau = 0.05, 0.10, \ldots, 0.95$), and additionally varied the cluster-dominance window (180–881 s), penalty strengths, and blockage requirements. In total, over 500 configurations were evaluated on the validation set. Figure 6.8 shows the resulting Pareto front of F_1 vs. median seed-to-alert delay, revealing a hard trade-off between accuracy and timeliness. Three Pareto-optimal configurations were selected along this curve: *Accuracy* (best F_1), *Balanced* (F_1 at speed), and *Speed* (fastest).

6.3 Operating Points

Table 6.1 summarizes the three operating points on the independent test set. Complete per-profile configurations are listed in Tables C.1 to C.3.

6.4 Ablation and Feature Analysis

Two complementary analyses characterize the contribution of individual features. The ablation in Table 6.2 disables one feature or mechanism at a time while keeping all remaining parameters at their calibrated values. The feature-isolation experiment in Figure 6.9 complements this by activating only one feature at a time, removing the weight-

Table 6.2: Ablation across the three operating points. Selected rows illustrate the principal feature contributions; ΔFP denotes the change in false positives relative to the respective baseline. Full per-profile tables are provided in Tables C.5 to C.7.

	Accuracy			*Balanced*			*Speed*		
Ablation	F_1	ΔFP	med	F_1	ΔFP	med	F_1	ΔFP	med
Full (baseline)	.750	—	9.8	.718	—	8.1	.653	—	6.1
w/o cluster dom.	.060	−15	17.3	.116	−19	20.3	.088	−36	24.1
w/o free_flow_ratio	.762	−2	10.7	.726	−5	9.0	.676	−7	7.4
w/o ds velocity	.736	−1	10.3	.682	−1	7.5	.657	−5	6.1
w/o full blockage	.708	−7	9.8	.704	−3	8.5	.667	−7	6.5
w/o velocity diff	.727	−3	9.8	.698	+0	7.7	.649	−1	6.1
w/o all penalties	.556	+68	10.6	.348	+191	8.2	.291	+239	6.7
w/o avg speed penalty	.750	+0	9.8	.699	+11	8.0	.614	+21	6.4

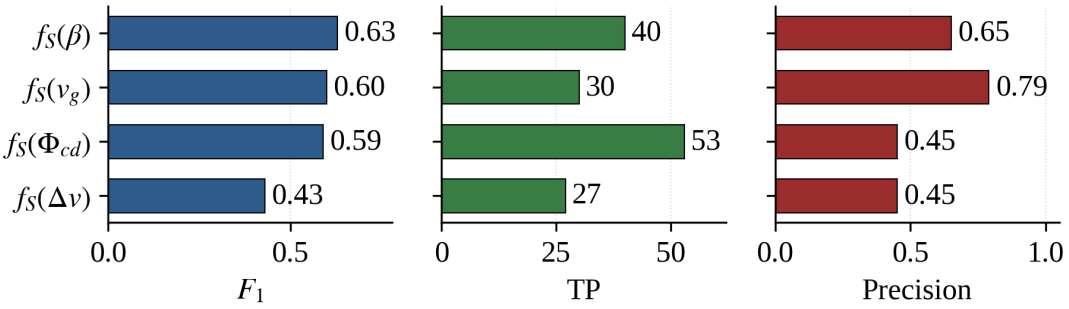


Figure 6.9: Feature-isolation experiment for the *Accuracy* configuration. Each bar shows the F_1 obtained when only the indicated feature is active, with all other features and penalties disabled.

bias introduced by the calibrated feature weights. Full per-profile tables are provided in Tables C.5 to C.8.

Removing cluster dominance collapses F_1 to ≤ 0.116 in every profile—its calibrated weight ($w_{\text{cd}} = 0.540$–0.632) is the only source capable of crossing $\tau = 0.50$ single-handedly. The feature-isolation view (Figure 6.9) provides the complementary picture: cluster dominance alone yields the highest recall ($R = 0.815$, median delay 5.8 min) but the *lowest* precision of any feature ($P = 0.445$, 66 FP). Full blockage attains the highest isolated F_1 (0.630, $P = 0.645$) and downstream velocity the highest isolated precision ($P = 0.789$). The system's effectiveness, therefore, derives from complementarity: cluster dominance provides broad, early-triggering recall, and full blockage, downstream velocity, and velocity difference act as precision-enhancing modifiers that suppress the false positives cluster dominance admits.

Table 6.3: Detection performance of the *Accuracy* configuration under varying fleet density. TP/FP are averaged across three random seeds per level; $\tilde{\Delta}t$ denotes the median seed-to-alert delay.

Density	TP	FP	$\tilde{\Delta}t$ [min]	F_1
25 %	35	17	13.2	0.598
50 %	45	17	12.3	0.709
75 %	46	15	10.8	0.730
100 %	48	15	9.8	0.750
125 %	48	16	9.6	0.744
150 %	47	16	9.4	0.734
175 %	47	16	9.1	0.734
200 %	47	17	9.0	0.729

All three profiles operate at $\tau = 0.50$, making the penalty terms the principal false-positive suppression mechanism. Disabling all penalties inflates FP by +68 in *Accuracy*, +191 in *Balanced*, and +239 in *Speed*—with magnitude scaling inversely to how conservatively each profile parametrizes its feature thresholds. The pre-jam free-flow ratio trades precision for timeliness (+0.9–1.3 min median delay when removed), making it most useful in speed-critical deployment contexts.

6.5 Sensitivity to Fleet Density

The calibrated features differ in how many independent vehicle observations they require before producing a reliable score. Cluster dominance computes a ratio over all classified segments in a spatiotemporal window; whether that window contains 50 or 200 segments, the ratio remains interpretable. Downstream velocity and blockage, by contrast, require corroboration across vehicles: the former needs consecutive near-zero readings along the downstream front, the latter multiple independent sessions to confirm sustained standstill. To quantify how these differences translate into detection performance, a controlled experiment varied the effective fleet density from 25 % to 200 % of the original test set.

Experimental setup. Density reduction (< 100 %) was achieved by randomly removing whole vehicle trajectories (sessions), preserving the temporal structure of retained vehicles. Density augmentation (> 100 %) duplicated randomly selected sessions with a +30 s time shift and ±30 m per-point position jitter, simulating additional vehicles in the same traffic stream.

Fleet density affects timeliness substantially more than precision. False positives remain within ±2 of the 100 % baseline across the entire 25–200 % range, indicating that the penalty mechanisms suppress spurious detections independently of fleet size. Median delay improves by 4.2 min between the extremes (13.2 → 9.0 min). Recall is similarly robust on the upper side of the 100 % reference: the system retains 46–48 of the 48 baseline true positives at every level from 75 % upward. Degradation concentrates in the lowest-density regime; at 25 % density, recall drops to 73 % of baseline (35/48) and median delay rises by 3.4 min. Augmenting beyond 100 % yields only marginal gains (0.8 min median-delay improvement between 100 % and 200 %), indicating that the system is close to its detection-latency floor at the native fleet density of the test set. The calibrated configuration is, therefore, well-matched to the penetration rates typical of current commercial FCD fleets (2–5 %).

6.6 Summary of Findings

Three findings characterize the rule-based detector.

First, the four features play complementary roles. Cluster dominance is the strongest single feature for recall and early triggering (isolated $R = 0.815$, median delay 5.8 min) but also produces the most false positives on its own. Full blockage, downstream velocity, and velocity difference do the opposite: each is weaker on recall but cleans up the false positives that cluster dominance admits.

Second, the Pareto-optimal operating points (Table 6.1) represent a real trade-off. *Accuracy* reaches $F_1 = 0.750$ at a median delay of 9.8 min, while *Speed* cuts the delay roughly in half to 5.8 min at the cost of 21 additional false positives.

Third, the detector handles reduced fleet density well. Precision stays roughly constant across the 25–200 % sweep, and recall drops only once the number of segments falls below the minimum that the cluster-dominance ratio needs to be statistically meaningful. Downstream velocity and blockage features stay informative when data is abundant.

7 Data-Driven Crash Detection Using Learning-Based Models

Data-driven approaches play an increasingly important role in traffic state estimation [61] and crash detection [204] due to the growing availability of high-resolution FCD. Learning-based models can extract complex spatiotemporal patterns directly from observations without requiring an analytical formulation of traffic flow dynamics [205]. In crash detection, this is particularly relevant, since crashes are rarely observed directly in FCD but become visible indirectly through characteristic changes in traffic flow [206]. This chapter considers CNNs and GNNs as two complementary model classes. CNN-based approaches operate on structured input representations such as spatiotemporal traffic maps derived from aggregated vehicle trajectories [e.g., 207], exploiting the regular grid structure through convolutional filters. The GNN-based approach constructs graphs directly from individual vehicle trajectories rather than from an explicit road network. Vehicle positions form nodes; edges encode interactions based on temporal overlap, spatial proximity, or similarity in traffic states. The resulting graph captures collective vehicle behavior within the traffic flow. Both model classes follow a hybrid modeling strategy. Physics-based structure, encoded in the input representations, graph construction rules, and model architecture, constrains the hypothesis space, while the model parameters are estimated from data.

7.1 Task Formulation and Shared Setup

7.1.1 Learning-Based Task Formulation

The task defined in Section 5.1.3 is realized as a supervised prediction problem. At each detection time t_{det}, the model observes all FCD trajectories within the spatiotemporal observation window $\mathcal{W}_{D,T}$. The model produces two outputs:

$$y_{\mathrm{cls}}^{\mathrm{pred}} \in (0,1), \qquad y_{\mathrm{loc}}^{\mathrm{pred}} \in (0,1). \tag{7.1}$$

The learning-based approach extends a master's thesis supervised by the author [175] and builds upon [9, 10].

A sliding sequence of observation windows is evaluated independently. Timeliness is determined by the earliest window for which $y_{\text{cls}}^{\text{pred}}$ exceeds the decision threshold.

7.1.2 Loss Functions

Crash classification is optimized using a class-weighted binary cross-entropy loss to account for the strong imbalance between crash and non-crash samples:

$$\ell_{\text{cls}} = \frac{1}{Z} \sum_{i=1}^{N} w_{y_{\text{cls}}^{\text{gt},(i)}} \Big[-y_{\text{cls}}^{\text{gt},(i)} \log y_{\text{cls}}^{\text{pred},(i)} - \big(1 - y_{\text{cls}}^{\text{gt},(i)}\big) \log\big(1 - y_{\text{cls}}^{\text{pred},(i)}\big) \Big], \tag{7.2}$$

with class weights $w_1 = 3$ and $w_0 = 1$. The normalization factor $Z = \sum_{i=1}^{N} w_{y_{\text{cls}}^{\text{gt},(i)}}$ ensures a consistent loss scale across batches with varying class ratios.

Crash localization is trained using the mean absolute error (MAE), evaluated only on crash samples:

$$\ell_{\text{loc}} = \frac{1}{N_{\text{pos}}} \sum_{i:\, y_{\text{cls}}^{\text{gt},(i)}=1} \big| y_{\text{loc}}^{\text{pred},(i)} - y_{\text{loc}}^{\text{gt},(i)} \big|. \tag{7.3}$$

Classification and localization models are optimized independently. As crash classification constitutes the primary objective of this study, model selection, hyperparameter tuning, and ablation experiments are primarily guided by ℓ_{cls}.

7.1.3 Experimental Setup

All experiments were implemented in PyTorch 2.8.0 [208], with GNN components based on PyTorch Geometric 2.6.1 [209]. Training was performed on an Amazon SageMaker [210] `ml.g5.2xlarge` instance equipped with a single NVIDIA A10G graphics processing unit (GPU) (24,GB memory, CUDA 12.6 support). Unless otherwise specified, all models were trained for up to 30 epochs using the Adam optimizer [211]. Gradient clipping with a maximum norm of 10 was applied to stabilize training. Model-specific learning rate schedules are described in their respective sections. Both pipelines operate on $\mathcal{W}_{D,T}$ with $D = 30\,\text{km}$ and $T = 30\,\text{min}$, meaning each model uses 30 minutes of traffic history along a 30 km freeway segment to predict crash occurrence and location. All ablation studies in this chapter were conducted on a reduced dataset, since generating and processing the full dataset is computationally expensive. The focus was on iterative model development and verifying the feasibility of individual components; using a smaller dataset for optimization and ablation was both sufficient and appropriate. Due to the high computational cost of

Table 7.1: Reduced training set used for model optimization and iterative development.

Split	Total windows	Crash windows	Non-crash windows
Train	32,946	16,304 (49.5%)	16,642 (50.5%)
Val	3,723	1,671 (44.9%)	2,052 (55.1%)

generating full test streams, all intermediate evaluations were performed on the validation set. Only the final best-performing models after ablation and hyperparameter selection were evaluated on the held-out test set. Table 7.1 summarizes the reduced ablation training dataset.

During development, both CNN and GNN models were found to be highly prone to overfitting. This is expected, as the dataset contains only 445 unique crash records (see Table 3.5), which introduces strong non-independence between samples. To mitigate this, validation was performed every few hundred training steps instead of once per epoch, enabling earlier detection of overfitting and more fine-grained model selection. Early stopping was applied based on the AUC-score, which served as the primary evaluation metric for model selection.

7.2 CNN-Based Crash Detection

CNNs operate on fixed-size grid inputs, making them a natural choice for spatiotemporal traffic data represented as space–time maps [207]. The U-Net architecture [155] is adopted for its encoder–decoder structure: the encoder captures multi-scale context through successive downsampling, while the decoder restores spatial resolution through skip connections, enabling joint crash classification and localization.

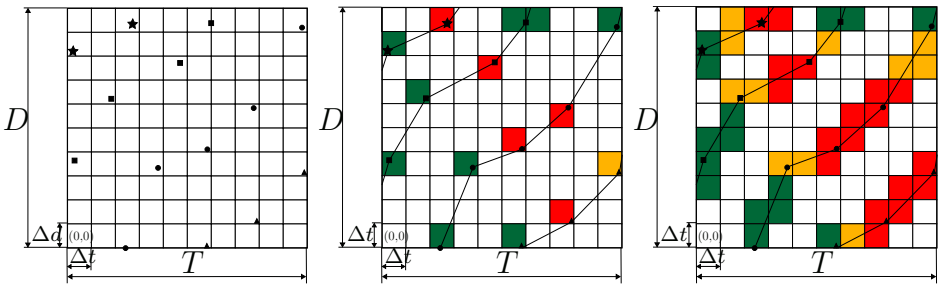
7.2.1 Grid Encoding

Building on the shared input preparation (Section 3.7), the raw FCD within each $\mathcal{W}_{D,T}$ are transformed into a structured tensor representation. The complete procedure is illustrated in Figure 7.1.

Each window is discretized into a regular time–distance grid with spatial resolution $\Delta d = 150\,\text{m}$ and temporal resolution $\Delta t = 10\,\text{s}$, yielding $p_d = 200$ spatial bins and $p_t = 180$ temporal bins. All trajectory points falling into a given cell (m, n) are assigned to that cell; speed is computed as the arithmetic mean of all observations, and traffic phase information is aggregated accordingly. Cells without observations remain empty and are marked as unknown.

The resulting two spatiotemporal matrices, one containing mean speeds and one containing traffic phase labels, are stacked along the channel dimension to form the input tensor $\boldsymbol{\mathcal{X}}_{\text{CNN}} \in \mathbb{R}^{p_d \times p_t \times 2}$, where the first channel represents speed and the second channel represents the one-hot encoded traffic phase.

Due to incomplete vehicle coverage, the resulting tensors are highly sparse: 76.0 % of all grid cells contain no observations. This sparsity reflects the limited penetration rate α and motivates dedicated strategies for handling missing data, described in the following subsections.



(a) Grid discretization and raw point mapping. (b) Pixel assignment and trajectory reconstruction. (c) Interpolation and phase encoding.

Figure 7.1: Step-by-step construction of the CNN input tensor from raw vehicle trajectories [175].

7.2.2 Sparsity Handling

Two strategies for handling missing grid cells are compared: input-level completion via PSM, and architecture-level masking via partial convolutions. The ablation in Section 7.2.4 evaluates their relative effectiveness.

PSM Completion

The first strategy fills missing entries by adapting PSM [159]. Since traffic phases are already determined during preprocessing (Chapter 3), the adapted procedure first estimates a phase probability map for missing cells and then completes speeds guided by these probabilities. The procedure is illustrated in Figure 7.2 and detailed in Algorithm 1.

For each missing cell (m, n), a probability vector $\mathbf{q}_{m,n} = \big(P(\text{F}),\, P(\text{S}),\, P(\text{J})\big)$ is estimated. WMJs propagate upstream at $v_g \approx -15\,\text{km}\,\text{h}^{-1}$ [59], so each observed J-pixel emits an upstream propagation sector, marking missing pixels likely belonging to phase J. An analogous sector for the synchronized phase S limits unbounded jam expansion. For a

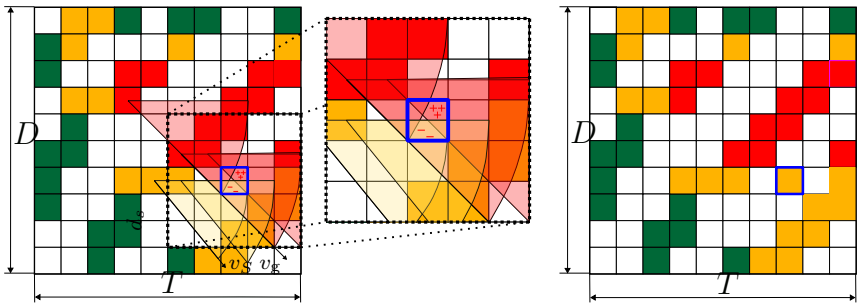


Figure 7.2: Sector-based propagation counting of jam pixels. The blue-bordered pixel falls within three J sectors and two S sectors, giving a net count of 1 below the threshold and is therefore not assigned to J. [175]

missing pixel, the net count $c = C_{\mathrm{J}} - C_{\mathrm{S}}$ determines the assignment: if $c \geq C_{\mathrm{psm}}$, the pixel is assigned to jam ($\mathbf{q}_{m,n} = (0, 0, 1)$).

Speed completion operates on reciprocal speeds for numerical stability in congested regions. For each phase Φ, an exponentially decaying convolution kernel $\mathbf{K}_\Phi$ [52] is parameterized by a temporal decay ℓ_Φ^t, a spatial decay ℓ_Φ^d, and a propagation speed c_Φ following 3PTT [42]: $c_{\mathrm{F}} = 80\,\mathrm{km\,h^{-1}}$, $c_{\mathrm{S}} = 0\,\mathrm{km\,h^{-1}}$, $c_{\mathrm{J}} = v_g = -15\,\mathrm{km\,h^{-1}}$. Each kernel is convolved with the reciprocal-speed field over observed pixels only and normalized, yielding a phase-specific estimate $\hat{v}_\Phi^{-1}$. The three estimates are fused via the phase probabilities:

$$\hat{v}^{-1}(m,n) = \sum_{\Phi \in \{\mathrm{F,S,J}\}} \mathbf{q}_{m,n,\Phi}\, \hat{v}_\Phi^{-1}(m,n) + v_{\mathrm{obs}}^{-1}(m,n). \tag{7.4}$$

Since $\mathbf{q}_{m,n} = \mathbf{0}$ at observed pixels, the second term preserves original measurements. Unfilled pixels are handled iteratively: newly completed pixels join the validity mask and the convolution–fusion cycle repeats until convergence.

Finally, the fused reciprocal-speed map is smoothed with a 3×3 Gaussian kernel, and each missing pixel is assigned the phase with the highest probability in $\mathbf{Q}$. The resulting labels are one-hot encoded and concatenated with the imputed reciprocal speed to form the completed four-channel input tensor $\boldsymbol{\mathcal{X}}_{\mathrm{CNN}}^{\mathrm{psm}} \in \mathbb{R}^{p_d \times p_t \times 4}$.

Partial Convolutions

The second strategy avoids explicit completion altogether. Partial convolutions [154] condition each convolutional operation on a binary validity mask $\mathbf{M} \in \{0,1\}^{p_d \times p_t}$ that tracks which cells contain observations. At each layer, the convolution output is renormalized by the fraction of valid pixels within the receptive field, and the mask is updated: any output pixel with at least one valid input is marked as valid. This allows the network to

process sparse inputs directly without introducing interpolation artifacts.

The input is constructed from the sparse grid $\boldsymbol{\mathcal{X}}_{\text{CNN}}$: speeds are inverted, phases are one-hot encoded with $\mathbf{0}$ for missing values, yielding the same four-channel format as the PSM variant.

7.2.3 U-Net with Transformer Head

The CNN-based model pairs a U-Net backbone [155] with a Transformer-based aggregation head for the final prediction. The CNN_{PSM} and $\text{CNN}_{\text{Partial}}$ variants share this architecture but differ in input construction.

U-Net backbone. All convolutional layers use 3×3 kernels with batch normalization and LeakyReLU ($\alpha = 0.1$). The encoder (Figure 7.3) consists of an input block and two downsampling blocks, each containing three residual sub-blocks with skip connections. Each downsampling block halves both spatial dimensions via stride-2 convolutions and doubles the channel depth, reaching $\mathbb{R}^{p_d/4 \times p_t/4 \times 4c_{\text{base}}}$.

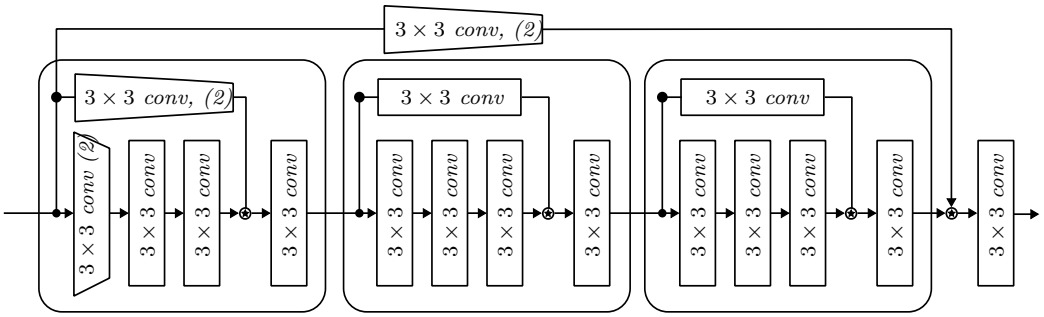


Figure 7.3: Encoder block structure. The trapezoid-marked convolution uses stride 1 in the input block and stride 2 in downsampling blocks. ⊛ denotes channel-wise concatenation [175].

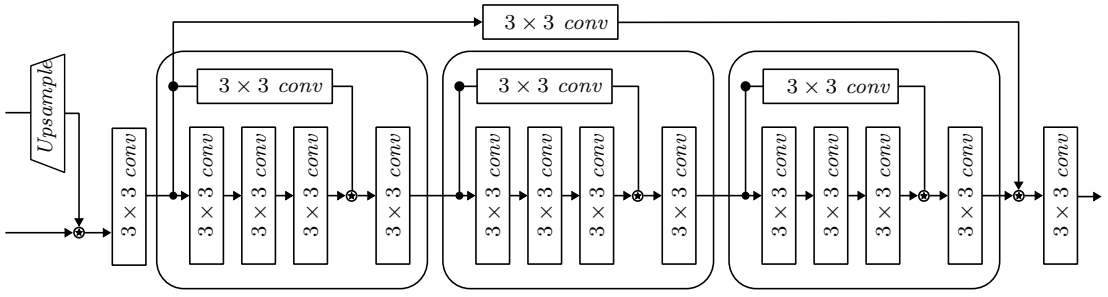


Figure 7.4: Decoder block structure. ⊛ denotes channel-wise concatenation [175].

The decoder (Figure 7.4) mirrors the encoder with nearest-neighbor upsampling and skip-connection concatenation, restoring the resolution to $\mathbb{R}^{p_d \times p_t \times c_{\text{base}}}$. A final stride-$(1, 4)$ convolution compresses the temporal axis, producing $\boldsymbol{\mathcal{O}} \in \mathbb{R}^{p_d \times p'_t \times 1}$ with $p'_t = p_t/4$. Figure 7.5 shows the complete backbone.

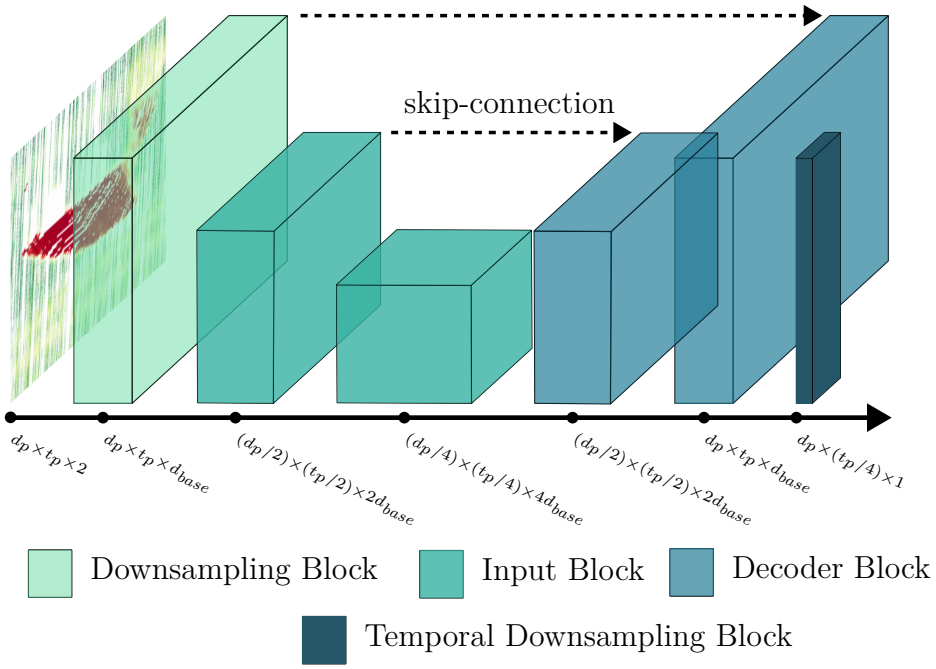


Figure 7.5: Overall U-Net backbone architecture [175].

Transformer attention aggregation head. The tensor $\boldsymbol{\mathcal{O}}$ is squeezed along the channel axis and rearranged into a matrix $\mathbf{Z} \in \mathbb{R}^{p_d \times p'_t}$, where each row $\mathbf{z}_m$ represents one spatial slice. A three-layer Transformer encoder [168] with five attention heads processes this sequence. A learnable query $\mathbf{r} \in \mathbb{R}^{p'_t}$ then aggregates the sequence via scaled dot-product attention [168]. A three-layer multilayer perceptron (MLP) with sigmoid activation maps the aggregated representation to the scalar output $y_{\text{cls}}^{\text{pred}} \in (0, 1)$ or $y_{\text{loc}}^{\text{pred}} \in (0, 1)$ depending on the task.

7.2.4 CNN Model Configuration and Ablation Study

Training uses an effective batch size of 32 (base 16 with two gradient accumulation steps), an initial learning rate of 1×10^{-4} decayed by $\gamma = 0.6$ every two epochs, and a base channel width $d_{\text{base}} = 12$ (Section D.3). Early stopping monitors binary classification loss on the validation set every 200 steps. Table 7.2 summarizes all hyperparameters.

Table 7.2: CNN pipeline: PSM hyperparameters and CNN architecture [175].

Category	**Setting**
Grid resolution $\Delta d, \Delta t$	$\Delta d = 150\,\mathrm{m}$, $\Delta t = 10\,\mathrm{s}$
Grid size $p_d \times p_t$	$p_d = 200$, $p_t = 180$
Sector parameters $\Delta j_a, \Delta i_a$	$\Delta j_S = 20$, $\Delta i_S = 10$; $\Delta j_J = 20$, $\Delta i_J = 20$
Sector threshold τ_{psm}	$\tau_{\mathrm{psm}} = 3$
Local probability window w_{prob}	$w_{\mathrm{prob}} = 60$
PSM convolution kernel size $k_{\mathrm{conv}}^{\mathrm{psm}}$	$k_{\mathrm{conv}}^{\mathrm{psm}} = 49$
Kernel temporal decay ℓ_Φ^t	$\ell_{\mathrm{F}}^t = 120\,\mathrm{s}$, $\ell_{\mathrm{S}}^t = 120\,\mathrm{s}$, $\ell_{\mathrm{J}}^t = 120\,\mathrm{s}$
Kernel spatial decay ℓ_Φ^d	$\ell_{\mathrm{F}}^d = 15\,\mathrm{m}$, $\ell_{\mathrm{S}}^d = 15\,\mathrm{m}$, $\ell_{\mathrm{J}}^d = 150\,\mathrm{m}$
Kernel propagation speed c_Φ	$c_{\mathrm{F}} = 80\,\mathrm{km\,h^{-1}}$, $c_{\mathrm{S}} = 0\,\mathrm{km\,h^{-1}}$, $c_{\mathrm{J}} = -15\,\mathrm{km\,h^{-1}}$
Base channel width d_{base}	$d_{\mathrm{base}} = 12$

Initialization strategy All convolution-type layers and linear layers followed by LeakyReLU use Kaiming uniform [212] initialization; all biases are set to zero; all normalization layers use unit scale with zero bias; learnable query vectors in the Transformer head are initialized with Xavier uniform [213].

Loss function Two identical models are trained separately: one for binary crash classification using the class-weighted binary cross-entropy (BCE) loss defined in Section 7.1.2, and one for crash localization using MAE on crash samples only ($y_{\mathrm{loc}} \in (0, 1)$).

Ablation Analysis

To assess the contribution of individual design factors, ablation experiments are conducted on the CNN base model (Section 7.2.3). Five variants are compared across two settings, with and without PSM, and evaluated on the validation set (Table 7.3); per-run results are provided in Table D.2.

- **With PSM.** (a) Full pipeline: PSM completion, phase input, and Transformer head. (b) Without phase input: only the velocity channel. (c) Without Transformer head: global average pooling replaces the attention aggregation.
- **Without PSM.** (d) Raw sparse input: no completion applied. (e) Partial convolution backbone: convolutional layers replaced by partial convolutions [154] to handle missing pixels directly.

Three observations emerge. First, comparing (a) with (d) and (e) shows that PSM provides limited gains; variant (e) outperforms the full pipeline by handling sparsity directly

Table 7.3: Ablation results on the validation set (threshold = 0.5). Best values in **bold**, second best underlined [175].

Variant	Val Loss	Precision	Recall	F1
With PSM				
(a) Full pipeline	$0.376_{\pm 0.002}$	$80.8\%_{\pm 2.6}$	$91.3\%_{\pm 1.1}$	$81.7\%_{\pm 0.2}$
(b) W/o phase input	$\underline{0.371_{\pm 0.015}}$	$\underline{82.4\%_{\pm 0.7}}$	$\underline{91.8\%_{\pm 2.2}}$	$\underline{83.5\%_{\pm 0.5}}$
(c) W/o Transformer head	$0.386_{\pm 0.015}$	$81.2\%_{\pm 2.2}$	$\mathbf{95.1\%_{\pm 4.0}}$	$82.1\%_{\pm 0.2}$
Without PSM				
(d) Raw sparse input	$0.372_{\pm 0.005}$	$82.8\%_{\pm 0.5}$	$91.6\%_{\pm 1.5}$	$83.1\%_{\pm 0.2}$
(e) Partial conv backbone	$\mathbf{0.360_{\pm 0.006}}$	$\mathbf{83.3\%_{\pm 0.9}}$	$90.6\%_{\pm 1.2}$	$\mathbf{83.7\%_{\pm 0.5}}$

through partial convolutions, suggesting that PSM completion alters the raw signal and removes useful detail. Second, replacing the Transformer head with global average pooling (a vs. c) degrades F1, confirming the value of learned spatial aggregation. Third, removing the phase channels improves performance (a vs. b), indicating that PSM-completed phase labels introduce noise rather than reliable information.

Based on these findings, the two best-performing variants are retained for test-set evaluation in the next chapter: (b) PSM with velocity input only, and (e) partial convolution backbone without PSM.

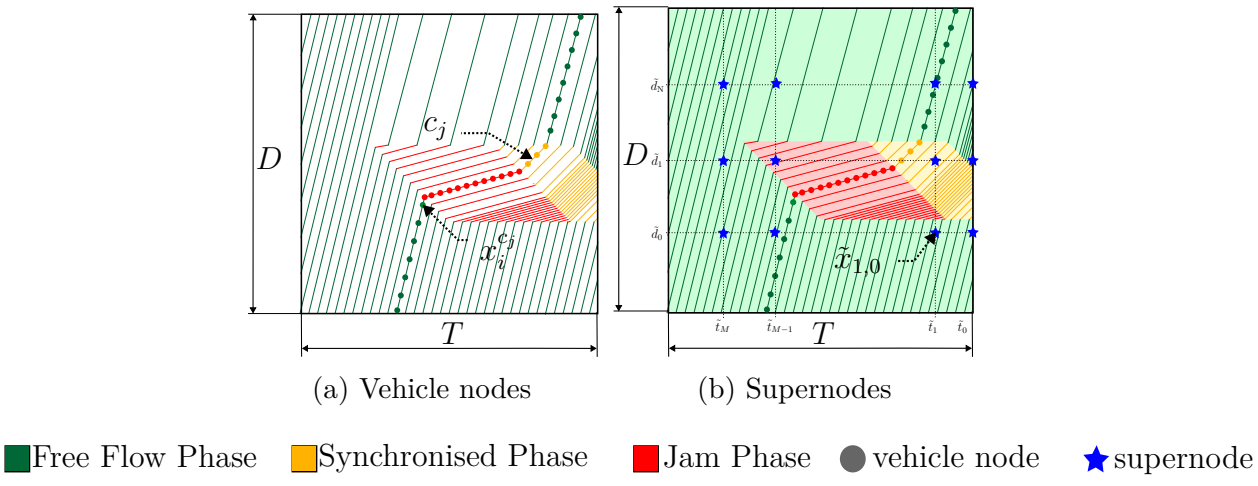
7.3 GNN-Based Crash Detection

Spatiotemporal extensions of GNNs enable joint modeling of evolving traffic dynamics and inter-vehicle relationships [170, 214]. The heterogeneous graph formulation adopted here supports multiple node and edge types within a unified structure, encoding individual vehicle kinematics, local traffic states, and directional propagation characteristics [10, 175].

7.3.1 Graph Construction

Unlike temporal GNNs that represent time as a sequence of synchronous graph snapshots $\{G^{(t)}\}$, a unified spatiotemporal graph is constructed from vehicle trajectories distributed across time and space. Each observation window is transformed into a heterogeneous directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$ with two node types and multiple edge types:

$\mathcal{V}$ is the node set, consisting of vehicle nodes $\mathbf{x}_i$ (individual trajectory data points) and supernodes $\tilde{\mathbf{x}}_{m,n}$ (virtual aggregation nodes). All nodes lie within the observation window $\mathcal{W}_{D,T}$.



(a) Vehicle nodes (b) Supernodes

Figure 7.6: Vehicle nodes and supernodes with phase context in $\mathcal{W}_{D,T}$ [175].

$\mathcal{E}$ is the edge set, comprising (i) vehicle spatial edges $\mathcal{E}_{\text{spatial}}$, (ii) WMJ propagation edges $\mathcal{E}_{\text{WMJ}}$, and (iii) supernode edges $\mathcal{E}_{\text{super}}$.

$\mathcal{L}$ is the set of temporal sequences, comprising vehicle trajectory sequences $\mathcal{L}_{\text{car}}$ and supernode history sequences $\mathcal{L}_{\text{super}}$. Concrete thresholds for all construction parameters are collected in Table 7.4.

Vehicle nodes. Each vehicle observation within $\mathcal{W}_{D,T}$ becomes a node whose feature vector $\mathbf{x}_i^{(\psi_j)} = \text{concat}(v_i,\ \mathbf{p}_i,\ t_i,\ d_i) \in \mathbb{R}^7$ concatenates scalar velocity v_i, a one-hot phase label $\mathbf{p}_i \in \{0,1\}^4$ encoding the four states (U, F, S, J) of the 3PTT, and spatiotemporal coordinates t_i, d_i. The superscript (ψ_j) identifies the parent trajectory. Figure 7.6a illustrates a single trajectory with its corresponding nodes.

Supernodes. Supernodes are placed at fixed spatiotemporal anchors $(\tilde{t}_m, \tilde{d}_n)$. Temporal anchors $\tilde{\mathcal{T}}$ are spaced at minute-level offsets before the detection time t_{det}; spatial anchors $\tilde{\mathcal{D}}$ divide the observation range D into equidistant segments upstream of d_{up}. Each supernode collects observations from an aggregation window $\mathcal{W}_{mn} = [\tilde{t}_m - \Delta t_{\text{agg}},\ \tilde{t}_m] \times [\tilde{d}_n - \Delta d_{\text{agg}},\ \tilde{d}_n + \Delta d_{\text{agg}}]$, retaining only the most recent record per vehicle. its feature vector $\tilde{\mathbf{x}}_{m,n} = \text{concat}(\tilde{v}_{m,n},\ \tilde{\rho}_{m,n},\ \tilde{t}_m,\ \tilde{d}_n) \in \mathbb{R}^4$ concatenates the mean speed $\tilde{v}_{m,n}$, the vehicle density $\tilde{\rho}_{m,n} = N_{m,n}/(2\,\Delta d_{\text{agg}})$, and the anchor coordinates. The resulting layout over $\tilde{\mathcal{T}} \times \tilde{\mathcal{D}}$ is shown in Figure 7.6b.

Spatial edges. Three families of directed edges capture multi-scale spatial interactions. All edge types include self-loops (omitted below for brevity) and enforce causality by

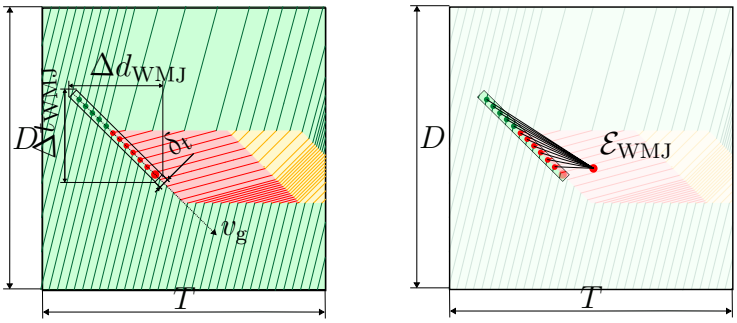


Figure 7.7: Illustration of vehicle WMJ edges $\mathcal{E}_{\mathrm{WMJ}}$ [175].

permitting only past-to-present connections.

(a) Vehicle spatial edges $\mathcal{E}_{\mathrm{spatial}}$ connect vehicle nodes that are spatially close and temporally adjacent. A directed edge from $\mathbf{x}_j$ to $\mathbf{x}_i$ exists if their spatial separation is at most Δd_{spa} and $\mathbf{x}_j$ occurs within a look-back horizon Δt_{spa} before $\mathbf{x}_i$:

$$(\mathbf{x}_j, \mathbf{x}_i) \in \mathcal{E}_{\mathrm{spatial}} \iff |d_j - d_i| \leq \Delta d_{\mathrm{spa}}, \quad t_j \in [t_i - \Delta t_{\mathrm{spa}},\ t_i]. \tag{7.5}$$

(b) Vehicle WMJ edges $\mathcal{E}_{\mathrm{WMJ}}$ encode the upstream propagation of WMJs ($v_g = -15\,\mathrm{km\,h^{-1}}$ [59]). A directed edge from $\mathbf{x}_j$ to $\mathbf{x}_i$ is created when $\mathbf{x}_j$ lies within the temporal and spatial horizons of $\mathbf{x}_i$ and falls within a tolerance of the ideal propagation line $t_i + (d_j - d_i)/v_g$ (Figure 7.7):

$$(\mathbf{x}_j, \mathbf{x}_i) \in \mathcal{E}_{\mathrm{WMJ}} \iff t_j \in [t_i - \Delta t_{\mathrm{WMJ}},\ t_i],\ d_j \in [d_i,\ d_i + \Delta d_{\mathrm{WMJ}}],\ \left|t_j - \left(t_i + \tfrac{d_j - d_i}{v_g}\right)\right| \leq \Delta t_{\mathrm{tol}}. \tag{7.6}$$

(c) Supernode edges $\mathcal{E}_{\mathrm{super}}$ provide hierarchical aggregation and consist of two subtypes. *Vehicle-to-supernode edges* connect every vehicle node whose coordinates fall inside $\mathcal{W}_{mn}$ to the corresponding supernode $\tilde{\mathbf{x}}_{m,n}$. *Supernode-to-supernode edges* link each supernode to its immediate temporal predecessor and to its spatial neighbors at the same time step. Both subtypes are processed jointly, so that each supernode simultaneously receives fine-grained vehicle information and coarser context from adjacent anchors (Figure 7.8). An alternative sequential design is evaluated in Section 7.3.4.

Temporal sequences. *Vehicle trajectory sequences* $\mathcal{L}_{\mathrm{car}}$ contain all nodes of a single vehicle in chronological order, encoding fine-grained trajectory dynamics (Figure 7.9a). *Supernode history sequences* $\mathcal{L}_{\mathrm{super}}$ group all supernodes sharing the same spatial anchor $\tilde{d}_n$ in temporal order (Figure 7.9b).

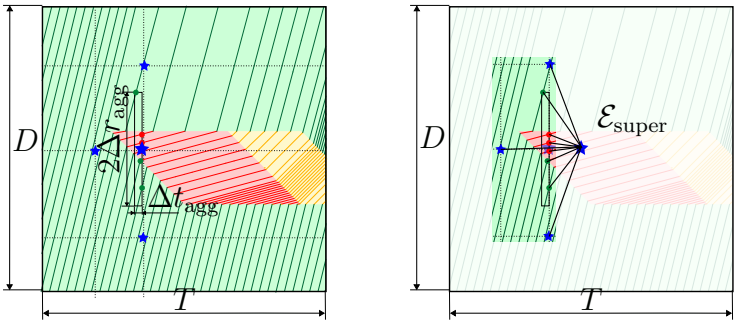
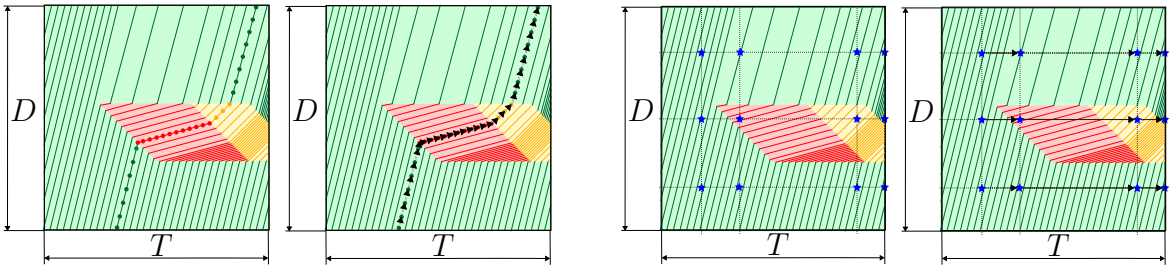


Figure 7.8: Illustration of supernode edges $\mathcal{E}_{\text{super}}$ [175].



(a) Vehicle trajectory sequence $\mathcal{L}_{\text{car}}$ (b) Supernode history sequence $\mathcal{L}_{\text{super}}$

Figure 7.9: Temporal sequences [175].

Graph construction procedure. The graph is built through an iterative, edge-driven expansion starting at the supernode anchors, progressively including only those vehicle nodes required for downstream message passing. All vehicle nodes inside $\mathcal{W}_{D,T}$ and all supernodes are collected into $\mathcal{V}$. Supernode edges $\mathcal{E}_{\text{super}}$ are established first; the incident vehicle nodes form the initial frontier $\mathcal{F}_1$. In each subsequent round k, spatial and WMJ edges are generated from $\mathcal{F}_k$, and newly reached nodes become $\mathcal{F}_{k+1}$. After K_{hop} rounds the accumulated edges are deduplicated, self-loops are added, and the temporal sequences $\mathcal{L}_{\text{car}}$ and $\mathcal{L}_{\text{super}}$ are constructed, yielding $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$. The procedure is listed in Algorithm 2.

To enable efficient batching despite large variation in graph size, a complexity index $\mathcal{C} = w_{\text{v}} |\mathcal{V}_{\text{car}}| + w_{\text{e}} |\mathcal{E}| + w_{\text{s}} |\mathcal{V}_{\text{super}}|$ is computed for each sample, and samples with similar complexity are grouped to reduce memory overhead; details are given in Section 7.3.3.

Ground-truth labeling. Each graph receives a binary classification label indicating whether a crash occurs within the prediction horizon, and, for positive samples, a normalized spatial location. In addition, each temporal supernode anchor receives an auxiliary binary label indicating whether the crash falls within its aggregation interval. Figure 7.10 illustrates an example: the crash occurs at t_{crash}=t_{det} - 10 min, so the two most recent anchors receive a positive label and the remaining ones a negative label. This auxiliary supervision is

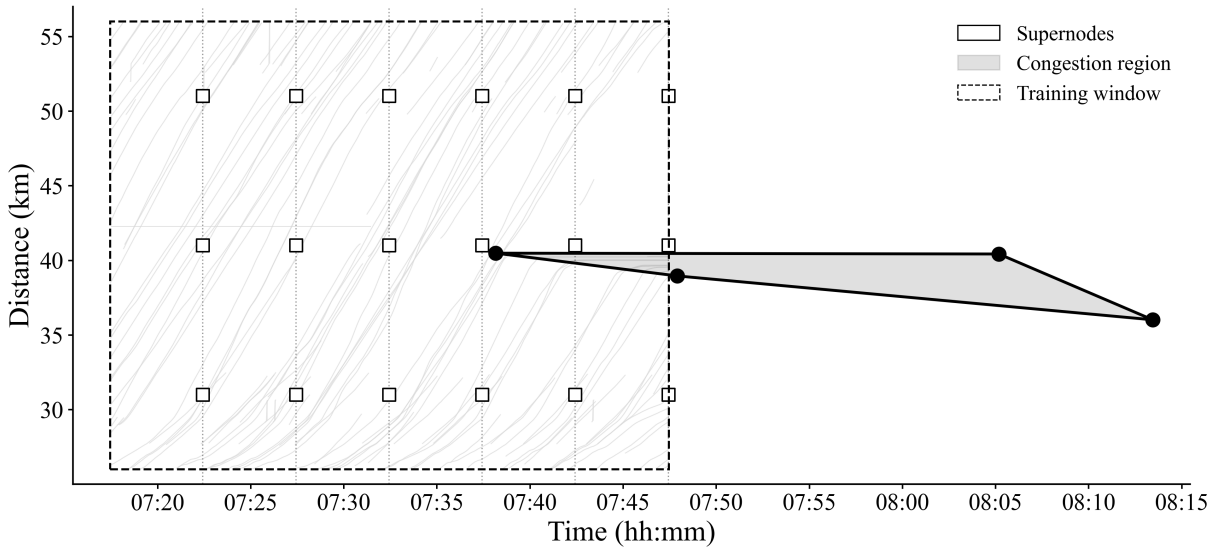


Figure 7.10: Ground-truth label construction.

exploited in the loss function (Section 7.3.3).

7.3.2 Model Architecture

The model operates on $\mathcal{G}$ as defined in Section 7.3.1. Type-specific three-layer MLPs project vehicle nodes ($\mathbf{x}_i \in \mathbb{R}^7$) and supernodes ($\tilde{\mathbf{x}}_{m,n} \in \mathbb{R}^4$) into a shared hidden space $\mathbb{R}^{f_{\text{hid}}}$ ($f_{\text{hid}} = 96$). Processing then proceeds through four stages – temporal encoding of vehicle trajectories, attention-based aggregation into supernodes, spatial message passing, and temporal encoding of supernode histories – before a Transformer-based output head produces predictions. Attention pooling transfers updated vehicle representations into supernodes via $\mathcal{E}_{\text{super}}$ after each of the first two stages. The overall information flow is shown in Figure 7.11; formal definitions of the standard components and the detailed node-level view are deferred to Section D.2. The remainder of this section describes the four design choices that depart from off-the-shelf building blocks: a partial causal convolution for sparse trajectories, a relation-fusion GATv2 block that treats the two vehicle-level edge types symmetrically, the attention-pooling scheduling, and a dual-head output with anchor-level deep supervision.

Partial causal convolution for sparse trajectories. Vehicle trajectory lengths T_j range from 1 to over 500 time steps and are heavily skewed toward short sequences. A standard causal TCN [166] would pad short sequences with zeros over long dilations, biasing the convolution toward padded positions. Following the partial convolution of Liu et al. [154],

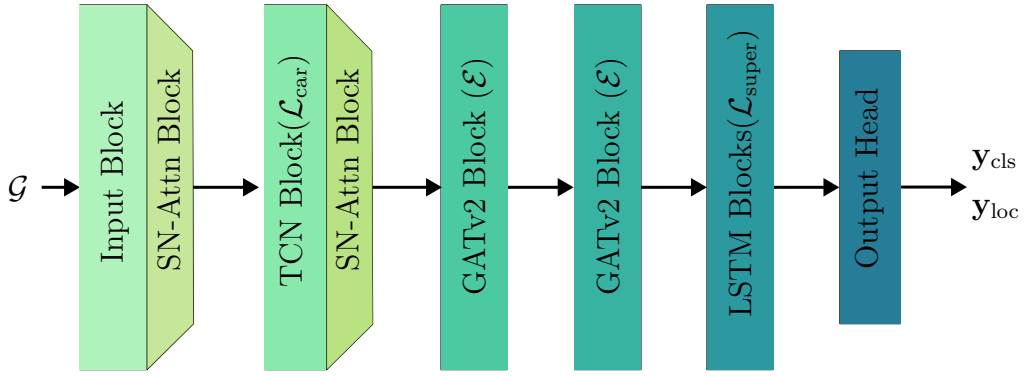


Figure 7.11: GNN model architecture overview [175].

each convolution is restricted to valid inputs and renormalized by the number of valid contributions:

$$\hat{\mathbf{x}}_t^{(\psi_j)} = \frac{1}{Z_t} \sum_{i=0}^{K_{\text{conv}}-1} \mathbf{W}_i \cdot \mathbf{x}_{t-\delta \cdot i}^{(\psi_j)} \cdot m_{t-\delta \cdot i}, \tag{7.7}$$

where $m_{t-\delta \cdot i} \in \{0, 1\}$ marks valid positions, $Z_t = \sum_i m_{t-\delta \cdot i}$, and outputs with $Z_t = 0$ are set to $\mathbf{0}$. Eight such layers with exponentially increasing dilations arranged in two residual groups yield a receptive field of 511 steps (17–128 min at 2–15 s sampling), covering the full $T = 30$ min window for all sampling intervals. The exact layer arrangement is given in Section D.2.

Relation-fusion GATv2 block. After temporal and attention encoding, spatial context is propagated using GATv2 [165] (formulation in Section 2.8.1). Rather than treating the three edge types identically, two GATv2 operators are applied in parallel over $\mathcal{E}_{\text{spatial}}$ and $\mathcal{E}_{\text{WMJ}}$ and their outputs fused via concatenation and a linear projection with LeakyReLU. A third GATv2 operator over $\mathcal{E}_{\text{super}}$ and a residual connection complete the block. This design keeps the two vehicle-level spatial relations on an equal footing while reserving the hierarchical aggregation path for supernode edges. Two blocks are stacked ($l = 1, 2$), extending each node's effective receptive field through multi-hop message exchange.

Two-pass attention pooling into supernodes. Single-head query-to-set attention [169] over the neighborhood defined by $\mathcal{E}_{\text{super}}$ injects vehicle information into each supernode twice: once after the trajectory TCN and once after the relation-fusion blocks. The first pass supplies the supernode with a temporally-encoded summary of its member vehicles before spatial propagation; the second pass updates it with post-propagation representations that already incorporate context from other edge types. Definition and batching are

Table 7.4: Graph-construction hyperparameters [175].

Parameter	Setting
Supernode time anchors $\tilde{\mathcal{T}}$	$\{\, t_{\text{det}} - 60\,\text{s} \cdot m \mid m \in \{0, 5, 10, 15, 20, 25\}\,\}$
Supernode location anchors $\tilde{\mathcal{D}}$	$d_{\text{up}} - \frac{1}{4}D,\ d_{\text{up}} - \frac{1}{2}D,\ d_{\text{up}} - \frac{3}{4}D$
Aggregation window $\mathcal{W}_{mn}$	$\Delta t_{\text{agg}} = 20\,\text{s};\ \Delta d_{\text{agg}} = 7.5\,\text{km}$
Vehicle spatial edges $\mathcal{E}_{\text{spatial}}$	$\Delta d_{\text{spa}} = 5\,\text{km};\ \Delta t_{\text{spa}} = 60\,\text{s};\ \text{causal}\ (t_j \le t_i)$
WMJ propagation edges $\mathcal{E}_{\text{WMJ}}$	$v_g = -15\,\text{km}\,\text{h}^{-1};\ \Delta t_{\text{WMJ}} = 300\,\text{s};\ \Delta d_{\text{WMJ}} = 5\,\text{km};$ $\Delta t_{\text{tol}} = 120\,\text{s}$
Multi-hop expansion	$K_{\text{hop}} = 2$ rounds

in Section D.2.

Dual-head output with anchor-level deep supervision. At every temporal anchor $\tilde{t}_m$, two instances with shared architecture but independent weights produce a crash probability ${y^{\text{pred}}}_{\text{cls},m} \in (0, 1)$ and a normalized crash location ${y^{\text{pred}}}_{\text{loc},m} \in (0, 1)$. Each instance applies a four-layer Transformer encoder [168] with eight attention heads and feed-forward width $2f_{\text{hid}}$ to the spatial slice of supernode embeddings $\tilde{\mathbf{X}}_m$, collapses it via a learnable query with scaled dot-product attention, and maps the result through a three-layer MLP with sigmoid activation. Deep supervision is applied at all six anchors during training – each anchor contributing a class-weighted term to the loss (Section 7.3.3) – while at inference only the prediction at $\tilde{t}_0$ is retained, consistent with the streaming protocol in Section 3.7. A two-layer unidirectional long short-term memory (LSTM) [167] applied in parallel across spatial anchors precedes the output head and encodes supernode histories; see Section D.2.

7.3.3 Training Configuration

All models are trained with the Adam optimizer (learning rate 1×10^{-4}), a StepLR scheduler ($\gamma_{\text{lr}} = 0.6$ every 3 epochs), gradient accumulation of 4, and step-based early stopping (every 200 steps on validation classification loss). Table 7.4 collects all graph-construction hyperparameters.

Initialization Linear layers with LeakyReLU activations use Kaiming uniform initialization [212]; all others use Xavier uniform [213]. Biases are initialized to zero; normalization layers (LayerNorm, GraphNorm) to unit scale and zero bias. LSTM input connections use Xavier uniform [213], recurrent connections use orthogonal initialization.

Dynamic Batching Node counts per graph range from 587 to 39 751, making a fixed batch size infeasible. Each batch is constrained to a maximum total node count of $B_{\text{cap}} = 500{,}000$, packed via a Best-Fit Decreasing variant with minimum-count tie-breaking [215] (Algorithm 3).

Regularization Despite containing only 1.5 M parameters (roughly one ninth of the CNN), the GNN is prone to overfitting. Three complementary strategies are applied: (i) data-level augmentation through trajectory downsampling, point subsampling, and feature perturbation (Section 3.7); (ii) graph-level stochastic regularization via edge dropout (30 %), edge rewiring (10 %), and feature masking (30 %); and (iii) softmax dropout (30 %) in all attention layers.

Loss Function The model outputs logits $y_{\text{cls},m} \in \mathbb{R}$ at six temporal anchors $\tilde{t}_m$ ($m = 0, \ldots, 5$). A weighted binary classification loss is applied with importance weights $w_m \in \{1.3, 1.2, 1.1, 1.0, 0.9, 0.8\}$, normalized as $\bar{w}_m = w_m / \sum_j w_j$:

$$\ell_{\text{cls}} = \frac{1}{B} \sum_{b=1}^{B} \sum_{m=0}^{5} \bar{w}_m \cdot \ell_{b,m}^{\text{cls}},$$

where $\ell_{b,m}^{\text{cls}}$ denotes the class-weighted BCE loss at anchor m for sample b. For crash localization, the MAE on positive samples is used.

7.3.4 Ablation Analysis

Module-wise ablation on the validation set (Table 7.5) removes one block at a time from the full pipeline with all other components held fixed. The TCN and the supernode-attention block incur the largest drops, confirming that trajectory-level temporal encoding and vehicle-to-supernode aggregation form the backbone of the pipeline. Removing any single GNN edge type also causes noticeable degradation, indicating that each captures a distinct aspect of the spatiotemporal structure. Supplementary ablations on input features, temporal-anchor placement, single- vs. multi-output supervision, and the parallel vs. sequential supernode-edge design are reported in Section D.4; the merged (parallel) design was selected on that basis.

Table 7.5: Module-wise ablation on the validation set (threshold = 0.5) [175].

Variant	**Val Loss**	**Precision**	**Recall**	**F1**
Full GNN pipeline	$\mathbf{0.3041_{\pm 0.0062}}$	$86.0\%_{\pm 1.9}$	$\mathbf{92.2\%_{\pm 1.8}}$	$\mathbf{88.9\%_{\pm 0.3}}$
Without SN-Attn Block	$0.3602_{\pm 0.0070}$	$84.8\%_{\pm 1.4}$	$90.4\%_{\pm 0.7}$	$83.3\%_{\pm 0.4}$
Without TCN Block	$0.3848_{\pm 0.0040}$	$79.5\%_{\pm 2.0}$	$83.5\%_{\pm 2.8}$	$81.4\%_{\pm 0.5}$
GATv2 Blocks				
Without $\mathcal{E}_{\text{spatial}}$	$0.3442_{\pm 0.0090}$	$85.6\%_{\pm 0.4}$	$88.2\%_{\pm 2.1}$	$84.3\%_{\pm 0.7}$
Without $\mathcal{E}_{\text{WMJ}}$	$0.3542_{\pm 0.0075}$	$83.7\%_{\pm 0.7}$	$86.8\%_{\pm 1.6}$	$83.2\%_{\pm 0.6}$
Without $\mathcal{E}_{\text{super}}$	$0.3665_{\pm 0.0062}$	$84.1\%_{\pm 0.5}$	$\underline{91.2\%_{\pm 1.5}}$	$82.7\%_{\pm 0.0}$
Without LSTM Block	$\underline{0.3346_{\pm 0.0058}}$	$\mathbf{86.4\%_{\pm 0.6}}$	$88.2\%_{\pm 1.8}$	$\underline{84.8\%_{\pm 0.2}}$
Without Transformer Head	$0.3456_{\pm 0.0044}$	$\underline{86.3\%_{\pm 0.4}}$	$89.1\%_{\pm 1.3}$	$84.5\%_{\pm 0.5}$

8 Comparison of Crash Detection Approaches

8.1 Summary of Model Selection

This chapter provides a systematic comparison of all four crash detection approaches developed in this work: the rule-based method (Section 6.4) and three deep learning architectures selected from the ablation studies in Sections 7.2.4 and 7.3.4. The ablation results reported there are based on a subset of the balanced dataset and are therefore preliminary. The three selected deep learning configurations are:

- **GNN Full Pipeline** (*GNN*): the complete architecture described in Section 7.3.2, including TCN block, SN-Attn block, GATv2 layers, LSTM, and Transformer head, trained with multi-output supervision and uniformly distributed temporal anchors.
- **CNN with PSM, velocity-only input** (*CNN (PSM)*): variant (b) in Table 7.3, which applies PSM-based input completion but omits the phase channel found to introduce noise.
- **CNN with partial convolutions, no PSM** (*CNN (PConv)*): variant (e) in Table 7.3, which processes the raw sparse input directly via partial convolutions without interpolation.

The **rule-based approach** serves as a non-learning baseline. Unlike the deep learning models, it applies detection rules grounded in 3PTT and requires only parameter calibration on a small labeled validation subset rather than end-to-end training on the full labeled dataset. All four methods are evaluated on the same held-out test set (Table 3.9b); the deep learning models were trained on the complete training set summarized in Table 3.9a. All models are benchmarked against the TPEG messages generated by the TMC system, which is the current state-of-the-art in RTTI and served as reference for the annotated ground truth (GT) set. To contextualize the proposed approaches, the TPEG system's own timeliness and localization accuracy are evaluated by matching the "TPEG – Raw Event" polygons

in the annotation set against the corresponding GT crash polygons ("TPEG – Mega jam"); this establishes a production-system reference for comparison with all four approaches (Section 8.3.1).

8.2 Prediction Workflow and Comparability of Approaches

The deep learning (DL) models are trained and evaluated on individual spatiotemporal windows $\mathcal{W}_{D,T}$, whereas the rule-based crash detection model (RB) detector operates directly on full-sample traffic records of arbitrary spatial and temporal extent. A direct comparison therefore requires a prediction pipeline that applies the DL models to the same unbounded observations as the RB baseline and exposes them to the long and complex congestion patterns encountered in realistic operation.

To bridge this gap, a sliding-window inference scheme is used. Each freeway record is tiled by consecutive windows $\mathcal{W}_{D,T}$ of $D = 30\,\text{km} \times T = 30\,\text{min}$ at a 10 s stride (Step 1, window sampling); for every window, an approach-specific input representation is constructed: a unified trajectory graph for the GNN, and a dense or sparse tensor pair for the two CNN variants (Step 2, input generation); and each input is passed through the corresponding classifier and localization head (Step 3, inference). The per-window outputs are then aggregated into record-level predictions that can be evaluated against the same ground truth used for the RB model. An example of the complete pipeline on an A6 crash case of October 10, 2024, including visualizations of the four input representations, is provided in Section E.1.

Training and evaluation protocol. All deep learning models are trained using average precision (AP) as the primary validation metric rather than F_1 at a fixed decision threshold. This decouples representation learning from threshold selection and offers two advantages. First, AP-based optimization encourages well-separated score distributions across the *entire* range of operating points, rather than concentrating discriminative power around a single threshold that may not transfer to unseen data. Second, the learned features are not coupled to a particular threshold: directly optimizing F_1 would make subsequent changes to the operating point, as is common in deployment when precision-recall trade-offs must be adjusted, degrade performance. After training, the best checkpoint is selected by validation AP, and a model-specific decision threshold is identified post-hoc by sweeping thresholds on the validation set and choosing the F_1-maximizing value. This two-stage protocol generalizes across operating points while still providing a concrete threshold optimized for balanced detection. All test-set results below use these per-model thresholds.

8.3 Quantitative Results

This section presents the final test-set results for all four approaches on both the classification and localization tasks. Raw metrics and rankings are reported here; mechanistic interpretation and behavioral characterization of each model are deferred to Section 8.5.1.

8.3.1 Classification Results

For each deep-learning model, the decision threshold maximizing the validation F_1 score was selected by grid search over the score range (Figure 8.1). All deep-learning predictions are smoothed with a causal exponential-decay filter spanning the preceding five minutes ($\tau = 2\,\text{min}$; see Section D.5). The rule-based approach is evaluated at its calibrated threshold $\tau = 0.5$ from the Accuracy operating point (Section 6.3).

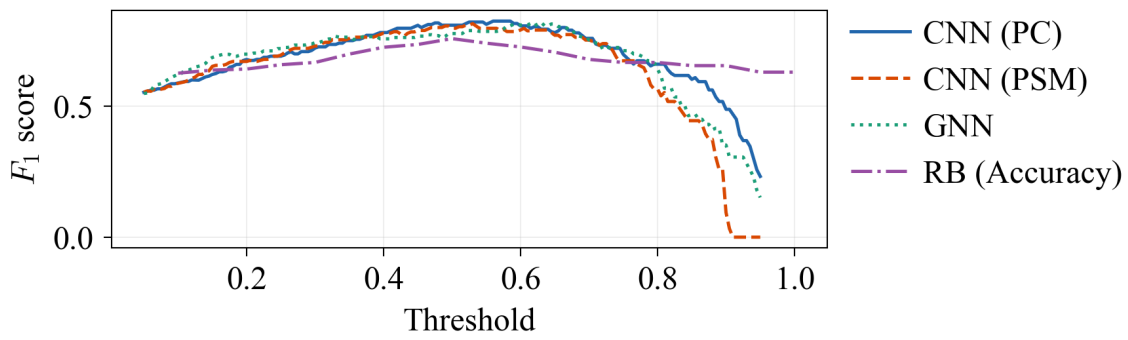


Figure 8.1: Validation F_1 score versus decision threshold τ for each detector. The rule-based curve corresponds to the Accuracy configuration

All three deep-learning models share an inverted-U response, peaking near $F_1 \approx 0.82$ at $\tau \approx 0.5$ and collapsing above $\tau = 0.8$ as positive predictions are increasingly suppressed. The rule-based curve is qualitatively flatter: F_1 stays above 0.6 across the full $\tau \in [0.1, 1.0]$ range and above 0.65 for $\tau \geq 0.2$. This stability follows from the bounded structure of the rule-based score (a saturated weighted sum of positive evidence and plausibility penalties) rather than from a sigmoid posterior, so extreme thresholds shift the operating point gradually rather than collapsing detection to zero. The deep-learning curves' sharp right-side decay, by contrast, reflects the long thin tail of their sigmoid output distributions: above some calibration-dependent threshold, almost no scores survive and recall vanishes faster than precision can compensate. For the CNN (PSM), this happens at $\tau = 0.92$.

The thresholds chosen for test-set evaluation and the resulting metrics are summarized in Table 8.2.

Table 8.1: Updated confusion matrices for all approaches on the test set (aligned to $GT = 66$). Rows denote predicted labels (p' positive, n' negative) and columns denote actual labels (p positive, n negative).

	Rule-based			**CNN (PConv)**			**CNN (PSM)**			**GNN**		
Pred.\Actual	p	n	Σ	p	n	Σ	p	n	Σ	p	n	Σ
p'	50	16	66	60	17	77	61	21	82	55	12	67
n'	16	1789	1805	6	95	101	5	91	96	11	100	111
Σ	66	1805	1871	66	112	178	66	112	178	66	112	178

Differences in evaluation granularity. True-negative counts are not directly comparable across detector families because they evaluate on different timelines. The deep-learning models operate on fixed-size detection windows from a 40 km × 5 h area of interest, with one decision per window, totaling 112 non-crash and 66 crash samples. The rule-based approach evaluates every detected congestion pattern (jam phase with ≥ 3 qualified trajectories), yielding 1,871 evaluable phases, 66 of them crashes. The larger rule-based TN count (1,805 vs. 112) therefore reflects finer evaluation granularity rather than superior specificity. The joint matrix reports the rule-based Accuracy profile; remaining profiles and extended metrics are in Table E.2.

Metric ranking. Table 8.2 summarizes the test-set metrics. Among the deep learning models, CNN (PConv) attains the best overall F_1 (83.3 %); the GNN leads on precision (82.3 %), accuracy (87.2 %), and both threshold-independent scores (AUC = 0.930, AP = 0.886); CNN (PSM) leads on recall (91.8 %). The GNN's recall (83.6 %) is the lowest of the three deep learning models. The rule-based approach achieves $F_1 = 75.0$ % with precision 76.2 % and recall 73.8 %. Its F_1 and recall fall below all three deep learning models; its precision sits between CNN (PSM) (74.7 %) and CNN (PConv) (77.5 %).

The fixed-threshold metrics above summarize detection performance at a single operating point per model. To complement this view, Figure 8.2 presents threshold-independent receiver operating characteristic (ROC) and Precision-Recall curves on the test set for the three deep learning models, alongside the rule-based Accuracy operating point swept across its own score range.

ROC curve analysis. All four detectors clear the chance diagonal by a wide margin (Figure 8.2a). The GNN attains the highest $\mathrm{AUC}_{\mathrm{ROC}} = 0.914$; CNN (PConv) at 0.899 and CNN (PSM) at 0.894 are essentially indistinguishable. The rule-based detector trails at $\mathrm{AUC}_{\mathrm{ROC}} = 0.806$, but its curve shape is qualitatively different: false positive rate (FPR)

Table 8.2: Test-set classification results of all four approaches. Best thresholds are determined on the validation set for deep learning models; the rule-based method uses its calibrated parameter set directly. The best and second-best scores among deep learning approaches are highlighted in **bold** and underline, respectively.

Approach	Thr.	F1	Prec.	Rec.	Acc.	AUC	AP
Rule-based	0.50	75.8%	75.8%	75.8%	75.0%	77.0%	81.3%
GNN	0.63	82.9%	**82.3%**	83.6%	**87.2%**	**93.0%**	**88.6%**
CNN (PSM)	0.53	82.4%	74.7%	**91.8%**	85.4%	89.6%	84.2%
CNN (PConv)	0.56	**83.3%**	77.5%	90.2%	86.6%	91.4%	86.0%

is bounded above at approximately 0.022 because the detector's architectural plausibility filters reject most non-crash phases before scoring, so lowering the threshold cannot admit further false positives. The full-range AUC integrates a horizontal extrapolation across operating points the detector cannot reach, which accounts for most of the gap to the learned models. Within the shared low-FPR envelope (FPR $\lesssim 0.05$), where deployment-realistic thresholds live, the rule-based curve tracks or sits above the learned curves.

Precision-Recall curve analysis. The GNN also leads on the precision-recall (PR) plane (Figure 8.2b, AP = 0.875), holding the highest average precision across operating points above recall ≈ 0.35. CNN (PConv) (AP = 0.849) and CNN (PSM) (AP = 0.845) track each other closely. The rule-based detector reaches AP = 0.846, competitive with both CNN variants despite the restricted shape of its curve: precision holds near 0.97 up to recall ≈ 0.81 — substantially higher than any learned detector in that range — before collapsing once the same architectural prerequisites cap further recall gains. AP is less sensitive to envelope width than ROC AUC, narrowing the gap to the learned detectors from ~ 0.11 to ~ 0.03. All three learned curves show minor step-wise non-monotonicity; the GNN sits above the CNNs across most of the plane.

Pareto Front Figure 8.3 shows the achievable trade-off between F_1 and detection latency. The deep-learning detectors reach lower minimum delays than the rule-based detector, which is bounded below at ~ 4.7 min by its architectural plausibility filters; the same filters cap recall and therefore F_1 at the rule-based detector's F_1-optimal trial. Within the operating envelope reachable to all four approaches, the Pareto fronts converge, with the GNN dominating at low-to-mid delays and CNN (PConv) at higher delays.

Three deployment-relevant regimes emerge. For the practically interesting target of high

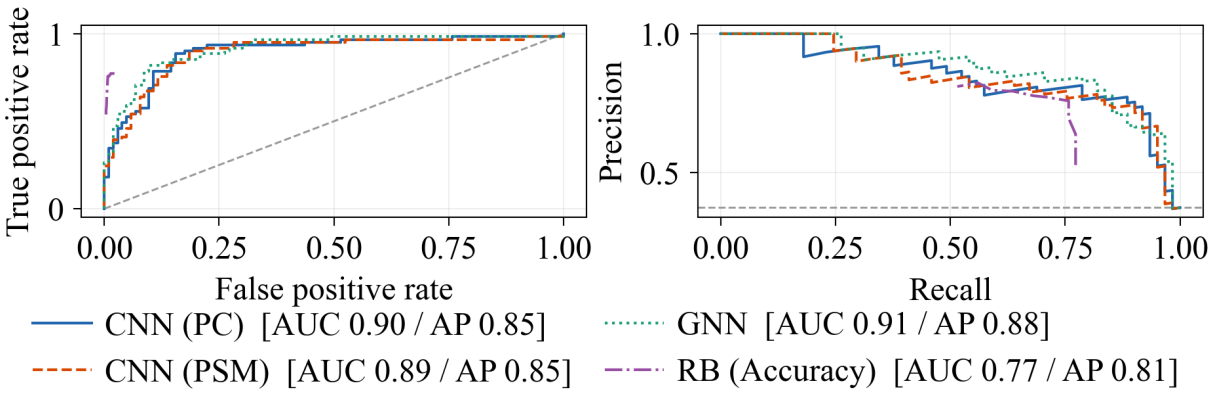


Figure 8.2: Threshold-independent evaluation of crash detection models on the test set. (a) ROC curves with corresponding AUC values. (b) PR curves with corresponding AUC values.

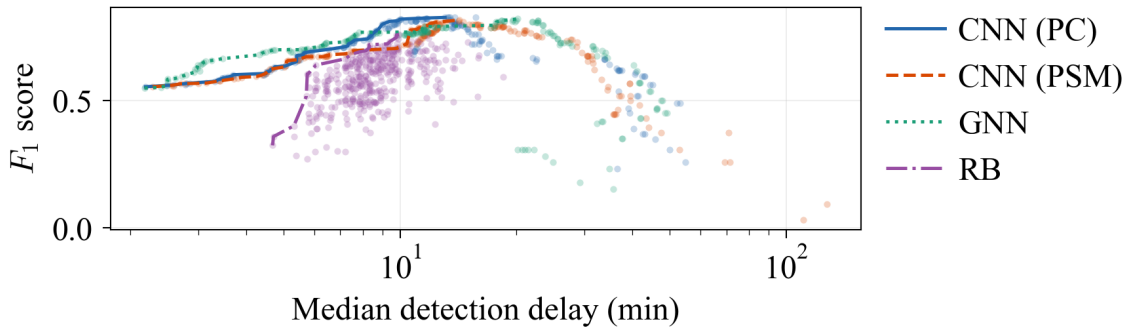


Figure 8.3: Pareto front of F_1 vs. median detection delay across the three deep-learning detectors (per-threshold sweeps on a single trained model each) and the rule-based detector's full calibration design space ($n = 387$ Optuna trials, each with its own calibrated parameter set).

accuracy at low latency ($F_1 \geq 0.7$ and median delay ≤ 7 min), the GNN is the only detector to deliver a viable operating point; under a strict ≤ 5 min latency budget it is again the only detector to reach competitive F_1. For maximum F_1 regardless of delay, CNN (PConv) wins overall with a peak of 0.824, which the GNN matches only at roughly double the delay. The rule-based detector's strongest low-delay trial is competitive with CNN (PSM) but trails the GNN by approximately 0.10 on F_1. A complete enumeration of the top ten Pareto-optimal operating points per detector is given in Table E.1.

Onset and clearance deviation. The Pareto analysis above sweeps the score threshold to characterize each detector's achievable envelope. To complement this view, onset and clearance are now examined at the per-model F_1-maximizing threshold (Table 8.2), using the temporal metrics from Section 3.7: crash onset deviation Δt_{on} (positive values indicate delayed detection) and clearance deviation Δt_{off} (negative values indicate premature clearance). For the rule-based method, onset is measured from the first positive detection after GT crash time and clearance from the last positive detection before GT clearance time. Statistics are reported in Tables 8.3 and 8.4, distributions in Figure 8.4.

Onset and clearance pose qualitatively different detection problems. Onset is essentially a threshold-crossing event: once sufficient evidence accumulates, all four approaches respond within a narrow band, and mean delays span only about seven minutes across the field. The defining result is the rule-based method's exceptional dispersion of $\sigma = 7.9$ min, roughly half that of either CNN variant and less than a quarter of the GNN's. This tightness follows directly from the staged scoring, whose activation is bounded by the evidence-accumulation windows; the same architectural mechanism that imposed the $\sim$4.7 min minimum-delay floor on the Pareto front structurally suppresses large deviations. CNN (PConv) edges out the rule-based mean and median by under two minutes but at almost double the variance, while the GNN's elevated mean reflects a heavy tail rather than a systematic offset, its median remaining close to the others. The TPEG baseline places between CNN (PSM) and the GNN.

For clearance the ranking inverts, and the spread of mean deviations exceeds thirty minutes across the field. CNN (PConv) is the only approach with a positive mean and exhibits a dispersion more than three times tighter than any other. The remaining three share a bimodal character, with medians close to zero but means pulled tens of minutes negative by a tail of extreme premature terminations reaching beyond eight hours in the worst cases. Typical-case clearance timing is therefore strong across all four approaches; the failure mode is a low-probability, high-magnitude early-clear event that only CNN (PConv) reliably avoids.

Table 8.3: Statistics of crash clearance detection delay Δt_{off} (in minutes). Negative values indicate premature clearance relative to ground truth. The TPEG reference ($n = 224$) is evaluated on the full annotation set; all other approaches on the held-out test set. Best and second-best values (closest to zero or smallest variance) are highlighted in **bold** and underline, respectively.

Approach	mean(Δt_{off})	median(Δt_{off})	std(Δt_{off})
TPEG (reference)	−28.44	−11.73	64.51
Rule-based	−26.8	−2.3	85.8
GNN	−22.10	**0.00**	89.40
CNN (PSM)	<u>−17.80</u>	<u>0.30</u>	<u>82.50</u>
CNN (PConv)	**+4.50**	2.00	**24.40**

Detector selection. The Pareto front and the onset/clearance statistics together identify distinct operating regimes rather than a single dominant detector. The GNN is the only approach satisfying tight latency targets, yet carries the heaviest onset tail and shares the bimodal clearance failure mode with CNN (PSM) and the rule-based detector. CNN (PConv) achieves the highest absolute F_1 at moderate delay and is uniquely reliable on clearance, but is not Pareto-optimal at low delay budgets. The rule-based detector offers the most predictable onset variance and exposes a structurally different operating-point manifold, namely a family of calibrated parameter sets rather than a single trained model with a sweepable threshold; where onset-time predictability or interpretable calibration is the operational priority, the rule-based detector is the appropriate choice despite trailing on peak F_1. The selection is therefore governed by which margin, low-delay F_1, peak F_1, onset variance, or clearance reliability, matters most for the deployment.

8.3.2 Regression Results

Crash location regression accuracy is evaluated on the test set using the MAE and the median absolute error, both reported in meters. Each crash event yields multiple spatiotemporal observation windows $\mathcal{W}_{D,T}$, so results are reported under three aggregation modes: *First* (the detection window that initially triggers a crash alert), *EvMean* (event-level mean across every window that fires), and *Best* (the single window with the smallest error per event).

Figure 8.5 and Table 8.5 reveal a clear separation among the deep learning models. CNN (PConv) is consistently the weakest, with First-mode errors roughly 1.3–1.4× larger

Table 8.4: Statistics of crash onset detection delay Δt_{on} (in minutes). Positive values indicate delayed detection relative to ground truth. The TPEG reference ($n = 224$) is evaluated on the full annotation set; all other approaches on the held-out test set. Best and second-best values are highlighted in **bold** and underline, respectively.

Approach	mean(Δt_{on})	median(Δt_{on})	std(Δt_{on})
TPEG (reference)	14.36	11.22	20.53
Rule-based	**12.30**	9.5	**7.9**
GNN	19.80	11.67	31.90
CNN (PSM)	14.00	11.00	13.50
CNN (PConv)	12.90	**8.30**	13.70

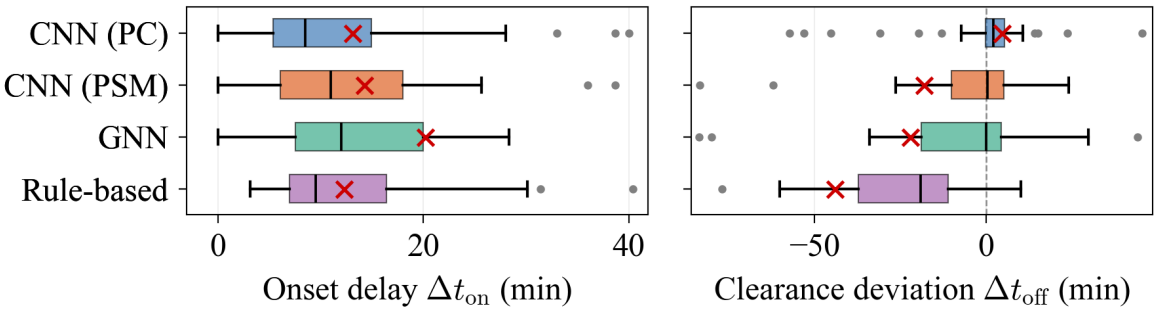


Figure 8.4: Distribution of crash onset delay Δt_{on} (left) and clearance deviation Δt_{off} (right) on the test set. The TPEG reference (dashed line) is shown for comparison. Positive onset values indicate delayed detection; negative clearance values indicate premature clearance. Boxes span the interquartile range with the median marked inside; whiskers extend to $1.5 \times$ IQR; points beyond are outliers. Numerical summaries are reported in Tables 8.3 and 8.4.

Table 8.5: Crash location regression errors on the test set (meters), stratified by evaluation mode. *First*: initial detection window; *EvMean*: event-level mean over all triggered windows; *Best*: lowest-error window per event. The TPEG reference reports the centroid deviation of matched Raw Event polygons against GT Mega Jam annotations ($n = 224$). Best and second-best values per column are marked in **bold** and underline.

Model	N	First		EvMean		Best	
		Mean	Median	Mean	Median	Mean	Median
TPEG (ref.)	224	—		2608.6	2393.2	—	
CNN (PC)	66	7607.9	5668.0	6960.5	5700.8	1748.7	1494.9
CNN (PSM)	66	5629.2	3335.9	5957.4	4851.3	1543.9	747.1
GNN	66	5775.7	2368.4	5880.3	4858.8	1872.2	996.1
RB (seed)	66	660.4	426.3	**660.4**	**426.3**	660.4	426.3
RB (DS front)	66	**587.3**	**255.8**	993.6	698.6	**404.7**	**61.5**

than those of CNN (PSM) and the GNN. Its pronounced First-to-Best improvement nonetheless indicates that the initial detection windows are spatially inaccurate while later windows recover a reasonable estimate, pointing to a temporal alignment issue rather than a fundamental representational deficit. CNN (PSM) and the GNN perform similarly in mean MAE across all three modes, yet the GNN attains a noticeably lower First-mode median, indicating a more concentrated error distribution with fewer moderate outliers on early detections. Under the Best-window criterion all three models improve substantially, with CNN (PSM) achieving the lowest mean and median.

The TPEG reference, evaluated by comparing its event centroid against the GT crash polygon centroid, outperforms all deep learning models but remains inferior to the rule-based estimates. This gap is partly structural: TPEG has access to the reported incident position, whereas the FCD-based approaches must infer the crash location from the observed congestion pattern alone. Both rule-based estimates nevertheless surpass TPEG by roughly a factor of four, reaching sub-700 m First-mode errors against the TPEG reference of about 2.6 km, with the downstream-front estimate slightly ahead of the seed position.

8.3.3 Inference Efficiency

Table 8.6 summarizes model size, average inference time, and throughput. The four detectors do not share a native unit of work, so a brief note on the measurement protocol is necessary before the comparison.

The deep-learning models are window-local: each forward pass consumes one $\mathcal{W}_{D,T}$

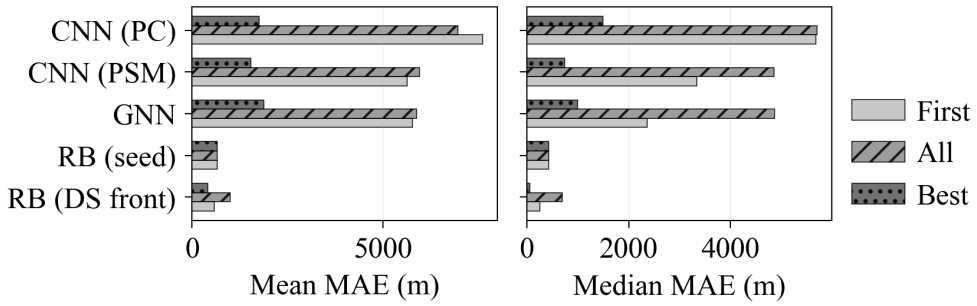


Figure 8.5: Mean and median MAE (in meters) for crash location regression, grouped by evaluation mode.

observation window in isolation, with no state carried across windows. Reported times correspond to a single GPU forward pass on a pre-assembled tensor or graph, averaged over multiple runs, with data loading and post-processing excluded. The rule-based method is not window-local; it operates on traffic-phase state built over an entire full-day FCD file, with S and J phases clustered across the full recording before any detection rule is evaluated. Clustering is performed offline as a cached preprocessing step in the current implementation, with no incremental variant in place. The reported timing therefore isolates rule evaluation, comprising feature assembly, scoring, gating, and plausibility filtering, against already-clustered state, averaged over five representative test datasets with three runs each. Clustering and data loading are excluded, matching the deep-learning protocol. To place the rule-based number on the same axis as the window-local models, total rule-evaluation wall time on a full-day file is divided by the number of non-overlapping 30 min windows the file spans. Runs are serial and CPU-bound.

Among the deep-learning models, parameter count is a weak predictor of inference cost. The CNN variants carry roughly an order of magnitude more parameters than the GNN, yet the GNN shows the highest relative timing variance, driven by variable graph sizes rather than by forward-pass arithmetic. CNN (PSM) is the slowest detector overall, with the PSM preprocessing stage accounting for roughly two-thirds of its total time; the CNN backbone alone is faster than the GNN despite carrying about eight times the parameters. CNN (PConv) avoids the PSM stage entirely and runs at less than half the total cost of CNN (PSM).

The rule-based method's amortized cost is lower than any deep-learning forward pass, but the comparison is asymmetric: this figure reflects the cost of scoring already-clustered state, not end-to-end processing from raw FCD to detection. Within that scope, the rule-based method operates on a compact per-phase feature vector (trajectory counts, front velocity

Table 8.6: Model size, average inference time, and throughput of the selected approaches. Deep-learning times are per 30 min window on GPU, covering the model forward pass only. The rule-based time is amortized per 30 min window from a full-day CPU run and covers rule evaluation against pre-clustered phase state; clustering and data loading are excluded (see text). For CNN (PSM), the reported time includes both PSM and CNN computation. Rule-based parameters refer to the count of optimized rule weights, thresholds, and gate settings rather than trained weights.

Model	Parameters	Avg. Inference Time [ms]	Throughput [FPS]
Rule-based	32	$24.87_{\pm 4.33}$	40.2
GNN	1,570,774	$73.4_{\pm 2.82}$	13.6
CNN (PSM): total	12,741,672	$113.3_{\pm 0.47}$	8.8
– PSM	0	$76.0_{\pm 0.0}$	13.2
– CNN	12,741,672	$37.3_{\pm 0.47}$	26.8
CNN (PConv)	11,519,078	$52.0_{\pm 1.25}$	19.2

and R^2, blockage flags, cluster-dominance score) rather than on a full space–time grid or graph, so per-window work is bounded regardless of input resolution. The comparatively large standard deviation reflects day-to-day variation in trajectory density and in the number of active J-phases under evaluation, not run-to-run noise.

8.4 Case Studies

This section complements the aggregate metrics reported in Section 8.3 with qualitative case studies, illustrating how the four approaches behave on real freeway data. A multi-sample overview is presented first (Section 8.4.1), followed by detailed probability-curve analyses of selected cases that expose the strengths and failure modes of each approach (Section 8.4.2). The detailed step-by-step prediction workflow for the deep learning models (window sampling, input generation, inference) is provided in Section E.1.

8.4.1 Multi-Sample Overview

Figure 8.6 shows nine full-day samples from German freeways. Each sample contains multiple congestion patterns and at least one annotated crash; across the nine samples, eleven ground-truth crashes are observed. The upper panel of each tile shows the TSD with

the three traffic phases (F, S, J) and the annotated GT post-crash congestion regions. The lower panel overlays the crash probability curves of all four detectors. Figure 8.7 shows three full-day samples from UK freeways. The samples show less recurring congestion compared to German freeways. All models detect the shown crashes.

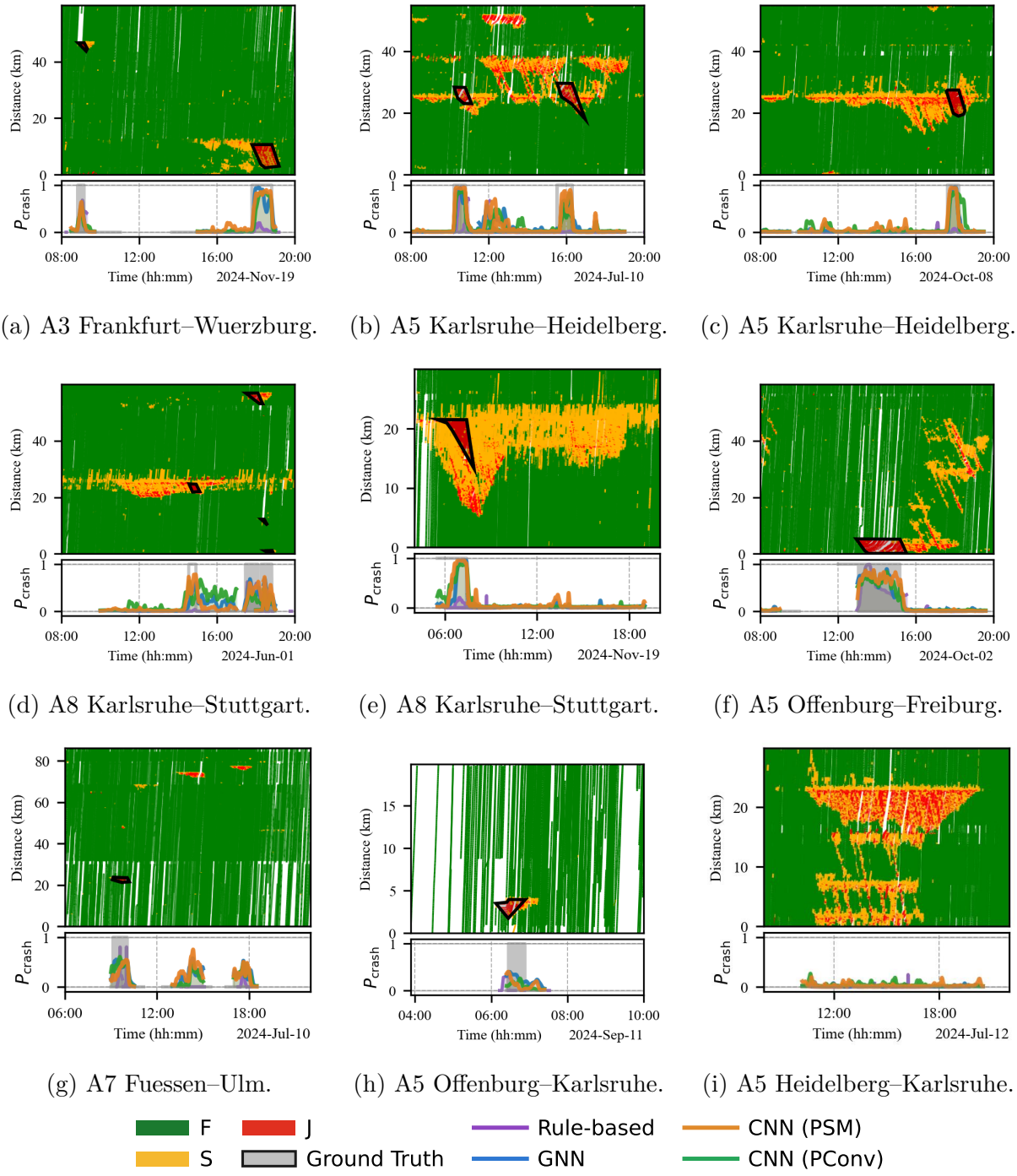
Two observations illustrate characteristic behavior across the grid. In Figure 8.6b, a congestion pattern around 12:00 near the 30 km mark exhibits clear Mega jam characteristics — sharp onset, strong upstream propagation, and a near-zero downstream-front velocity — all four models respond with elevated probabilities and the event is counted as a FP. No crash is annotated at this location in the dataset, but an unreported incident cannot be ruled out. In Figure 8.6e, by contrast, the congestion forms gradually and all models react only after a recognizable pattern has developed, resulting in a delayed detection. This reflects an intrinsic limitation of congestion-based crash detection: the signal only becomes discriminative once the congestion has reached a sufficient spatiotemporal extent. Figure 8.6i shows extended congestion with multiple WMJs. All models only show low probabilities, correctly distinguishing WMJs from Mega jams.

8.4.2 Detailed Probability-Curve Analysis

Four samples are examined in detail below. Each combines a challenging congestion configuration with at least one true positive (TP), FP, or false negative (FN) outcome, and together they cover the full range of model behavior observed in the test set.

A7 Fuessen–Ulm, 2024-07-10 (one TP, two FPs). Figure 8.8a shows three congestion events. All four models produce moderate probabilities for each event, but cross their respective decision thresholds at different points. Only CNN (PConv) triggers on the 14:00 pattern producing a FP; for the 17:30 pattern, only the rule-based approach produces a FP. Both congestions share similarities with post-crash congestion — high ϕ_{cd}, $v_g \approx 0$, and a large pre-jam free-flow area r_f — which makes them hard to distinguish from crash-induced jams. The crash at 09:00 is detected only by the rule-based approach and CNN (PSM); the other two models show elevated probabilities but do not exceed their thresholds.

A5 Karlsruhe–Heidelberg, 2024-07-10 (two TPs, one ambiguous FP). Figure 8.8d contains two correctly detected crashes at 10:30 and 16:00 — all four models cross their thresholds. A third event at 12:00, detected by the rule-based approach and CNN (PConv), is annotated as non-crash and thus counted as FP. As in the previous case, its feature signature (high ϕ_{cd}, $v_g \approx 0$, high r_f) is crash-like, suggesting either an unreported incident or a genuine discrimination limit of the current feature set.

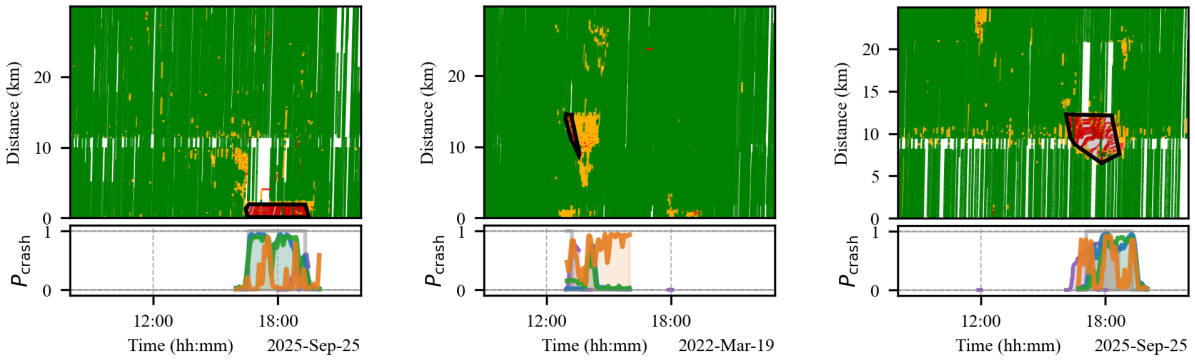


(a) A3 Frankfurt–Wuerzburg. (b) A5 Karlsruhe–Heidelberg. (c) A5 Karlsruhe–Heidelberg.

(d) A8 Karlsruhe–Stuttgart. (e) A8 Karlsruhe–Stuttgart. (f) A5 Offenburg–Freiburg.

(g) A7 Fuessen–Ulm. (h) A5 Offenburg–Karlsruhe. (i) A5 Heidelberg–Karlsruhe.

Figure 8.6: Crash detection results across nine German freeway corridor samples. Annotated GT crashes are shaded in gray. Upper panels show phase maps with F, S, and J phases; lower panels show crash probability curves from the four approaches.



(a) M11 Woodford Cambridge. (b) M25 Slough Bricketwood. (c) M3 Shepperton Canberley.

Figure 8.7: Crash detection results across three UK freeway corridor samples. Annotated GT crashes are shaded in gray. Upper panels show phase maps with F, S, and J phases; lower panels show crash probability curves from the four approaches.

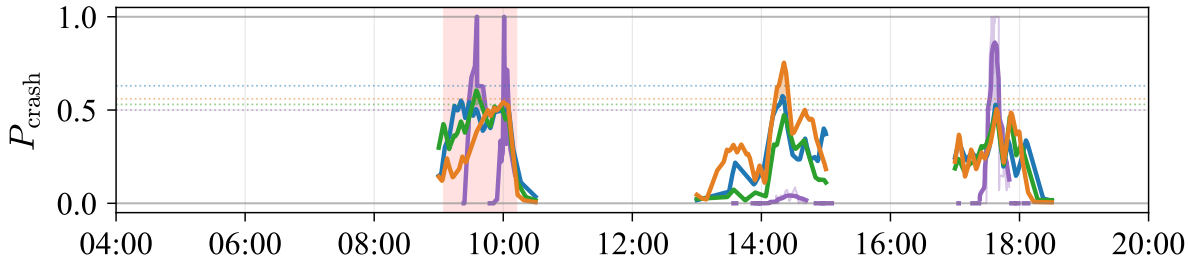
A5 Offenburg–Karlsruhe, 2024-09-11 (one FN). Figure 8.8b shows an annotated crash at 06:10 that is only detected by the rule-based approach, despite a transient FB and a high pre-jam free-flow area r_f. The probability curves of the remaining detectors remain well below their thresholds throughout the event. The shortness of the blockage and the rapid dissipation of the resulting queue appear to prevent the models from correct identification.

A5 Heidelberg–Karlsruhe, 2024-07-12 (sustained true negatives (TNs)). Figure 8.6i shows an extended congestion between 11:00 and 19:00 spanning roughly 25 km. All four models correctly maintain low probabilities throughout, yielding a sustained sequence of true negatives. The component WMJs are individually well-defined and display characteristic recurring-congestion behavior ($v_g < 0$, moderate ϕ_{cd}, and multiple prior WMJs along the corridor), which allows the detectors to attribute the congestion to traffic dynamics rather than to an incident. This case is representative of the class of dense-traffic scenarios that the rule-based approach and the deep learning models jointly reject, and confirms that the elevated probabilities observed on the FP cases above are not merely a consequence of congestion severity.

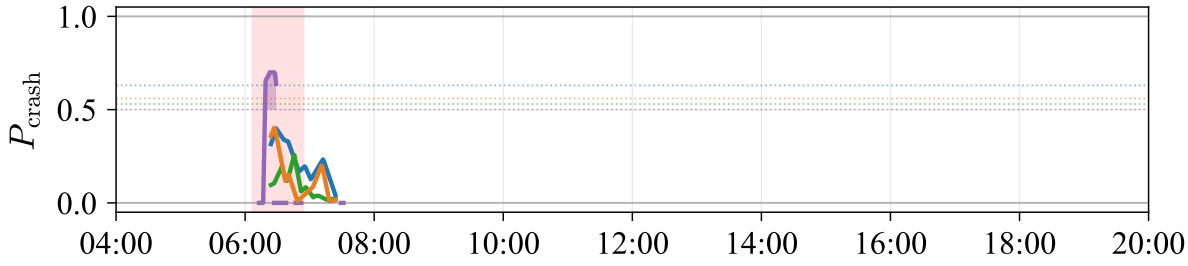
8.5 Analysis and Discussion

8.5.1 Model Performance Analysis

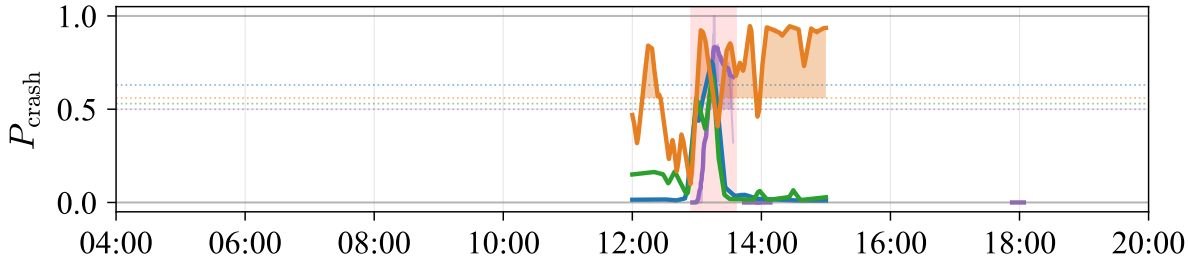
Rule-based approach. The rule-based method provides a transparent and interpretable baseline grounded in 3PTT. Its parameters are calibrated on a small labeled validation



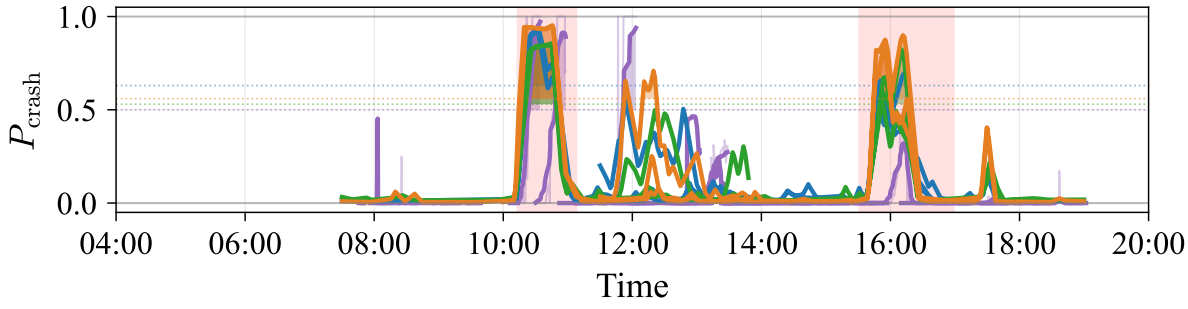
(a) Probability curves for sample Figure 8.6g.



(b) Probability curves for sample Figure 8.6h.



(c) Probability curves for sample Figure 8.7b.



(d) Probability curves for sample Figure 8.6b.

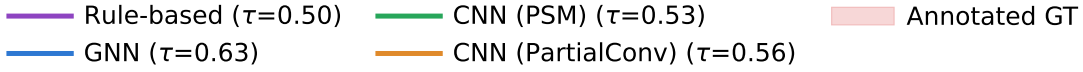


Figure 8.8: Crash-probability curves $P_{\text{crash}}(t)$ for the rule-based detector and the deep-learning baselines on four representative motorway segments. Red shading marks annotated ground-truth crash intervals; horizontal dotted lines indicate the per-model decision thresholds τ. Filled areas highlight time spans where $P_{\text{crash}} \geq \tau$.

subset using multi-objective Optuna and grid search (Section 6.2), rather than being trained end-to-end on the full labeled dataset; this substantially reduces the risk of overfitting to annotation artifacts, but does not eliminate it, since the calibration objective itself depends on crash labels. On the test set it achieves $F_1 = 75.0\,\%$, with precision (76.2 %) close to that of CNN (PConv) and recall (73.8 %) below all three deep learning models. False negatives arise when crash-induced congestion does not develop into a clearly structured mega jam; false positives occur when severe recurrent congestion produces "Mega jam"-like structures. Two principal limitations follow from its design rather than from its performance numbers. First, the calibrated parameter set is fixed at deployment: adapting to site-specific conditions (e.g., a different FCD penetration regime) requires re-running the calibration procedure rather than online adaptation. Second, the aggregated score $y_{\text{cls}}^{\text{pred}} \in [0, 1]$ produced by equation (6.13) is a thresholdable decision score but it is not probabilistically calibrated in the statistical sense and should not be interpreted as a posterior crash probability. For localization, the upstream-seed heuristic yields a mean MAE of 0.660 km and the downstream-front estimate 0.587 km, both substantially more accurate than any deep learning model and the TPEG reference.

GNN model. The GNN responds fastest to both crash onset and clearance, as its relational structure directly captures vehicle-level temporal dynamics. This responsiveness is reflected in its leading threshold-independent scores (AUC = 0.930, AP = 0.886) and its highest precision (82.3 %), indicating that explicit modelling of spatiotemporal relations and inter-vehicle interactions is beneficial for separating crash from non-crash congestion. However, the high sensitivity reduces prediction stability for two main reasons: the dynamically constructed graph varies substantially with traffic density, and incomplete vehicle coverage at the downstream congestion front can cause premature clearance detection (visible in the long left tail of Δt_{off} in Figure 8.4).

CNN (PSM) model. The CNN using PSM shows the slowest temporal response, with both rise and decline of crash probability lagging behind the other models. It achieves the highest recall (91.8 %) of any model but the lowest precision (74.7 %) among the deep learning variants; together with its smooth probability curves, this suggests that the PSM completion step propagates crash-indicative signal across the grid but also broadens the activated region, at the cost of more false positives. CNN (PSM) nonetheless produces the lowest localization error (1543.9 m) of the DL models, indicating that when the model eventually concentrates its prediction, it does so accurately.

Table 8.7: Summary of approach performance. The TPEG production system is included as an operational reference. Two check marks (✓✓) denote the best performance, one (✓) indicates moderate, and a cross (×) marks the weakest. "Interpretability" indicates whether the detection logic is fully transparent and auditable.

Approach	F1	Speed	Stability	Location	Interpretability
TPEG (ref.)	—	✓	×	✓	—
Rule-based	×	✓	✓✓	✓✓	✓✓
GNN	✓✓	✓✓	×	✓	×
CNN (PSM)	✓	×	✓	×	×
CNN (PConv)	✓✓	✓	✓✓	✓	×

CNN (PConv) model. CNN (PConv) responds faster than the PSM variant to both crash onset and clearance while maintaining stable probability trends. Despite ranking lower in the earlier small-scale ablation, it attains the best overall test-set F_1 (83.3 %), the best onset timing (12.9 min mean, 8.3 min median), and the tightest onset/clearance distributions (Figure 8.4). The divergence between the small-scale ablation ranking and the full-test ranking suggests that the partial-convolution handling of sparsity generalizes better to the full data distribution than the ablation subset indicated.

Crash location. For localization, both rule-based estimates outperform every deep learning model and the TPEG reference by roughly 4–10× (0.587–0.660 km vs. 2.6 km for TPEG, 5.6–7.6 km for the deep learning models in First mode). Among deep learning models, CNN (PConv) achieves the smallest Best-mode error, CNN (PSM) consistently predicts crash locations downstream of the labeled positions due to amplified local artifacts during PSM completion, and the GNN performs similarly to CNN (PSM) on mean error but with a noticeably lower First-mode median (2368.4 m vs. 3335.9 m).

Overall comparison. Table 8.7 consolidates the qualitative assessment. The GNN achieves the highest F_1-score among deep learning models with the fastest response but the lowest stability. The rule-based approach provides full interpretability and the highest inference throughput without labeled end-to-end training, along with the best localization accuracy, at the cost of a rigid binary decision and a non-calibrated score. CNN (PSM) produces smooth, high-recall outputs with reasonable F_1 but the poorest First-mode localization among deep learning models. CNN (PConv) provides stable probability evolution, the best test-set F_1, and the tightest timing distributions, with localization that is weakest in First mode but recovers under Best-mode aggregation.

TPEG production-system reference. The TPEG system provides an important operational baseline. Its onset delay ($\overline{\Delta t_{\text{on}}} = +14.4\,\text{min}$, $\sigma = 20.5\,\text{min}$) is comparable to the deep learning models (at max. F_1 setting), which is notable given that TPEG relies on upstream incident reports rather than real-time traffic analysis. However, TPEG clearance timing ($\overline{\Delta t_{\text{off}}} = -28.4\,\text{min}$) is the worst among all approaches, reflecting the operational practice of closing events upon incident clearance rather than upon congestion dissipation. For localization, TPEG (MAE = 2.6 km) outperforms the deep learning models but is substantially less accurate than the rule-based approach (0.6–0.7 km), despite having access to the reported incident position. All four proposed approaches thus improve upon the TPEG production system in at least one dimension, with the rule-based method offering the most substantial improvements in localization and clearance timing simultaneously.

Limits of data quality The annotation quality of the dataset bounds the achievable performance of all approaches, since the models are trained and evaluated against imperfect labels that do not always align with the underlying traffic dynamics. This label noise propagates into both false positives and false negatives across all models, which may learn to associate patterns with crashes on the basis of flawed annotations. Manual labeling of the congestion patterns mitigates the shortcomings of TPEG, but the actual crash events and the precise duration of the post-crash congestion cannot be determined from FCD alone. This inherent uncertainty in the ground truth imposes an upper bound on the performance attainable by any model trained on this dataset. Figure 8.9 illustrates a representative case of label noise. The TPEG reference (blue) indicates a crash at 09:50, while the underlying congestion structure exhibits a triangle-shaped mega-jam pattern (red) from 09:45 to 10:07, as recorded during manual labeling. The RB model detects a pattern whose onset coincides with the GT annotation but that persists until 10:27. A second congestion pattern (black) detected by the RB model at 10:25 may correspond to a false positive, a secondary crash event, or a continuation of the initial event; the data cannot resolve this ambiguity. This case is one of many in which the annotation does not align cleanly with the observed traffic dynamics, illustrating the challenges of training and evaluating models on noisy labels. The remaining congestion patterns not detected by the RB model (grey) further illustrate the impact of clustering on detection sensitivity: in particular, the congestion from 10:50 to 11:00 may equally well constitute a continuation of the initial crash pattern, but is treated as a separate congestion pattern and evaluated in isolation.

These annotation issues affect all four approaches, though in different ways. The deep learning models may learn to associate the noisy labels with specific input patterns,

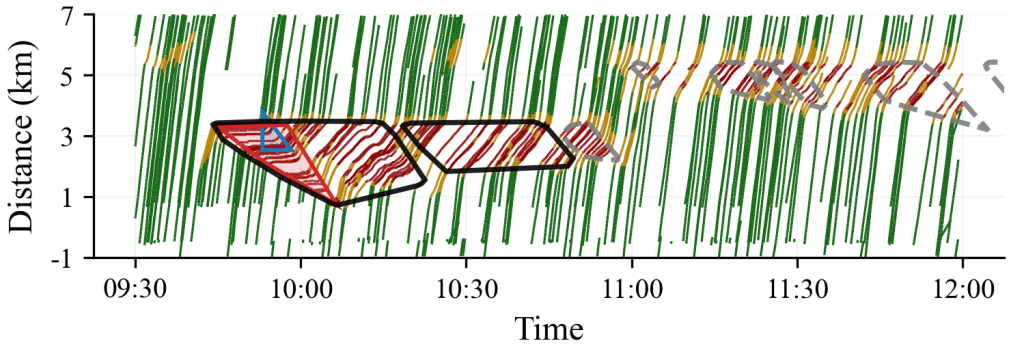


Figure 8.9: Example of label noise in the dataset. The TPEG reference (blue), the annotated GT (red) the detected crashes by the RB model (black) and the remaining clustered congestion patterns (grey) are shown.

potentially reducing generalization. The rule-based approach, which does not learn feature representations from labels, is less susceptible to propagating annotation bias into its decision logic; its test-set evaluation metrics, however, are equally affected by label noise, as both true and false detection rates are computed against the same imperfect ground truth.

9 Post-Crash Congestion Prediction

9.1 Introduction

While crash detection provides early awareness of traffic disruptions, its downstream value for routing and travel-time estimation depends on predicting how the resulting congestion evolves in space and time [45]. Crashes account for approximately 72 % of non-recurring congestion events [197], producing delays whose spatiotemporal extent is poorly described by crash duration alone.

Prior work on post-crash congestion prediction has largely separated two concerns. One line of research models the *extent* of incident-induced congestion, typically via data-driven spatiotemporal architectures [170, 216–219]; these approaches rarely produce an explicit spatial delineation usable for estimated time of arrival (ETA) computation. A second line focuses on *travel time* itself, either through production ETA systems [220, 221] or through incident-duration and delay-propagation studies [222–224]; these typically treat the incident as a point effect rather than a spatiotemporally evolving region. The two concerns remain largely decoupled in the literature [225, 226].

This chapter presents a prediction stage that sits directly downstream of the crash-detection pipeline developed in earlier chapters and couples both concerns within a single 3PTT-grounded model. Given a detected crash, the method estimates a convex-hull approximation of the crash-induced congestion region in the (t, d) plane, progressively refined as characteristic transition points become observable in the FCD stream. The same hull is then used to derive per-vehicle travel-time delay estimates, producing a unified framework in which spatial extent and ETA are two projections of the same representation. Validation is performed on the test and validation set, totaling 128 crashes (Table 3.9b). The performance is benchmarked against TPEG congestion delay reports for the same crash-induced congestion.

Parts of this chapter are based on previous publications by the author [6].

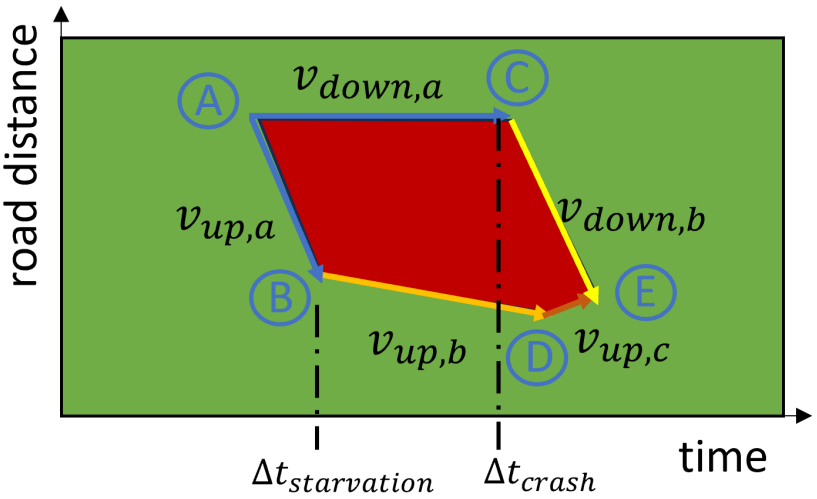


Figure 9.1: Schematic representation of post-crash congestion evolution with transition points (A) onset, (B) upstream starvation, (C) crash clearance, (D) upstream dissipation, and (E) congestion resolution.

9.2 Methodology

9.2.1 Congestion Evolution and Prediction Stages

The prediction is built around five spatiotemporal markers of a post-crash congestion event (Figure 9.1):

(A) *Congestion onset*: initial formation of congestion at the bottleneck location.

(B) *Upstream starvation*: the point at which upstream front velocity decreases, typically driven by rerouting.

(C) *Crash clearance*: removal of the incident from the road.

(D) *Upstream dissipation*: a secondary change in upstream front velocity during extended events, signaling accelerated recovery.

(E) *Congestion resolution*: full dissolution of the congested region.

Markers (A), (B), and (C) are *detection anchors* observed directly in the FCD stream and trigger hull updates. Marker (D) appears only in a minority of extended episodes and is handled within the same regression pipeline used for (B), as discussed in Section 9.2.3. Marker (E) follows deterministically from the intersection of the upstream and downstream fronts. The prediction is accordingly organized into three hull-producing stages: an initial

estimate at detection (Stage 1), a refinement at starvation (Stage 2), and a final estimate at clearance (Stage 3). Not every event exhibits (B); freeway stretches without off-ramps in particular may lack it, in which case Stage 2 coincides with Stage 1.

9.2.2 Congestion Area Estimation

To produce an early estimate, the initially unknown transition points (A)–(E) are computed from default parameters (Section 9.2.4). Once observations become available, measured values replace the defaults.

Let (t_0, d_0) denote the spatiotemporal seed point of the detected congestion pattern (A), and let Δt_{starv} and Δt_{crash} denote the elapsed times from onset to upstream starvation and crash clearance, respectively. Front propagation is characterized by four velocity parameters:

- $v_{\text{up,a}}$, $v_{\text{up,b}}$: upstream front velocities before and after starvation,
- $v_{\text{down,a}}$, $v_{\text{down,b}}$: downstream front velocities before and after clearance.

An additional $v_{\text{up,c}}$ enters the hull when marker (D) is observed; see Section 9.2.3.

The upstream starvation point (B) is computed from (A) and $v_{\text{up,a}}$:

$$\begin{aligned} t_{\text{us}} &= t_0 + \Delta t_{\text{starv}}, \\ d_{\text{us}} &= d_0 + v_{\text{up,a}}\,\Delta t_{\text{starv}}. \end{aligned} \tag{9.1}$$

The crash clearance point (C) follows from (B) and $v_{\text{down,a}}$:

$$\begin{aligned} t_{\text{ce}} &= t_{\text{us}} + \Delta t_{\text{crash}}, \\ d_{\text{ce}} &= d_0 + v_{\text{down,a}}\,\Delta t_{\text{crash}}. \end{aligned} \tag{9.2}$$

The congestion resolution point (E) corresponds to the intersection of the upstream and downstream fronts after clearance:

$$\begin{aligned} \Delta t_{\text{conv}} &= \left| \frac{d_{\text{us}} - d_{\text{ce}}}{v_{\text{down,b}} - v_{\text{up,b}}} \right|, \\ t_{\text{conv}} &= t_{\text{ce}} + \Delta t_{\text{conv}}, \\ d_{\text{conv}} &= d_{\text{ce}} + v_{\text{down,b}}\,\Delta t_{\text{conv}}. \end{aligned} \tag{9.3}$$

The total spatiotemporal extent is the area of the convex hull of the transition points:

$$A = \text{Area}\big(\text{ConvexHull}\{(t_0, d_0),\ (t_{\text{us}}, d_{\text{us}}),\ (t_{\text{ce}}, d_{\text{ce}}),\ (t_{\text{conv}}, d_{\text{conv}})\}\big). \tag{9.4}$$

Table 9.1: Parameters used in each prediction stage and their availability. Parentheses indicate limited reliability.

Parameter	Stage 1	Stage 2	Stage 3
$(t_0,\, d_0)$	✓	✓	✓
$v_{\mathrm{up,a}}$	✓	✓	✓
$v_{\mathrm{down,a}}$	–	✓	✓
$(t_{\mathrm{us}},\, d_{\mathrm{us}})$	–	✓	✓
$v_{\mathrm{up,b}}$	–	(✓)	✓
$(t_{\mathrm{ce}},\, d_{\mathrm{ce}})$	–	–	✓
$v_{\mathrm{down,b}}$	–	–	(✓)

Table 9.1 summarizes parameter availability at each prediction stage. Velocities shown in parentheses indicate limited reliability: they are derived from slope changes identified in few data points and are therefore sensitive to noise. For crash clearance, $v_{\mathrm{down,b}}$ is initialized at the empirically supported default of $-15\,\mathrm{km\,h^{-1}}$ [59, 78] and blended with the measured slope as observations accumulate.

As a safeguard against a degenerate geometry, when the predicted upstream and downstream fronts disagree severely (e.g., near-parallel slopes shortly after clearance), the naive intersection Δt_{conv} from equation (9.3) can fall an implausibly large distance past t_{ce}. Δt_{conv} is therefore capped at a fixed offset beyond t_{ce}, preventing a single noisy slope estimate from producing a runaway spatial error.

Iterative refinement. Between the discrete stage updates triggered by (B) and (C), the predicted hull is re-evaluated every $\Delta t_{\mathrm{update}} = 2\,\mathrm{min}$ from the incoming FCD stream. At each step, the front velocities observed are re-estimated by linear regression over the most recent front positions, and the hull is re-computed from equations (9.1) to (9.3). Parameters whose transition point has not yet been observed remain at their defaults; observed parameters are replaced by their running estimates. This two-timescale scheme—discrete between-stage updates and continuous within-stage refinement—prevents small errors in an early slope estimate from compounding into a large spatial offset at the prediction horizon. Figure 9.2 illustrates the effect during the starvation stage of a German A8 event: as additional trajectories enter the regression window, the upstream velocity estimate converges and the predicted front aligns with the observed one.

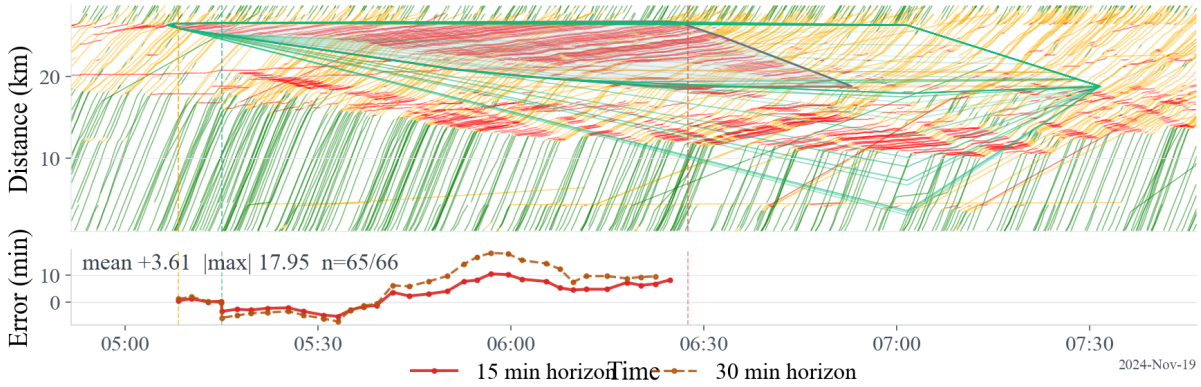


Figure 9.2: Within-stage refinement during the upstream-starvation stage (A8, November 19, 2024). The predicted hull (red) is re-estimated every two minutes; the upstream front velocity converges toward the observed propagation as new FCD trajectories enter the regression window.

9.2.3 Detection of Transition Points

Both the upstream starvation point (B) and the crash clearance point (C) are detected by continuous slope estimation on the corresponding fronts: upstream-entering FCD transitions for (B) and downstream-leaving transitions for (C). The method reuses the regression pipeline described in Section 6.1.4.

Upstream starvation (B). As drivers reroute in response to a detected crash, upstream inflow decreases and the upstream front velocity v_u exhibits a sustained reduction. A rolling comparison of the current regression slope against previously observed values identifies the starvation point (Figure 9.3). Rerouting via off-ramps is the most common cause, but any factor reducing overall demand—e.g., the end of a rush-hour period—can produce the same signature; an empirical example of off-ramp-driven starvation on the German A81 is provided in Section F.1. Once identified, the slope change updates the upstream front prediction (orange arrow in Figure 9.1).

Upstream dissipation (D). In extended congestion episodes, the upstream front velocity may undergo a *second* change—approaching zero or turning positive—as upstream inflow drops below the queue discharge rate. Although observed in only a minority of events, it matters when present, because the associated $v_{\mathrm{up,c}}$ replaces $v_{\mathrm{up,b}}$ in equation (9.3) and shortens the predicted tail of the event. (D) is detected within the same slope-regression framework used for (B), gated on (C) already being observed and conditioned on a joint data-gap-and-slope-collapse check: the gap between the most recent upstream transition

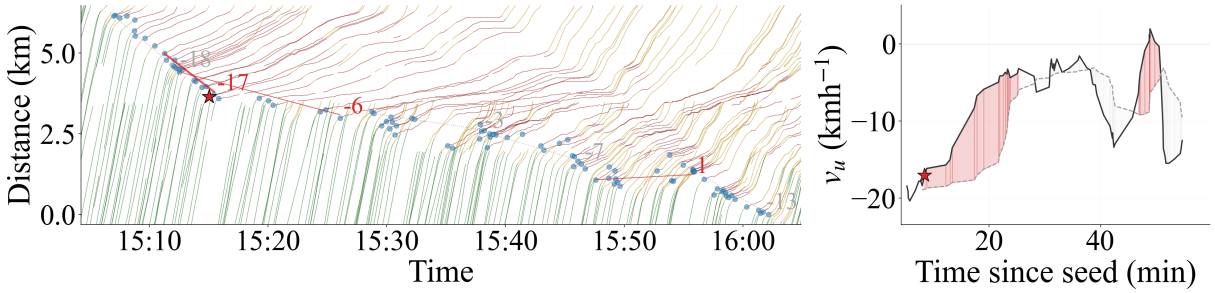


Figure 9.3: Detection of upstream starvation via continuously evaluated slopes of the upstream front velocity v_u. The red star marks the detected starvation point.

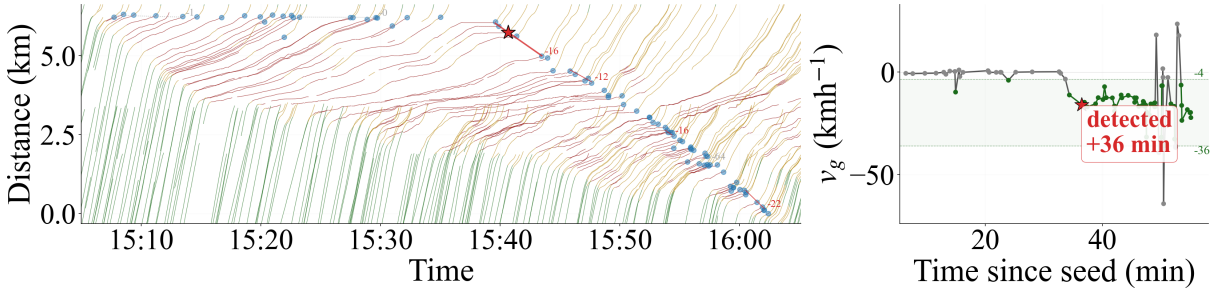


Figure 9.4: Crash clearance detection based on downstream front velocity. Gray points: detected front positions; colored lines: regression slopes; red dotted line: clearance point.

and the latest non-upstream observation must exceed a multiple of the mean post-starvation inter-arrival gap, *and* the current upstream-front slope must satisfy $v_{\text{up}} \geq v_{\text{collapse}}$ (on the order of $-3\,\text{km}\,\text{h}^{-1}$). When both hold, the last observed trajectory anchors the dissipation point and the corresponding final-window slope is recorded as $v_{\text{up,c}}$.

Crash clearance (C). The downstream front slope is evaluated continuously; clearance is confirmed once a configurable number of consecutive estimates fall within a classification-specific velocity band centered on $v_g \approx -15\,\text{km}\,\text{h}^{-1}$ [59] (Figure 9.4). With both front velocities then known, the resolution point (E) follows from equation (9.3).

9.2.4 Default Prediction Parameters

On first detection, transition points and velocities that have not yet been observed are initialized from data-driven defaults derived from the full set of 445 labeled congestion events (Table 3.6). A full exploration of category-specific parameterization—K-means clustering on a nine-dimensional feature vector, analysis of variance (ANOVA) effect-size analysis, and an early-classification experiment—is documented in Section F.2. The analysis

Table 9.2: Global default prediction parameters (medians over 445 events). Times in minutes, velocities in $\mathrm{km\,h^{-1}}$.

$\Delta t_{\mathrm{starv}}$	$\Delta t_{\mathrm{crash}}$	$v_{\mathrm{up,a}}$	$v_{\mathrm{up,b}}$	$v_{\mathrm{down,a}}$	$v_{\mathrm{down,b}}$
23	62	−8.6	−0.8	0.0	−15.2

shows that upstream propagation velocities are effectively group-independent ($\eta^2 < 0.05$) and that the features which *do* separate clusters are cumulative quantities that reveal themselves only as the event unfolds. A single set of global medians is therefore adopted (Table 9.2); these values carry substantial initial uncertainty, which the iterative refinement procedure progressively resolves as observations accumulate.

9.3 Travel-Time Delay Estimation

The predicted delay D_j^{pred} derives from $\mathcal{H}_{\mathrm{pred}}$ by virtual-trajectory simulation: departure times $t_j = t_{\min} + (j-1)\Delta t$ are sampled uniformly across the hull's temporal extent, mapped to entry positions d_j on the lower boundary by uniform arc-length sampling, then propagated at v_{cong} inside the hull and v_{free} outside, yielding

$$D_j^{\mathrm{pred}} = T_j^{\mathrm{cong}} - T_j^{\mathrm{free}}. \tag{9.5}$$

The ground-truth reference is reconstructed from FCD probes traversing $\mathcal{H}_{\mathrm{true}}$. Each vehicle's longest contiguous in-hull run yields entry $(t_k^{\mathrm{entry}}, d_k^{\mathrm{entry}})$, exit $(t_k^{\mathrm{exit}}, d_k^{\mathrm{exit}})$, transit duration τ_k, and distance ℓ_k, with the delay taken as the excess over free-flow propagation,

$$D_k^{\mathrm{true,traj}} = \tau_k - \frac{\ell_k}{v_{\mathrm{free}}}. \tag{9.6}$$

This measures the same physical quantity the hull simulation forecasts, the additional time a driver entering at tick T actually experiences inside the jam, and is what an ETA consumer needs.

The free-flow reference v_{free} is drawn from FCD probes in the same spatial range as $\mathcal{H}_{\mathrm{true}}$ but outside its temporal extent within ±2 h, with the corpus median adopted. For Section 9.4.3, $v_{\mathrm{cong}} = 8.6\,\mathrm{km\,h^{-1}}$ and $v_{\mathrm{free}} = 105.5\,\mathrm{km\,h^{-1}}$ are extracted directly from the test set; MAE is insensitive to v_{cong} over a 4× range (Table F.2).

Table 9.3: IoU scores across prediction stages.

Prediction stage	IoU
Stage 1 (Onset)	0.456
Stage 2 (Starvation)	0.588
Stage 3 (Clearance)	0.913

9.4 Results

9.4.1 Dataset and Validation Setup

Each case is triggered three minutes after the ground-truth crash onset, decoupling prediction error from detection latency and false-alarm effects. In production, the alert delay would be dictated by the detection component.

The hull simulation is compared against TPEG congestion-delay reports already distributed to original equipment manufacturers (OEMs) and map providers. Since TPEG emits no forward simulation, its most recent value at tick T is held forward as the horizon-h estimate, corresponding to the implicit forecast seen by downstream consumers between feed updates. The comparison is restricted to Mega Jam events and evaluated at the native 2 min tick cadence on the held-out test set.

The IoU,

$$\mathrm{IoU} = \frac{|\mathcal{H}_{\mathrm{true}} \cap \mathcal{H}_{\mathrm{pred}}|}{|\mathcal{H}_{\mathrm{true}} \cup \mathcal{H}_{\mathrm{pred}}|}, \tag{9.7}$$

quantifies spatiotemporal hull overlap. The MAE of travel-time delay, evaluated at horizons of 5, 10, 15, and 30 min against both references in Section 9.3, quantifies value for routing and ETA.

9.4.2 Intersection over Union Performance

Table 9.3 reports IoU across the three prediction stages. Scores improve monotonically as each new anchor replaces a defaulted parameter with a measured one. Stage 3 is tightest because clearance anchors the downstream boundary and the upstream front has stabilised over the full starvation stage; Stage 1 relies almost entirely on the data-driven defaults of Section 9.2.4.

IoU is, however, only an indirect proxy for the routing-relevant quantity — the delay experienced by vehicles crossing the hull. A moderate IoU can correspond to acceptable delay error when the temporal boundaries are correctly placed even if spatial extent drifts.

Table 9.4: Corpus filtering from matched Mega jams to the curated paired-comparison subset. The first block removes events where TPEG cannot supply a usable baseline; the second block applies surrounding-congestion classification, used to partition retained events into Group A (with surrounding congestion that confounds TPEG's section-level delay reading) and Group B (free-flow surroundings).

Stage	Events	Share
Matched Mega jam events (IoU matching)	128	100.0%
No TPEG reports in matched window	−46	−36.0%
TPEG present but ≥ 70% tick gaps	−42	−32.8%
Events with sufficient TPEG coverage	40	31.3%
Group A, with surrounding congestion	16	12.5%
Group B, free-flow surroundings	24	18.75%

9.4.3 Travel-Time Delay Prediction Accuracy

The paired-comparison corpus is reduced from 128 IoU-matched Mega jam events to 40 through two compounded filters, detailed in Table 9.4. The remaining events partition into 16 in Group A (surrounding congestion) and 24 in Group B (free-flow surroundings). Since TPEG reports a full section delay rather than a per-jam delay, the comparison is restricted to Group B, where the persistence assumption is consistent with the underlying free-flow condition; reporting against Group A would conflate the hull's per-jam forecast with TPEG's aggregated section reading and overstate the hull's advantage. Note that TPEG delay is broadcasted through a different channel than the event feed, explaining the presence of delays for only a subset of the mega jam events.

On the reduced dataset (Table 9.5), the hull simulation beats TPEG at every horizon through h=30, with pooled MAE of 9.6 min at the nowcast and 15.4 min at h=30 against TPEG's 12.0 and 17.1 min.The two estimators fail in opposite directions: the hull under-reports by −3.6 to −12.2 min, since the floored v_{cong} decays the predicted queue too aggressively, while TPEG stays close to unbiased on Group B at +0.4 to +2.1 min across the horizon range. Splitting the rolling errors by lifecycle stage (Section F.6), the hull's gain sits at the dynamic transitions, cutting MAE by 5.4 min at onset and 10.4 min at crash-end, and the two methods are within 0.2 min on MAE through the upstream-starvation plateau, where a held-forward delay is approximately what is happening physically.

Performance by jam size. Grouping the h=15 in-jam paired records by the actual delay at the forecast target (Table 9.6), the hull cuts MAE by 50% in the 5–15 bucket (5.8 vs

Table 9.5: Travel-time delay prediction error against the TPEG congestion-delay baseline, evaluated on the active-phase records of the corpus subsets defined in Table 9.4. Hull statistics pooled per 1-minute tick across 38 events (4,228 ticks total); TPEG statistics pooled across 17 events (1,693 ticks). MAE, root mean squared error (RMSE) and signed bias in minutes. Ground truth: real trajectory; simulator parameters: per-stage measured v_{cong} with floor 8 km/h, $v_{\text{free}} = 110$ km/h.

	Hull simulation, $n = 40$ events				TPEG, $n = 24$ events		
h [min]	MAE	RMSE	Bias	n	MAE	RMSE	Bias
0	9.62	13.24	−3.58	932	11.98	16.31	+1.66
5	10.80	14.61	−4.29	820	13.45	17.57	+1.04
10	11.93	15.81	−5.82	769	14.14	18.37	+0.89
15	12.81	16.95	−7.49	721	14.71	18.75	+0.76
30	15.39	19.79	−12.19	457	17.12	22.01	+0.42
60	19.58	25.05	−18.65	101	18.38	22.97	+2.10

Table 9.6: In-jam h=15 MAE stratified by realised delay at the target forecast time. The ≥ 60 min bucket is empty for paired Group B records at h=15 in the present corpus.

Bucket [min]	TPEG MAE	Hull MAE
0–5	7.56	7.63
5–15	11.61	5.78
15–30	16.21	13.69
30–60	24.13	26.18
pooled	14.68	12.29

11.6 min) and by 16% in the 15–30 bucket (13.7 vs 16.2 min). On 30–60 min jams the two methods are comparable in pooled MAE, but the hull severely under-reports, with bias −26.2 min against TPEG's −13.0 min.

Error growth with horizon. Hull MAE grows from 9.6 min at h=0 to 19.6 min at h=60, with bias growing monotonically more negative from −3.6 to −18.7 min over the same range; TPEG MAE grows more gradually from 12.0 to 18.4 min, with bias remaining within +0.4 to +2.1 min throughout. The crossover at h=60 where TPEG falls below the hull reflects the cumulative effect of the v_{cong}-floor on long-horizon predictions; a horizon-aware floor schedule or a queue-shape correction at long lead would address this trade-off, as reported in the robustness sweep in Table F.2.

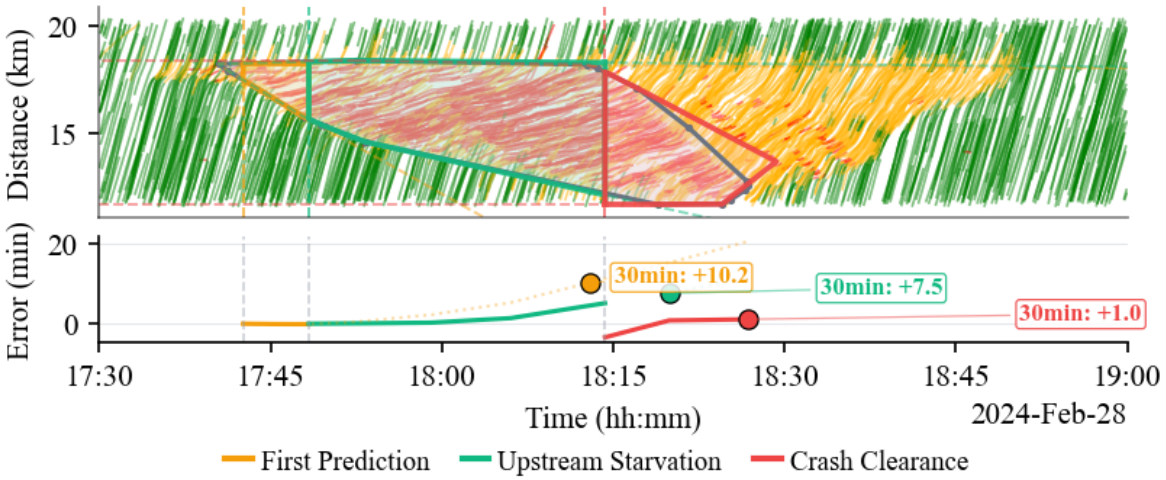


Figure 9.5: Predicted congestion area for US I-294, February 24, 2024. The gray area represents the actual congestion hull; orange, green, and red show the three prediction stages. Trajectories are colored by 3PTT phase. The rolling error and 30 min error per stage are shown in the lower panel.

9.4.4 Case Studies

Three events from three countries illustrate the method's behavior across penetration rates, road geometries, and congestion morphologies. Each figure shows the ground-truth hull (gray) alongside the three prediction stages (orange, green, red) with trajectories colored by 3PTT phase; the lower panel gives the rolling prediction error defined in Section 9.4.1 and the corresponding per-stage summary. Two further case studies—(i) compounding-error behavior without within-stage refinement on the German A81 (March 18, 2022), and (ii) an extended starvation stage without error accumulation on NJ 3, United States (March 3, 2024)—are provided in Section F.3. A visual TPEG delay report analysis can be found in Figure F.5.

I-294, Illinois, United States (February 24, 2024). The I-294 case demonstrates the direct benefit of within-stage updates (Figure 9.5). An early slump in the upstream front velocity is picked up by the rolling regression, which prevents the Stage 2 upstream boundary from drifting past the Stage 3 downstream boundary—the failure mode in which the predicted fronts would otherwise fail to meet.

M3, United Kingdom (September 25, 2025). The M3 case tests the method at a substantially lower FCD penetration rate (Figure 9.6). Despite the sparser trajectory data,

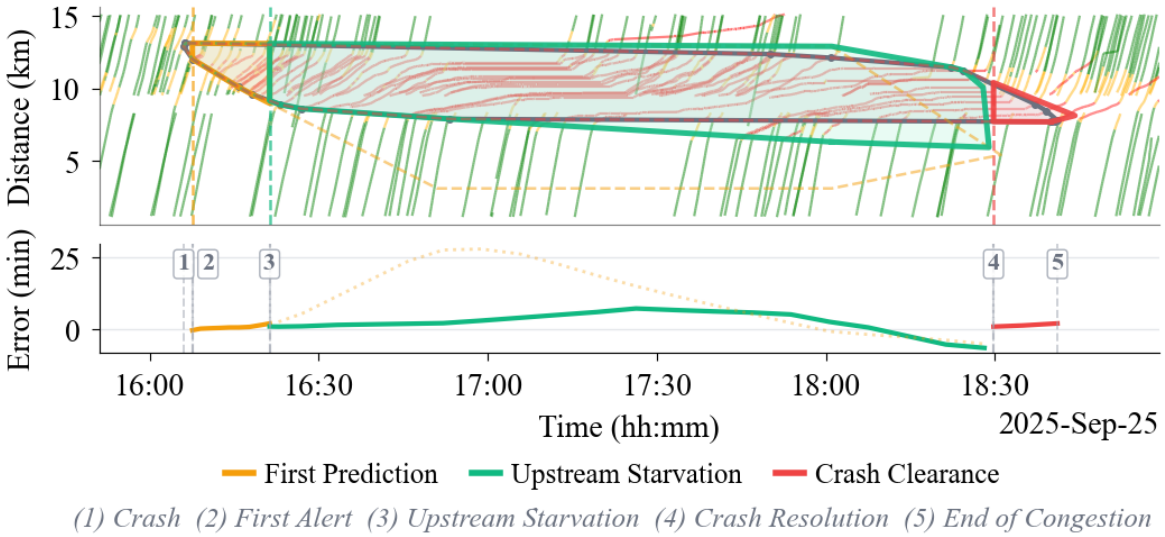


Figure 9.6: Predicted congestion area for UK M3, September 25, 2025.

the within-stage regression remains stable enough to prevent error growth, and the rolling error stays in the same low range. This indicates that the method degrades gracefully under reduced data availability, provided the penetration suffices to identify the front positions reliably.

A8, Germany (November 19, 2024). The A8 event exposes the sensitivity of the final shape to the clustering that underlies front identification (Figure 9.7). In Stage 3, the detected upstream front deviates from the GT shape because it tracks a WMJ that the GT excluded, which propagates into a larger error in the final stage. However, the meeting-point cap of Section 9.2.2— introduced precisely to suppress unreasonably distant convergence points when the estimated fronts disagree—bounds the spatial reach of this deviation, so the final stage recovers: per-stage IoU values are [0.564, 0.601, 0.910].

A8, Germany (March 20, 2022). Figure 9.8 overlays actual and predicted delay. Both TPEG and the staged congestion track the ground truth closely, though TPEG underestimates delay in the first 20 min and overestimates it in the last 20 min, a signature of sluggish adaptation to the evolving congestion. Post-crash congestion method also overestimates the end of congestion, as the crash end is only detected with a delay.

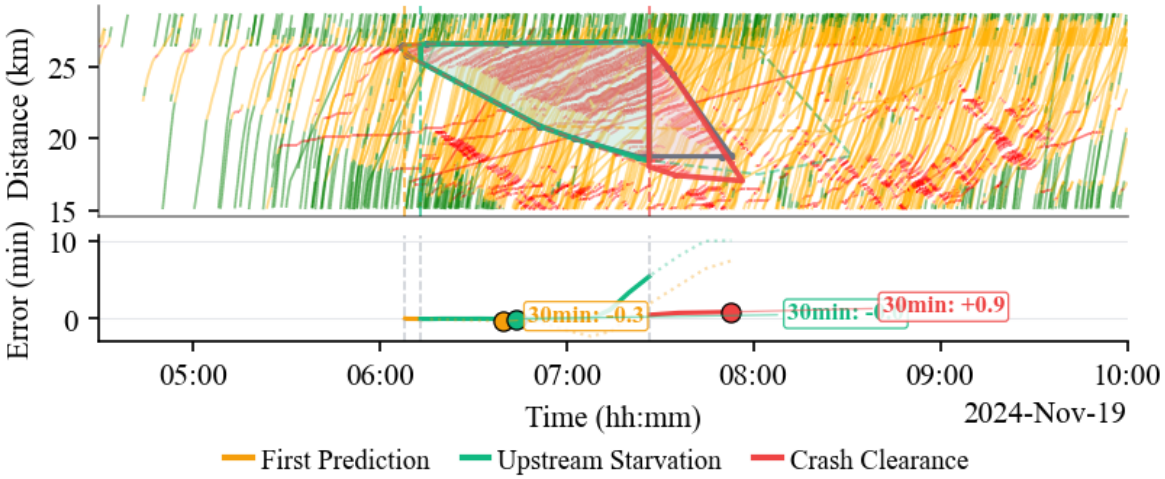


Figure 9.7: Predicted congestion area for German A8, November 19, 2024.

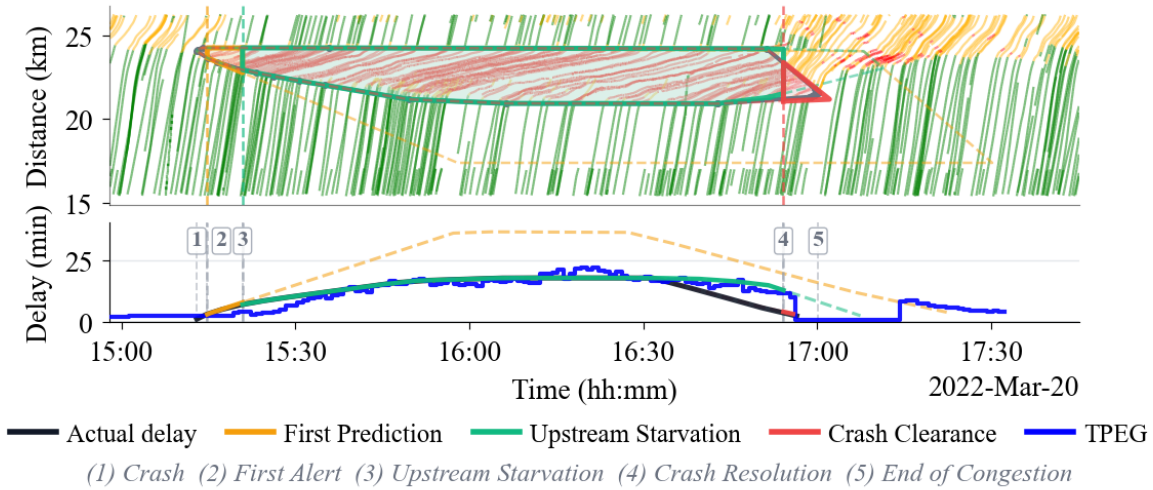


Figure 9.8: Predicted congestion area for German A8, March 20, 2022. TPEG Reference in blue.

9.5 Discussion

9.5.1 Limitations

The convex-hull representation assumes that the congestion region can be approximated as a simply connected polygon in the (t, d) plane, which may not hold for fragmented or multi-nucleus patterns.

Travelers who receive delay warnings may reroute, shift departure times, or switch modes, causing the forecast to deviate from observed traffic in ways that can be self-defeating (relieving one corridor while creating bottlenecks elsewhere). The results reported here do not capture this effect, as predictions were not disseminated during the observation period. The iterative refinement procedure partly mitigates the concern in practice—the initial estimate is conservative and tightened only as observations accumulate, reducing the risk of drivers overreacting to an early low-confidence forecast—but quantifying the effect requires live-system validation with active dissemination.

9.5.2 Extensions and Future Work

Two natural extensions follow from the present work. First, complex episodes with secondary incidents or strong demand fluctuations exhibit multiple upstream slope shifts; incorporating detection of additional turning points beyond (B) and (D) would tighten the predicted hull in those cases. Second, coupling the prediction with a pilot deployment by a traffic-service provider would both extend the validation set and address the behavioral-feedback question raised above.

10 Discussion and Conclusion

This thesis investigated whether FCD can serve as a practical, infrastructure-independent data source for detecting, predicting, and explaining non-recurring freeway congestion. Chapters 4 to 9 show that the developed approach matches the currently deployed reference sources and outperforms them on timeliness and localization.

10.1 Revisiting the Research Questions

RQ1: Crash Detection and Localization

Can FCD detect and localize freeway crashes without roadside infrastructure or involvement of the crashed vehicle?

The rule-based pipeline Chapter 6 reaches $F_1 = 75.0\,\%$ with the tightest onset distribution of any approach examined ($\sigma = 7.9\,\text{min}$, roughly half the CNN variants and a quarter of the GNN), which follows from its staged evidence accumulation. Localization reaches a mean error of 587–660 m, improving on TPEG (2,609 m) by a factor of four and on every deep learning model by a factor of eight to thirteen. A 3PTT-grounded detector therefore reaches competitive detection scores and the sharpest localization of any method examined here, without end-to-end supervised training.

The three deep learning models trade spatial accuracy for higher classification scores and lower minimum latency. CNN (PConv) leads on peak F_1 (83.3 % at 13.1 min) and clearance dispersion ($\sigma = 24.4\,\text{min}$ against 82–89 min for the rest); the GNN leads on threshold-independent scores (AUC = 0.930, AP = 0.886) and matches CNN (PConv)'s peak at nearly double the delay; CNN (PSM) leads on recall (91.8 %). The rule-based detector is bounded below at $\sim 4.7\,\text{min}$ by its plausibility filters, which also cap F_1 near 0.76, whereas the deep-learning detectors reach $\sim 2\,\text{min}$ at $F_1 \approx 0.55$. Within the 5–10 min envelope reachable to all four approaches, the GNN dominates the Pareto front below $\sim 8\,\text{min}$ and CNN (PConv) above. The GNN is the only detector clearing $F_1 \geq 0.7$ at $\leq 7\,\text{min}$ delay ($F_1 = 0.724$ at 6.43 min) and the only one reaching $F_1 \geq 0.69$ under a $\leq 5\,\text{min}$ budget ($F_1 = 0.698$ at 4.69 min).

All four approaches improve on TPEG in at least one operational dimension. Mean onset delay drops from +14.4 min (TPEG) to +12.3 min (rule-based) and +12.9 min (CNN (PConv)), clearance bias from −28.4 min to +4.5 min under CNN (PConv), and localization from 2.6 km to below 700 m with the rule-based estimator despite TPEG having direct access to the reported incident position. The strengths are complementary: the rule-based pipeline gives the most predictable onset timing and the sharpest localization, the GNN the fastest accurate detections, and CNN (PConv) the highest peak F_1. No single method wins on every metric, so the open question is how to combine them rather than which one to choose. A hybrid stack that issues an early GNN alert, refines the spatial estimate with the rule-based localizer, and confirms with CNN (PConv) should inherit the strengths of each and outperform any single component.

RQ2: Post-Crash Congestion Prediction

Can the spatiotemporal propagation of post-crash congestion be predicted with sufficient accuracy to provide actionable travel-time estimates during the early phase of an incident?

The iterative convex-hull framework Chapter 9 passes through three stages tied to observable transition points (onset, upstream starvation, clearance), with hull overlap rising from IoU = 0.456 at onset to 0.588 after starvation and 0.913 after clearance. Against the TPEG congestion-delay feed currently distributed to OEMs and map providers, the hull beats TPEG at every horizon through h=30, with pooled MAE running from 9.6 min at the nowcast to 15.4 min at h=30 against TPEG's 12.0 and 17.1 min. Broken down by jam size, the gain is uneven: the hull cuts MAE by 50% on 5–15 min jams (5.8 vs. 11.6 min) and 16% on 15–30 min jams (13.7 vs. 16.2 min), but under-reports on 30–60 min jams (bias −26.2 vs. TPEG's −13.0 min), where the floored v_{cong} decays the predicted delay too quickly against a queue that is still long. The framework therefore gives usable travel-time estimates over the routing-relevant 5–30 min horizons for most jams in the corpus, and fails in a known way on multi-hour jams.

RQ3: Weather-Induced Traffic Breakdown

How do localized rain cells influence freeway traffic breakdown, and does the movement direction of a rain cell relative to the traffic stream govern the severity of the resulting congestion?

The adjusted odds ratio for a one-log-unit intensity increase is 2.43 (CI_{95} $[1.61, 3.67]$), while area loses significance once intensity is controlled for. Direction matters once

congestion severity is held constant, with upstream-moving LRCs yielding 2.45× the odds of constrained propagation over downstream-moving ones (CI_{95} [1.15, 5.21]), and relative velocity gives a sharper account of the mechanism. Moderate upstream velocities lock the LRC onto the MB with an 89.2 % constraint rate (against 62.1 % for fast upstream, $OR = 5.04$), whereas faster downstream LRCs extend post-rain persistence without raising the already low constraint rate, the most problematic configuration identified. Breakdowns cluster at LRC entry and exit fronts rather than the interior, producing the observed U-shape. A crash rate ratio of 1.52 under rainfall, consistent with published ranges [41, 201, 202], couples the weather and crash analyses: rainfall raises the probability of the events RQ1 targets. The LRC–congestion coupling is direction-dependent and operationally relevant, and rainfall intensity together with LRC velocity relative to traffic forms a compact predictor set for breakdown risk.

10.2 Methodological Observations

Two points run across the three problems and shape how the results above should be read.

Chapter 5 uses phase labels to define the detection vocabulary (jam extent, phase-transition propagation, downstream-front velocity consistency); Chapter 7 feeds the same phase labels as the CNN's second channel and the GNN's node features, carrying the inductive bias into the learned representation; Chapter 4 interprets LRCs as moving bottlenecks within the same framework. Raw trajectories are projected into a theory-informed state space before any detection, learning, or analysis. This projection is cheap and sensor-free, and a single shared preprocessing framework (Chapter 3) serves all three problems.

The composition argument made in RQ1 is not specific to detection. The same pattern recurs in Chapter 9, where rule-based transition-point detection gates an otherwise data-driven hull refinement: the IoU progression from 0.456 at onset to 0.913 at clearance tracks how each new rule-based anchor tightens the data-driven prediction. The recommendation throughout is composition rather than replacement, with rule-based stages providing interpretable, low-latency structural anchors and learned stages providing classification refinement, clearance confirmation, and hull continuation between those anchors.

10.3 Practical Implications

The results address known limitations of current TMC operations. The clearance bias of +4.5 min under CNN (PConv), against −28.4 min for TPEG, gives operators a more faithful

estimate of when congestion actually dissipates, and a rule-based localization error below 700 m, against 2,609 m, provides a more precise spatial anchor for dispatching response resources. The direction-dependent mechanism from Chapter 4 further suggests that variable speed-limit and route-guidance systems can exploit LRC motion relative to traffic: tracking the dry-to-wet front under upstream-moving precipitation, and shifting focus to residual-congestion dissolution under downstream-moving precipitation. For navigation and ETA services, the iterative hull prediction beats the deployed TPEG baseline at every horizon through h=30, with the gain sharpening on the 5–30 min jams that make up most of the corpus, where it cuts MAE by 16 to 50%. The stage-gated refinement also reduces the risk of overreacting to early, high-uncertainty forecasts, a failure that Waze-style crowdsourced systems cannot easily avoid, and gives a principled way to release progressively tighter estimates as transition points become observable in the FCD stream. On longer jams the post-clearance decay geometry breaks: on 30–60 min jams the two methods are comparable in pooled MAE, but the hull severely under-reports (bias −26.2 min against TPEG's −13.0 min). This is a calibration limitation of the post-clearance plateau model, not of the framework itself, and remains a deployment caveat. None of the proposed methods requires roadside sensors, vehicle involvement at the crash site, or pre-installed detection equipment. This is the main practical argument for the framework in regions where fixed sensing is sparse or absent, which includes most of the road networks where the underreporting figures (Chapter 1) are highest.

10.4 Limitations

Data coverage and sampling. The dataset is dominated by German freeways with a smaller international subset (UK, Italy, US), so transferability to urban, mixed-class, or behaviorally distinct networks is plausible in principle but not empirically verified. Connected-vehicle penetration near 5 % is sufficient for pattern-level analysis but introduces sampling bias for short-lived events and for trajectory coverage at the downstream congestion front; clearance detection is disproportionately affected, visible in the long left tail of Δt_{off} for three of the four approaches (Figure 8.4).

Label ceiling. The annotation workflow in Chapter 3 mitigates the known shortcomings of TPEG as a reference, but the actual crash instant and the precise duration of post-crash congestion are not directly observable, leaving an irreducible uncertainty in the labels themselves. This sets a ceiling on F_1 in Table 8.2 that binds learned and rule-based detectors alike, though not symmetrically (Figure 8.9), so comparative rankings are more

reliable than absolute scores.

Representational scope. The convex-hull representation in Chapter 9 assumes a simply connected congestion region in the (t, d) plane and degrades on multi-nucleus or fragmented patterns. The deep-learning detectors operate on $30\,\text{km} \times 30\,\text{min}$ windows; events exceeding these bounds rely on the sliding-window aggregation of Section 8.2, which carries no cross-window state and cannot exploit dependencies between windows in a single forward pass. The weather analysis is restricted to rainfall; snow, fog, and freezing rain involve distinct physical mechanisms and are out of scope. LRC identification relies on DBSCAN, so boundary cases between constraint categories are sensitive to its parameter choice, and statistical power is additionally limited for the subset of events with estimable precipitation velocity.

Deployment readiness. None of the four detection pipelines or the post-crash predictor has been evaluated inside a production system with active information dissemination. Reported errors reflect open-loop, offline performance; closed-loop effects, where operator action or driver-facing broadcasts alter the routing decisions of FCD-providing vehicles and thereby perturb the input stream, remain unmeasured and are an open question for follow-on work.

10.5 Future Work

The most immediate extension is the composite detection pipeline sketched in RQ1. The complementarity identified in Chapter 8 has not yet been exploited in a joint system, and a direct benchmark of such a stack against its individual components on the same test set is the natural next step.

A more fundamental change is to reframe crash detection as anomaly detection rather than supervised classification. The deep-learning approaches here frame crash detection as supervised binary classification, which assumes sufficient positive examples to learn a discriminative boundary. Crashes are rare by construction, the dataset is correspondingly imbalanced, and the available positive labels carry the spatial and temporal inaccuracies characterized in Section 8.5. Training exclusively on non-crash traffic and flagging deviations from normal dynamics would remove the dependence on crash-labeled data and reduce the objective to modeling the distribution of regular traffic states. Two versions can be built on the existing architectures: a reconstruction-based autoencoder on the CNN's spatiotemporal grid or the GNN's graph structure, with crash-induced patterns yielding

elevated reconstruction errors that replace the classification head; and a self-supervised alternative that learns embeddings of normal congestion, with crash samples detected as outliers via distance scoring on the pretrained embeddings.

The remaining directions broaden scope. The convex-hull representation can be generalized via hull unions or alpha shapes, handling multi-nucleus and fragmented patterns within the same framework. The rain-cell analysis can be extended to snow, fog, and freezing rain, whose distinct physical mechanisms fall outside the rainfall-only scope of Chapter 4. Transferability to urban and lower-class roads should be validated empirically, since the preprocessing pipeline is road-class-agnostic in principle but its phase thresholds and scoring constants are calibrated on freeway traffic. A joint model drawing on the shared FCD across both weather and crash questions, using weather information to sharpen the crash detector and masking crash-induced congestion to clean the rain-cell analysis, would bring the three contributions together under one detection objective. A field trial with a provider of sufficient connected-vehicle coverage, finally, would quantify the feedback loop and stress-test the sliding-window inference scheme of Section 8.2 under streaming FCD, closing the open-loop gap noted in the limitations.

10.6 Closing Remarks

Infrastructure-independent freeway crash detection from FCD is feasible today. A single position-trace stream, projected through 3PTT into a phase-aware state space, supports detection, localization, and post-crash prediction at levels that exceed the timeliness and localization accuracy of the TPEG feed currently distributed to OEMs and map providers, while using none of its incident-reporting inputs. Within the framework, no detector wins on every metric, and the practical value lies in combining complementary estimators on the shared FCD stream rather than in any single method. The same state space also makes the coupling between localized rain cells and freeway breakdown direction-dependent and measurable, with the movement of a rain cell relative to traffic governing whether congestion stays constrained.

A Data and Preprocessing

A.1 TPEG Event Codes

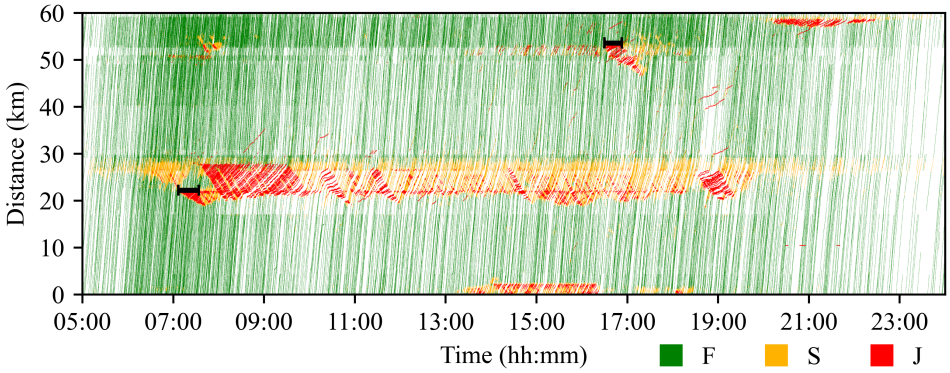
TPEG defines 31 event codes covering traffic conditions, incidents, and regulatory information [132]. For reference, these codes are grouped thematically in Table A.1.

Table A.1: Thematic grouping of TPEG event codes.

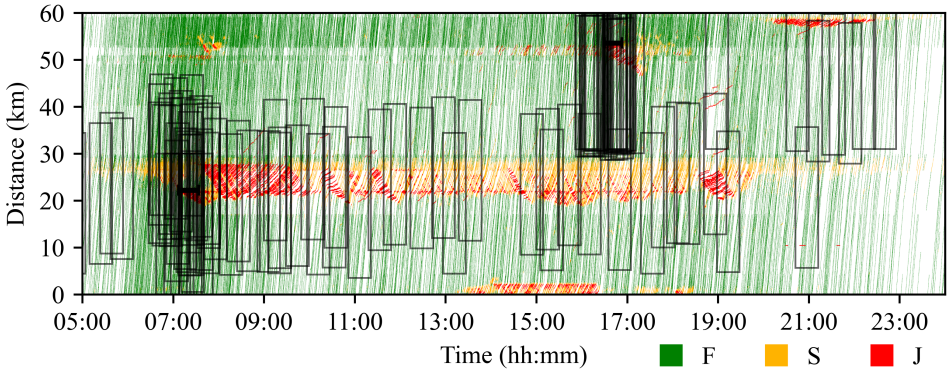
Category	**Event codes**
Traffic state and flow	traffic congestion (1), narrow lanes (4), impassibility (5), delay (29), slow-moving vehicle (26), dangerous end of queue (27)
Crashes and incidents	crash (2), fire (8), broken-down vehicle (13), vehicle on wrong carriageway (14), objects on the road (10), animals on roadway (11), people on roadway (12), reckless person (20), risk of fire (28)
Weather and visibility	slippery road (6), aquaplaning (7), hazardous driving conditions (9), extreme weather conditions (17), visibility reduced (18), precipitation (19)
Traffic management	roadworks (3), rescue and recovery work in progress (15), regulatory measure (16), traffic regulations changed (22), traffic light or signs not working (31), police checkpoint (30), no parking (25), overheight warning system triggered (21)
Events and disruptions	major event (23), service suspended (24)

A.2 Detection Window Sampling

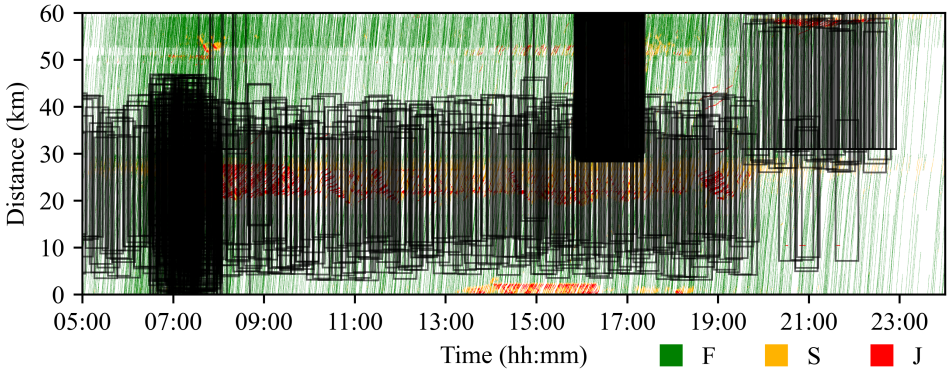
For one road section and day, Figure A.1 shows the three sampling modes' congestion windows, with None (a), 100 (b) and all 763 (c) windows shown for clarity. Crash events are oversampled, low-speed regions less so, and free-flow conditions undersampled.



(a) Phase-colored TSD showing crashes and recurring congestion.



(b) Overlay of 100 randomly sampled windows across the same road section.



(c) Overlay of all randomly sampled windows across the same road section.

Figure A.1: Visualization of the detection window sampling process on the A8 freeway on 7 October 2024.

B Weather-Induced Traffic Breakdown

This appendix collects the full regression models and robustness checks underlying the statistical results reported in Section 4.5.

Table B.1: Logistic regression predicting rain-related congestion among wet observations ($n = 509$). Reported in Section 4.5.1.

Predictor	Coef.	SE	z	p	95 % CI
Intercept	−5.40	1.05	−5.14	< 0.001	[−7.46, −3.34]
log(Intensity)	0.89	0.21	4.21	< 0.001	[0.47, 1.30]
log(Area)	−0.36	0.21	−1.74	0.083	[−0.76, 0.05]

Table B.2: Adjusted logistic regression predicting rain-front constraint ($n = 190$). Upstream direction is the primary predictor; covariates control for congestion severity. Reported in Section 4.5.2.

Predictor	Coef.	SE	z	p	95 % CI
Intercept	6.14	1.53	4.00	< 0.001	[3.13, 9.15]
Upstream (binary)	0.90	0.39	2.32	0.020	[0.14, 1.65]
log(Extent)	−0.62	0.29	−2.17	0.030	[−1.18, −0.06]
log(Duration)	−0.59	0.30	−1.95	0.051	[−1.18, 0.00]
log(Intensity)	−0.79	0.41	−1.91	0.056	[−1.60, 0.02]
log(LRC extent)	0.30	0.25	1.19	0.233	[−0.19, 0.78]

Table B.3: Sensitivity of the direction–constraint association to the velocity threshold used for classifying upstream vs. downstream LRCs. All tests are unadjusted Fisher exact tests. Reported in Section 4.5.2.

Threshold	n	n_{up}	n_{dn}	Constr. up (%)	Constr. dn (%)	OR (p)
±1	228	110	118	78.2	65.3	1.91 (0.040)
±3	203	98	105	77.6	66.7	1.73 (0.089)
±5	190	90	100	76.7	66.0	1.69 (0.112)
±10	135	66	69	77.3	62.3	2.06 (0.064)

Table B.4: Tests for concentration of breakdowns near the dry-to-wet front ($n = 93$, $I > 10\,\mathrm{mm/h}$). The binomial test result is reported in the main text (Section 4.5.3); Wilcoxon and KS tests are listed here for completeness.

Test	Statistic	p	Interpretation
Binomial	57/93 in front half (61.3 %)	0.019	More F→S at front
Wilcoxon signed-rank	$W = 1651$, Cohen's $d = 0.21$	0.020	Front distance < 0.5
Kolmogorov–Smirnov	$D = 0.142$	0.043	Non-uniform distribution

C Rule-Based Detection

C.1 Complete Parameter Configurations

Tables C.1 to C.3 list every parameter for the three Pareto-optimal operating points described in Section 6.3. Table C.4 collects parameters that are shared across all profiles.

C.2 Detailed Ablation Results

Tables C.5 to C.7 report the complete ablation results for each operating point. In each row, a single feature or mechanism is disabled while all remaining parameters are held at their calibrated values.

C.3 Feature-Isolation Experiment

Table C.8 reports the detection performance of each feature in isolation, using a fixed weight of $w = 0.8$ (with `free_flow_ratio` set to $w = 0.4$) and all other features disabled. This complements the ablation study by showing the standalone discriminative power of each indicator.

Cluster dominance stands out as the only feature with high standalone recall (0.815), detecting 53 of 65 crashes at a fast median delay of 5.8 min, albeit with 66 false positives. Downstream velocity achieves high precision (0.789) but low recall (0.462), indicating that it serves as a precision-enhancing supplement rather than a primary detector. Full blockage achieves the highest isolated F_1 (0.630) through a more balanced precision–recall trade-off, but at a median delay of 14.1 min—too slow for early warning. The pre-jam free-flow ratio produces no detections at the tested weight. These isolation results are consistent with the feature-importance ranking observed in the ablation study (Table 6.2).

Table C.1: Complete parameter configuration: Accuracy (Best F_1).

Group	Parameter	Value
Feature weights	Cluster dominance w_{cd}	0.540
	Pre-jam free-flow ratio w_{r_f}	0.220
	Downstream velocity w_{ds}	0.152
	Full blockage w_{fb}	0.146
	Velocity difference w_{vd}	0.040
Penalty weights	Prior WMJ w_{wmj}	−0.500
	Queue w_{queue}	−0.500
	Upstream R^2 w_{R^2}	−0.300
	Growth rate w_{growth}	−0.300
DS velocity scorer	Band v_{band}	0.517 km/h
	Ramp full c_{full}	2
	Max readings	10
Cluster dominance	Look-back Δt_{cd}	881 s
Alert logic	Threshold τ	0.50
Blockage	FB session count	3

Table C.2: Complete parameter configuration: Balanced.

Group	Parameter	Value
Feature weights	Cluster dominance w_{cd}	0.632
	Pre-jam free-flow ratio w_{r_f}	0.232
	Downstream velocity w_{ds}	0.201
	Full blockage w_{fb}	0.151
	Velocity difference w_{vd}	0.032
Penalty weights	Prior WMJ w_{wmj}	−0.700
	Queue w_{queue}	−0.500
	Upstream R^2 w_{R^2}	−0.300
	Growth rate w_{growth}	−0.300
	Avg. jam speed $w_{\text{avg-speed}}$	−0.300
	Ramp range	5–15 km/h
DS velocity scorer	Band v_{band}	0.823 km/h
	Ramp full c_{full}	3
	Max readings	10
Cluster dominance	Look-back Δt_{cd}	300 s
Alert logic	Threshold τ	0.50
Blockage	FB session count	3

Table C.3: Complete parameter configuration: Speed (Fastest).

Group	Parameter	Value
Feature weights	Cluster dominance w_{cd}	0.632
	Pre-jam free-flow ratio w_{r_f}	0.232
	Downstream velocity w_{ds}	0.201
	Full blockage w_{fb}	0.151
	Velocity difference w_{vd}	0.032
Penalty weights	Prior WMJ w_{wmj}	−0.500
	Queue w_{queue}	−0.500
	Upstream R^2 w_{R^2}	−0.300
	Growth rate w_{growth}	−0.300
	Avg. jam speed $w_{\text{avg-speed}}$	−0.400
	Ramp range	5–15 km/h
DS velocity scorer	Band v_{band}	0.823 km/h
	Ramp full c_{full}	3
	Max readings	10
Cluster dominance	Look-back Δt_{cd}	300 s
Alert logic	Threshold τ	0.50
Blockage	FB session count	2

Table C.4: Frozen parameters (shared across all operating points).

Group	Parameter	Value
Plausibility filters	Min. phase age	2.0 min
	Min. upstream R^2	0.0
	Max. jam growth rate	3000 m^2/min
	Min. qualified trajectories	3
Minimum queue	Min. vehicles	2
	Standstill threshold v_q	1.3 km/h
Score multiplier γ		1.0
Prior WMJ window	Distance	3000 m
	Time	1800 s
Pre-jam window	Distance	2500 m
	Time	1200 s
Scoring ranges	Free-flow ratio r_f	$[0.50, 0.95]$
	Upstream velocity	$[-10.0, 2.0]$ km/h
Blockage (shared)	FB standstill velocity	0.5 km/h
	FB duration	1 min
	FB min. segments	5

Table C.5: Ablation: Accuracy (Best F_1). Baseline: $F_1 = 0.750$, TP = 48, FP = 15, median delay = 9.8 min.

Setting	Time (s)	F_1	TP	FP	Med (min)	ΔF_1	ΔFP
Full (baseline)	109	0.750	48	15	9.8	+0.000	+0
w/o cluster dom	166	0.060	2	0	17.3	−0.690	−15
w/o free_flow_ratio	157	0.762	48	13	10.7	+0.012	−2
w/o ds velocity	161	0.736	46	14	10.3	−0.014	−1
w/o full blockage	146	0.708	40	8	9.8	−0.042	−7
w/o velocity diff	166	0.727	44	12	9.8	−0.023	−3
w/o all penalties	160	0.556	57	83	10.6	−0.194	+68
w/o avg speed penalty	166	0.750	48	15	9.8	+0.000	+0

Table C.6: Ablation: Balanced. Baseline: $F_1 = 0.718$, TP = 47, FP = 19, median delay = 8.1 min.

Setting	Time (s)	F_1	TP	FP	Med (min)	ΔF_1	ΔFP
Full (baseline)	150	0.718	47	19	8.1	+0.000	+0
w/o cluster dom	159	0.116	4	0	20.3	−0.602	−19
w/o free_flow_ratio	157	0.726	45	14	9.0	+0.008	−5
w/o ds velocity	165	0.682	43	18	7.5	−0.035	−1
w/o full blockage	146	0.704	44	16	8.5	−0.014	−3
w/o velocity diff	166	0.698	45	19	7.7	−0.020	+0
w/o all penalties	172	0.348	58	210	8.2	−0.369	+191
w/o avg speed penalty	166	0.699	51	30	8.0	−0.019	+11

Table C.7: Ablation: Speed (Fastest). Baseline: $F_1 = 0.653$, TP = 49, FP = 36, median delay = 6.1 min.

Setting	Time (s)	F_1	TP	FP	Med (min)	ΔF_1	ΔFP
Full (baseline)	169	0.653	49	36	6.1	+0.000	+0
w/o cluster dom	154	0.088	3	0	24.1	−0.565	−36
w/o free_flow_ratio	167	0.676	48	29	7.4	+0.023	−7
w/o ds velocity	165	0.657	47	31	6.1	+0.004	−5
w/o full blockage	146	0.667	47	29	6.5	+0.013	−7
w/o velocity diff	166	0.649	48	35	6.1	−0.005	−1
w/o all penalties	178	0.291	58	275	6.7	−0.362	+239
w/o avg speed penalty	167	0.614	54	57	6.4	−0.040	+21

Table C.8: Performance of individual features in isolation (fixed $w = 0.8$, threshold $\tau = 0.50$). Evaluated on the independent test set (65 ground-truth crashes).

Feature	F_1	**P**	**R**	**TP**	**FP**	**Med (min)**
free_flow_ratio only	0.000	0.000	0.000	0	0	—
cluster dom ($w = 0.8$)	0.576	0.445	0.815	53	66	5.8
ds velocity ($w = 0.8$)	0.583	0.789	0.462	30	8	12.2
full blockage ($w = 0.8$)	0.630	0.645	0.615	40	22	14.1
velocity diff ($w = 0.8$)	0.432	0.450	0.415	27	33	7.0

C.4 Confusion Matrices and Extended Timeliness Statistics

All three operating points are evaluated on an independent test set containing 65 ground-truth crash events across 50 datasets, yielding 2152 total jam phases of which 1737 have ≥ 3 qualified trajectories.

Table C.9: Confusion matrices and extended detection metrics for the three operating points.

Metric	Accuracy	Balanced	Speed
TP	48	47	49
FP	15	19	36
FN	17	18	16
TN	1657	1653	1636
Precision	0.762	0.712	0.577
Recall	0.738	0.723	0.754
Specificity	0.991	0.989	0.978
FPR	0.009	0.011	0.022
Accuracy	0.982	0.979	0.970
F_1	0.750	0.718	0.653
F_2	0.743	0.721	0.710
Mean delay	13.0 min	12.9 min	10.7 min
Median delay	9.8 min	8.1 min	6.1 min
< 7 min	12/48	17/47	32/49
< 10 min	27/48	31/47	39/49
25th pctl	7.1 min	5.9 min	4.9 min
75th pctl	16.7 min	15.5 min	13.9 min
Min delay	3.1 min	2.7 min	2.4 min
Max delay	47.4 min	58.6 min	47.1 min

D Learning-Based Crash Detection

D.1 Implementation Details

D.1.1 PSM Inference Pipeline

Algorithm 1 PSM Inference Pipeline [159, 175]

Input: Velocity map $V \in \mathbb{R}^{p_d \times p_t}$, phase map $P_{\text{CNN}} \in \{0,1,2,3\}^{p_d \times p_t}$ (0 = missing)
Output: Completed input tensor ${\boldsymbol{\mathcal{X}}_{\text{CNN}}}^{\text{psm}}$
Estimate phase probabilities: $\mathbf{Q} \leftarrow \texttt{EstimatePhaseProb}(P_{\text{CNN}})$
Initialize valid mask: $\mathbf{M}_{\text{valid}} \leftarrow \mathbf{1}[P_{\text{CNN}} > 0]$
Initialize reciprocal speed: $V^{-1}(i,j) \leftarrow 1/\max(v_{\min}, V(i,j))$ where valid, else 0
$k \leftarrow 0$
while new pixels filled and $k <$ `max_iters` **do**
    for $\Phi \in \{\text{F}, \text{S}, \text{J}\}$ **do**
        Convolve: $\tilde{V}_{\Phi}^{-1} \leftarrow \mathbf{K}_{\Phi} * V^{-1}, \quad \text{Norm}_{\Phi} \leftarrow \mathbf{K}_{\Phi} * \mathbf{M}_{\text{valid}}$
        Normalize: $\hat{V}_{\Phi}^{-1}(i,j) \leftarrow \tilde{V}_{\Phi}^{-1}(i,j)/\text{Norm}_{\Phi}(i,j)$ if $\text{Norm}_{\Phi}(i,j) > 0$, else 0
    Fuse: $\hat{V}^{-1}(i,j) \leftarrow \sum_{\Phi} \mathbf{q}_{m,ni,j,\Phi} \cdot \hat{V}_{\Phi}^{-1}(i,j) + V^{-1}(i,j)$
    Update mask: $\mathbf{M}_{\text{valid}} \leftarrow \mathbf{M}_{\text{valid}} \vee \mathbf{1}[\hat{V}^{-1} > 0]$
    $k \leftarrow k + 1$
Smooth reciprocal velocity with Gaussian filter: $\hat{V}^{-1} \leftarrow \texttt{GaussianSmooth}(\hat{V}^{-1})$
Finalize phase labels: $P_{\text{CNN}}(i,j) \leftarrow \arg\max_{\Phi} \mathbf{Q}_{i,j,\Phi}$ if $P_{\text{CNN}}(i,j) = 0$
One-hot encode: $\mathbf{P}_{\text{one-hot}} \leftarrow \texttt{OneHot}(P_{\text{CNN}})$
Assemble completed tensor: ${\boldsymbol{\mathcal{X}}_{\text{CNN}}}^{\text{psm}} \leftarrow \texttt{Concat}(\hat{V}^{-1}, \mathbf{P}_{\text{one-hot}})$
return ${\boldsymbol{\mathcal{X}}_{\text{CNN}}}^{\text{psm}}$

D.1.2 Regularization Details

Cross-trajectory subsampling. To regularize training and limit graph size, each spatio–temporal window $\mathcal{W}_{D,T}$ is independently subsampled along the trajectory dimension. Let $n = |\mathcal{C}|$ denote the number of unique vehicles in $\mathcal{W}_{D,T}$, with empirical mean $\mu \approx 121.2$ and standard deviation $\sigma \approx 55.2$. Windows with $n < \mu$ are left intact; for larger windows the

retention probability decays smoothly as a function of the z-scored vehicle count:

$$z = \frac{n - \mu}{\sigma}\,, \tag{D.1}$$

$$p(n) \;=\; \mathrm{clip}\Big(p_{\max} - \lambda\, z \;+\; \epsilon,\; p_{\min},\; p_{\max}\Big), \qquad \epsilon \sim \mathcal{U}(-\delta,\, \delta)\,, \tag{D.2}$$

where $p_{\max} = 1.0$ is the maximum retention rate, $p_{\min} = 0.4$ is the floor, $\lambda = 0.2$ controls the decay per unit z-score, and $\delta = 0.05$ introduces stochastic jitter so that repeated epochs observe different subgraphs of the same window. The retained vehicle set is then

$$\mathcal{C}_{\mathrm{keep}} = \begin{cases} \mathcal{C}, & n < \mu, \\ \mathrm{RandomSubset}\big(\mathcal{C},\, \lfloor\, p(n)\, n\, \rfloor\big), & n \geq \mu, \end{cases} \tag{D.3}$$

where $\mathrm{RandomSubset}(\mathcal{C}, k)$ samples k vehicles uniformly without replacement. Because both ϵ and the random subset are resampled every epoch, the model observes a different subgraph of the same window across training iterations, acting as a data-level form of dropout.

Design rationale and trade-offs. Windows at or below the population mean pass through unaltered, so that roughly half of all training samples retain their full trajectory set.

A notable consequence of this strategy is that it attenuates the correlation between vehicle count and the observation window's traffic density. One could argue that higher vehicle penetration is itself an informative signal—denser traffic conditions are empirically associated with elevated crash risk—and that subsampling erases this signal. In a *prediction* setting, where the goal is to forecast whether a crash will occur, this concern would be significant: the model would lose access to a potentially causal feature.

In the present *detection* setting, however, the objective is to identify whether a crash has already manifested within the current window. The relevant evidence therefore lies in the kinematic and structural signatures of the trajectories that *are* observed, rather than in the absolute count of observed vehicles. Subsampling thus serves a genuinely regularizing purpose: it forces the model to detect crashes from partial observations, improving robustness to varying probe-vehicle penetration rates that are inevitable in real-world floating-car data. Simultaneously, it prevents the model from relying on vehicle density as a shortcut feature that may not transfer across different road segments, time-of-day regimes, or fleet compositions.

D.2 GNN Architecture Details

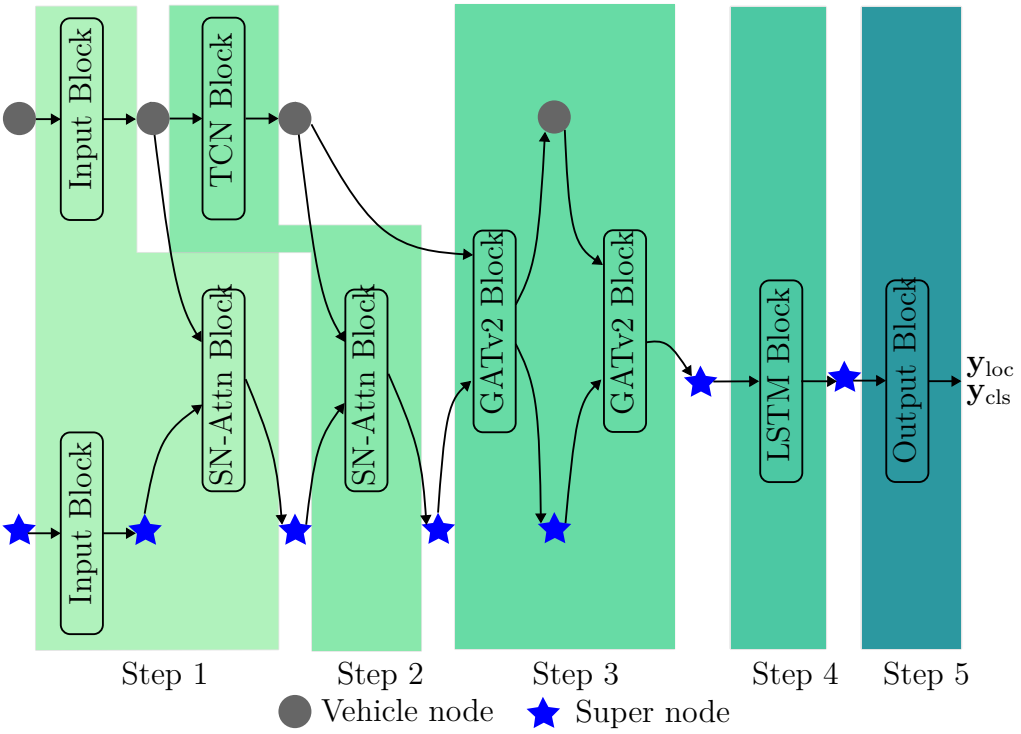


Figure D.1: Node flow of vehicle nodes and supernodes through the GNN model (Section 7.3.2) [175].

D.3 CNN Ablation Details

Table D.1: CNN: Validation results with different base channel widths d_{base} (threshold = 0.5) [175].

d_{base}	Val Loss	Precision	Recall	Accuracy	F1-score
8 (1 run)	0.4022	79.6%	87.6%	**82.5%**	81.1%
12 (3 runs)	**0.3758**	80.8%	**91.3%**	**82.5%**	**81.7%**
16 (1 run)	0.3913	**81.7%**	90.5%	**82.5%**	81.5%

Table D.2: CNN: Per-run validation results of ablation variants (threshold = 0.5) [175].

Variant	Run	Val Loss	Precision	Recall	Accuracy	F1-score
a	1	0.3783	80.44%	**92.70%**	82.41%	**81.88%**
a	2	0.3765	**84.08%**	89.95%	**82.70%**	81.43%
a	3	**0.3728**	77.81%	91.26%	82.46%	81.75%
b	1	0.3758	81.60%	89.05%	83.99%	82.97%
b	2	**0.3516**	**83.31%**	91.98%	**84.80%**	**84.11%**
b	3	0.3863	82.17%	**94.31%**	83.80%	83.37%
c	1	0.4017	82.61%	89.59%	82.89%	**82.46%**
c	2	**0.3658**	**83.00%**	**98.68%**	**83.62%**	82.05%
c	3	0.3903	78.11%	97.13%	82.68%	81.92%
d	1	0.3776	**83.25%**	**93.66%**	84.31%	83.15%
d	2	**0.3653**	82.08%	89.95%	**84.64%**	**83.34%**
d	3	0.3744	83.17%	91.32%	84.37%	82.86%
e	1	0.3572	83.78%	**92.10%**	**85.33%**	**84.43%**
e	2	**0.3536**	**84.00%**	90.37%	84.72%	83.52%
e	3	0.3685	82.08%	89.29%	84.56%	83.20%

Table D.3: GNN: Validation results with different hidden dimensions d_{hid} (threshold = 0.5) [175].

d_{hid}	Val Loss	Precision	Recall	Accuracy	F1-score
96 (1 run)	**0.2608**	**75.6%**	**84.5%**	89.14%	72.7%
128 (1 run)	0.2897	71.4%	83.7%	89.15%	**72.9%**
512 (1 run)	0.2703	72.1%	82.8%	**89.19%**	71.7%

Table D.4: GNN: Validation results with different loss weights for minority samples (threshold = 0.5) [175].

Loss Weight	Val Loss	Precision	Recall	Accuracy	F1-score
2 (1 run)	0.2714	**80.6%**	73.3%	88.80%	69.8%
3 (1 run)	**0.2608**	75.6%	84.5%	**89.14%**	**72.7%**
4 (1 run)	0.2772	73.7%	**88.8%**	88.45%	71.8%

Algorithm 2 Spatiotemporal graph construction [175].

Input: window $\mathcal{W}_{D,T}$; supernode anchors $(\tilde{t}_m, \tilde{d}_n)$; hop depth K_{hop}
Initialize supernode set $\{\tilde{\mathbf{x}}_{m,n}\}$
$\mathcal{V} \leftarrow \{\text{all vehicle nodes in } \mathcal{W}_{D,T}\} \cup \{\tilde{\mathbf{x}}_{m,n}\}$
Compute $\mathcal{E}_{\text{super}}$ by connecting vehicle nodes within each $\mathcal{W}_{mn}$
$\mathcal{F}_1 \leftarrow$ vehicle nodes incident to $\mathcal{E}_{\text{super}}$; $\mathcal{E} \leftarrow \mathcal{E}_{\text{super}}$; $\mathcal{F}_{\text{used}} \leftarrow \mathcal{F}_1$
for $k = 1$ **to** K_{hop} **do**
    Construct ${\mathcal{E}_{\text{spatial}}}^{(k)}$ and ${\mathcal{E}_{\text{WMJ}}}^{(k)}$ from nodes in $\mathcal{F}_k$
    $\mathcal{F}_{k+1} \leftarrow$ newly reached nodes $\setminus \mathcal{F}_{\text{used}}$
    $\mathcal{F}_{\text{used}} \leftarrow \mathcal{F}_{\text{used}} \cup \mathcal{F}_{k+1}$; $\mathcal{E} \leftarrow \mathcal{E} \cup {\mathcal{E}_{\text{spatial}}}^{(k)} \cup {\mathcal{E}_{\text{WMJ}}}^{(k)}$
Deduplicate $\mathcal{E}$; add self-loops
Build $\mathcal{L}_{\text{car}}$ and $\mathcal{L}_{\text{super}}$
Output: $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$

D.4 GNN Ablation Details

D.4.1 Hyperparameter Selection

D.4.2 Design Choice Ablations

Table D.5: Multi-output vs. single-output supervision (threshold = 0.5) [175].

Variant	Val Loss	Precision	Recall	F1
Multi-output	$\mathbf{0.3041_{\pm 0.0062}}$	$\mathbf{86.00\%_{\pm 1.94}}$	$\mathbf{92.15\%_{\pm 1.78}}$	$\mathbf{88.94\%_{\pm 0.27}}$
Single-output	$0.3218_{\pm 0.0057}$	$85.14\%_{\pm 0.44}$	$90.78\%_{\pm 1.08}$	$87.87\%_{\pm 0.35}$

Table D.6: Temporal anchor distribution ablation (threshold = 0.5) [175].

Setting	Val Loss	Precision	Recall	F1
Front	$0.3412_{\pm 0.0035}$	$84.40\%_{\pm 1.24}$	$88.83\%_{\pm 1.99}$	$86.54\%_{\pm 0.32}$
Late	$0.3633_{\pm 0.0047}$	$81.73\%_{\pm 0.98}$	$90.76\%_{\pm 1.35}$	$86.00\%_{\pm 0.08}$
Uniform	$\mathbf{0.3041_{\pm 0.0063}}$	$\mathbf{86.00\%_{\pm 1.93}}$	$\mathbf{92.15\%_{\pm 1.78}}$	$\mathbf{88.94\%_{\pm 0.27}}$

Table D.7: Supernode edge processing ablation (threshold = 0.5) [175].

Variant	Val Loss	Precision	Recall	F1
Merged (parallel)	**$0.3041_{\pm 0.0062}$**	**$86.00\%_{\pm 1.94}$**	**$92.15\%_{\pm 1.78}$**	**$88.94\%_{\pm 0.27}$**
Separated (sequential)	$0.3123_{\pm 0.0031}$	$85.06\%_{\pm 0.28}$	$92.06\%_{\pm 0.51}$	$88.42\%_{\pm 0.24}$

Table D.8: GNN: Feature ablation results (Groups A–G, threshold = 0.6) [175].

Group	Val Loss	Precision	Recall	Accuracy	F1-score
A (Baseline)	**$0.2995_{\pm 0.0060}$**	**$88.24\%_{\pm 0.41}$**	$88.79\%_{\pm 1.53}$	$88.00\%_{\pm 0.45}$	$88.51\%_{\pm 0.55}$
B	$0.3142_{\pm 0.0035}$	$87.55\%_{\pm 0.49}$	$87.72\%_{\pm 0.77}$	$87.10\%_{\pm 0.21}$	$87.63\%_{\pm 0.24}$
C	$0.3005_{\pm 0.0091}$	$87.95\%_{\pm 0.45}$	**$89.25\%_{\pm 1.39}$**	**$88.04\%_{\pm 0.83}$**	**$88.60\%_{\pm 0.87}$**
D	$0.3225_{\pm 0.0093}$	$87.65\%_{\pm 0.37}$	$86.13\%_{\pm 1.28}$	$86.46\%_{\pm 0.74}$	$86.88\%_{\pm 0.79}$
E	$0.3041_{\pm 0.0062}$	$88.05\%_{\pm 1.86}$	$89.07\%_{\pm 2.64}$	$87.98\%_{\pm 0.18}$	$88.52\%_{\pm 0.39}$
F	$0.3238_{\pm 0.0146}$	$87.73\%_{\pm 1.51}$	$86.47\%_{\pm 4.83}$	$86.61\%_{\pm 1.31}$	$87.01\%_{\pm 1.76}$
G	$0.3132_{\pm 0.0093}$	$87.12\%_{\pm 1.23}$	$88.99\%_{\pm 1.10}$	$87.40\%_{\pm 0.27}$	$88.04\%_{\pm 0.11}$

Table D.9: GNN: Feature ablation results (Groups A–G, threshold = 0.7) [175].

Group	Val Loss	Precision	Recall	Accuracy	F1-score
A (Baseline)	**$0.2995_{\pm 0.0060}$**	**$90.34\%_{\pm 0.55}$**	$84.00\%_{\pm 2.58}$	**$86.98\%_{\pm 0.90}$**	**$87.03\%_{\pm 1.12}$**
B	$0.3142_{\pm 0.0035}$	$89.45\%_{\pm 0.35}$	$82.97\%_{\pm 0.53}$	$86.04\%_{\pm 0.19}$	$86.09\%_{\pm 0.22}$
C	$0.3005_{\pm 0.0091}$	$90.08\%_{\pm 0.39}$	**$83.63\%_{\pm 1.69}$**	$86.69\%_{\pm 0.97}$	$86.73\%_{\pm 1.08}$
D	$0.3225_{\pm 0.0093}$	$89.35\%_{\pm 0.45}$	$80.55\%_{\pm 0.81}$	$84.87\%_{\pm 0.54}$	$84.72\%_{\pm 0.59}$
E	$0.3041_{\pm 0.0062}$	$89.96\%_{\pm 1.38}$	$83.89\%_{\pm 3.35}$	$86.70\%_{\pm 0.81}$	$86.77\%_{\pm 1.16}$
F	$0.3238_{\pm 0.0146}$	$89.68\%_{\pm 0.93}$	$82.00\%_{\pm 6.76}$	$85.68\%_{\pm 2.65}$	$85.54\%_{\pm 3.36}$
G	$0.3132_{\pm 0.0093}$	$88.77\%_{\pm 1.10}$	$85.22\%_{\pm 1.68}$	$86.67\%_{\pm 0.26}$	$86.94\%_{\pm 0.41}$

Table D.10: GNN: Feature ablation per-run results (Groups A–G, threshold = 0.5) [175].

Group	Run	Val Loss	Precision	Recall	Accuracy	F1-score
A	1	0.3010	86.37%	90.95%	87.81%	88.60%
A	2	**0.2929**	86.13%	**92.59%**	**88.38%**	**89.25%**
A	3	0.3045	**86.46%**	91.13%	87.95%	88.74%
B	1	**0.3108**	85.47%	**91.45%**	**87.45%**	**88.36%**
B	2	0.3139	**86.53%**	89.89%	**87.45%**	88.18%
B	3	0.3178	85.76%	90.86%	87.38%	88.23%
C	1	0.3009	85.60%	92.68%	88.07%	89.00%
C	2	**0.2912**	**86.42%**	**92.82%**	**88.67%**	**89.51%**
C	3	0.3094	85.40%	91.22%	87.31%	88.22%
D	1	**0.3139**	**86.06%**	**90.63%**	**87.48%**	**88.29%**
D	2	0.3211	85.94%	88.84%	86.62%	87.37%
D	3	0.3324	85.56%	88.84%	86.38%	87.17%
E	1	0.3060	83.97%	**94.15%**	87.60%	88.77%
E	2	**0.2971**	**87.83%**	90.72%	**88.62%**	**89.25%**
E	3	0.3091	86.19%	91.59%	87.98%	88.81%
F	1	0.3223	85.99%	90.12%	**87.21%**	88.01%
F	2	0.3391	**87.59%**	86.15%	86.43%	86.86%
F	3	**0.3100**	84.38%	**92.36%**	87.12%	**88.19%**
G	1	0.3180	85.14%	**91.72%**	87.36%	88.31%
G	2	**0.3025**	**86.44%**	90.67%	**87.74%**	**88.51%**
G	3	0.3190	84.86%	**91.72%**	87.17%	88.16%

Table D.11: GNN: Output head ablation per-run results (threshold = 0.5) [175].

Variant	Run	Val Loss	Precision	Recall	Accuracy	F1-score
Multi-output	1	0.3060	83.97%	**94.15%**	87.60%	88.77%
Multi-output	2	**0.2971**	**87.83%**	90.72%	**88.62%**	**89.25%**
Multi-output	3	0.3091	86.19%	91.59%	87.98%	88.81%
Single-output	1	0.3188	84.68%	**92.00%**	**87.17%**	**88.19%**
Single-output	2	0.3284	85.19%	89.94%	86.62%	87.50%
Single-output	3	**0.3182**	**85.55%**	90.40%	87.05%	87.91%

Table D.12: GNN: Temporal anchor distribution per-run results (threshold = 0.5) [175].

Setting	Run	Val Loss	Precision	Recall	Accuracy	F1-score
Front	1	**0.3384**	**85.73%**	86.63%	85.79%	86.17%
Front	2	0.3402	83.28%	**90.51%**	85.86%	**86.75%**
Front	3	0.3451	84.18%	89.37%	**85.97%**	86.69%
Late	1	0.3646	80.61%	**92.30%**	85.01%	**86.06%**
Late	2	0.3672	**82.30%**	90.13%	**85.34%**	86.04%
Late	3	**0.3581**	82.29%	89.85%	85.22%	85.90%
Uniform	1	0.3060	83.97%	**94.15%**	87.60%	88.77%
Uniform	2	**0.2971**	**87.83%**	90.72%	**88.62%**	**89.25%**
Uniform	3	0.3091	86.19%	91.59%	87.98%	88.81%

Table D.13: GNN: Supernode edge processing per-run results (threshold = 0.5) [175].

Variant	Run	Val Loss	Precision	Recall	Accuracy	F1-score
Merged (parallel)	1	0.3060	83.97%	**94.15%**	87.60%	88.77%
Merged (parallel)	2	**0.2971**	**87.83%**	90.72%	**88.62%**	**89.25%**
Merged (parallel)	3	0.3091	86.19%	91.59%	87.98%	88.81%
Separated (sequential)	1	0.3112	**85.36%**	91.68%	87.48%	88.40%
Separated (sequential)	2	0.3158	84.80%	91.86%	87.19%	88.19%
Separated (sequential)	3	**0.3100**	85.02%	**92.64%**	**87.67%**	**88.67%**

Table D.14: GNN: Module-wise ablation per-run results (threshold = 0.5) [175].

Variant	**Run**	**Val Loss**	**Precision**	**Recall**	**Accuracy**	**F1-score**
Full GNN pipeline	1	0.3060	83.97%	**94.15%**	87.60%	88.77%
Full GNN pipeline	2	**0.2971**	**87.83%**	90.72%	**88.62%**	**89.25%**
Full GNN pipeline	3	0.3091	86.19%	91.59%	87.98%	88.81%
Without SN-Attn Block	1	0.3645	82.89%	90.37%	84.26%	83.35%
Without SN-Attn Block	2	**0.3503**	**86.27%**	**91.26%**	**84.81%**	**83.96%**
Without SN-Attn Block	3	0.3656	85.10%	89.53%	84.79%	83.07%
Without TCN Block	1	0.3892	79.00%	83.20%	82.00%	81.05%
Without TCN Block	2	0.3840	77.90%	**86.40%**	82.40%	**81.93%**
Without TCN Block	3	**0.3813**	**81.70%**	80.80%	**82.80%**	81.25%
Without $\mathcal{E}_{\text{spatial}}$	1	0.3568	85.01%	**90.54%**	84.34%	83.36%
Without $\mathcal{E}_{\text{spatial}}$	2	0.3395	85.75%	85.40%	85.70%	84.59%
Without $\mathcal{E}_{\text{spatial}}$	3	**0.3363**	**86.01%**	88.51%	**85.81%**	**84.92%**
Without $\mathcal{E}_{\text{WMJ}}$	1	**0.3459**	**84.37%**	**88.45%**	**84.32%**	**83.91%**
Without $\mathcal{E}_{\text{WMJ}}$	2	0.3526	82.72%	87.37%	84.04%	83.36%
Without $\mathcal{E}_{\text{WMJ}}$	3	0.3641	83.88%	84.56%	83.49%	82.36%
Without $\mathcal{E}_{\text{super}}$	1	0.3628	**84.72%**	90.37%	**84.26%**	**82.76%**
Without $\mathcal{E}_{\text{super}}$	2	0.3752	83.43%	**93.24%**	83.46%	82.74%
Without $\mathcal{E}_{\text{super}}$	3	**0.3615**	84.05%	90.01%	83.51%	82.66%
Without LSTM Block	1	0.3427	85.75%	85.76%	85.53%	84.57%
Without LSTM Block	2	**0.3298**	**87.21%**	88.99%	85.62%	84.77%
Without LSTM Block	3	0.3313	86.22%	**89.95%**	**85.64%**	**85.12%**
Without Transformer Head	1	0.3456	**86.77%**	87.37%	84.79%	83.88%
Without Transformer Head	2	0.3511	86.23%	89.53%	85.06%	84.55%
Without Transformer Head	3	**0.3402**	85.82%	**90.48%**	**85.59%**	**84.99%**

Algorithm 3 BFD-MinCount batch packing [175].

Input: graph samples $\{\mathcal{G}^{(i)}\}_{i=1}^{N_{\text{samp}}}$ with node counts $|\mathcal{V}^{(i)}|$; batch capacity B_{cap}
Shuffle sample order: $I \leftarrow \texttt{shuffle}([1, \ldots, N_{\text{samp}}])$
Initialize batch layout: $\mathcal{B} \leftarrow [\,]$
for each $i \in I$ **do**
    Select candidate batches: $\mathcal{C} \leftarrow \left\{ b \in \mathcal{B} \;\middle|\; \sum_{\mathcal{G}^{(j)} \in b} |\mathcal{V}^{(j)}| + |\mathcal{V}^{(i)}| \leq B_{\text{cap}} \right\}$
    if $\mathcal{C} \neq \emptyset$ **then**
        Compute minimal graph count: $n_{\min} \leftarrow \min \{ \texttt{len}(b) \mid b \in \mathcal{C} \}$
        Extract tie set: $\mathcal{C}' \leftarrow \{ b \in \mathcal{C} \mid \texttt{len}(b) = n_{\min} \}$
        Select batch with max space: $b^\star \leftarrow \arg\max_{b \in \mathcal{C}'} \left(B_{\text{cap}} - \sum_{\mathcal{G}^{(j)} \in b} |\mathcal{V}^{(j)}| \right)$
        Assign sample: $b^\star \leftarrow b^\star \cup \{\mathcal{G}^{(i)}\}$
    else
        Create new batch: $b_{\text{new}} \leftarrow \{\mathcal{G}^{(i)}\}$
        Append batch: $\mathcal{B} \leftarrow \mathcal{B} \cup \{b_{\text{new}}\}$
Output: batch layout $\mathcal{B}$ where each batch satisfies $\sum_{\mathcal{G}^{(j)} \in b} |\mathcal{V}^{(j)}| \leq B_{\text{cap}}$

D.4.3 Feature Ablation at Additional Thresholds

D.4.4 Per-Run Validation Results

D.5 Exponential Decay Smoothing Function

To stabilize short-term fluctuations in model outputs while maintaining causality, all classification results are based on exponentially smoothed prediction sequences. At each time step t_i, the smoothed prediction $\tilde{y}_i$ is computed as a weighted average over the preceding 5 minutes of predictions:

$$\tilde{y}_i = \frac{\displaystyle\sum_{j:\, 0 \leq t_i - t_j \leq \Delta t_{\text{win}}} \exp\left(-\frac{t_i - t_j}{\tau_{\text{smooth}}} \right) y_j}{\displaystyle\sum_{j:\, 0 \leq t_i - t_j \leq \Delta t_{\text{win}}} \exp\left(-\frac{t_i - t_j}{\tau_{\text{smooth}}} \right)},$$

where $\Delta t_{\text{win}} = 5\,\text{min}$ defines the smoothing window and $\tau_{\text{smooth}} = 2\,\text{min}$ is the exponential time constant controlling the rate of decay. Larger τ_{smooth} values result in slower decay (stronger temporal averaging), while smaller τ_{smooth} values make the smoothing more responsive to recent changes. This formulation ensures causal processing, as each smoothed value $\tilde{y}_i$ depends only on past predictions $\{y_j \,|\, t_j \leq t_i\}$ [175].

E Crash Detection Comparison

This appendix provides supplementary material for the comparison of crash detection approaches presented in Chapter 8.

E.1 Prediction Workflow

This section details the sliding-window inference pipeline summarized in Section 8.2, illustrated on a representative crash case on the A6 freeway between Heilbronn and Mannheim on October 10, 2024, following [175]. The workflow proceeds in three stages: window sampling, approach-specific input generation, and model inference.

Step 1: Window sampling. The observation interval is defined from 06:00 to 19:00 within the selected freeway record. With $\mathcal{W}_{D,T}$ spanning $D = 30\,\text{km}$ and $T = 30\,\text{min}$, a sequence of consecutive spatiotemporal windows is generated using a sliding stride of 10 s. Figure E.1 visualizes seven representative windows extracted at 08:00, 10:00, 12:00, 14:00, 15:00, 16:00, and 18:00. In practice, the full 12-hour interval yields a much denser series of windows continuously covering the entire period.

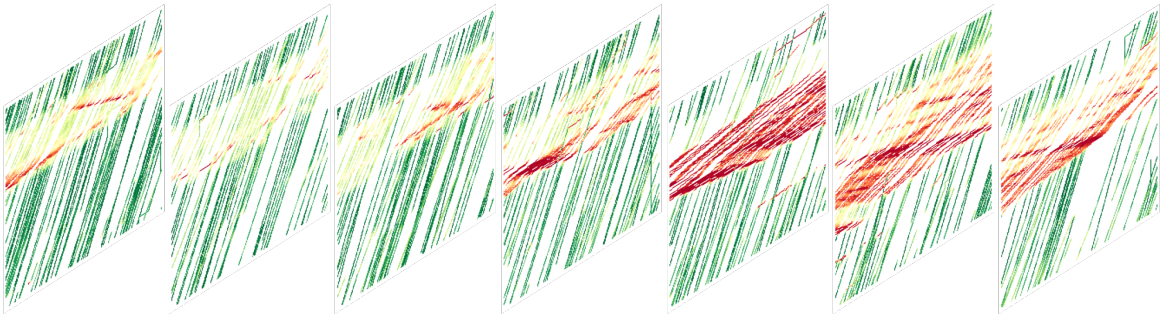


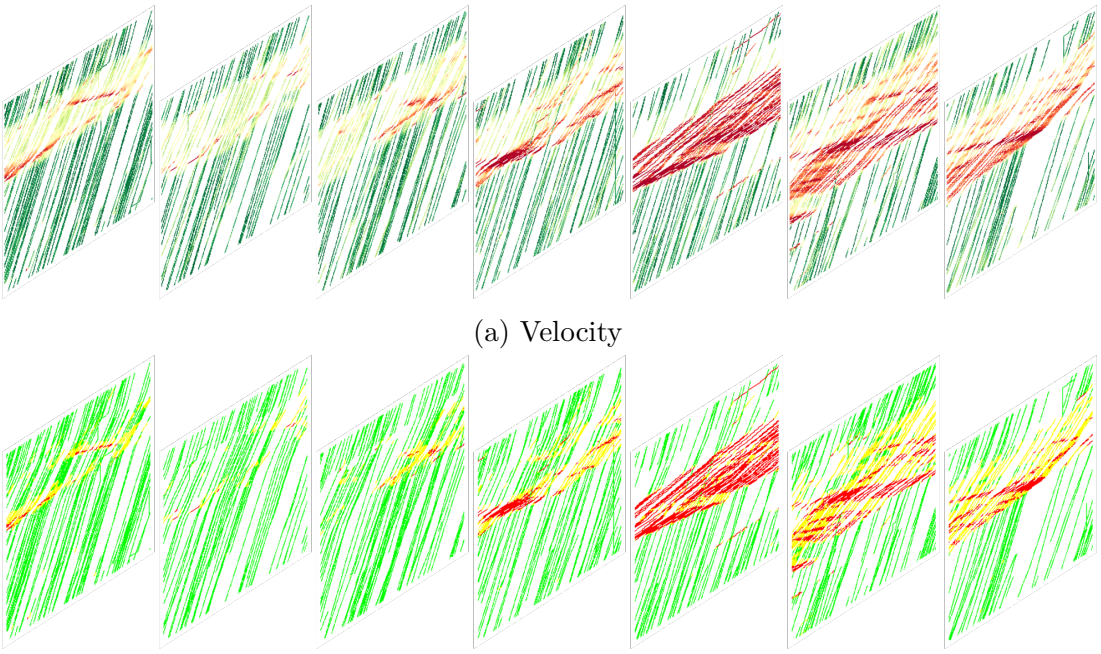
Figure E.1: Sampling of spatiotemporal observation windows for the A6 crash case. Seven representative windows are shown; in practice a new 30 km × 30 min window is generated every 10 s [175].

Step 2: Model input generation. For each sampled window, inputs are constructed according to the representation expected by each approach. All four pipelines share the same underlying observation $\mathcal{W}_{D,T}$ but differ in how the enclosed trajectories are structured.

- **Rule-based.** The rule-based approach operates on the phase-labeled trajectory data on region level. Each vehicle trajectory is classified into traffic phases (F, S, J) according to the thresholds defined in Chapter 6. The detection logic then evaluates conditions on the spatial extent, temporal persistence, and propagation characteristics of the identified congested phases. No grid discretization or graph construction is required.
- **GNN.** In the graph-based pipeline, all vehicle trajectories intersecting $\mathcal{W}_{D,T}$ are converted into a unified graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$, where $\mathcal{V}$, $\mathcal{E}$, and $\mathcal{L}$ denote nodes, edges, and temporal sequences, respectively.
- **CNN (PSM).** The PSM method (Section 7.2.2) completes the sparse reciprocal-velocity map using the phase map as auxiliary guidance. The resulting dense tensor serves as the CNN input (Figure E.3).
- **CNN (PartialConv).** This model operates directly on the sparse input without explicit completion. Sparsity is addressed implicitly through the partial-convolution mechanism (Section 7.2.2). The input consists of the reciprocal-velocity channel, the one-hot encoded phase tensor, and a binary mask (Figure E.2).

Step 3: Model inference. The processed inputs are fed into the corresponding models for crash classification and localization. Each deep learning pipeline consists of two independently trained networks: a binary classifier producing a crash probability and a regression model predicting the crash location within $\mathcal{W}_{D,T}$. The rule-based approach evaluates its detection rules sequentially and, upon a positive detection, reports the upstream seed of the identified congestion pattern as the crash location. The per-window outputs are aggregated across the sliding-window sequence to produce record-level predictions, which form the basis of the quantitative comparison in Section 8.3.

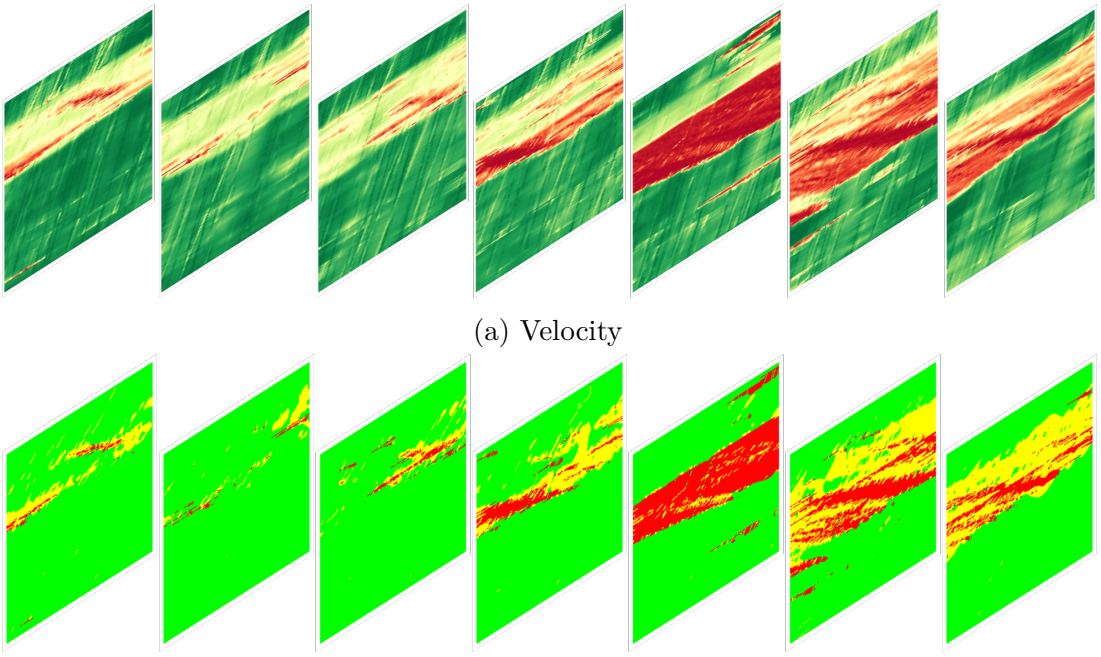
E.2 Additional Material



(a) Velocity

(b) Phase

Figure E.2: Partial-convolution inputs across the seven representative windows: sparse velocity channel and one-hot encoded phase tensor derived from the TSDs [175].



(a) Velocity

(b) Phase

Figure E.3: PSM inputs across the seven representative windows: completed velocity map and associated phase tensor derived from the TSDs [175].

Table E.1: Top ten Pareto-optimal operating points per detector, sampled evenly along the achievable F_1 range. The deep-learning detectors are listed by score-threshold τ on a single trained model; the rule-based detector is listed by v10 calibration trial number, since each Pareto-optimal point uses a different parameter set rather than a different threshold.

Operating point	F_1	median delay (min)
GNN		
$\tau = 0.050$	0.548	2.18
$\tau = 0.135$	0.656	3.25
$\tau = 0.150$	0.686	4.40
$\tau = 0.195$	0.698	5.55
$\tau = 0.255$	0.724	6.43
$\tau = 0.310$	0.747	7.56
$\tau = 0.320$	0.756	7.64
$\tau = 0.510$	0.777	11.34
$\tau = 0.520$	0.788	11.59
$\tau = 0.630$	0.820	20.18
CNN (PConv)		
$\tau = 0.050$	0.553	2.19
$\tau = 0.105$	0.591	3.58
$\tau = 0.135$	0.605	4.50
$\tau = 0.170$	0.634	5.05
$\tau = 0.215$	0.678	5.61
$\tau = 0.280$	0.708	7.12
$\tau = 0.320$	0.736	7.63
$\tau = 0.375$	0.770	8.63
$\tau = 0.435$	0.797	8.99
$\tau = 0.560$	0.824	13.08
CNN (PSM)		
$\tau = 0.050$	0.553	2.29
$\tau = 0.100$	0.587	3.80
$\tau = 0.125$	0.611	4.53
$\tau = 0.155$	0.656	5.13
$\tau = 0.180$	0.670	5.72
$\tau = 0.250$	0.695	7.61
$\tau = 0.270$	0.721	10.41
$\tau = 0.425$	0.775	10.64
$\tau = 0.440$	0.783	11.19
$\tau = 0.530$	0.815	14.26
Rule-based (v10 design space)		
trial 295	0.323	4.67
trial 1	0.358	4.70
trial 380	0.400	5.31
trial 225	0.516	5.72
trial 241	0.603	5.75
trial 315	0.604	5.78
trial 3	0.638	6.06
trial 120	0.662	7.20
trial 8	0.718	8.32
trial 0	0.758	9.93

Table E.2: Ten Pareto-optimal operating points selected from the calibration study ($n = 387$ completed Optuna trials). Each row reports one trial's confusion-matrix counts (TP, FP, FN), derived metrics (F_1, F_2, $F_{0.5}$, precision P, recall R), and median detection delay. The total positive count is constant ($TP + FN = 66$ ground-truth crashes); the total negative count varies between trials with persistence parameters and is therefore not reported. Rows are ordered by detection delay. Multiple labels indicate that one trial satisfies several criteria.

trial	**role**	**TP**	**FP**	**FN**	$\boldsymbol{F_1}$	$\boldsymbol{F_2}$	$\boldsymbol{F_{0.5}}$	$\boldsymbol{P}$	$\boldsymbol{R}$	$\widetilde{\boldsymbol{\Delta t}}$ **[min]**
295	fastest	54	214	12	0.323	0.508	0.237	0.201	0.818	4.67
1	best F_1 ($\Delta t \leq 5$)	51	168	15	0.358	0.528	0.271	0.233	0.773	4.70
241	knee	44	36	22	0.603	0.640	0.570	0.550	0.667	5.75
3	knee / best F_1 ($\Delta t \leq 7$)	52	45	14	0.638	0.720	0.573	0.536	0.788	6.06
8	knee	47	18	19	0.718	0.714	0.721	0.723	0.712	8.32
2	best TP / best recall	59	90	7	0.549	0.714	0.446	0.396	0.894	9.07
4	best F_2	51	18	15	0.756	0.766	0.746	0.739	0.773	9.78
5	best P ($R \geq 0.7$)	47	13	19	0.746	0.725	0.768	0.783	0.712	9.78
0	best F_1	50	16	16	0.758	0.758	0.758	0.758	0.758	9.93
11	best $F_{0.5}$	45	10	21	0.744	0.705	0.787	0.818	0.682	11.83

F Post-Crash Congestion Prediction

This appendix collects supplementary material for the post-crash congestion prediction method of Chapter 9: an empirical example of off-ramp-driven upstream flow starvation (Section F.1); the cluster analysis that justifies adopting a single set of global default prediction parameters (Section F.2); two additional qualitative case studies omitted from the main text (Section F.3); the sensitivity of the hull simulation's error to the assumed simulator velocities (Section F.5); and the lifecycle-stage decomposition of the ETA comparison (Section F.6).

F.1 Empirical Example of Upstream Flow Starvation

The upstream starvation point (B) of Section 9.2.1 is typically driven by rerouting: once a crash is reported, drivers divert upstream of the congestion, reducing inflow and slowing the upstream propagation of the congested region. On freeways with off-ramps in the affected corridor, this effect is directly observable in the flow data.

Figure F.1 illustrates the effect on German freeway A81 on March 19, 2022. The top panel shows connected-vehicle trajectories colored by 3PTT phase in a distance–time diagram; the bottom panel shows fleet flow rates measured at 5 km upstream of the congestion and at 7 km within it. An off-ramp between the two measurement points absorbs a substantial share of the diverted traffic, causing a visible drop in upstream front velocity at approximately 11:20. While rerouting via off-ramps is the most common cause, other factors—the end of a rush-hour period, a temporary reduction in overall demand, or metering upstream—can also contribute to the same signature.

F.2 Cluster Analysis of Congestion Parameters

This section documents the cluster analysis underlying the decision to adopt a single set of global default prediction parameters (Section 9.2.4 and Table 9.2) rather than a category-specific parameterization. The analysis proceeds in three steps: an unsupervised clustering of the 445 labeled congestion events, an ANOVA effect-size analysis of per-cluster parameter

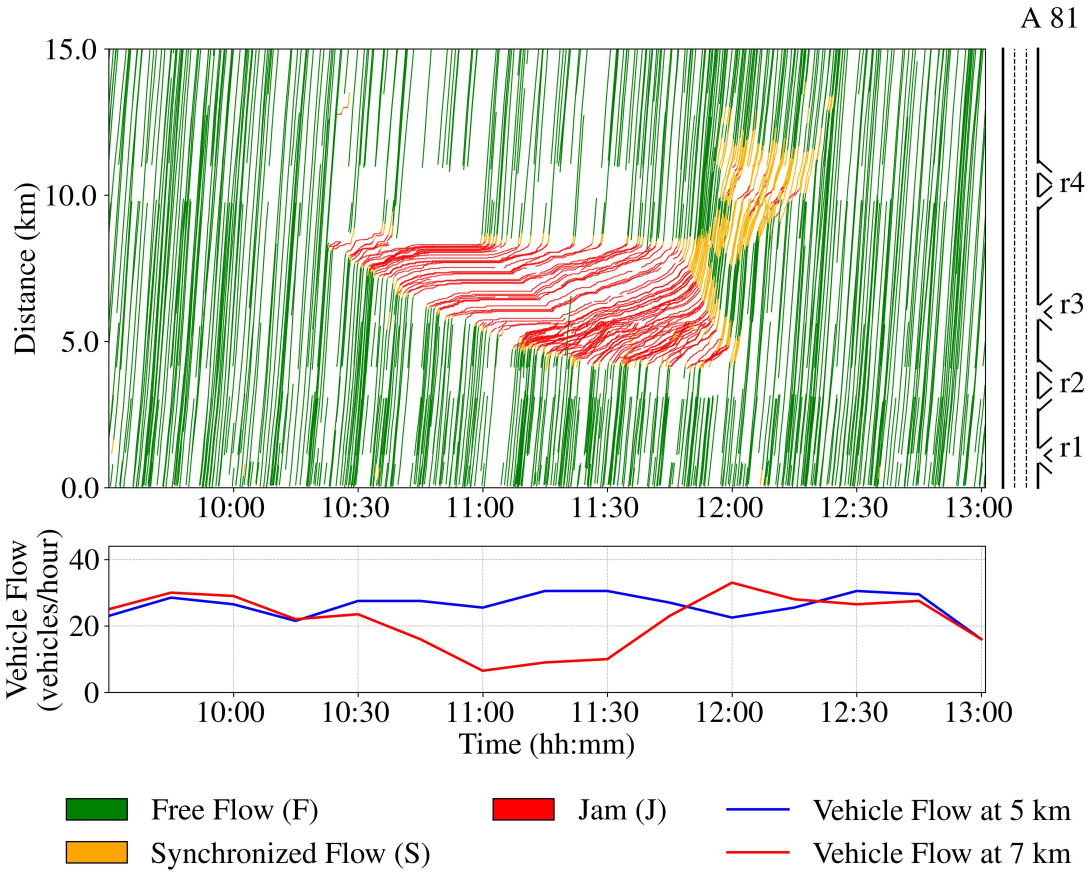


Figure F.1: Empirical example of upstream flow starvation (B) on the German A81, March 19, 2022. *Top:* connected-vehicle trajectories colored by 3PTT phase; freeway corridor and ramps shown on the right axis. *Bottom:* fleet flow rate at two measurement locations, with the drop at the upstream measurement point coinciding with the detected starvation time.

Table F.1: Median parameter values per congestion type. Times in minutes, velocities in $\mathrm{km\,h^{-1}}$.

Group	n	$\Delta t_{\text{starv.}}$	Δt_{crash}	$v_{\text{up,a}}$	$v_{\text{up,b}}$	$v_{\text{down,a}}$	$v_{\text{down,b}}$
C0	290	24	70	−8.6	−0.9	0.0	−15.2
C2	134	24	54	−8.9	−0.3	0.0	−15.9
C1	21	12	12	−7.2	0.0	0.1	−12.9
Global	445	23	62	−8.6	−0.8	0.0	−15.2

differences, and an early-classification experiment that asks whether the identified clusters can be recovered from partial observations at the time a prediction would need to be made.

F.2.1 K-Means Clustering and Group Structure

K-means clustering is applied to a nine-dimensional feature vector comprising total duration, spatial extent, area, starvation onset time, and the four propagation velocities $v_{\text{up,a}}$, $v_{\text{up,b}}$, $v_{\text{down,a}}$, $v_{\text{down,b}}$ (plus $v_{\text{up,c}}$ where marker (D) is observed). Five validation metrics evaluated for $k = 2, \ldots, 10$ consistently favor $k = 3$, with a silhouette score of 0.370 and a sharp drop to 0.220 at $k = 4$.

Post hoc, the three clusters exhibit distinct spatiotemporal profiles: C0 ($n = 428$, median duration 59 min, extent 4.1 km) contains short-to-medium events; C2 ($n = 134$, 173 min, 7.2 km) represents extended episodes; and C1 ($n = 21$) is a small group with atypical downstream-front behavior. Table F.1 reports median parameter values for the three dominant congestion types present in the dataset.

F.2.2 ANOVA Effect-Size Analysis

To test whether category-specific parameterization would improve prediction, an ANOVA effect-size analysis quantifies how much of the per-parameter variance is explained by cluster membership. The per-parameter η^2 values are shown in Figure F.2.

The upstream propagation velocities $v_{\text{up,a}}$ ($\eta^2 = 0.048$) and $v_{\text{up,b}}$ ($\eta^2 = 0.022$) are effectively group-independent: both fall well below the conventional small-effect threshold of $\eta^2 = 0.06$ and together explain less than 5 % of between-group variance. The features that *do* separate the clusters—total duration, spatial extent, and congestion area—are cumulative quantities that only reveal themselves as the event unfolds and are therefore unavailable at the moment a prediction is needed.

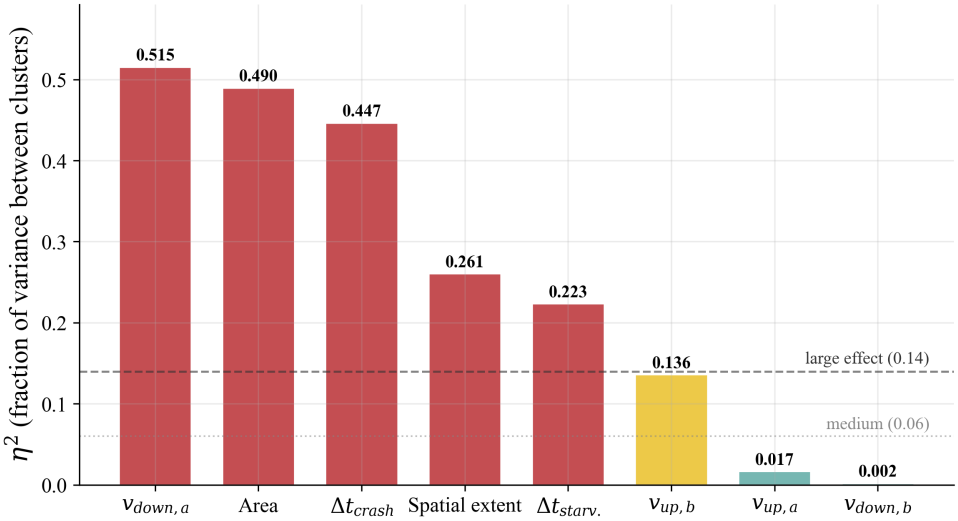


Figure F.2: ANOVA effect size η^2 per prediction parameter for $k = 3$ clusters over $n = 445$ events. Red: large effect ($\eta^2 \geq 0.14$); yellow: medium effect; blue: small effect. Propagation-velocity parameters fall in the small-effect range.

F.2.3 Early-Classification Experiment

The second test asks whether the clusters could be recovered early—at a point when a category-specific default set could actually be selected. Each congestion polygon is clipped to the first τ minutes after onset and seven early-observable features are extracted; a classifier is then trained to predict cluster membership from these partial observations.

At $\tau = 10$ min no feature reaches even a medium effect size ($\eta^2 < 0.06$), and the best classifier barely exceeds the majority-class baseline (balanced accuracy 0.36 versus 0.333). Even at $\tau = 30$ min, balanced accuracy reaches only 0.44. The clusters are not reliably identifiable from partial observations within the horizon on which the prediction must operate.

F.2.4 Implications for Parameterization

The two findings jointly rule out category-specific parameterization in practice: the parameters that *could* be personalized per cluster (the upstream velocities) are group-independent, and the features that *would* allow selecting a cluster (duration, extent, area) are not observable in time. A single set of global medians is therefore adopted (Table 9.2 of the main chapter), derived as the median over all 445 events and dominated by the most prevalent cluster C0 (73 % of events).

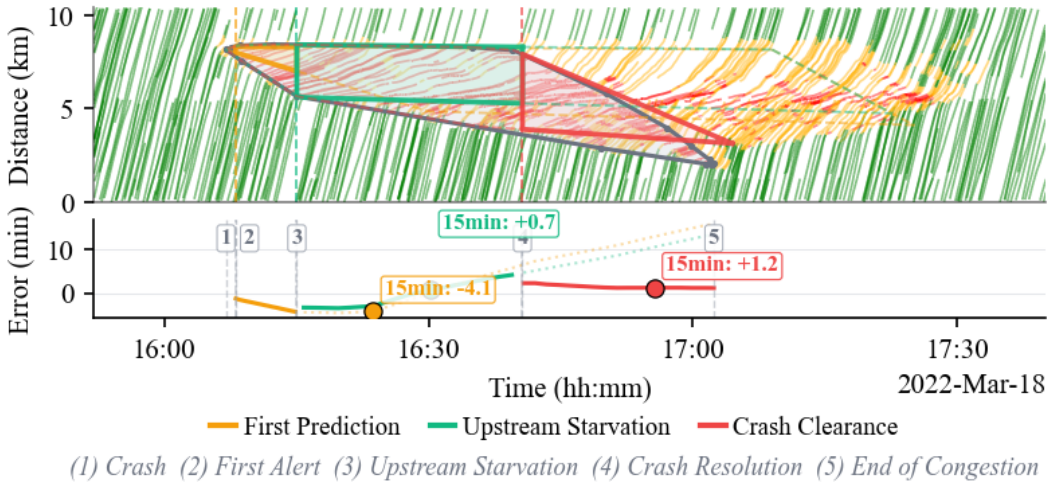


Figure F.3: Predicted congestion area for the German A81, March 18, 2022, shown *without* within-stage refinement. The gray area represents the actual congestion hull; orange, green, and red show the three prediction stages. Trajectories are colored by 3PTT phase. Rolling error is reported only at the 15-minute horizon, since beyond 30 minutes the compounded spatial error grows rapidly.

F.3 Additional Case Studies

The case studies presented in Section 9.4.4 (I-294, M3, and A8) illustrate the method's behavior on three countries and across a range of FCD penetration rates. Two additional events are documented here: the German A81 on March 18, 2022 (Section F.3.1), which demonstrates the compounding-error behavior obtained *without* within-stage refinement and therefore motivates the iterative update of Section 9.2.2, and NJ 3, United States, on March 3, 2024 (Section F.3.2), which shows an extended starvation stage in which the iterative scheme prevents error accumulation over approximately 40 minutes.

F.3.1 A81, Germany (March 18, 2022): Behavior without Within-Stage Refinement

This case shows the three stage predictions *without* within-stage refinement and therefore isolates the compounding error that motivates the iterative update of Section 9.2.2 (Figure F.3). Beyond a 30-minute horizon, the spatial deviation grows rapidly as small slope errors accumulate; for this reason the error in the lower panel is reported only at the 15-minute horizon.

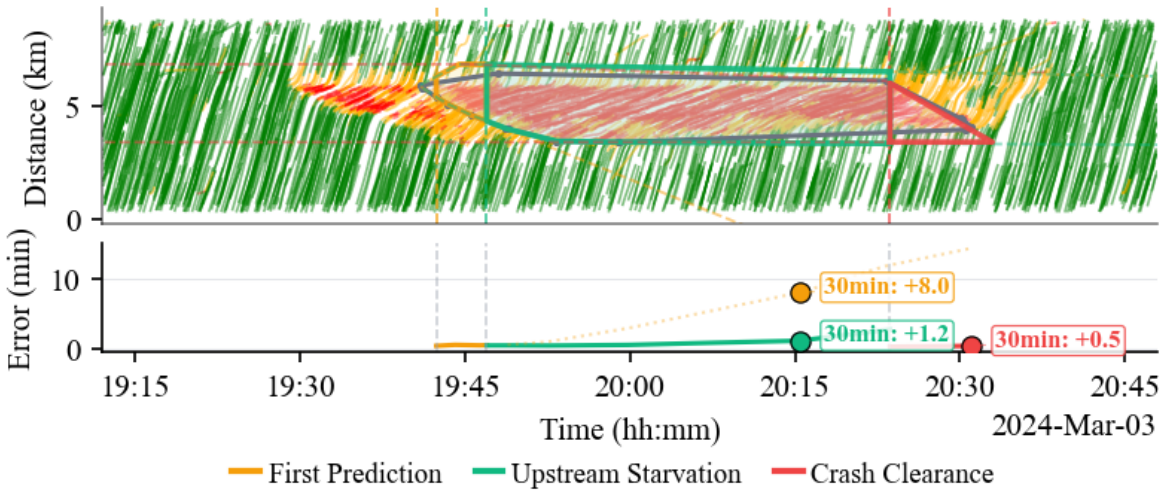


Figure F.4: Predicted congestion area for US NJ 3, March 3, 2024. The gray area represents the actual congestion hull; orange, green, and red show the three prediction stages. Trajectories are colored by 3PTT phase. The rolling error and 30-minute error per stage are shown in the lower panel.

F.3.2 NJ 3, United States (March 3, 2024): Extended Starvation without Error Growth

NJ 3 demonstrates the full benefit of the iterative scheme (Figure F.4). The upstream-starvation stage extends for approximately 40 minutes, yet no compounding error develops: the rolling prediction error stays around one minute throughout the stage, as each two-minute update re-estimates the front velocity from the latest trajectories before any drift can accumulate.

F.4 TPEG-based delay error analysis

Figure F.5 shows additional delay prediction samples for mega jams on German freeways, illustrating the limits of TPEG-based ETA. Figure F.5 (1) shows why TPEG comparison needed to be limited to Mega jams in free flow, since TPEG correctly predicts the delay of the congestion beyond the mega jam which is out of scope of the methods task, and thus needs to be excluded from the comparison. Figure F.5 (2) shows a sample of a mega jam with a long starvation stage. TPEG's delayed onset is clearly visible. TPEG also overestimates the delay when in jam vehicle speed is low. Figure F.5 (3) shows a sample of a mega jam with a short starvation stage. Since this is a short full blockage, TPEG has

limited data and thus a delayed onset combined with an overestimation of the delay.

F.5 Robustness to Simulator Velocities

Table F.2 reports the hull simulation's MAE as a function of $(v_{\text{cong}}, v_{\text{free}})$ on the same corpus and active-phase restriction as Section 9.4.3. Across a 3.5× range in v_{cong} (10–35 km/h), MAE at every horizon varies by no more than 1.8 min ($\leq 13\%$ of the best-case value). The minimum lies at $v_{\text{cong}} = 15$ km/h, edging the default 10 km/h on every horizon by 0.2 to 0.7 min, well below the per-event variability indicated by the RMSE in Table 9.5. Signed bias at the nowcast grows monotonically more negative with v_{cong}, from −10.6 min at 10 km/h to −17.6 min at 35 km/h: a higher assumed congested speed compresses simulated travel time and underestimates realised delay, but MAE stays bounded because the inflated bias is partly offset by a smaller variance.

Table F.2: Hull simulation's prediction error as a function of $(v_{\text{cong}}, v_{\text{free}})$ on the same corpus and active-phase restriction as Table 9.5.

v_{cong}	v_{free}	Hull MAE [min] at horizon h					Bias
[km/h]	[km/h]	h=0	h=5	h=15	h=30	h=60	@ h=0
10.0	110.0	16.33	13.42	14.59	13.39	**15.19**	−10.64
15.0	110.0	**16.13**	**12.96**	**13.90**	**13.39**	15.38	−13.51
20.0	110.0	16.51	13.35	14.17	14.07	15.61	−15.17
25.0	110.0	17.01	13.99	14.69	14.59	15.85	−16.28
30.0	110.0	17.50	14.62	15.25	15.14	16.06	−17.05
35.0	110.0	17.93	15.18	15.79	15.65	16.26	−17.63

F.6 Lifecycle-Stage Decomposition of the ETA Comparison

Table F.3 decomposes the rolling comparison into the three forecast-relevant lifecycle stages. The hull's advantage is concentrated at the dynamic transitions. At onset the hull cuts MAE by 5.4 min (12.2 vs 17.6 min), reflecting its kinematic queue-formation model relative to a baseline that holds the latest report forward. At crash-end the hull cuts MAE by 10.4 min (7.0 vs 17.4 min), reflecting the rapid contraction of the simulated hull at clearance against the characteristic lag of persistence into the post-jam window. At the upstream-starvation stage in the bulk of the jam, the two estimators are within 0.2 min on

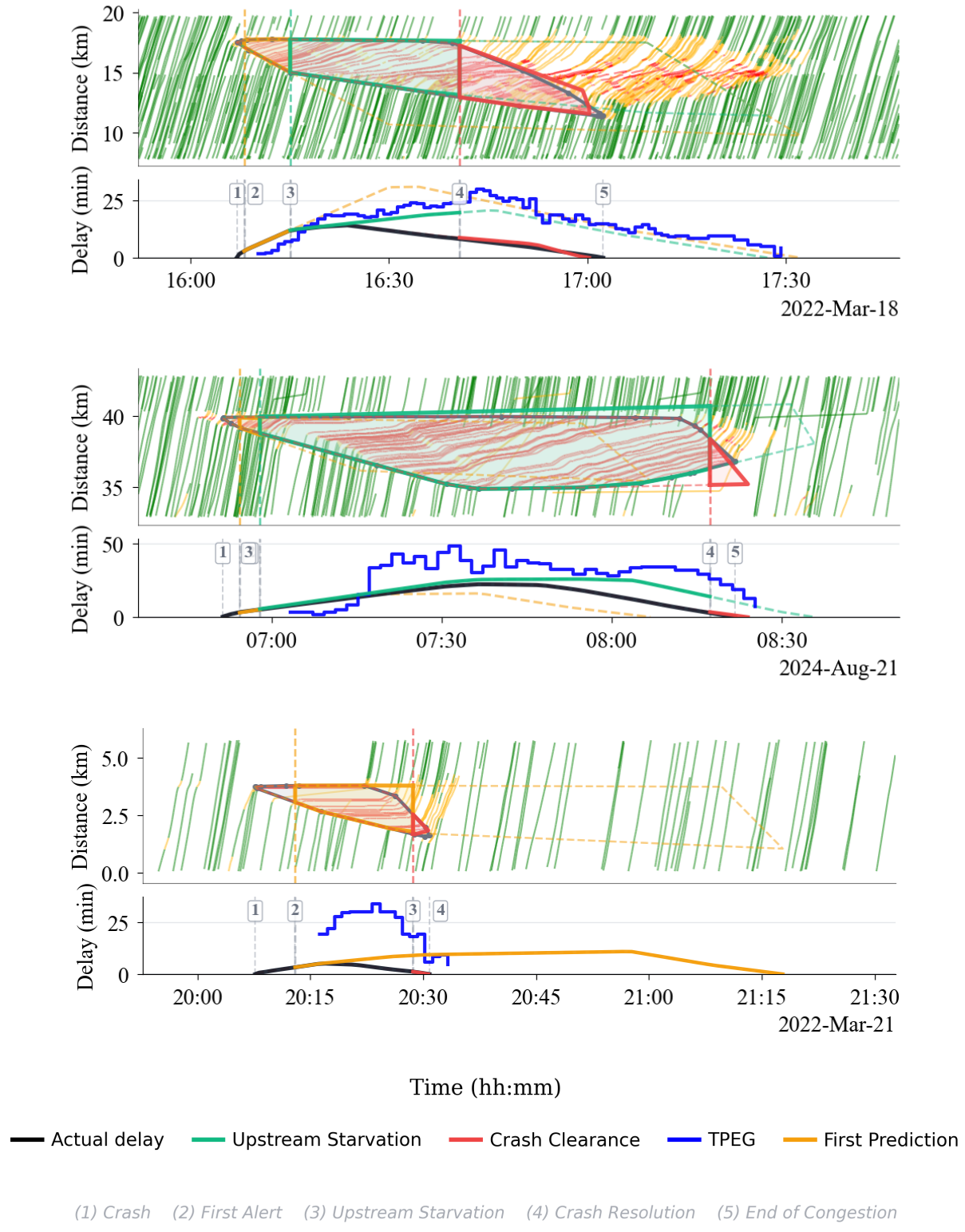


Figure F.5: Multiple delay prediction samples for Mega-Jams on German freeways showing limits of TPEG-based ETA prediction for mega jams.

Table F.3: Rolling pooled error by lifecycle stage. Onset covers the alert tick to the upstream-starvation transition, starvation covers the bulk of the jam where the upstream front is back-propagating through the queue, and crash-end covers the tail decay after the obstruction clears. Hull pooled across 40 events; TPEG pooled across 24 events.

	Hull simulation				TPEG baseline		
Stage	MAE	RMSE	Bias	n	MAE	RMSE	Bias
Onset	12.23	16.84	−3.97	1,034	17.61	22.79	−9.91
Upstream starvation	12.32	16.26	−7.47	2,490	12.53	15.57	+5.23
Crash-end	7.00	10.20	−5.40	276	17.40	24.46	+0.25

MAE, with the hull biased −7.5 min and TPEG biased +5.2 min. The starvation result reflects the regime persistence is built for, a quasi-stable queue length over several minutes is exactly the held-forward assumption, and on Group B that assumption is approximately satisfied through the plateau phase of the jam. The hull's advantage is structural, in the dynamics that arise from simulating fronts; TPEG is competitive only where those dynamics are quiescent.

List of Own Publications

The following publications have been produced by the author in the context of this dissertation.

Journal Articles & Conference Contributions

[1] L. Querfurth, H. Rehborn, B. Bernhardt, Y. Guan, and S. Hoffmann, "Large Data Analysis: How Can Accidents Be Detected Based on Probe Vehicle Data?" In *Transportation Research Procedia*, vol. 86, Lund, 2024, pp. 451–458, isbn: 2352-1465. doi: 10.1016/j.trpro.2025.04.057

[2] L. Querfurth, H. Rehborn, B. Bernhardt, Y. Guan, W. Huang, and S. Hoffmann, "Spatio-Temporal Accident Detection through Traffic Impact Analysis Using Connected Vehicle Data," in *Transportation Research Board*, Washington, DC, Jun. 2025.

[3] L. Querfurth, H. Rehborn, B. Bernhardt, Y. Guan, W. Huang, and S. Hoffmann, "Crash localization and traffic impact assessment via spatio-temporal analysis of connected vehicle data," *Accident Analysis & Prevention*, vol. 214, p. 107 956, May 2025, issn: 0001-4575. doi: 10.1016/j.aap.2025.107956

[4] L. Querfurth, H. Rehborn, M. Zürn, and S. Hoffmann, "Spatio-Temporal Classification of Post-Crash Freeway Congestion Using Fleet Data," in *ITS European Congress 2025*, Seville, May 2025.

[5] L. Querfurth, H. Rehborn, M. Zürn, and S. Hoffmann, "Impact of Localized Rain Events on Traffic Breakdown," in *Transportation Research Procedia*, vol. 95, Edinburgh: Elsevier, Sep. 2025, pp. 97–104, isbn: 2352-1465. doi: 10.1016/j.trpro.2026.02.013

[6] L. Querfurth, H. Rehborn, M. Zürn, and S. Hoffmann, "Predicting the Spatio-Temporal Development of Non-Recurring Congestion Patterns," in *Transportation Research Board*, Wahington D.C., Jan. 2026.

[7] L. Querfurth, H. Rehborn, M. Zürn, and S. Hoffmann, "Unfallerkennung mit Flottendaten," *Straßenverkehrstechnik*, Kirschbaum Verlag GmbH, no. 02, pp. 85–88, Feb. 2026, issn: 0039-2219. doi: 10.53184/SVT2-2026-2 Accessed: Feb. 12, 2026.

[8] L. Querfurth, H. Rehborn, M. Zürn, and S. Hoffmann, "Anisotropic, Phase-Aware DBSCAN for Spatio-Temporal Clustering of Congested Traffic Regions in Floating Car Data," in *ITS European Congress*, Istanbul, Turkey, Apr. 2026.

[9] L. Querfurth, S. Cheng, H. Rehborn, M. Zürn, and S. Hoffmann, "CNN-Based Spatio-Temporal Crash Detection Using Connected Vehicle Trajectory Data," in *IEEE ITSC 2026*, Naples, Sep. 2026.

[10] L. Querfurth, S. Cheng, M. Zürn, H. Rehborn, and S. Hoffmann, "Physics-informed graph neural network for real-time freeway crash detection from connected-vehicle data," *Transportation Research Part C: Emerging Technologies*, 2026, submitted.

[11] L. Querfurth, M. Zürn, H. Rehborn, and S. Hoffmann, "Wenn lokale Starkregenzellen den Verkehr beeinflussen: Richtungsabhängige Verkehrszusammenbrüche auf Autobahnen," *Straßenverkehrstechnik*, 2026, submitted.

Patents

[12] H. Rehborn, M. Koller, and L. Querfurth, "Verfahren zur Übermittlung von Positionsdaten eines Fahrzeuges," DE102025122480A1, Jul. 2025. Accessed: Feb. 12, 2026.

[13] H. Rehborn, L. Querfurth, M. Ghanem, and M. Zürn, "Verfahren zur Gestattung von Nebentätigkeiten beim autonomen Fahren ohne Kontrollen/Bestrafung," 2025.

[14] H. Rehborn, L. Querfurth, and M. Zürn, "Verfahren zur richtungsabhängigen verkehrlichen Bewertung von Extremniederschlagsereignissen für ein Fahrzeug," DE102024003651B3, Dec. 2025. Accessed: Feb. 12, 2026.

[15] S. Cheng, L. Querfurth, H. Rehborn, and M. Zürn, "Automatische Unfallerkennung in vernetzten Fahrzeugflotten mittels graphbasiertem Lernen auf einem rollierenden Positionsgraphen," 02025141130.1, 2025.

[16] L. Querfurth, "Verfahren zur optimierten Verkehrssteuerung durch Vorhersage des zeitlich-räumlichen Staugebietes nach einem Vorfall," 2025P00376 DE, 2025.

Abbreviations

3PTT	three phase traffic theory
ADAC	Automobile Club of Germany
ANOVA	analysis of variance
AP	average precision
AUC	area under the curve
BCE	binary cross-entropy
CI	confidence interval
CNN	convolutional neural network
DBSCAN	density-based spatial clustering of applications with noise
DL	deep learning
DWD	German Meteorological Service (*Deutscher Wetterdienst*)
ETA	estimated time of arrival
F	Uncongested traffic at low density, moving near the speed limit with unrestricted lane changing and overtaking.
FB	A *full blockage* is a road state in which lanes are impassable, producing a temporary stoppage of traffic at the blockage location.
FCD	floating car data
FN	false negative
FP	false positive
FPR	false positive rate
GAN	generative adversarial network
GAT	graph attention network
GCN	graph convolution network
GNN	graph neural network
GNSS	global navigation satellite system

GP	A *general congestion pattern* is a congestion pattern in which both synchronized flow and WMJs emerge, typically fixed to an infrastructure bottleneck.
GPU	graphics processing unit
GT	ground truth
IoU	intersection over union
J	Low-speed congestion propagating upstream at a characteristic velocity v_g, independent of any bottleneck. Realized as WMJs and, when multiple jams merge, as mega jams.
KS	Kolmogorov-Smirnov
LP	A *localized congestion pattern* is a spatially confined congestion event, shorter and less severe than a GP or SP, often caused by transient disturbances.
LRC	A *localized rainfall cell* is a spatially bounded precipitation region.
LSTM	long short-term memory
LWR	The *Lighthill-Whitham-Richards* model is the classical first-order continuum traffic flow model, describing traffic as a conserved fluid with a flow-density relation.
MAE	mean absolute error
MB	A *moving bottleneck* is a traveling capacity restriction—such as a slow vehicle, a rolling roadworks convoy, or any other moving obstruction—that produces congestion propagating with the obstacle.
MLP	multilayer perceptron
MP	A *moving congestion pattern* is a congestion structure that propagates along the road rather than being anchored to a fixed bottleneck.
OEM	original equipment manufacturer
OR	odds ratio
PR	precision-recall
PSM	The *phase-based smoothing method* reconstructs spatiotemporal speed fields from sparse FCD by smoothing along characteristic directions associated with free flow and congested phases.

RADOLAN	Radar-Online-Adjustment (German: *RAdar-OnLine-ANeichung*)
RB	The *rule-based crash detection model* combines features grounded in 3PTT through an additive probability scoring scheme.
RMSE	root mean squared error
ROC	receiver operating characteristic
RTTI	real time traffic information
S	Congested traffic with continuous flow at reduced speeds, anchored at a downstream bottleneck that drives its formation and dissipation.
SP	A *synchronized flow congestion pattern* consists of S alone, without the emergence of WMJs, and is typically tied to a bottleneck.
TCN	temporal convolutional network
TMC	traffic message channel
TN	true negative
TP	true positive
TPEG	transport protocol experts group
TSD	A *time-space diagram* visualizes traffic state across road position (space axis) and time, typically color-coded by speed to reveal the propagation of congestion fronts.
WMJ	A *wide moving jam* is a localized region of very low speed that propagates upstream at a characteristic velocity largely independent of the surrounding traffic state, as defined in 3PTT.

Symbols

D	Length of the freeway segment, with spatial extent $[d_{\min}, d_{\max}]$
Δd_{agg}	Half-width of the spatial aggregation window for supernodes
Δd_{pert}	Maximum spatial perturbation for window jittering
Δd_{spa}	Spatial horizon for vehicle spatial edges
Δt_{agg}	Duration of the temporal aggregation window for supernodes
Δt_{pert}	Maximum temporal perturbation for window jittering
Δt_{spa}	Temporal look-back horizon for vehicle spatial edges
$\mathcal{G}$	Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$
G_{safe}	Safe following distance
$\mathbf{H}$	Node-feature matrix
I	Rainfall intensity
$\mathbf{M}$	Binary validity mask for partial convolution
N	Number of vehicles
$P_{\Phi}(t, d)$	Phase-aware gap field for phase Φ
T_{blk}	Blockage duration
T_{cr}	Cumulative duration of crash-related congestion events
T_{obs}	Total duration of the observation period
$\mathcal{W}_{D,T}$	Spatiotemporal observation window defined by spatial horizon D and temporal horizon T
$\mathcal{W}_{mn}$	Spatiotemporal aggregation window at anchor (m, n)
$\boldsymbol{\mathcal{X}}_{\text{CNN}}$	CNN input tensor
$*$	Convolution operation (e.g., in CNNs)
β	Blockage rate
ϕ_{cd}	Cluster dominance (fraction of the window occupied by the *current* cluster)
r_f	Pre-jam free-flow ratio

d	Spatial position along roadway
Δt	Time interval or time step
$\Delta t_{\text{dwell}}^{\Phi \to \Phi'}$	Minimum dwell time required to validate a phase transition from Φ to Φ'
η_{cr}	Crash ratio (proportion of crash-related congestion time to total observation time)
$f_S(\cdot)$	Scoring function (returns value in $[0, 1]$)
$\mathcal{H}$	Hull function returning the set of points within the convex hull of a given set
k	Sample or sequence index (e.g., along a trajectory γ_ψ or within a measurement series)
n_{lane}	Number of lanes on the road segment
q	Traffic flow rate
$q_{\max}$	Maximum capacity
$q_{\max}^{\text{free}}$	Maximum free flow capacity
q_{out}	Jam outflow capacity
r_{core}	Local radius threshold for core-distance neighborhood
s	Space headway between consecutive vehicles (front-to-front distance)
S_i	Score for feature i
t	Time variable
t_{crash}	Ground-truth crash onset time
t_{det}	Detection time at which the observation window is evaluated
v	Vehicle speed
$v_{\text{F,low}}$	Lower velocity threshold for the free-flow phase
$v_{\text{S,low}}$	Lower velocity threshold for the synchronized phase
v_g	Jam downstream front propagation velocity
v_{ideal}	Ideal propagation velocity for phase-aware clustering
v_{lim}	Posted speed limit
$v_{\min}$	Minimum speed threshold
$\tilde{v}_{m,n}$	Aggregated speed at anchor (m, n)
v_u	Jam upstream front propagation velocity
w_i	Weight for feature i
$\mathbf{x}_i$	Node i in graph
$\mathbf{x}_j$	Node j in graph

$\tilde{\mathbf{x}}_{m,n}$	Supernode at temporal anchor m and spatial anchor n
y_{cls}	Classification label or prediction
$y_{\text{cls}}^{\text{pred}}$	Predicted crash classification probability
y^{pred}	Predicted label
Φ	Traffic phase label, $\Phi \in \{\text{F}, \text{S}, \text{J}\}$
$\Phi_{\text{congested}}$	Traffic phase label, $\Phi_{\text{congested}} \in \{\text{S}, \text{J}\}$
Φ_{down}	Downstream congestion front; concrete instances: S_{down} ($\text{S} \to \text{F}$), J_{down} ($\text{J} \to \text{F}$)
Φ_{up}	Upstream congestion front; concrete instances: S_{up} ($\text{F} \to \text{S}$), J_{up} ($\text{S} \to \text{J}$)
α	Fleet penetration rate (ratio of connected to total vehicles)
γ_ψ	Continuous trajectory of trip ψ
ψ	Anonymized session identifier used to group positional samples into a unique trajectory
ρ	Traffic density
$\rho_{\max}$	Maximum (jam) density
$\rho_{\min}$	Minimum density in jam
$\tilde{\rho}_{m,n}$	Aggregated density at anchor (m, n)
σ	Empirical standard deviation (unit depends on the underlying variable)
$\sigma_\parallel$	Scaling semi-axis along the propagation direction
$\sigma_\perp$	Scaling semi-axis perpendicular to the propagation direction
τ	Time headway between vehicles
$\tau_{\text{del}}^{(a)}$	Acceleration delay time for vehicles starting from a jam
τ_{safe}	Safe time headway
τ_{smooth}	Exponential smoothing time constant for prediction sequences
$\to$	Phase transition indicator (e.g., $F \to S$)
α_{ij}	Attention coefficient from node j to node i in GATv2
c_Φ	Propagation speed of the convolution kernel for phase Φ
C_{psm}	Sector count threshold for phase-sector map assignment
$\tilde{\mathcal{D}}$	Set of spatial supernode anchors

d_{up}	Upstream boundary of the observation range
ℓ_{Φ}^{d}	Spatial decay length-scale of the convolution kernel for phase Φ
ℓ_{Φ}^{t}	Temporal decay length-scale of the convolution kernel for phase Φ
$\mathcal{E}$	Edge set of the spatiotemporal graph
$\mathcal{E}_{\text{spatial}}$	Spatial edges between nearby vehicle nodes
$\mathcal{E}_{\text{super}}$	Edges connecting supernodes to nearby vehicle nodes
$\mathcal{E}_{\text{WMJ}}$	Propagation edges reflecting wide moving jam patterns
$\mathbf{h}_i$	Feature vector of node i in a graph neural network
$\mathbf{K}_{\Phi}$	Exponentially decaying convolution kernel for phase Φ
$\mathcal{L}_{\text{car}}$	Vehicle trajectory sequences (time-ordered vehicle nodes)
$\mathcal{L}$	Set of temporal node sequences in the spatiotemporal graph
$\mathcal{L}_{\text{super}}$	Supernode history sequences (time-ordered supernodes)
$\mathcal{N}_i$	Set of in-neighbours of node i in a directed graph
p_d	Number of spatial bins (partitions) in the discretized observation grid
Ψ	GNN layer mapping $\Psi\colon \mathbb{R}^{\lvert\mathcal{V}\rvert\times d} \to \mathbb{R}^{\lvert\mathcal{V}\rvert\times d'}$; subscript denotes edge type, superscript denotes layer depth
p_t	Number of temporal bins (partitions) in the discretized observation grid
$\mathbf{q}_{m,n}$	Phase probability vector at grid cell (m, n)
$\tilde{\mathcal{T}}$	Set of temporal supernode anchors
$\mathcal{V}$	Node set of the spatiotemporal graph, comprising vehicle nodes $\mathbf{x}_i$ and supernodes $\tilde{\mathbf{x}}_{m,n}$

References

[17] P. Christidis and N. Rivas, *Measuring road congestion*, 2012. doi: 10.2791/15282 Accessed: Feb. 25, 2026.

[18] ADAC, *ADAC Staubilanz 2024: Fast 450.000 Stunden Stillstand*, https://presse.adac.de/meldungen/adac-ev/verkehr/-adac-staubilanz-2024.html, Feb. 2025. Accessed: Feb. 25, 2026.

[19] U. F. H. Administration, *FHWA Operations - Reducing Non-Recurring Congestion*, https://ops.fhwa.dot.gov/program_areas/reduce-non-cong.htm, 2024. Accessed: Oct. 1, 2025.

[20] K. Jha and L. Albert, *Congestion Pie Chart for Different Sources of Congestion*, Aug. 2021.

[21] U.S. Department of Transportation, Federal Highway Administration, "Incorporating Travel-Time Reliability into the Congestion Management Process (CMP): A Primer Chapter 1. Introduction - FHWA Office of Operations," U.S. Department of Transportation, Federal Highway Administration, Office of Operations, Washington, DC, Tech. Rep. FHWA-HOP-14-034, 2015. Accessed: Jul. 23, 2025.

[22] WHO, *Global status report on road safety*, 2023.

[23] J. Kwon, M. Mauch, and P. Varaiya, "Components of Congestion: Delay from Incidents, Special Events, Lane Closures, Weather, Potential Ramp Metering Gain, and Excess Demand," 2006. Accessed: Oct. 1, 2025.

[24] M. Karlaftis, S. P. Latsoki, N. J. Richards, and K. C. Sinha, "ITS Impacts on Safety and Traffic Management: An Investigation of Secondary Crash Causes," *ITS Journal - Intelligent Transportation Systems Journal*, vol. 5, no. 1, pp. 39–52, Jan. 1999, issn: 1024-8072. doi: 10.1080/10248079908903756 Accessed: Feb. 25, 2026.

[25] N. Becker, H. W. Rust, and U. Ulbrich, "Weather impacts on various types of road crashes: A quantitative analysis using generalized additive models," *European Transport Research Review*, vol. 14, no. 1, pp. 1–18, Dec. 2022, issn: 1866-8887. doi: 10.1186/s12544-022-00561-2 Accessed: Sep. 9, 2025.

[26] F. Nemry and H. Demirel, *Impacts of Climate Change on transport: A focus on road and rail transport infrastructures*, 2012. doi: 10.2791/15504 Accessed: Oct. 1, 2025.

[27] M. Nokkala, P. Leviäkangas, K. Oiva, A.-M. Hietajärvi, A. Vajda, S. Athanasatos, S. Michaelides, M. Papadakis, M. Kreuz, T. Mühlhausen, J. Ludvigson, and R. Klæboe, "Summary report on the costs of extreme weather for the European transport system," Jan. 2012.

[28] F. De Michiel, J. Iskandar, P. R. Abecasis, and T. Haanperä, *A study on the market for in-vehicle data in the EU*, Dec. 2024. Accessed: Oct. 1, 2025.

[29] A. Amer, E. Roberts, U. Mangar, W. H. Kraft, J. T. Wanat, P. C. Cusolito, J. R. Hogan, and X. Zhao, "Traffic Incident Management Gap Analysis Primer: Technical Documentation Page - FHWA Office of Operations," Federal Highway Administration Office of Operations, Washington, Draft Report FHWA-HOP-15-007, Mar. 2015. Accessed: Oct. 1, 2025.

[30] *Regulation (EU) 2015/758 of the European Parliament and of the Council of 29 April 2015 concerning type-approval requirements for the deployment of the eCall in-vehicle system based on the 112 service and amending Directive 2007/46/EC*, Apr. 2015. Accessed: Mar. 23, 2026.

[31] World Bank, *Socio-Economic Costs and Human Impacts of Road Accidents in Azerbaijan*. World Bank, Washington, DC, Jun. 2021. doi: 10.1596/35986 HDL: 10986/35986. Accessed: Jan. 21, 2026.

[32] K. H. Janstrup, S. Kaplan, T. Hels, J. Lauritsen, and C. G. Prato, "Understanding traffic crash under-reporting: Linking police and medical records to individual and crash characteristics," *Traffic Injury Prevention*, vol. 17, no. 6, pp. 580–584, Aug. 2016, issn: 1538-957X. doi: 10.1080/15389588.2015.1128533

[33] R. Elvik and A. B. Mysen, "INCOMPLETE ACCIDENT REPORTING: META-ANALYSIS OF STUDIES MADE IN 13 COUNTRIES," *Transportation Research Record*, no. 1665, 1999, issn: 0361-1981. Accessed: Jan. 21, 2026.

[34] J. Bigham, G. Strang, and S. H. Oum, "Evaluation of the Accuracy of Global Positioning System Coordinates for Collision Locations in California," Jul. 2014. Accessed: Jan. 21, 2026.

[35] A. Coursey, J. Ji, M. Quinones-Grueiro, W. Barbour, Y. Zhang, T. Derr, G. Biswas, and D. B. Work, *FT-AED: Benchmark Dataset for Early Freeway Traffic Anomalous Event Detection*, Jun. 2024. doi: 10.48550/arXiv.2406.15283 arXiv: 2406.15283 [cs]. Accessed: Jan. 21, 2026.

[36] M. Amin-Naseri, P. Chakraborty, A. Sharma, S. B. Gilbert, and M. Hong, "Evaluating the Reliability, Coverage, and Added Value of Crowdsourced Traffic Incident Reports from Waze," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2672, no. 43, pp. 34–43, Dec. 2018, issn: 0361-1981, 2169-4052. doi: 10.1177/0361198118790619 Accessed: Jan. 21, 2026.

[37] W. Vandenberghe, E. Vanhauwaert, S. Verbrugge, I. Moerman, and P. Demeester, "Feasibility of expanding traffic monitoring systems with floating car data technology," *IET Intelligent Transport Systems*, vol. 6, no. 4, pp. 347–354, Dec. 2012. doi: 10.1049/iet-its.2011.0221 Accessed: Mar. 1, 2026.

[38] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: Where we are and where we're going," *Transportation Research Part C: Emerging Technologies*, Special Issue on Short-term Traffic Flow Forecasting, vol. 43, pp. 3–19, Jun. 2014, issn: 0968-090X. doi: 10.1016/j.trc.2014.01.005 Accessed: Oct. 1, 2025.

[39] M. T. Newaz, M. Ershadi, M. Jefferies, and P. Davis, "A critical review of the feasibility of emerging technologies for improving safety behavior on construction sites," *Journal of Safety Research*, vol. 89, pp. 269–287, Jun. 2024, issn: 0022-4375. doi: 10.1016/j.jsr.2024.04.006 Accessed: Oct. 2, 2025.

[40] M. A. Sahraei and S. R. M. Al Mamari, "A Review of Internet of Things Approaches for Vehicle Accident Detection and Emergency Notification," *Sustainability*, vol. 17, no. 14, p. 6510, Jan. 2025, issn: 2071-1050. doi: 10.3390/su17146510 Accessed: Oct. 2, 2025.

[41] D. Eisenberg, "The mixed effects of precipitation on traffic crashes," *Accident Analysis & Prevention*, vol. 36, no. 4, pp. 637–647, Jul. 2004, issn: 0001-4575. doi: 10.1016/S0001-4575(03)00085-X Accessed: Oct. 1, 2025.

[42] B. S. Kerner, *The Physics of Traffic: Empirical Freeway Pattern Features, Engineering Applications, and Theory* (Understanding Complex Systems (UCS)). Berlin: Springer, 2004, isbn: 978-3-540-40986-1.

[43] B. Greenshields, "A Study of Traffic Capacity.," *Highway Research Board*, Scientific Research Publishing, vol. 14, pp. 448–477, 1935. Accessed: Sep. 18, 2025.

[44] A. D. May, *Traffic Flow Fundamentals*. Prentice Hall, 1990, isbn: 978-0-13-926072-8.

[45] M. Treiber and A. Kesting, *Traffic Flow Dynamics: Data, Models and Simulation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, isbn: 978-3-642-32459-8 978-3-642-32460-4. doi: 10.1007/978-3-642-32460-4 Accessed: Oct. 8, 2024.

[46] M. J. Cassidy, "Bivariate relations in nearly stationary highway traffic," *Transportation Research Part B: Methodological*, vol. 32, no. 1, pp. 49–59, Jan. 1998, issn: 0191-2615. doi: 10.1016/S0191-2615(97)00012-X Accessed: Feb. 26, 2026.

[47] J. G. Wardrop, "Road paper. some theoretical aspects of road traffic research.," *Proceedings of the Institution of Civil Engineers*, vol. 1, no. 3, pp. 325–362, May 1952, issn: 1753-7789. doi: 10.1680/ipeds.1952.11259 Accessed: Jan. 8, 2026.

[48] G. Leduc, *Road Traffic Data: Collection Methods and Applications* (JRC-IPTS Working Papers). Nov. 2008.

[49] J. Guerrero-Ibáñez, S. Zeadally, and J. Contreras-Castillo, "Sensor Technologies for Intelligent Transportation Systems," *Sensors*, vol. 18, no. 4, p. 1212, Apr. 2018, issn: 1424-8220. doi: 10.3390/s18041212 Accessed: Feb. 26, 2026.

[50] E. Brockfeld and P. Wagner, "Testing and Benchmarking of Microscopic Traffic Flow Models," in *Proceedings of the 10th World Conference on Transport Research*, vol. abstra, Istanbul (Turkey), 2004, pp. 775–776. Accessed: Feb. 26, 2026.

[51] J. C. Herrera, D. B. Work, R. Herring, X. (Ban, Q. Jacobson, and A. M. Bayen, "Evaluation of traffic data obtained via GPS-enabled mobile phones: The Mobile Century field experiment," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 4, pp. 568–583, Aug. 2010, issn: 0968090X. doi: 10.1016/j.trc.2009.10.006 Accessed: Jan. 28, 2026.

[52] M. Treiber, A. Kesting, and R. E. Wilson, "Reconstructing the Traffic State by Fusion of Heterogeneous Data," *Computer-Aided Civil and Infrastructure Engineering*, vol. 26, no. 6, pp. 408–419, 2011, issn: 1467-8667. doi: 10.1111/j.1467-8667.2010.00698.x Accessed: Feb. 16, 2026.

[53] J. Hong, X. Zhang, Z. Wei, Li Li, and Y. Ren, "Spatial and Temporal Analysis of Probe Vehicle-based Sampling for Real-time Traffic Information System," in *2007 IEEE Intelligent Vehicles Symposium*, Istanbul, Turkey: IEEE, Jun. 2007, pp. 1234–1239, isbn: 978-1-4244-1067-5 978-1-4244-1068-2. doi: 10.1109/IVS.2007.4290287 Accessed: Sep. 19, 2025.

[54] B. Kerner, C. Demir, R. Herrtwich, S. Klenov, H. Rehborn, M. Aleksi, and A. Haug, "Traffic state detection with floating car data in road networks," in *Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005.*, Vienna, Austria: IEEE, 2005, pp. 700–705, isbn: 978-0-7803-9215-1. doi: 10.1109/ITSC.2005.1520133 Accessed: Sep. 19, 2025.

[55] H. Bar-Gera, "Evaluation of a cellular phone-based system for measurements of traffic speeds and travel times: A case study from Israel," *Transportation Research Part C: Emerging Technologies*, vol. 15, no. 6, pp. 380–391, Dec. 2007, issn: 0968090X. doi: 10.1016/j.trc.2007.06.003 Accessed: Sep. 19, 2025.

[56] C. De Fabritiis, R. Ragona, and G. Valenti, "Traffic Estimation And Prediction Based On Real Time Floating Car Data," in *2008 11th International IEEE Conference on Intelligent Transportation Systems*, Beijing, China: IEEE, Oct. 2008, pp. 197–203, isbn: 978-1-4244-2111-4. doi: 10.1109/ITSC.2008.4732534 Accessed: Sep. 19, 2025.

[57] B. S. Kerner, H. Rehborn, R.-P. Schäfer, S. L. Klenov, J. Palmer, S. Lorkowski, and N. Witte, "Traffic dynamics in empirical probe vehicle data studied with three-phase theory: Spatiotemporal reconstruction of traffic phases and generation of jam warning messages," *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 1, pp. 221–251, Jan. 2013, issn: 0378-4371. doi: 10.1016/j.physa.2012.07.070 Accessed: Feb. 27, 2026.

[58] S. Breitenberger, B. Grüber, M. Neuherz, and R. Kates, "Traffic information potential and necessary penetration rates," *Traffic Engineering & Control*, vol. 45.11, Dec. 2004.

[59] B. S. Kerner and H. Rehborn, "Experimental properties of complexity in traffic flow," *Physical Review E*, vol. 53, no. 5, R4275–R4278, May 1996, issn: 1063-651X, 1095-3787. doi: 10.1103/PhysRevE.53.R4275 Accessed: Jun. 30, 2025.

[60] M. Schönhof and D. Helbing, "Empirical Features of Congested Traffic States and Their Implications for Traffic Modeling," *Transportation Science*, vol. 41, no. 2, pp. 135–166, 2007, issn: 0041-1655. JSTOR: 25769343. Accessed: Oct. 15, 2025.

[61] T. Seo, A. M. Bayen, T. Kusakabe, and Y. Asakura, "Traffic state estimation on highway: A comprehensive survey," *Annual Reviews in Control*, vol. 43, pp. 128–151, Jan. 2017, issn: 1367-5788. doi: 10.1016/j.arcontrol.2017.03.005 Accessed: Feb. 29, 2024.

[62] M. J. Lighthill and G. B. Whitham, "On kinematic waves II. A theory of traffic flow on long crowded roads," *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 229, no. 1178, pp. 317–345, Jan. 1955. doi: 10.1098/rspa.1955.0089 Accessed: Oct. 8, 2024.

[63] P. I. Richards, "Shock Waves on the Highway," *Operations Research*, vol. 4, no. 1, pp. 42–51, 1956, issn: 0030-364X. JSTOR: 167515. Accessed: Sep. 18, 2025.

[64] H. J. Payne, "MODELS OF FREEWAY TRAFFIC AND CONTROL.," *MATHEMATICAL MODELS OF PUBLIC SYSTEMS*, 1971. Accessed: Sep. 18, 2025.

[65] G. B. (B. Whitham, *Linear and Nonlinear Waves*. New York, Wiley, 1974, isbn: 978-0-471-94090-6. Accessed: Sep. 18, 2025.

[66] C. F. Daganzo, "Requiem for second-order fluid approximations of traffic flow," *Transportation Research Part B: Methodological*, vol. 29, no. 4, pp. 277–286, Aug. 1995, issn: 0191-2615. doi: 10.1016/0191-2615(95)00007-Z Accessed: Feb. 26, 2026.

[67] A. Aw and M. Rascle, "Resurrection of "second order" models of traffic flow," *SIAM J. Appl. Math.*, vol. 60, no. 3, pp. 916–938, Feb. 2000, issn: 0036-1399. doi: 10.1137/S0036139997332099 Accessed: Sep. 19, 2025.

[68] H. M. Zhang, "A theory of nonequilibrium traffic flow," *Transportation Research Part B: Methodological*, vol. 32, no. 7, pp. 485–498, Sep. 1998, issn: 0191-2615. doi: 10.1016/S0191-2615(98)00014-9 Accessed: Sep. 19, 2025.

[69] I. Prigogine and R. Herman, *Kinetic Theory of Vehicular Traffic*. American Elsevier Publishing Company, 1971, isbn: 978-0-444-00082-8.

[70] D. Helbing, "Improved fluid-dynamic model for vehicular traffic," *Physical Review E*, vol. 51, no. 4, pp. 3164–3169, Apr. 1995, issn: 1063-651X, 1095-3787. doi: 10.1103/PhysRevE.51.3164 Accessed: Nov. 3, 2025.

[71] G. F. Newell, "Nonlinear Effects in the Dynamics of Car Following," *Operations Research*, vol. 9, no. 2, pp. 209–229, Apr. 1961, issn: 0030-364X. doi: 10.1287/opre.9.2.209 Accessed: Feb. 26, 2026.

[72] M. Bando, K. Hasebe, A. Nakayama, A. Shibata, and Y. Sugiyama, "Dynamical model of traffic congestion and numerical simulation," *Physical Review E*, vol. 51, no. 2, pp. 1035–1042, Feb. 1995. doi: 10.1103/PhysRevE.51.1035 Accessed: Sep. 25, 2025.

[73] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical Review E*, vol. 62, no. 2, pp. 1805–1824, Aug. 2000. doi: 10.1103/PhysRevE.62.1805 Accessed: Sep. 25, 2025.

[74] K. Nagel and M. Schreckenberg, "A cellular automaton model for freeway traffic," *Journal de Physique I*, vol. 2, no. 12, pp. 2221–2229, Dec. 1992, issn: 1155-4304, 1286-4862. doi: 10.1051/jp1:1992277 Accessed: Mar. 1, 2026.

[75] L. Elefteriadou, R. P. Roess, and W. R. McShane, "Probabilistic nature of breakdown at freeway merge junctions," *Transportation research record*, vol. 1484, pp. 80–89, 1995, issn: 0361-1981. Accessed: Feb. 26, 2026.

[76] W. Brilon, J. Geistefeldt, and H. Zurlinden, "Implementing the Concept of Reliability for Highway Capacity Analysis," *Transportation Research Record*, vol. 2027, pp. 1–8, Dec. 2007. doi: 10.3141/2027-01

[77] B. S. Kerner, *Introduction to Modern Traffic Flow Theory and Control: The Long Road to Three-Phase Traffic Theory* (Understanding Complex Systems (UCS)). Berlin: Springer, 2009, isbn: 978-3-642-02605-8.

[78] B. S. Kerner, *Understanding Real Traffic*. Cham: Springer, 2021, isbn: 978-3-030-79601-3.

[79] J. Palmer, H. Rehborn, and B. S. Kerner, "Application of Three-Phase Traffic Theory Processing Data From Moving Vehicles," *Traffic Engineering & Control*, vol. 52, no. 4, Apr. 2011, issn: 0041-0683.

[80] J. Ishimaru, M. E. Hallenbeck, and J. Nee, *Measurement of Recurring Versus Non-Recurring Congestion: Technical Report*, https://rosap.ntl.bts.gov, Oct. 2003. Accessed: Sep. 24, 2025.

[81] B. S. Kerner, "Empirical Features of Congested Patterns at Highway Bottlenecks," *Transportation Research Record*, vol. 1802, no. 1, pp. 145–154, Jan. 2002, issn: 0361-1981. doi: 10.3141/1802-17 Accessed: Oct. 10, 2025.

[82] C. F. Daganzo, "The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory," *Transportation Research Part B: Methodological*, vol. 28, no. 4, pp. 269–287, Aug. 1994, issn: 0191-2615. doi: 10.1016/0191-2615(94)90002-7 Accessed: Sep. 18, 2025.

[83] R. Kühne, "Foundation of Traffic Flow Theory I: Greenshields Legacy Highway Traffic," in *Proceedings Symposium on the Fundamental Diagram: 75 Years*, Woods Hole, Massachusetts, 2008. Accessed: Oct. 2, 2025.

[84] H. S. Mahmassani, "Dynamic Network Traffic Assignment and Simulation Methodology for Advanced System Management Applications," *Networks and Spatial Economics*, vol. 1, no. 3, pp. 267–292, Sep. 2001, issn: 1572-9427. doi: 10.1023/A:1012831808926 Accessed: Oct. 2, 2025.

[85] W. Huang, Y. Dülgar, H. Rehborn, B. A. Bernhardt, and J. Xu, "Identification and Characterization of Mobile Work Zones using Floating Car Data," in *Transportation Research Board 100th Annual MeetingTransportation Research BoardTransportation Research Board*, 2021. Accessed: Aug. 30, 2024.

[86] B. S. Kerner, *Breakdown in Traffic Networks: Fundamentals of Transportation Science*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017, isbn: 978-3-662-54471-6 978-3-662-54473-0. doi: 10.1007/978-3-662-54473-0 Accessed: Feb. 17, 2025.

[87] Intergovernmental Panel On Climate Change (Ipcc), *Climate Change 2021 – The Physical Science Basis: Working Group I Contribution to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*, 1st ed. Cambridge University Press, Jul. 2023, isbn: 978-1-009-15789-6. doi: 10.1017/9781009157896 Accessed: Sep. 11, 2025.

[88] R. P. Allan and B. J. Soden, "Atmospheric Warming and the Amplification of Precipitation Extremes," *Science*, vol. 321, no. 5895, pp. 1481–1484, Sep. 2008. doi: 10.1126/science.1160787 Accessed: Sep. 11, 2025.

[89] P. A. O'Gorman, "Precipitation Extremes Under Climate Change," *Current Climate Change Reports*, vol. 1, no. 2, pp. 49–59, Jun. 2015, issn: 2198-6061. doi: 10.1007/s40641-015-0009-3 Accessed: Sep. 11, 2025.

[90] B. N. J. Persson, U. Tartaglino, O. Albohr, and E. Tosatti, "Rubber friction on wet and dry road surfaces: The sealing effect," *Physical Review B*, vol. 71, no. 3, p. 035 428, Jan. 2005. doi: 10.1103/PhysRevB.71.035428 Accessed: Sep. 8, 2025.

[91] J. Xu, W. Lv, C. Gao, T. Xin, X. Liu, and Y. Hou, "Prediction of pavement water film depth and estimation of critical rainfall conditions for refined road safety management: A simulation study," *PLOS ONE*, vol. 20, no. 2, e0318228, Feb. 2025, issn: 1932-6203. doi: 10.1371/journal.pone.0318228 Accessed: Sep. 8, 2025.

[92] J. N. Morden, F. Caraffini, I. Kypraios, A. H. Al-Bayatti, and R. Smith, "Driving in the Rain: A Survey toward Visibility Estimation through Windshields," *International Journal of Intelligent Systems*, vol. 2023, no. 1, p. 9 939 174, 2023, issn: 1098-111X. doi: 10.1155/2023/9939174 Accessed: Sep. 8, 2025.

[93] T. Jin, X. Liu, C. Chen, Y. Xia, X. Liu, M. Lv, and L. Li, "The impact of environmental noise on drivers' cognitive abilities: A case study on in-vehicle voice interaction interfaces," *Applied Ergonomics*, vol. 117, p. 104 247, May 2024, issn: 1872-9126. doi: 10.1016/j.apergo.2024.104247

[94] W. Y. Pao, J. Howorth, L. Li, M. Agelin-Chaab, L. Roy, J. Knutzen, A. Baltazar-y-Jimenez, and K. Muenker, "Investigation of Automotive LiDAR Vision in Rain from Material and Optical Perspectives," *Sensors (Basel, Switzerland)*, vol. 24, no. 10, p. 2997, May 2024, issn: 1424-8220. doi: 10.3390/s24102997 Accessed: Sep. 8, 2025.

[95] Y. Qiu, Y. Lu, Y. Wang, and C. Yang, "Visual Perception Challenges in Adverse Weather for Autonomous Vehicles: A Review of Rain and Fog Impacts," in *2024 IEEE 7th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Chongqing, China: IEEE, Sep. 2024, pp. 1342–1348, isbn: 979-8-3503-7080-5. doi: 10.1109/ITNEC60942.2024.10733168 Accessed: Sep. 8, 2025.

[96] J. H. Hogema, "Effects of Rain on Daily Traffic Volume and on Driving Behaviour," no. TNO-TM-96-8019, Dec. 1996. Accessed: Mar. 11, 2025.

[97] M. M. Ahmed and A. Ghasemzadeh, "The impacts of heavy rain on speed and headway Behaviors: An investigation using the SHRP2 naturalistic driving study data," *Transportation Research Part C: Emerging Technologies*, vol. 91, pp. 371–384, Jun. 2018, issn: 0968090X. doi: 10.1016/j.trc.2018.04.012 Accessed: Aug. 12, 2024.

[98] L. Qiu and W. A. Nixon, "Effects of Adverse Weather on Traffic Crashes: Systematic Review and Meta-Analysis," *Transportation Research Record*, vol. 2055, no. 1, pp. 139–146, Jan. 2008, issn: 0361-1981. doi: 10.3141/2055-16 Accessed: Sep. 9, 2025.

[99] M. Agarwal, T. H. Maze, and R. R. Souleyrette, "Impacts of Weather on Urban Freeway Traffic Flow Characteristics and Facility Capacity," 2005.

[100] H. Rakha, M. Arafeh, and S. Park, "Modeling Inclement Weather Impacts on Traffic Stream Behavior," *International Journal of Transportation Science and Technology*, vol. 1, no. 1, pp. 25–47, Mar. 2012, issn: 20460430. doi: 10.1260/2046-0430.1.1.25 Accessed: Mar. 11, 2025.

[101] J. Seeherman and A. Skabardonis, *Quantification of Weather Influences on Freeway Bottlenecks*, 2013. Accessed: Mar. 11, 2025.

[102] D. Zechin, F. Caleffi, and H. B. B. Cybis, "Influence of Rain on Highway Breakdown Probability," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2674, no. 8, pp. 687–695, Aug. 2020, issn: 0361-1981, 2169-4052. doi: 10.1177/0361198120919754 Accessed: Sep. 26, 2024.

[103] I. Stamos, J. M. Salanova Grau, E. Mitsakis, and G. Aifadopoulou, "Modeling Effects of Precipitation on Vehicle Speed: Floating Car Data Approach," *Transportation Research Record*, vol. 2551, no. 1, pp. 100–110, Jan. 2016, issn: 0361-1981. doi: 10.3141/2551-12 Accessed: Sep. 9, 2024.

[104] S. S. Dhaliwal, X. Wu, J. Thai, and X. Jia, "Effects of Rain on Freeway Traffic in Southern California," *Transportation Research Record*, vol. 2616, no. 1, pp. 69–80, Jan. 2017, issn: 0361-1981. doi: 10.3141/2616-08 Accessed: Sep. 9, 2024.

[105] WMO, "Guide to Meteorological Instruments and Methods of Observation (WMO-No. 8)," in *World Meterological Organization*, Geneva, Switzerland, 2018. Accessed: Nov. 14, 2024.

[106] L. Luo, H. Xiao, H. Yang, H. Chen, J. Guo, Y. Sun, and L. Feng, "Raindrop size distribution and microphysical characteristics of a great rainstorm in 2016 in Beijing, China," *Atmospheric Research*, vol. 239, p. 104 895, Jul. 2020, issn: 0169-8095. doi: 10.1016/j.atmosres.2020.104895 Accessed: Aug. 22, 2025.

[107] H. Rehborn and M. Koller, "A study of the influence of severe environmental conditions on common traffic congestion features," *Journal of Advanced Transportation*, vol. 48, no. 8, pp. 1107–1120, Dec. 2014, issn: 0197-6729, 2042-3195. doi: 10.1002/atr.1254 Accessed: Mar. 11, 2025.

[108] destatis, *Verkehrsunfälle*, https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Verkehrsunfaelle/_inhalt.html, 2025. Accessed: Mar. 11, 2025.

[109] M. Attallah, J. Kianfar, and Y. Wang, "Impact of High Resolution Radar-Obtained Weather Data on Spatio-Temporal Prediction of Freeway Speed," *Sustainability*, vol. 14, no. 22, p. 14 932, Nov. 2022, issn: 2071-1050. doi: 10.3390/su142214932 Accessed: Mar. 7, 2025.

[110] Y. Yao, D. Wu, Y. Hong, D. Chen, Z. Liang, Q. Guan, X. Liang, and L. Dai, "Analyzing the Effects of Rainfall on Urban Traffic-Congestion Bottlenecks," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 504–512, 2020, issn: 2151-1535. doi: 10.1109/JSTARS.2020.2966591 Accessed: Sep. 8, 2025.

[111] H. Mohammed A and J. Ben-Edigbe, "Highway Capacity Prediction in Adverse Weather," *Journal of Applied Sciences*, vol. 11, no. 12, pp. 2193–2199, Jun. 2011, issn: 18125654. doi: 10.3923/jas.2011.2193.2199 Accessed: Mar. 11, 2025.

[112] J. Kim, H. S. Mahmassani, R. Alfelor, Y. Chen, T. Hou, L. Jiang, M. Saberi, Ö. Verbas, and A. Zockaie, "Implementation and Evaluation of Weather-Responsive Traffic Management Strategies: Insight from Different Networks," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2396, no. 1, pp. 93–106, Jan. 2013, issn: 0361-1981, 2169-4052. doi: 10.3141/2396-11 Accessed: Mar. 11, 2025.

[113] W. Qiao, A. Haghani, M. Hamedi, and C. Shao, "Freeway Path and Multi-Step Ahead Travel Time Prediction under Various Weather Scenarios," Jan. 2013.

[114] E. Chung, "Does weather affect highway capacity?" In *Proceedings of the 5th International Symposium on Highway Capacity and Quality of Service.* 2006.

[115] E. Hooper, L. Chapman, and A. Quinn, "The impact of precipitation on speed–flow relationships along a UK motorway corridor," *Theoretical and Applied Climatology*, vol. 117, no. 1-2, pp. 303–316, Jul. 2014, issn: 0177-798X, 1434-4483. doi: 10.1007/s00704-013-0999-5 Accessed: Sep. 9, 2024.

[116] T. Lomax, S. Turner, G. Shunk, H. Levinson, R. H. Pratt, P. Bay, and G. B. Douglas, "QUANTIFYING CONGESTION. VOLUME 1: FINAL REPORT," *NCHRP Report*, 1997. Accessed: Oct. 2, 2025.

[117] B. Wright, Y. Zou, and Y. Wang, "Impact of Traffic Incidents on Reliability of Freeway Travel Times," *Transportation Research Record*, vol. 2484, no. 1, pp. 90–98, Jan. 2015, issn: 0361-1981. doi: 10.3141/2484-10 Accessed: Oct. 10, 2025.

[118] A. E. Stewart and J. H. Lord, "Motor vehicle crash versus accident: A change in terminology is necessary," *Journal of Traumatic Stress*, vol. 15, no. 4, pp. 333–335, Aug. 2002, issn: 0894-9867. doi: 10.1023/A:1016260130224

[119] N. Knechel, "When a Crash Is Really an Accident: A Concept Analysis," *Journal of Trauma Nursing: The Official Journal of the Society of Trauma Nurses*, vol. 22, no. 6, pp. 321–329, 2015, issn: 1078-7496. doi: 10.1097/JTN.0000000000000167

[120] G. L. Reagle, *A Crash is Not an Accident | FMCSA*, https://www.fmcsa.dot.gov/newsroom/crash-not-accident, Governmental, Sep. 1997. Accessed: Oct. 2, 2025.

[121] R. Carpenter, "Subject: ADOPT THE TERMS "CRASH," "COLLISION," AND "INCIDENT" AND DISCONTINUE THE TERM "ACCIDENT" WHEN REFERRING TO TRAFFIC CRASHES," 2021.

[122] H. Park and A. Haghani, "Real-time prediction of secondary incident occurrences using vehicle probe data," *Transportation Research Part C: Emerging Technologies*, vol. 70, pp. 69–85, Sep. 2016, issn: 0968090X. doi: 10.1016/j.trc.2015.03.018 Accessed: Apr. 19, 2025.

[123] R. Li, F. C. Pereira, and M. E. Ben-Akiva, "Overview of traffic incident duration analysis and prediction," *European Transport Research Review*, vol. 10, no. 2, p. 22, Jun. 2018, issn: 1867-0717, 1866-8887. doi: 10.1186/s12544-018-0300-1 Accessed: Feb. 17, 2025.

[124] M. Houbraken, S. Logghe, M. Schreuder, P. Audenaert, D. Colle, and M. Pickavet, "Automated Incident Detection Using Real-Time Floating Car Data," *Journal of Advanced Transportation*, vol. 2017, e8241545, Dec. 2017, issn: 0197-6729. doi: 10.1155/2017/8241545 Accessed: Feb. 9, 2024.

[125] A. J. Khattak, X. Wang, H. Zhang, and M. Cetin, "Primary and Secondary Incident Management: Predicting Durations in Real Time," *Virginia Center for Transportation Innovation and Research*, 2009.

[126] B. L. Smith, L. Qin, and R. Venkatanarayana, "Characterization of Freeway Capacity Reduction Resulting from Traffic Accidents," *Journal of Transportation Engineering*, vol. 129, no. 4, pp. 362–368, Jul. 2003, issn: 0733-947X. doi: 10.1061/(ASCE)0733-947X(2003)129:4(362) Accessed: Oct. 10, 2025.

[127] S.-M. Chin, O. Franzese, D. L. Greene, H. L. Hwang, and R. Gibson, *Temporary Losses of Highway Capacity and Impacts on Performance: Phase 2*, https://rosap.ntl.bts.gov/view/dot/37083, Nov. 2004. Accessed: Oct. 10, 2025.

[128] D. Yang, Y. Wu, F. Sun, J. Chen, D. Zhai, and C. Fu, "Freeway accident detection and classification based on the multi-vehicle trajectory data and deep learning model," *Transportation Research Part C: Emerging Technologies*, vol. 130, p. 103 303, Sep. 2021, issn: 0968-090X. doi: 10.1016/j.trc.2021.103303 Accessed: Feb. 13, 2024.

[129] J. White, C. Thompson, H. Turner, B. Dougherty, and D. C. Schmidt, "WreckWatch: Automatic Traffic Accident Detection and Notification with Smartphones," *Mobile Networks and Applications*, vol. 16, no. 3, pp. 285–303, Jun. 2011, issn: 1572-8153. doi: 10.1007/s11036-011-0304-8 Accessed: Feb. 26, 2026.

[130] I. A. Zualkernan, F. Aloul, F. Basheer, G. Khera, and S. Srinivasan, "Intelligent accident detection classification using mobile phones," in *2018 International Con-*

ference on Information Networking (ICOIN), Jan. 2018, pp. 504–509. doi: 10.1109/ICOIN.2018.8343170 Accessed: Feb. 26, 2026.

[131] A. Bonyar, A. Geczy, O. Krammer, H. Santha, B. Illes, J. Kaman, Z. Szalay, P. Hanak, and G. Harsanyi, "A review on current eCall systems for autonomous car accident detection," in *40th International Spring Seminar on Electronics Technology (ISSE)*, Sofia, Bulgaria: IEEE, May 2017, pp. 1–8, isbn: 978-1-5386-0582-0. doi: 10.1109/ISSE.2017.8000985 Accessed: Mar. 19, 2024.

[132] *Intelligent transport systems — Traffic and Travel Information (TTI) via Transport Protocol Experts Group (TPEG) data-streams — Part 1: Introduction, numbering and versions*, 2025. Accessed: Oct. 24, 2025.

[133] C. L. Dudek, C. J. Messer, and N. B. Nuckles, "Incident detection on urban freeways," *Transportation research record*, vol. 495, pp. 12–24, 1974, issn: 0361-1981. Accessed: Feb. 26, 2026.

[134] H. J. Payne, E. D. Helfenbein, and H. C. Knobel, *DEVELOPMENT AND TESTING OF INCIDENT DETECTION ALGORITHMS, VOLUME 2: RESEARCH METHODOLOGY AND DETAILED RESULTS*. Apr. 1976. Accessed: Feb. 26, 2026.

[135] H. J. Payne and S. C. Tignor, "FREEWAY INCIDENT-DETECTION ALGORITHMS BASED ON DECISION TREES WITH STATES," in *Transportation Research Record*, 1978, isbn: 978-0-309-02827-1. Accessed: Feb. 26, 2026.

[136] M. Levin and G. M. Krause, "Incident detection: A Bayesian approach," *Transportation research record*, vol. 682, pp. 52–58, 1978, issn: 0361-1981. Accessed: Feb. 26, 2026.

[137] R. L. Cheu and S. G. Ritchie, "Automated detection of lane-blocking freeway incidents using artificial neural networks," *Transportation Research Part C: Emerging Technologies*, vol. 3, no. 6, pp. 371–388, Dec. 1995, issn: 0968-090X. doi: 10.1016/0968-090X(95)00016-C Accessed: Feb. 26, 2026.

[138] H. Dia and G. Rose, "Development and evaluation of neural network freeway incident detection models using field data," *Transportation Research Part C: Emerging Technologies*, vol. 5, no. 5, pp. 313–331, Oct. 1997, issn: 0968-090X. doi: 10.1016/S0968-090X(97)00016-8 Accessed: Feb. 26, 2026.

[139] B. Abdulhai and S. G. Ritchie, "Enhancing the universality and transferability of freeway incident detection using a Bayesian-based neural network," *Transportation Research Part C: Emerging Technologies*, vol. 7, no. 5, pp. 261–280, Oct. 1999, issn: 0968-090X. doi: 10.1016/S0968-090X(99)00022-4 Accessed: Feb. 26, 2026.

[140] F. Yuan and R. L. Cheu, "Incident detection using support vector machines," *Transportation Research Part C: Emerging Technologies*, Traffic Detection and Estimation, vol. 11, no. 3, pp. 309–328, Jun. 2003, issn: 0968-090X. doi: 10.1016/S0968-090X(03)00020-2 Accessed: Feb. 26, 2026.

[141] P. Chakraborty, A. Sharma, and C. Hegde, "Freeway Traffic Incident Detection from Cameras: A Semi-Supervised Learning Approach," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov. 2018, pp. 1840–1845. doi: 10.1109/ITSC.2018.8569426 Accessed: Feb. 26, 2026.

[142] D. Tian, C. Zhang, X. Duan, and X. Wang, "An Automatic Car Accident Detection Method Based on Cooperative Vehicle Infrastructure Systems," *IEEE Access*, vol. 7, pp. 127 453–127 463, 2019, issn: 2169-3536. doi: 10.1109/ACCESS.2019.2939532 Accessed: Feb. 13, 2024.

[143] Y. Yao, M. Xu, Y. Wang, D. J. Crandall, and E. M. Atkins, *Unsupervised Traffic Accident Detection in First-Person Videos*, Jul. 2019. doi: 10.48550/arXiv.1903.00618 arXiv: 1903.00618 [cs]. Accessed: Mar. 19, 2024.

[144] M. M. Karim, R. Qin, and Z. Yin, *An Attention-guided Multistream Feature Fusion Network for Localization of Risky Objects in Driving Videos*, Sep. 2022. doi: 10.48550/arXiv.2209.07922 arXiv: 2209.07922 [cs]. Accessed: Feb. 26, 2026.

[145] P. Mehrannia, S. S. G. Bagi, B. Moshiri, and O. A. Al-Basir, "Deep representation of imbalanced spatio-temporal traffic flow data for traffic accident detection," *IET Intelligent Transport Systems*, vol. 17, no. 3, pp. 606–619, Mar. 2023, issn: 1751-956X. doi: 10.1049/itr2.12287 Accessed: Aug. 21, 2024.

[146] F. Hajri and H. Fradi, "Vision Transformers for Road Accident Detection from Dashboard Cameras," in *2022 18th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Nov. 2022, pp. 1–8. doi: 10.1109/AVSS56176.2022.9959545 Accessed: Feb. 26, 2026.

[147] P. Moriano, A. Berres, H. Xu, and J. Sanyal, "Spatiotemporal features of traffic help reduce automatic accident detection time," *Expert Systems with Applications*, vol. 244, p. 122 813, Jun. 2024, issn: 09574174. doi: 10.1016/j.eswa.2023.122813 Accessed: Aug. 26, 2024.

[148] B. N. Persaud and F. L. Hall, "Catastrophe theory and patterns in 30-second freeway traffic data— Implications for incident detection," *Transportation Research Part A: General*, vol. 23, no. 2, pp. 103–113, Mar. 1989, issn: 0191-2607. doi: 10.1016/0191-2607(89)90071-X Accessed: Feb. 26, 2026.

[149] H. Rehborn and J. Palmer, "ASDA/FOTO based on Kerner's three-phase traffic theory in North Rhine-Westphalia and its integration into vehicles," in *2008 IEEE Intelligent Vehicles Symposium*, Jun. 2008, pp. 186–191. doi: 10.1109/IVS.2008.4621192 Accessed: Feb. 26, 2026.

[150] Y. Zhang, F. Liu, S. Yue, Y. Li, and Q. Dong, "Accident Detection and Flow Prediction for Connected and Automated Transport Systems," *Journal of Advanced Transportation*, 2023.

[151] Z. Shi, Y. Wang, D. Guo, F. Jiao, H. Zhang, and F. Sun, "The Urban Intersection Accident Detection Method Based on the GAN-XGBoost and Shapley Additive Explanations Hybrid Model," *Sustainability*, vol. 17, no. 2, p. 453, Jan. 2025, issn: 2071-1050. doi: 10.3390/su17020453 Accessed: Feb. 26, 2026.

[152] S. Chandar, A. Reddy, M. Mansoor, and S. Jamadagni, "Road Accident Proneness Indicator Based On Time, Weather And Location Specificity Using Graph Neural Networks," in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec. 2020, pp. 1527–1533. doi: 10.1109/ICMLA51294.2020.00235 Accessed: Feb. 26, 2026.

[153] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," *Genetic Programming and Evolvable Machines*, vol. 19, no. 1, pp. 305–307, Jun. 2018, issn: 1573-7632. doi: 10.1007/s10710-017-9314-z Accessed: Feb. 26, 2026.

[154] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, *Image Inpainting for Irregular Holes Using Partial Convolutions*, Dec. 2018. doi: 10.48550/arXiv.1804.07723 arXiv: 1804.07723 [cs]. Accessed: Jan. 28, 2026.

[155] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., Cham: Springer International Publishing, 2015, pp. 234–241, isbn: 978-3-319-24574-4. doi: 10.1007/978-3-319-24574-4_28

[156] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," in *Proceedings of the IEEE Conference on*

Computer Vision and Pattern Recognition, 2017, pp. 2117–2125. Accessed: Feb. 26, 2026.

[157] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, Apr. 2018, issn: 1939-3539. doi: 10.1109/TPAMI.2017.2699184 Accessed: Feb. 26, 2026.

[158] J.-B. Cordonnier, A. Mahendran, A. Dosovitskiy, D. Weissenborn, J. Uszkoreit, and T. Unterthiner, "Differentiable Patch Selection for Image Recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2351–2360. Accessed: Feb. 26, 2026.

[159] F. Rempe, P. Franeck, U. Fastenrath, and K. Bogenberger, "A phase-based smoothing method for accurate traffic speed estimation with floating car data," *Transportation Research Part C: Emerging Technologies*, vol. 85, pp. 644–663, Dec. 2017, issn: 0968-090X. doi: 10.1016/j.trc.2017.10.015 Accessed: Feb. 28, 2024.

[160] F. Rempe, P. Franeck, and K. Bogenberger, "On the estimation of traffic speeds with Deep Convolutional Neural Networks given probe data," *Transportation Research Part C: Emerging Technologies*, vol. 134, p. 103 448, Jan. 2022, issn: 0968-090X. doi: 10.1016/j.trc.2021.103448 Accessed: Oct. 23, 2024.

[161] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A Comprehensive Survey on Graph Neural Networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, Jan. 2021, issn: 2162-237X, 2162-2388. doi: 10.1109/TNNLS.2020.2978386 arXiv: 1901.00596 [cs]. Accessed: Feb. 20, 2026.

[162] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural Message Passing for Quantum Chemistry," in *Proceedings of the 34th International Conference on Machine Learning*, PMLR, Jul. 2017, pp. 1263–1272. Accessed: Mar. 1, 2026.

[163] T. N. Kipf and M. Welling, *Semi-Supervised Classification with Graph Convolutional Networks*, Feb. 2017. doi: 10.48550/arXiv.1609.02907 arXiv: 1609.02907 [cs]. Accessed: Feb. 20, 2026.

[164] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, *Graph Attention Networks*, Feb. 2018. doi: 10.48550/arXiv.1710.10903 arXiv: 1710.10903 [stat]. Accessed: Feb. 26, 2026.

[165] S. Brody, U. Alon, and E. Yahav, *How Attentive are Graph Attention Networks?* Jan. 2022. doi: 10.48550/arXiv.2105.14491 arXiv: 2105.14491 [cs]. Accessed: Feb. 20, 2026.

[166] S. Bai, J. Z. Kolter, and V. Koltun, *An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling*, Apr. 2018. doi: 10.48550/arXiv.1803.01271 arXiv: 1803.01271 [cs]. Accessed: Feb. 20, 2026.

[167] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, issn: 0899-7667. doi: 10.1162/neco.1997.9.8.1735 Accessed: Feb. 20, 2026.

[168] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, *Attention Is All You Need*, Aug. 2023. doi: 10.48550/arXiv.1706.03762 arXiv: 1706.03762 [cs]. Accessed: Feb. 20, 2026.

[169] J. Lee, Y. Lee, J. Kim, A. R. Kosiorek, S. Choi, and Y. W. Teh, *Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks*, May 2019. doi: 10.48550/arXiv.1810.00825 arXiv: 1810.00825 [cs]. Accessed: Feb. 20, 2026.

[170] B. Yu, H. Yin, and Z. Zhu, "Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, Stockholm, Sweden: International Joint Conferences on Artificial Intelligence Organization, Jul. 2018, pp. 3634–3640, isbn: 978-0-9992411-2-7. doi: 10.24963/ijcai.2018/505 Accessed: Jun. 27, 2025.

[171] C. Zhang, J. J. Q. Yu, and Y. Liu, "Spatial-Temporal Graph Attention Networks: A Deep Learning Approach for Traffic Forecasting," *IEEE Access*, vol. 7, pp. 166 246–166 256, 2019, issn: 2169-3536. doi: 10.1109/ACCESS.2019.2953888 Accessed: Feb. 20, 2026.

[172] K. R. Radhika, S. V. Sheela, T. Jha, V. S. Sushant, and S. Rajibul, "Dynamic Graph Convolutional Network-based Long Short-Term Memory for Taxi Trip Time Prediction," in *2024 First International Conference on Software, Systems and Information Technology (SSITCON)*, Oct. 2024, pp. 1–4. doi: 10.1109/SSITCON62437.2024.10796568 Accessed: Feb. 20, 2026.

[173] Z. Hu, Y. Dong, K. Wang, and Y. Sun, *Heterogeneous Graph Transformer*, Mar. 2020. doi: 10.48550/arXiv.2003.01332 arXiv: 2003.01332 [cs]. Accessed: Feb. 20, 2026.

[174] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, *Structural-RNN: Deep Learning on Spatio-Temporal Graphs*, Apr. 2016. doi: 10.48550/arXiv.1511.05298 arXiv: 1511.05298 [cs]. Accessed: Feb. 20, 2026.

[175] S. Cheng, "Data-driven Spatio-temporal Crash Detection using Connected Vehicle Data," M.S. thesis, University of Stuttgart, Stuttgart, 2025.

[176] BASt, *Jahresauswertung der automatischen Verkehrszählung*, https://www.bast.de/DE/Themen/Digitales/HF_1/Massnahmen/verkehrszaehlung/Daten/2022_1/Jawe2022.html?nn=414410, 2022.

[177] J. Paczia, B. M. Staes, R. L. Bertini, S. Hoffmann, H. Rehborn, O. Stein, B. A. Bernhardt, and W. Huang, "Congested Pattern Analysis in the Framework of Kerner's Three-Phase Traffic Theory: A Comparison Between U.S. and German Probe Vehicle Data," in *Transportation Research Board 102nd Annual Meeting*, 2023. Accessed: Mar. 19, 2024.

[178] Mapzen, *Valhalla: Open Source Routing Engine*, https://github.com/valhalla/valhalla, 2024. Accessed: Jun. 14, 2024.

[179] OpenStreetMap contributors, *OpenStreetMap Contributors. Planet dump retrieved from https://planet.osm.org*, https://www.openstreetmap.org/#map=6/51.330/10.453, 2024. Accessed: Jun. 14, 2024.

[180] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD'96, Portland, Oregon: AAAI Press, Aug. 1996, pp. 226–231. Accessed: Jul. 11, 2024.

[181] H. Jiang, J. Jang, and J. Łącki, *Faster DBSCAN via subsampled similarity queries*, 2020. doi: 10.48550/ARXIV.2006.06743 Accessed: Jul. 11, 2024.

[182] D. Birant and A. Kut, "ST-DBSCAN: An algorithm for clustering spatial–temporal data," *Data & Knowledge Engineering*, Intelligent Data Mining, vol. 60, no. 1, pp. 208–221, Jan. 2007, issn: 0169-023X. doi: 10.1016/j.datak.2006.01.013 Accessed: Nov. 5, 2025.

[183] R. J. G. B. Campello, D. Moulavi, and J. Sander, "Density-Based Clustering Based on Hierarchical Density Estimates," in *Advances in Knowledge Discovery and Data Mining*, J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, Eds., Berlin, Heidelberg: Springer, 2013, pp. 160–172, isbn: 978-3-642-37456-2. doi: 10.1007/978-3-642-37456-2_14

[184] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for Hyper-Parameter Optimization," in *Advances in Neural Information Processing Systems*, vol. 24, Curran Associates, Inc., 2011. Accessed: Dec. 5, 2025.

[185] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19, New York, NY, USA: Association for Computing Machinery, Jul. 2019, pp. 2623–2631, isbn: 978-1-4503-6201-6. doi: 10.1145/3292500.3330701 Accessed: Nov. 7, 2025.

[186] DWD, *Open Data Server [CDC Grids Germany]*, https://opendata.dwd.de/climate_environment/, 2025. Accessed: Mar. 19, 2025.

[187] J. Kreklow, B. Tetzlaff, B. Burkhard, and G. Kuhnt, "Radar-Based Precipitation Climatology in Germany—Developments, Uncertainties and Potentials," *Atmosphere*, vol. 11, no. 2, p. 217, Feb. 2020, issn: 2073-4433. doi: 10.3390/atmos11020217 Accessed: Feb. 27, 2026.

[188] J. Zeder and E. M. Fischer, "Observed extreme precipitation trends and scaling in Central Europe," *Weather and Climate Extremes*, vol. 29, p. 100 266, Sep. 2020, issn: 2212-0947. doi: 10.1016/j.wace.2020.100266 Accessed: Feb. 27, 2026.

[189] W. Forum, "Transport Protocol Experts Group (TPEG) Overview and Technical Introduction," WorldDAB, Tech. Rep., 2017. Accessed: Oct. 14, 2025.

[190] *Intelligent transport systems — Traffic and travel information messages via traffic message codingPart 1: Coding protocol for Radio Data System-Traffic Message Channel (RDS-TMC) using ALERT-C*, 2021.

[191] TISA, *TPEG - Transport Protocol Experts Group*, 2014. Accessed: Jan. 7, 2025.

[192] E. EIP, "Reference Handbook for harmonized ITS Core Service Deployment in Europe," 2021.

[193] E. Commission, "Technical Guidance for the Implementation of Safety-Related Traffic Information (SRTI) and Real-Time Traffic Information (RTTI)," Tech. Rep., 2022.

[194] TISA, "RTTI 5-Star Rating Scheme: Evaluation Methodology Workshop," Traveller Information Services Association (TISA), Tech. Rep., 2025.

[195] M. Tkachenko, M. Malyuk, A. Holmanyuk, and N. Liubimov, *Label Studio: Data labeling software*, Dec. 2025.

[196] T. Afrin and N. Yodo, "A Survey of Road Traffic Congestion Measures towards a Sustainable and Resilient Transportation System," *Sustainability*, vol. 12, no. 11, p. 4660, Jan. 2020, issn: 2071-1050. doi: 10.3390/su12114660 Accessed: Feb. 27, 2026.

[197] A. Skabardonis, P. Varaiya, and K. F. Petty, "Measuring Recurrent and Nonrecurrent Traffic Congestion," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1856, no. 1, pp. 118–124, Jan. 2003, issn: 0361-1981, 2169-4052. doi: 10.3141/1856-12 Accessed: Apr. 28, 2025.

[198] Y. Chung, "Assessment of non-recurrent congestion caused by precipitation using archived weather and traffic flow data," *Transport Policy*, vol. 19, no. 1, pp. 167–173, Jan. 2012, issn: 0967-070X. doi: 10.1016/j.tranpol.2011.10.001 Accessed: Feb. 27, 2026.

[199] A. Rahman and N. E. Lownes, "Analysis of rainfall impacts on platooned vehicle spacing and speed," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 15, no. 4, pp. 395–403, Jul. 2012, issn: 1369-8478. doi: 10.1016/j.trf.2012.03.004 Accessed: Mar. 19, 2026.

[200] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*. John Wiley & Sons, Feb. 2013, isbn: 978-1-118-54835-6.

[201] S. E. Stevens, C. J. Schreck, S. Saha, J. E. Bell, and K. E. Kunkel, "Precipitation and Fatal Motor Vehicle Crashes: Continental Analysis with High-Resolution Radar Data," *Bulletin of the American Meteorological Society*, vol. 100, no. 8, pp. 1453–1461, Aug. 2019, issn: 0003-0007, 1520-0477. doi: 10.1175/BAMS-D-18-0001.1 Accessed: Mar. 11, 2026.

[202] S. Tafazzol, H. Sharif, M. Gholikhani, M. Ahsan, D. Ghebreyesus, K. Billah, and C. Furl, "Relative crash risk and road safety during rainfall in Texas from 2006 to 2021," *Scientific Reports*, vol. 15, no. 1, p. 36 749, Oct. 2025, issn: 2045-2322. doi: 10.1038/s41598-025-20760-w Accessed: Mar. 11, 2026.

[203] B. S. Kerner, *Features of Traffic Congestion caused by bad Weather Conditions or Accident*, 2007. doi: 10.48550/ARXIV.0712.1728 Accessed: Dec. 16, 2024.

[204] Y. Ali, F. Hussain, and M. M. Haque, "Advances, challenges, and future research needs in machine learning-based crash prediction models: A systematic review," *Accident Analysis & Prevention*, vol. 194, p. 107 378, Jan. 2024, issn: 0001-4575. doi: 10.1016/j.aap.2023.107378 Accessed: Oct. 2, 2025.

[205] S. Wang, J. Cao, and P. S. Yu, *Deep Learning for Spatio-Temporal Data Mining: A Survey*, Jun. 2019. doi: 10.48550/arXiv.1906.04928 arXiv: 1906.04928 [cs]. Accessed: Feb. 24, 2025.

[206] J. Golze, U. Feuerhake, C. Koetsier, and M. Sester, "IMPACT ANALYSIS OF ACCIDENTS ON THE TRAFFIC FLOW BASED ON MASSIVE FLOATING CAR DATA," in *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLIII-B4-2021, Copernicus GmbH, Jun. 2021, pp. 95–102. doi: 10.5194/isprs-archives-XLIII-B4-2021-95-2021 Accessed: Feb. 16, 2024.

[207] X. Ma, Z. Dai, Z. He, J. Na, Y. Wang, and Y. Wang, *Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction*, Apr. 2017. doi: 10.48550/arXiv.1701.04245 arXiv: 1701.04245 [cs]. Accessed: Feb. 18, 2026.

[208] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems*, vol. 32, Curran Associates, Inc., 2019. Accessed: Apr. 8, 2026.

[209] M. Fey and J. E. Lenssen, *Fast Graph Representation Learning with PyTorch Geometric*, Apr. 2019. doi: 10.48550/arXiv.1903.02428 arXiv: 1903.02428 [cs]. Accessed: Apr. 8, 2026.

[210] *Das Zentrum für all Ihre Daten, Analytik und KI – Amazon SageMaker – AWS*, https://aws.amazon.com/de/sagemaker/. Accessed: Apr. 8, 2026.

[211] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, Jan. 2017. doi: 10.48550/arXiv.1412.6980 arXiv: 1412.6980 [cs]. Accessed: Jan. 28, 2026.

[212] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034. Accessed: Mar. 2, 2026.

[213] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, JMLR Workshop and Conference Proceedings, Mar. 2010, pp. 249–256. Accessed: Mar. 2, 2026.

[214] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 922–929, Jul. 2019, issn: 2374-3468. doi: 10.1609/aaai.v33i01.3301922 Accessed: Feb. 20, 2026.

[215] D. S. Johnson, "Fast algorithms for bin packing," *Journal of Computer and System Sciences*, vol. 8, no. 3, pp. 272–314, Jun. 1974, issn: 0022-0000. doi: 10.1016/S0022-0000(74)80026-7 Accessed: Mar. 3, 2026.

[216] Z. Chen, X. C. Liu, and G. Zhang, "Non-recurrent congestion analysis using data-driven spatiotemporal approach for information construction," *Transportation Research Part C: Emerging Technologies*, vol. 71, pp. 19–31, Oct. 2016, issn: 0968090X. doi: 10.1016/j.trc.2016.07.002 Accessed: Apr. 19, 2025.

[217] J. Li, H. Yang, and S. Razavi, "Predicting freeway non-recurring congestion via a spatio-temporal deep learning approach," *Transportmetrica A: Transport Science*, pp. 1–29, Dec. 2024, issn: 2324-9935, 2324-9943. doi: 10.1080/23249935.2024.2438314 Accessed: Apr. 19, 2025.

[218] A. Abdi, S. Seyedabrishami, and S. O'Hern, "A Two-Stage Sequential Framework for Traffic Accident Post-Impact Prediction Utilizing Real-Time Traffic, Weather, and Accident Data," *Journal of Advanced Transportation*, vol. 2023, T. Campisi, Ed., pp. 1–16, Jun. 2023, issn: 2042-3195, 0197-6729. doi: 10.1155/2023/8737185 Accessed: Feb. 17, 2025.

[219] J. Zhu, Y. Song, L. Zhao, and H. Li, "A3T-GCN: Attention Temporal Graph Convolutional Network for Traffic Forecasting," *ISPRS International Journal of Geo-Information*, vol. 10(7), p. 485, 2021. doi: 10.48550/ARXIV.2006.11583 Accessed: Jun. 27, 2025.

[220] A. Derrow-Pinion, J. She, D. Wong, O. Lange, T. Hester, L. Perez, M. Nunkesser, S. Lee, X. Guo, B. Wiltshire, P. W. Battaglia, V. Gupta, A. Li, Z. Xu, A. Sanchez-Gonzalez, Y. Li, and P. Veličković, "ETA Prediction with Graph Neural Networks in Google Maps," 2021. doi: 10.48550/ARXIV.2108.11482 Accessed: Jun. 27, 2025.

[221] M. Dannehy, *TomTom Real Time Data and Emergency Management*, https://proceedings.esri.com/library/userconf/proc16/papers/566_455.pdf, 2016. Accessed: Jun. 27, 2025.

[222] A. Mystakidis, P. Koukaras, and C. Tjortjis, "Advances in Traffic Congestion Prediction: An Overview of Emerging Techniques and Methods," *Smart Cities*,

vol. 8, no. 1, p. 25, Feb. 2025, issn: 2624-6511. doi: 10.3390/smartcities8010025 Accessed: Apr. 19, 2025.

[223] A. Grigorev, S. Shafiei, H. Grzybowska, and A.-S. Mihăiţă, "Predicting the Duration of Traffic Incidents for Sydney Greater Metropolitan Area using Machine Learning Methods," *International Journal of Intelligent Transportation Systems Research*, vol. 23, no. 1, pp. 104–125, Apr. 2025, issn: 1348-8503, 1868-8659. doi: 10.1007/s13177-024-00437-w Accessed: Jun. 27, 2025.

[224] T. Lian and B. P. Loo, "Cost of travel delays caused by traffic crashes," *Communications in Transportation Research*, vol. 4, p. 100 124, Dec. 2024, issn: 27724247. doi: 10.1016/j.commtr.2024.100124 Accessed: Jun. 27, 2025.

[225] H. Rehborn, S. L. Klenov, and M. Koller, "Traffic Prediction of Congested Patterns," in *Encyclopedia of Complexity and Systems Science*, R. A. Meyers and B. S. Kerner, Eds., New York, NY: Springer, 2019, pp. 501–557, isbn: 978-3-642-27737-5. doi: 10.1007/978-3-642-27737-5 Accessed: Jul. 3, 2025.

[226] R. Dai, S. Xu, Q. Gu, C. Ji, and K. Liu, "Hybrid Spatio-Temporal Graph Convolutional Network: Improving Traffic Prediction with Navigation Data," in *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20)*, San Diego, CA, USA: arXiv, 2020, pp. 3074–3082. doi: 10.48550/ARXIV.2006.12715 Accessed: Jun. 27, 2025.

Declaration of AI Use

Large language models (Anthropic, Claude; OpenAI ChatGPT) were used during the preparation of this manuscript to support the exploration of the relevant literature, to assist with code development, and to validate and compare the statistical methods applied, as well as for language editing and grammar correction. All scientific content, analyses, and conclusions remain the sole responsibility of the author.

Lennart Querfurth

List of Figures

2.1 Qualitative TSD of microscopic FCD. Dots denote position reports; consecutive points of the same vehicle are connected to establish trajectories. Colors indicate speed, with warmer colors representing higher speeds. The diagram illustrates how trajectories converge in congested regions and how the slope of trajectories corresponds to vehicle speed. 8
2.2 Qualitative explanation of traffic phases, their transitions and characteristics (adapted from [78]). 11
2.3 Empirical characteristics of metastable F and breakdown probability [42]. Left: J-line demonstrating the metastability of F at high flow rates. Right: Breakdown probability as a function of traffic flow. 12
2.4 Qualitative illustration of diverse congestion patterns and spontaneous breakdowns (F→S) at a MB in the space–time plane [78]. 15
2.5 CNN principle 20

3.1 Velocity TSD for the A81 segment between Stuttgart and Bad-Dürrheim on 2022-03-18. Colors encode instantaneous speed (kmh^{-1}); horizontal axis is time, vertical axis is distance along the route. 28
3.2 Phase-colored TSD for the A81 segment between Stuttgart and Bad-Dürrheim on 2022-03-18. Colors encode traffic phases following [42]. 28
3.3 Clustering of congested regions based on traffic patterns on A81 near Stuttgart. Colors indicate distinct clusters identified by the DBSCAN algorithm; horizontal axis is time, vertical axis is distance along the route. 31
3.4 Comparison of region clustering results for three freeway segments in Germany using different DBSCAN variants: (a) AP-DBSCAN (this work), (b) ST-DBSCAN [182], (c) HDBSCAN [183], and (d) standard DBSCAN [180]. Each row corresponds to a different freeway segment and date, with colors indicating distinct clusters identified by each algorithm. Horizontal axis is time, vertical axis is distance along the route. 33
3.5 Dataset including rain data, infrastructural bottlenecks, traffic flow and FCD. A5 between Karlsruhe and Offenburg, on July 10, 2024. 34

3.6 TSD files used for labeling are plotted without axes or borders to allow easy matching of annotated pixel positions to the time-space domain of the dataset. The example shows two raw TPEG event codes and a TPEG – Mega jam aligned with the second event code. 36
3.7 Examples for each labeling category present in the dataset. TPEG event code is annotated in $(a) - (d)$ 37
3.8 Velocity-weighted sampling on a speed-distance diagram, with $N_d = 3$ and $N_t = 6$. Two candidate cells are highlighted, with sampling probabilities inversely proportional to their mean speeds. The red cell shows a congested region with higher sampling probability, while the blue cell shows a free-flow region with lower sampling probability. 42
3.9 Crash-centred sampling around a manually annotated crash event. Windows are generated within a temporal buffer around the event, covering pre-crash, crash, and recovery conditions. 43
3.10 Sequential congestion sampling. Windows are placed at regular intervals along the time axis within a labelled congestion region; spatial and temporal perturbations are applied during training. 43

4.1 (a) Breakdown probability shift and habituation effect, (b) Increased safe following distance in rainy conditions [5]. 48
4.2 Conceptual illustration of congestion dynamics for upstream- and downstream-moving LRCs. 49
4.3 Conceptual explanation of direction-dependent congestion patterns induced by LRCs. Left: downstream-moving LRC with prolonged exposure. Right: upstream-moving LRC with limited congestion growth. 49
4.4 Both travel directions of German freeway A7 on June 2, 2024. Identical LRCs act as upstream-MBs in one direction and downstream-MBs in the other. (Top) shows the full distance. (Bottom) shows zoom-in at 80 km clearly showing direction-dependent impact. 51
4.5 The same LRC propagates downstream relative to traffic on the right and upstream on the left, showing the direction-dependent congestion response. 52
4.6 A7 between Füssen and Ulm on August 18, 2024. Opposite driving directions. Low-intensity and high-intensity LRC cells interact differently depending on their propagation direction relative to traffic. 53
4.7 A81 between Bad Dürrheim and Stuttgart on September 8, 2024. Same driving direction shown with precipitation overlay (left) and without (right). A low-intensity LRC reduces vehicle speed without triggering traffic breakdown. 54

4.8 A5 between Karlsruhe and Heidelberg on August 18, 2024. Opposite driving directions. A high-intensity LRC cell embedded in a broader low-intensity field triggers localized breakdown, while the surrounding low-intensity precipitation only reduces speed. 54
4.9 Breakdown onsets over normalized distance to the dry-to-wet front. The dashed line marks uniform expectation, the solid curve a quadratic fit. Breakdowns concentrate at both transition fronts; the effect is most pronounced for downstream-moving LRCs. 58

5.1 Congestion features in recurrent (left) and crash-induced (right) congestion. Dashed lines indicate individual vehicle trajectories [3]. 62
5.2 Spatial and temporal deviation of the proposed method (black line) compared to TPEG-based crash information (blue square). The red area indicates a mega jam evolving from a crash at the upstream seed of the congestion pattern [3]. 62
5.3 Categories of crash-induced congestion patterns. 64

6.1 Pre-jam free-flow ratio for selected jam phases. Left: phase-colored trajectories with the upstream query windows shown as rectangles. Right: stacked bar chart of the F/S/J composition per window. Crash-induced jams (1-3) exhibit a higher free-flow share, while recurring congestion (4-10) shows a higher congested share. 68
6.2 Downstream front velocity evolution for a crash-induced congestion (1-3) and a WMJ (4). The downstream front remains near zero until the crash is cleared, reaching full confidence after four consecutive in-band readings. The WMJ exhibits a negative propagation velocity. 69
6.3 Blockage detection for a crash with full road closure (1). Trajectories entering the congested region decelerate to standstill (1) and remain blocked beyond the duration threshold. The blockage rate β rises as additional sessions confirm sustained standstill. Crashes without full blockage (2-4) and recurring congestion (5-8) show low or zero β and do not reach the session count threshold. 71
6.4 Cluster dominance for crash-induced congestion (1-3) and WMJs (4, 5). The crash shows a high cluster dominance due to the single coherent cluster, while the WMJ exhibits lower dominance as the jam fragments into multiple clusters. 72

6.5 Velocity difference between upstream and downstream fronts for crash-induced congestion (1-3) and WMJs (4, 5). The crash shows a negative velocity difference due to the fixed downstream front and upstream-propagating front, while the WMJ exhibits a near-zero velocity difference as both fronts propagate upstream. 72
6.6 Mean speed within the jam phase for crash-induced congestion (1–3) and WMJs (4,5). Crashes exhibit near-zero mean speed and, therefore, no penalty, while WMJs retain residual flow that activates the penalty ramp. 74
6.7 End-of-crash detection. Left: downstream front transitions with regression lines for the stationary crash phase (solid, $\hat{v}_g \approx 0$) and the dissipation phase (dashed, $\hat{v}_g \approx -15\,\mathrm{km\,h^{-1}}$); the diamond marks the detected regime transition. Right: downstream front velocity over time, showing the transition from spatial fixation to upstream propagation. 76
6.8 Pareto front of F_1 vs. median seed-to-alert delay for the 224 configurations evaluated in the grid search. 78
6.9 Feature-isolation experiment for the *Accuracy* configuration. Each bar shows the F_1 obtained when only the indicated feature is active, with all other features and penalties disabled. 79

7.1 Step-by-step construction of the CNN input tensor from raw vehicle trajectories [175]. 86
7.2 Sector-based propagation counting of jam pixels. The blue-bordered pixel falls within three J sectors and two S sectors, giving a net count of 1 below the threshold and is therefore not assigned to J. [175] 87
7.3 Encoder block structure. The trapezoid-marked convolution uses stride 1 in the input block and stride 2 in downsampling blocks. $\circledast$ denotes channel-wise concatenation [175]. 88
7.4 Decoder block structure. $\circledast$ denotes channel-wise concatenation [175]. . . . 88
7.5 Overall U-Net backbone architecture [175]. 89
7.6 Vehicle nodes and supernodes with phase context in $\mathcal{W}_{D,T}$ [175]. 92
7.7 Illustration of vehicle WMJ edges $\mathcal{E}_{\mathrm{WMJ}}$ [175]. 93
7.8 Illustration of supernode edges $\mathcal{E}_{\mathrm{super}}$ [175]. 94
7.9 Temporal sequences [175]. 94
7.10 Ground-truth label construction. 95
7.11 GNN model architecture overview [175]. 96

8.1 Validation F_1 score versus decision threshold τ for each detector. The rule-based curve corresponds to the Accuracy configuration 103
8.2 Threshold-independent evaluation of crash detection models on the test set. (a) ROC curves with corresponding AUC values. (b) PR curves with corresponding AUC values. 106
8.3 Pareto front of F_1 vs. median detection delay across the three deep-learning detectors (per-threshold sweeps on a single trained model each) and the rule-based detector's full calibration design space ($n = 387$ Optuna trials, each with its own calibrated parameter set). 106
8.4 Distribution of crash onset delay Δt_{on} (left) and clearance deviation Δt_{off} (right) on the test set. The TPEG reference (dashed line) is shown for comparison. Positive onset values indicate delayed detection; negative clearance values indicate premature clearance. Boxes span the interquartile range with the median marked inside; whiskers extend to $1.5 \times$ IQR; points beyond are outliers. Numerical summaries are reported in Tables 8.3 and 8.4. 109
8.5 Mean and median MAE (in meters) for crash location regression, grouped by evaluation mode. 111
8.6 Crash detection results across nine German freeway corridor samples. Annotated GT crashes are shaded in gray. Upper panels show phase maps with F, S, and J phases; lower panels show crash probability curves from the four approaches. 114
8.7 Crash detection results across three UK freeway corridor samples. Annotated GT crashes are shaded in gray. Upper panels show phase maps with F, S, and J phases; lower panels show crash probability curves from the four approaches. 115
8.8 Crash-probability curves across four scenarios. 116
8.9 Example of label noise in the dataset. The TPEG reference (blue), the annotated GT (red) the detected crashes by the RB model (black) and the remaining clustered congestion patterns (grey) are shown. 120

9.1 Schematic representation of post-crash congestion evolution with transition points (A) onset, (B) upstream starvation, (C) crash clearance, (D) upstream dissipation, and (E) congestion resolution. 122
9.2 Within-stage refinement during the upstream-starvation stage (A8, November 19, 2024). The predicted hull (red) is re-estimated every two minutes; the upstream front velocity converges toward the observed propagation as new FCD trajectories enter the regression window. 125

9.3 Detection of upstream starvation via continuously evaluated slopes of the upstream front velocity v_u. The red star marks the detected starvation point. 126
9.4 Crash clearance detection based on downstream front velocity. Gray points: detected front positions; colored lines: regression slopes; red dotted line: clearance point. 126
9.5 Predicted congestion area for US I-294, February 24, 2024. The gray area represents the actual congestion hull; orange, green, and red show the three prediction stages. Trajectories are colored by 3PTT phase. The rolling error and 30 min error per stage are shown in the lower panel. 131
9.6 Predicted congestion area for UK M3, September 25, 2025. 132
9.7 Predicted congestion area for German A8, November 19, 2024. 133
9.8 Predicted congestion area for German A8, March 20, 2022. TPEG Reference in blue. 133

A.1 Visualization of the detection window sampling process on the A8 freeway on 7 October 2024. 142

D.1 Node flow of vehicle nodes and supernodes through the GNN model (Section 7.3.2) [175]. 153

E.1 Sampling of spatiotemporal observation windows for the A6 crash case. Seven representative windows are shown; in practice a new 30 km × 30 min window is generated every 10 s [175]. 161
E.2 Partial-convolution inputs across the seven representative windows: sparse velocity channel and one-hot encoded phase tensor derived from the TSDs [175]. 163
E.3 PSM inputs across the seven representative windows: completed velocity map and associated phase tensor derived from the TSDs [175]. 163

F.1 Empirical example of upstream flow starvation (B) on the German A81, March 19, 2022. *Top:* connected-vehicle trajectories colored by 3PTT phase; freeway corridor and ramps shown on the right axis. *Bottom:* fleet flow rate at two measurement locations, with the drop at the upstream measurement point coinciding with the detected starvation time. 167
F.2 ANOVA effect size η^2 per prediction parameter for $k = 3$ clusters over $n = 445$ events. Red: large effect ($\eta^2 \geq 0.14$); yellow: medium effect; blue: small effect. Propagation-velocity parameters fall in the small-effect range. . . . 169

F.3 Predicted congestion area for the German A81, March 18, 2022, shown *without* within-stage refinement. The gray area represents the actual congestion hull; orange, green, and red show the three prediction stages. Trajectories are colored by 3PTT phase. Rolling error is reported only at the 15-minute horizon, since beyond 30 minutes the compounded spatial error grows rapidly. 170
F.4 Predicted congestion area for US NJ 3, March 3, 2024. The gray area represents the actual congestion hull; orange, green, and red show the three prediction stages. Trajectories are colored by 3PTT phase. The rolling error and 30-minute error per stage are shown in the lower panel. 171
F.5 Multiple delay prediction samples for Mega-Jams on German freeways showing limits of TPEG-based ETA prediction for mega jams. 173

List of Tables

2.1 Phase detection parameters adapted from [79]. Speed thresholds in $\mathrm{km h}^{-1}$; dwell times in seconds. 13
2.2 Rain intensity classifications [105]. 16

3.1 Geographic and temporal coverage of the trajectory dataset. 26
3.2 Tuned AP-DBSCAN hyperparameters for German freeway sections. 30
3.3 Comparison of clustering results. Best score in **bold**, second best underlined. Arrows indicate optimization direction. 32
3.4 Rain-cluster characteristics of wet section-days (peak intensity $> 10\,\mathrm{mm/h}$, $n = 509$). 34
3.5 Distribution of labeled congestion events by source and country. 38
3.6 Overview of crash-related congestion categories, cumulative durations, and membership in ground-truth sets. Rows are not mutually exclusive and must not be summed (see text). 39
3.7 Crash-related congestion time and crash ratio per ground-truth set. 40
3.8 Contribution of each sampling mode to the training set. 44
3.9 Classification dataset statistics. The test set is shared across both pipelines and evaluated at the region level. 44

4.1 Rain–congestion constraint categories by LRC direction (onset in rain, $|v_{\mathrm{LRC}}| > 5\,\mathrm{km\,h}^{-1}$). 56
4.2 Constraint rate by upstream LRC speed ($v_{\mathrm{LRC}} < -1\,\mathrm{km\,h}^{-1}$, $n = 110$). . . 56
4.3 Key statistical results. Effect sizes, test statistics, and p-values for all main findings. 59

6.1 Pareto-optimal operating points on the independent test set (65 ground-truth crashes, 50 datasets). Delays are seed-relative; $\tilde{\Delta}t$ denotes the median. . . 78
6.2 Ablation across the three operating points. Selected rows illustrate the principal feature contributions; ΔFP denotes the change in false positives relative to the respective baseline. Full per-profile tables are provided in Tables C.5 to C.7. 79

6.3 Detection performance of the *Accuracy* configuration under varying fleet density. TP/FP are averaged across three random seeds per level; $\tilde{\Delta}t$ denotes the median seed-to-alert delay. 80

7.1 Reduced training set used for model optimization and iterative development. 85
7.2 CNN pipeline: PSM hyperparameters and CNN architecture [175]. 90
7.3 Ablation results on the validation set (threshold = 0.5). Best values in **bold**, second best underlined [175]. 91
7.4 Graph-construction hyperparameters [175]. 97
7.5 Module-wise ablation on the validation set (threshold = 0.5) [175]. 99

8.1 Updated confusion matrices for all approaches on the test set (aligned to $GT = 66$). Rows denote predicted labels (p' positive, n' negative) and columns denote actual labels (p positive, n negative). 104
8.2 Test-set classification results of all four approaches. Best thresholds are determined on the validation set for deep learning models; the rule-based method uses its calibrated parameter set directly. The best and second-best scores among deep learning approaches are highlighted in **bold** and underline, respectively. 105
8.3 Statistics of crash clearance detection delay Δt_{off} (in minutes). Negative values indicate premature clearance relative to ground truth. The TPEG reference ($n = 224$) is evaluated on the full annotation set; all other approaches on the held-out test set. Best and second-best values (closest to zero or smallest variance) are highlighted in **bold** and underline, respectively. 108
8.4 Statistics of crash onset detection delay Δt_{on} (in minutes). Positive values indicate delayed detection relative to ground truth. The TPEG reference ($n = 224$) is evaluated on the full annotation set; all other approaches on the held-out test set. Best and second-best values are highlighted in **bold** and underline, respectively. 109
8.5 Crash location regression errors on the test set (meters), stratified by evaluation mode. *First*: initial detection window; *EvMean*: event-level mean over all triggered windows; *Best*: lowest-error window per event. The TPEG reference reports the centroid deviation of matched Raw Event polygons against GT Mega Jam annotations ($n = 224$). Best and second-best values per column are marked in **bold** and underline. 110

8.6 Model size, average inference time, and throughput of the selected approaches. Deep-learning times are per 30 min window on GPU, covering the model forward pass only. The rule-based time is amortized per 30 min window from a full-day CPU run and covers rule evaluation against pre-clustered phase state; clustering and data loading are excluded (see text). For CNN (PSM), the reported time includes both PSM and CNN computation. Rule-based parameters refer to the count of optimized rule weights, thresholds, and gate settings rather than trained weights. 112
8.7 Summary of approach performance. The TPEG production system is included as an operational reference. Two check marks (✓✓) denote the best performance, one (✓) indicates moderate, and a cross (×) marks the weakest. "Interpretability" indicates whether the detection logic is fully transparent and auditable. 118

9.1 Parameters used in each prediction stage and their availability. Parentheses indicate limited reliability. 124
9.2 Global default prediction parameters (medians over 445 events). Times in minutes, velocities in $\mathrm{km\,h^{-1}}$. 127
9.3 IoU scores across prediction stages. 128
9.4 Corpus filtering from matched Mega jams to the curated paired-comparison subset. The first block removes events where TPEG cannot supply a usable baseline; the second block applies surrounding-congestion classification, used to partition retained events into Group A (with surrounding congestion that confounds TPEG's section-level delay reading) and Group B (free-flow surroundings). 129
9.5 Travel-time delay prediction error against the TPEG congestion-delay baseline, evaluated on the active-phase records of the corpus subsets defined in Table 9.4. Hull statistics pooled per 1-minute tick across 38 events (4,228 ticks total); TPEG statistics pooled across 17 events (1,693 ticks). MAE, RMSE and signed bias in minutes. Ground truth: real trajectory; simulator parameters: per-stage measured v_{cong} with floor 8 km/h, $v_{\mathrm{free}} = 110$ km/h. . 130
9.6 In-jam h=15 MAE stratified by realised delay at the target forecast time. The ≥ 60 min bucket is empty for paired Group B records at h=15 in the present corpus. 130

A.1 Thematic grouping of TPEG event codes. 141

B.1 Logistic regression predicting rain-related congestion among wet observations ($n = 509$). Reported in Section 4.5.1. 143
B.2 Adjusted logistic regression predicting rain-front constraint ($n = 190$). Upstream direction is the primary predictor; covariates control for congestion severity. Reported in Section 4.5.2. 143
B.3 Sensitivity of the direction–constraint association to the velocity threshold used for classifying upstream vs. downstream LRCs. All tests are unadjusted Fisher exact tests. Reported in Section 4.5.2. 144
B.4 Tests for concentration of breakdowns near the dry-to-wet front ($n = 93$, $I > 10\,\mathrm{mm/h}$). The binomial test result is reported in the main text (Section 4.5.3); Wilcoxon and KS tests are listed here for completeness. . . 144

C.1 Complete parameter configuration: Accuracy (Best F_1). 146
C.2 Complete parameter configuration: Balanced. 146
C.3 Complete parameter configuration: Speed (Fastest). 147
C.4 Frozen parameters (shared across all operating points). 147
C.5 Ablation: Accuracy (Best F_1). Baseline: $F_1 = 0.750$, TP = 48, FP = 15, median delay = 9.8 min. 148
C.6 Ablation: Balanced. Baseline: $F_1 = 0.718$, TP = 47, FP = 19, median delay = 8.1 min. 148
C.7 Ablation: Speed (Fastest). Baseline: $F_1 = 0.653$, TP = 49, FP = 36, median delay = 6.1 min. 148
C.8 Performance of individual features in isolation (fixed $w = 0.8$, threshold $\tau = 0.50$). Evaluated on the independent test set (65 ground-truth crashes). 149
C.9 Confusion matrices and extended detection metrics for the three operating points. 150

D.1 CNN: Validation results with different base channel widths d_{base} (threshold = 0.5) [175]. 153
D.2 CNN: Per-run validation results of ablation variants (threshold = 0.5) [175]. 154
D.3 GNN: Validation results with different hidden dimensions d_{hid} (threshold = 0.5) [175]. 154
D.4 GNN: Validation results with different loss weights for minority samples (threshold = 0.5) [175]. 154
D.5 Multi-output vs. single-output supervision (threshold = 0.5) [175]. 155
D.6 Temporal anchor distribution ablation (threshold = 0.5) [175]. 155
D.7 Supernode edge processing ablation (threshold = 0.5) [175]. 156

D.8 GNN: Feature ablation results (Groups A–G, threshold = 0.6) [175]. 156
D.9 GNN: Feature ablation results (Groups A–G, threshold = 0.7) [175]. 156
D.10 GNN: Feature ablation per-run results (Groups A–G, threshold = 0.5) [175]. 157
D.11 GNN: Output head ablation per-run results (threshold = 0.5) [175]. 157
D.12 GNN: Temporal anchor distribution per-run results (threshold = 0.5) [175]. 158
D.13 GNN: Supernode edge processing per-run results (threshold = 0.5) [175]. . 158
D.14 GNN: Module-wise ablation per-run results (threshold = 0.5) [175]. 159

E.1 Top ten Pareto-optimal operating points per detector, sampled evenly along the achievable F_1 range. The deep-learning detectors are listed by score-threshold τ on a single trained model; the rule-based detector is listed by v10 calibration trial number, since each Pareto-optimal point uses a different parameter set rather than a different threshold. 164
E.2 Ten Pareto-optimal operating points selected from the calibration study ($n = 387$ completed Optuna trials). Each row reports one trial's confusion-matrix counts (TP, FP, FN), derived metrics (F_1, F_2, $F_{0.5}$, precision P, recall R), and median detection delay. The total positive count is constant ($TP + FN = 66$ ground-truth crashes); the total negative count varies between trials with persistence parameters and is therefore not reported. Rows are ordered by detection delay. Multiple labels indicate that one trial satisfies several criteria. 165

F.1 Median parameter values per congestion type. Times in minutes, velocities in $\mathrm{km\,h^{-1}}$. 168
F.2 Hull simulation's prediction error as a function of $(v_{\mathrm{cong}}, v_{\mathrm{free}})$ on the same corpus and active-phase restriction as Table 9.5. 172
F.3 Rolling pooled error by lifecycle stage. Onset covers the alert tick to the upstream-starvation transition, starvation covers the bulk of the jam where the upstream front is back-propagating through the queue, and crash-end covers the tail decay after the obstruction clears. Hull pooled across 40 events; TPEG pooled across 24 events. 174